



Splunk 管理者マニュアル

バージョン : 4.0.3

作成日 : 2009 年 8 月 24 日 午後 5 時 1 分

Copyright Splunk, Inc. All Rights Reserved

目次

はじめに	1
本マニュアルについて	1
Splunk とは	1
準備	2
ブート時間に Splunk を起動する設定	2
Splunk Web で Splunk Manager を検索	3
ライセンスのインストール	3
デフォルト値の変更	5
Splunk の起動	9
Windows で Splunk を起動	9
UNIX で Splunk を起動	9
Splunk Web の起動	10
Splunk Web および Splunk Apps について	11
Splunk Web とは	11
App とは？	11
App の入手先	14
App アーキテクチャとオブジェクトの所有権	15
App オブジェクトの管理	16
設定の前に	18
設定方法	18
Splunk Manager について	19
設定ファイルについて	20
設定ファイルの場所	23
データの追加と入力の設定	25
Splunk の監視対象	25
ファイルとディレクトリの監視	27
ネットワークポートの監視	35
Windows イベントログデータを監視	39
Windows レジストリデータの監視	41
WMI データの監視	44
アクティブディレクトリの監査	48
ファイルシステムへの変更を監視	50
クローリングで監視対象事項を詳細検索	54
SNMP イベントを Splunk に送信	56

カスタム(スクリプト)入力の設定	56
ホワイトリストまたはブラックリスト専用の受信データ	58
ログローテーションの処理方法	60
インデキシングとイベント処理	62
イベントとは何か	62
インデキシングのしくみ	62
セグメンテーションによるデータ圧縮の向上	64
ホストフィールドの値の設定方法	64
入力用にホストの値を設定	65
イベントデータに基づいてホストの値を設定	67
イベントにメタデータをダイナミックに割り当てる	69
複数行イベントのインデックス	69
キャラクタセットエンコードの設定	72
sed によるデータの匿名化	72
Splunk 構文解析をサポートするカスタムログの設定	73
タイムスタンプ	74
タイムスタンプのしくみ	74
タイムスタンプ認識の設定	75
複数のタイムスタンプを持つイベントのタイムスタンプ抽出を設定	79
タイムスタンプにタイムゾーンオフセットを適用	80
ローカライズされたタイムスタンプフォーマット(ヨーロッパなど)の認識	81
Splunk にタイムスタンプを認識するよう指示	81
インデキシング性能を向上させるタイムスタンプ抽出の調整	85
ユーザーと役割について	87
ユーザーの追加と役割の割り当て	87
Splunk によるユーザー認証の設定	91
LDAP によるユーザー認証の設定	92
Splunk で PAM または RADIUS 認証を使用するための設定	100
CLI を使用したユーザアカウントの削除	102
インデックスの管理	104
インデックスの管理について	104
複数インデックスの設定	105
サマリーインデキシングによるレポート効率の向上	107
ディスク使用量の制限を設定	107
ディスク使用を管理するセグメンテーションの設定	108
ホスト、ソース、ソースタイプに対するカスタムセグメンテーションの設定	110
インデックスの移動	111

インデックスデータに別のパーティションを使用	112
Splunk からインデックスされたデータを削除	113
アラートの定義	116
アラートのしくみ	116
savedsearches.conf によるアラートの設定	117
スクリプトアラートの設定	120
SNMP トラップを他のシステムに送信	121
高度な条件付アラート	124
バックアップと保存方針の設定	126
バックアップの対象	126
必要な空き容量	126
インデックスされたデータのバックアップ	127
設定情報のバックアップ	130
ローテーションとアーカイブポリシーの設定	130
アーカイブの自動化	131
アーカイブデータの修復	133
データセキュリティの設定	135
Splunk で何が保護できるか	135
HTTPS で Splunk に安全にアクセスする	136
SSL で Splunk サーバーに安全にアクセスする	136
検索ピアに対する証明書の配布	136
ファイルシステムに対する変更の監視	137
アーカイブ署名の設定	141
暗号署名監査イベント	143
Splunk 動作の監査	144
フォワーディングと受信の設定	147
フォワーディングと受信について	147
受信の設定	150
フォワーディングの設定	151
フォワーダと受信ホストとの間で SSL 暗号化を使用	155
自動ロードバランシングの設定	157
ラウンドロビン・データバランシングの設定	162
コンテンツに基づいて異なる場所にデータをルーティング	162
イベントの特定キューへのルーティング	165
イベントを特定のインデックスにルーティング	167
データをサードパーティシステムにルーティング	169
クローニングされたデータを複数の受信ホストに転送	171

syslog または HTTP フォーマットでの転送	172
他の Splunk インスタンスへの展開	174
デプロイメントサーバーについて	174
デプロイメントの計画	174
サーバークラスの定義	175
デプロイメントクライアントの設定	178
マルチテナント環境への展開	180
Apps および設定の展開または更新	181
分散検索の設定	183
分散検索とは何か？	183
分散検索の設定	183
特定の Splunk サーバを分散検索から除外する	187
検索ジョブの管理	188
ジョブおよびジョブ管理について	188
Splunk Web でのジョブの管理	188
OS でのジョブを管理する	189
Splunk のコマンドラインインタフェース(CLI)を使用する	191
CLI について	191
CLI でヘルプ情報を得る	192
設定ファイルレファレンス	193
admon.conf	193
alert_actions.conf	194
app.conf	197
audit.conf	200
authentication.conf	203
authorize.conf	208
commands.conf	211
crawl.conf	214
deploymentclient.conf	217
distsearch.conf	220
eventdiscoverer.conf	224
eventtypes.conf	225
fields.conf	227
indexes.conf	229
inputs.conf	234
limits.conf	247
literals.conf	254

macros.conf	255
multikv.conf	257
outputs.conf	261
procmon-filters.conf	272
props.conf	274
pubsub.conf	285
regmon-filters.conf	286
restmap.conf	288
savedsearches.conf	291
searchbnf.conf	296
segmenters.conf	299
server.conf	301
serverclass.conf	307
serverclass.seed.xml.conf	311
source-classifier.conf	313
sourcetypes.conf	314
sysmon.conf	316
tags.conf	317
tenants.conf	319
times.conf	320
transactiontypes.conf	322
transforms.conf	324
user-seed.conf	330
web.conf	331
wmi.conf	337
トラブルシューティング	341
コンタクトサポート	341
サポートに送るデータサンプルの匿名化	343
探しているイベントが見つからない場合	346
SuSE Linux: サーバーから正しくフォーマットされた応答を取得できません	346
サポートの指示により使用するコマンドラインツール	347

はじめに

本マニュアルについて

マニュアルの内容

本書では、**Splunk administrator** の情報および使用方法を説明します。利用者自身または他のユーザーへのサービスとして Splunk を設定、実行、およびメンテナンスを行う責任者のためのマニュアルです。ユーザーの追加、セキュリティの設定、データバックアップ、および管理業務の方法を説明します。

イベントタイプおよびソースタイプに関する情報はどこですか？

新バージョンの Splunk では、異なるアプローチを採用しました。Splunk の情報知識を扱う担当者のためだけに、別のマニュアルを作成しました。該当する方は、情報知識監督者マニュアルをご覧ください。このマニュアルに関するみなさまからのご意見・ご感想もお待ちしております。

Splunk 検索に関するヘルプを探しているのですが

検索関連事項は、**ユーザーマニュアル**および**検索レファレンスマニュアル**をご覧ください。特に、一般例のリストをお探しの場合には、**検索チートシート**をご覧ください。

Splunk とは

Splunk とは

Splunk は IT 検索エンジンです。

- Splunk は、アプリケーション、サーバー、およびネットワークデバイスからリアルタイムで IT データを検索およびナビゲートする場合にご利用いただけます。
- データソースには、ログ、設定、メッセージ、アラート、スクリプト、コード、メトリクスなどが含まれます。
- Splunk では、Splunk Web を利用して、リアルタイムで全 IT データの検索、ナビゲート、アラート、レポートが可能です。

準備

ブート時間に Splunk を起動する設定

ブート時間に Splunk を起動する設定

Windows では、マシンスタートアップ時に Splunk がデフォルトで起動します。その他のプラットフォームでは、手動設定をしなければいけません。これを無効にする方法に関しては、本題の終わりを参照してください。

Splunk では、システム起動時に Splunk が起動するよう、システム起動設定を更新するユーティリティを提供しています。このユーティリティは、適切な init スクリプトを作成します(または、OS の種類によっては、同等の設定変更を行います)。

ルートで実行：

```
$SPLUNK_HOME/bin/splunk enable boot-start
```

Splunk をルートとして起動しない場合、Splunk を起動するユーザーを `-user` パラメータで指定して渡すことができます。例えば、Splunk が、ユーザーボブとして起動する場合、ルートとして以下のように起動します。

```
$SPLUNK_HOME/bin/splunk enable boot-start -user bob
```

システム開始時に Splunk の実行を停止したい場合は、以下を実行します：

```
$SPLUNK_HOME/bin/splunk disable boot-start
```

`$SPLUNK_HOME/etc/init.d/README`、およびコマンドラインに `help boot-start` を入力すると、更なる情報をご利用いただけます。

マックユーザの皆様へ

Splunk は、`/System/Library/StartupItems` ディレクトリにスクリプト、および環境設定ファイルを自動作成します。このスクリプトはシステム起動時に実行され、システムシャットダウン時に Splunk を自動停止します。

注意： Mac の OS をお使いの場合、必ずルートレベルの権限(または `sudo` の使用)が必要です。`sudo` の使用には管理者アクセスが必要です。

例：

Mac OS でシステム起動時に Splunk の起動を有効にするためには以下を利用します。

CLI のみ

```
./splunk enable boot-start
```

sudo と CLI

```
sudo ./splunk enable boot-start
```

Windows で起動時のブートを無効にする

デフォルトでは、Windows マシン起動時に Splunk が自動的に起動します。Splunk プロセス(SplunkWeb および Splunkd) を Windows サービスマネージャーから手動で起動するよう設定できます。

Splunk Web で Splunk Manager を検索

Splunk Web で Splunk Manager を検索

Splunk Web は、Splunk 操作のほとんどを管理する便利なインタフェース、**管理**を提供しています。**管理**にアクセスするには、Splunk Web の右上隅にあるリンクをご覧ください。

[INSERT GRAPHIC HERE]

管理を使って Splunk を設定および維持する方法について詳しく説明します。

ライセンスのインストール

ライセンスのインストール

初めて Splunk をダウンロードすると、登録するよう求められます。

登録すると、1 日に最大 500MB のインデックスをご利用いただけるエンタープライズ版の評価用ライセンス(60 日間有効)を取得します。このライセンスはダウンロードに付随しています。

注意： Splunk 4.0 の「無料」ライセンスはありません。無料ライセンスは次回のリリースでご利用いただける予定です。

エンタープライズライセンスでは、以下の機能をご利用いただけます：

- 複数ユーザーアカウントおよびアクセス制御
- 分散検索およびデータルーティング
- デプロイメント管理

重要： 同じエンタープライズライセンスは複数のサーバーで使用できません。Splunk のライセンス(無料またはエンタープライズ)に関係なく、各インスタンスごと(フォワーダーを含む)に固有のライセンスが必要です。これには、複数のフォワーディングインスタンスにインストール可能な、1 日 1MB のフォワードオンリーライセンスは例外です。

ライセンス版へのアクセス

無料ライセンス(`splunk-free.license`)またはエンタープライズライセンス(`splunk.license`)に関係なく、全ての Splunk サーバーのライセンスが、`$SPLUNK_HOME/etc/`に保存されます。

Splunk ライセンスの例

```
user@company.com;EQ/GQXW/J7u9VLJShPsW4m8yi+5a+geRrof4Bep70j32xsBpq  
JIItM5pdntRfl4auply366BAjTMnfTB6JyzJOZLplyBQijk02fQjgKjakl0ol4N5G6Wr  
09ufnSe3iOXVAay24hzFfgDkaijOnkoGOPJqnHaVzaWC9dxIuKUvDpt3UcKtkDv0Gka  
Q4EZxAvZKAFImvOF4PmDoNaMiBgLLkWikGhezFTTDh10PLl9kyeVThGzAyN23J512pVM  
3xqNIg3pFcd2aJf31xspt1HRdSwofkfnuCVpzildy3qMbae4g85KpCfND+aJ6z2LoUu3  
RQ40V4SpxMXEZ4PgSGZ6dwA==
```

新しいライセンスはどこですか？

新しいライセンスを要求すると、Splunk から e メールでライセンスを取得できます。また、その新しいライセンスは、splunk.com の My Orders(注文)ページからアクセスできます。新しいライセンスをインストールする(または既存のライセンスの変更および更新を行う)には、既存のライセンスを新しいライセンスに切り替えます。

Splunk Web の **管理 > ライセンス** ページまたは、CLI でライセンスのインストールおよび更新が可能です。

Splunk Web からインストール

Splunk Web を使用してライセンスのインストールまたは更新を行うには、

1. Splunk を起動し、サポートされているブラウザで Splunk Web を開きます。
2. ダッシュボードの右上隅にある **管理** をクリックします
3. **ライセンス** をクリックします。

License & Usage ページは、ライセンスレベル、ピーク使用量、およびライセンス違反を表示します。

1. ライセンスの変更をクリックします。
ライセンスの変更ページが開き、既存のライセンスキー、または splunk.license ファイルを表示します。
2. 新しいライセンスキーをコピーして、既存のライセンスに貼り付け(上書き)します。
3. 保存をクリックします。
4. Splunk サーバーを再起動して新しいライセンスを適用します。

注意 : Splunk Web からサーバーを再起動できます。**管理 > サーバーコントロール** ページで、**Splunk 再起動** をクリックします。

CLI からインストール

CLI を使用してライセンスをインストールまたは更新するには、

1. splunk.license というファイル名でファイルを作成します。
2. 新しいライセンスキーをコピーして、splunk.license ファイル内に貼り付けます。
3. ライセンスファイル splunk.license を、\$SPLUNK_HOME/etc/ディレクトリに移動します。

```
mv splunk.license $SPLUNK_HOME/etc/
```

注意 : このディレクトリに既に splunk.license ファイルが存在する場合は、mv により実行の確認なしにファイルを上書き

します。このとき、無料ライセンス splunk-free.license は上書きされません。ただし、splunk.license が存在する場合、デフォルトで Splunk は無料ライセンスを無視します。

1. Splunk サーバーを再起動して、新しいライセンスを適用します。

```
$SPLUNK_HOME/bin/splunk restart
```

新しい評価用またはエンタープライズライセンスを適用した後の最初のログイン

新しい評価ライセンスまたはエンタープライズライセンスを適用した後の最初のログインでは、デフォルトユーザー名「admin」、およびパスワード「changeme」を使用します。後でユーザーデータを消去(リセット)すると、ユーザー名とパスワードはこのデフォルトにリセットされます。

ライセンス違反

ライセンスで許可されるインデックス最大容量を超えると違反となります。暦上の 1 日でライセンスで許可される 1 日の許可容量を超えると、違反警告を受けます。そのメッセージは 14 日間持続します。30 日周期で違反数が 7 を超えると、検索ができなくなります。それ以前の 30 日間の違反数が 7 つ以下、またはより大きな容量制限を持つ新しいライセンスを適用した場合に、検索能力が再びできるようになります。

注意：ライセンス違反期間中でも Splunk はデータのインデックスを停止しません。ライセンスの許容量を超えた場合にアクセスをブロックするのみです。

デフォルト値の変更

デフォルト値の変更

環境に合わせて Splunk の設定を始める前に、以下のデフォルト設定の中に変更する項目がないかご確認ください。

デフォルトの管理者パスワードの変更

エンタープライズ版 Splunk には、デフォルトの管理者アカウントとパスワードがあります。Splunk では、そのデフォルト設定を変更することを強くお勧めします。これは、Splunk の CLI または Splunk Web で変更可能です。

Splunk Web の場合

- Splunk Web に admin ユーザーとしてログインします。
- 画面右上の**管理**をクリックします。
- **ユーザー**をクリックします
- **管理者ユーザー**をクリックします
- パスワードを更新して、**保存**をクリックします。

Splunk CLI の場合

次の Splunk CLI コマンドを使います。

```
# splunk edit user
```

注意：変更の前に既存のパスワードで認証を行う必要があります。CLI から Splunk にログインする、または `-auth` パラメータを使用してください。

例：

```
# splunk edit user admin -password foo -roles administrator -auth  
admin:changeme
```

このコマンドで、管理者パスワードを、*changeme* から *foo* に変更します。

ネットワークポートの変更

Splunk は 2 つのポートを使用します。次のポートがデフォルト設定されています。

- 8000 – Splunk Web 用 HTTP または HTTPS ソケット。
- 8089 – Splunkd 管理ポート。 *splunkd* デーモンとの通信に使用します。 Splunk Web は、コマンドラインインタフェースやその他のサーバーからの分散接続と同様にこのポートで *splunkd* と通信します。

注意：インストール時にこれらのポートを変更できます。

Splunk Web の場合

- admin ユーザーで Splunk Web にログインします。
- 画面右上の**管理**をクリックします。
- システムコンフィギュレーションタブをクリックします。
- **システム設定**をクリックします。
- **ウェブポート**の値を変更し、**保存**をクリックします。

Splunk CLI の場合

Splunk CLI でポート設定を変更するには、CLI コマンド `set` を使用します。

```
# splunk set web-port 9000
```

このコマンドは、Splunk Web ポートを 9000 に設定します。

```
# splunk set splunkd-port 9089
```

このコマンドは、splunkd ポートを 9089 に設定します。

デフォルトの Splunk サーバー名の変更

Splunk サーバー名の設定は、Splunk Web に表示される名前、および分散設定でその他の Splunk Servers に送信した名前の両方を制御します。

デフォルト名は、Splunk サーバーホストの DNS または IP アドレスから取得しています。

Splunk Web の場合

- admin ユーザーで Splunk Web にログインします。
- 画面右上の**管理**をクリックします。
- システムコンフィギュレーションタブをクリックします。
- **システム設定**をクリックします。
- Splunk サーバー名の値を変更し、**保存**をクリックします。

Splunk CLI の場合

CLI でサーバー名を変更するには、以下を入力します。

```
# splunk set servername foo
```

このコマンドは、サーバー名を foo に設定します。

データ保存位置の変更

データ保存は、Splunk サーバーが全てのインデックスデータ、ユーザアカウント、作業ファイルを保存するトップレベルディレクトリです。

注意：このディレクトリを変更すると、サーバーは、古いデータ保存ファイルを移動しません。代わりに、新しい位置で新たに始めます。

データを他のディレクトリに移動するには、インデックスの移動にある説明に従ってください。

Splunk Web の場合

- admin ユーザーで Splunk Web にログインします。
- 画面右上の**管理**をクリックします。
- システムコンフィギュレーションタブをクリックします。
- **システム設定**をクリックします。
- **インデックスのパス**のパスを変更し、**保存**をクリックします。

Splunk CLI の場合

CLI でサーバー名を変更するには、以下を入力します。

```
# splunk set datastore-dir /var/splunk/
```

このコマンドは、データ保存ディレクトリを、/var/splunk/に設定します。

ディスクの最低空き容量の設定

ディスクの最低空き容量設定は、Splunk でインデックスを停止する前にデータ保存位置のディスク空き容量がどこまで少なくなるかを制御します。

空き容量が大きくなると、Splunk はインデックスを再開します。

Splunk Web の場合

- admin ユーザーで Splunk Web にログインします。
- 画面右上の**管理**をクリックします。
- システムコンフィギュレーションタブをクリックします。
- **システム設定**をクリックします。
- インデックスを一時停止する空き容量の下限 の値を変更し、**保存**をクリックします。

Splunk CLI の場合

CLI でサーバー名を変更するには、以下を入力します。

```
# splunk set minfreemb 2000
```

このコマンドは、最低空き容量を 2000MB に設定します。

Splunk の起動

Windows で Splunk を起動

Windows で Splunk を起動

Windows では、Splunk は、デフォルトで、C:\Program Files\Splunk にインストールされます。Splunk 関連の文書に記される多くの例では、\$SPLUNK_HOME を Splunk 本体またはホームディレクトリとして示しています。Splunk をデフォルトディレクトリにインストールする場合は、文字列 \$SPLUNK_HOME を C:\Program Files\Splunk に置き換えることができます。

Windows サービスマネージャから以下の Splunk プロセスを開始・停止します。

- サーバーデーモン : splunkd
- ウェブインタフェース : splunkweb

また、開始または停止、または両方のプロセスを同時に実行するには、\Program Files\Splunk\bin に移動して、以下を入力します。

```
# splunk [start|stop|restart]
```

UNIX で Splunk を起動

UNIX で Splunk を起動

本章では Splunk の起動について簡単な説明をします。Splunk を初めてお使いになる方は、ユーザーマニュアルを先に読みください。

Splunk の起動

Splunk サーバーホスト上のシェルプロンプトから、次のコマンドを実行します。

```
# splunk start
```

これにより、splunkd(インデックスマシンおよびその他のバックエンド処理)および splunkweb (Splunk Web のインタフェース)の両方を起動します。個別に起動するには、以下を実行します。

または、

```
# splunk start splunkd
```

注意 : startwebserver が、web.conf に設定されている場合は、手動で Splunkweb を起動すると、その設定が無効になりません。環境設定ファイルで無効になっていると、起動しません。

Splunk(splunkd または splunkweb)を再起動するには、以下を入力します。

```
# splunk restart
```

```
# splunk restart splunkd
# splunk restart splunkweb
```

Splunk の停止

Splunk をシャットダウンするには、以下のコマンドを実行します。

```
# splunk stop
```

splunkd および Splunk Web を個別に停止するには、以下を入力します。

```
# splunk stop splunkd
```

または、

```
# splunk stop splunk web
```

Splunk の実行状態を確認する

Splunk が実行中かどうかを確認するには、サーバーホストのシェルプロンプトで以下のコマンドを入力します。

```
# splunk status
```

次の出力が表示されます。

```
splunkd is running (PID: 3162).
splunk helpers are running (PIDs: 3164).
splunkweb is running (PID: 3216).
```

または、**ps** を使用して、実行中の Splunk プロセスを確認します。

```
# ps aux | grep splunk | grep -v grep
```

Solaris ユーザーは、**aux** の代わりに **-ef** と入力します。

```
# ps -ef | grep splunk | grep -v grep
```

Splunk Web の起動

Splunk Web の起動

以下に移動します。

```
http://mysplunkhost:8000
```

インストールで選択したホストおよびポートを使用します。

エンタープライズライセンスで Splunk に初めてログインするときには、ユーザー名(*admin*)およびパスワード(*changeme*)を使用します。無料ライセンスの Splunk には、アクセス制御がありません。

Splunk Web および Splunk Apps について

Splunk Web とは

Splunk Web とは

Splunk Web は、Splunk のダイナミックでインタラクティブなグラフィカル・ユーザ・インタフェース (GUI) です。Web ブラウザを使用してアクセスする Splunk Web は、問題調査、結果報告、および 1 つまたは複数の Splunk デプロイメントの管理に使用する主要なインタフェースです。サポートされているオペレーティングシステムおよびブラウザの一覧は、動作環境をご覧ください。

Splunk Web を起動するには、以下に移動します。

```
http://<mysplunkhost>:8000
```

インストールで選択したホストとポートを使用します。デフォルトポートは 8000 です。ただし、このポートを既に使用している場合には、インストーラが別のポートを選ぶよう指示します。

エンタープライズライセンスで Splunk に初めてログインするときには、ユーザ名 (*admin*) およびパスワード (*changeme*) を使用します。無料ライセンスの Splunk では、アクセス制御または、複数ユーザーアカウントをサポートしていません。

Launcher

初めて Splunk Web を起動すると、**Launcher** が表示されます。ここでユーザーは現在利用可能なアプリケーションのリストから 1 つの **App** を選択できます。特に、「Getting Started」App を試す場合、実行している OS によって、Windows または UNIX 専用の App が表示されます。また、Splunk App Store to では、さらに別の App をブラウズおよびダウンロードすることができます。

Apps について詳しく説明します。

App とは？

App とは？

Splunk App は、1 つまたは複数のイベントタイプに対する定義、検索、および保存済み検索を集めた単純なものから、Splunk の外観を完全に再設定する新しいビューやダッシュボードを含むものなど、さらには、Splunk の REST API を使用した完全に新しいプログラムのような複雑なものまでさまざまです。

Splunk を使用することは、常にアプリケーションを使用することです。その状態を通常、App「使用中」と呼びます。

App の用途は

App を使うと、1 つの Splunk インスタンス上に異なる環境を構築できます。組織の中で、異なる Splunk ユーザーのコミュニティ用インタフェース、例えば、E メールサーバーのトラブルシューティング用、Web 分析用などが作成できます。こうすることにより、すべてのユーザーが同じ Splunk インスタンスを使用しながら、各ユーザーの関心のある分野に関

係したデータのみを閲覧できます。

App の種類

初めて Splunk をインストールしてログインすると、App Launcher が表示されます。この画面は、プレインストールされているアプリケーションの一覧を表示します。デフォルトのアプリケーションの 1 つは、Getting Started App です。このアプリケーションは新規ユーザーに Splunk の機能を紹介するために開発されています。Splunk 初心者の方は、ぜひこのアプリケーションをご利用ください。また、ご意見ご感想などございましたらお寄せください。



Launcher を迂回する

Splunk へのログインで Launcher を表示したくない場合は、各ユーザー別にデフォルトアプリケーションを起動するように設定を変更できます

- 以下のユーザー用ローカルディレクトリに `user-prefs.conf` という名前のファイルを作成します。

```
etc/users/<user>/user-prefs/local/user-prefs.conf
```

- `user-prefs.conf` ファイルに以下のラインを入力します。

```
default_namespace = search
```

例：

- admin ユーザーのファイルは以下のとおりです。

```
etc/users/admin/user-prefs/local/user-prefs.conf
```

- test ユーザーのファイルは以下のとおりです。

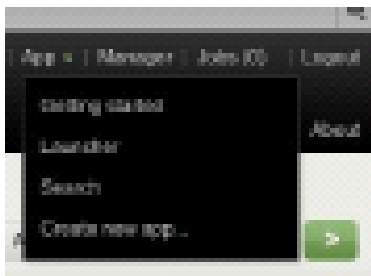
etc/users/test/user-prefs/local/user-prefs.conf

その他のデフォルト設定について

Splunk には、デフォルトで検索アプリケーション(Search App)および OS をサポートするアプリケーションが装備されています。

- Search App は、Splunk のコア機能を提供するインターフェースで、多目的使用できるよう設計されています。これまでに Splunk をお使いいただいた方は、Search App 以前のバージョンの Splunk Web のメイン機能と同じとお考えください。Search App では、検索バーとグラフを多く使うダッシュボードが表示されます。Search App を使用中、ウィンドウ左上の **Dashboard** および **Views** ドロップダウンメニューから新しく選択すると、ダッシュボードまたはビューを変更できます。
- OS 用の App(Windows 用 Splunk または *NIX 用 Splunk)は、ご使用のプラットフォームで Splunk を最大限活用できるようにするダッシュボードおよびプリビルド検索を提供します。これらは、デフォルトで無効にされていますが、Splunk Manager の Apps セクションから有効にすることができます。

使用する App を変更する場合には、左上の App ドロップダウンメニューから新しいものを選択します。



Launcher に戻ってから別の App を選択することもできます。

他の App を入手

Launcher または Apps メニューの App 一覧に、別の App を追加できます。例えば、データ操作作業の大部分が、管理や PCI コンプライアンスの変更などに関連する業務を伴う場合、Splunk には、専用の App があります。

ダウンロード可能な App をを見つけるには、Launcher の **Browse More Apps** タブをクリックします。

App 関連の Splunk knowledge の保存と共有

Splunk ナレッジは、保存済み検索、イベントタイプ、タグなどの Splunk データの品質を高め、必要な情報の検索を簡単にするものです。Splunk では、これらのナレッジ項目はオブジェクトとも呼ばれています。

Splunk Web にログインしたすべてのユーザーが、使用している App の本人のユーザーディレクトリにこれらのオブジェクトを作成および保存できます(ユーザーに正式な権限があることが前提)。

ユーザーに正式な権限があることを前提に、ユーザーが App 用にオブジェクトを保存すると、以下のいずれかの処理を

しない限りそのファイルはそのユーザーが App を使用している間のみ利用可能です。

- 同じ App で、オブジェクトを他の特定の役割またはユーザと共有する
- 該当する App のアクセス権があるすべてのユーザーが利用できるようオブジェクトを利用可能にする
- すべての App(およびユーザー)が利用できるようにオブジェクトを利用可能にする

本書の App アーキテクチャおよびオブジェクト所有権についてお読みください。

App の入手先

App の入手先

新しい App は Aplunk.com から入手できます。また、Splunk.com で入手可能なすべての App は Launcher に表示されるため、Splunk から直接 App をダウンロードおよびインストールできます。

Splunk Web にログインすると、App Launcher がデフォルトで表示されます。これが変更されている場合は、Splunk が提供するすべての App のメインページの右上にある App メニューから Launcher に戻ることができます。

インターネットに接続している場合

Splunk サーバーまたはクライアントマシンが、インターネットに接続されている場合は、Splunk の Launcher から App を直接ダウンロードできます。

1. Launcher の **Browse Apps** タブをクリックします。Splunk App Store に接続し、このバージョンの Splunk で利用可能な他の App を検索およびダウンロードできます。
2. 希望する App を選び、**Install App** を選択します。手動でインストールをする場合には、**Download App** を選択します。
3. Splunk.com のユーザー名およびパスワードでログインするよう促されます(これは、Splunk のユーザー名およびパスワードではありません)。
4. これで App のインストールが完了します。App に Web GUI コンポーネントが含まれている場合(App によってはイベントタイプ定義のようなナレッジオブジェクトのみで構成されているものがある)、Launcher から App を誘導できます。

インターネットに接続していない場合

Splunk サーバーやクライアントにインターネット接続がない場合には、SplunkBase から App をダウンロードして、サーバーにコピーする必要があります。

1. インターネットに接続されているコンピュータを使い、Splunk App Store で希望する App を検索します。
2. App をダウンロードします。
3. この App を Splunk サーバーにコピーします。
4. App を、`$SPLUNK_HOME / etc / apps` ディレクトリに挿入します。
5. App を解凍します(-xvf 解凍を実行する、または tar および ungzp ファイルを解凍するツールを使用する)。Splunk App は、tar および gzip で圧縮されていますが、.SPL の拡張子です。ツールが強制的にこの拡張子を認識するよう設定しなければいけない場合があります。

6. App の内容によっては、Splunk を再起動させる必要があります。
7. これで App はインストールが完了し、Launcher から利用できます(Web GUI コンポーネントを含む場合)。

App アーキテクチャとオブジェクトの所有権

Splunk ナレッジは、通常 Splunk ナレッジで構成されています。Splunk ナレッジとは、保存済み検索、イベントタイプ、タグなど、Splunk データの品質を高め、必要な情報の検索を簡単にするためのものです。Splunk では、これらのナレッジ項目をオブジェクトとも呼んでいます。

Splunk Web にログインしたすべてのユーザーが、使用している App のユーザーディレクトリにこれらのオブジェクトを作成および保存できます(ユーザーに正式な権限があることが前提)。これはデフォルトの動作で、ユーザーがオブジェクトを保存すると常にその App のユーザーディレクトリを使用します。ユーザーディレクトリは、`$Splunk_HOME/etc/users/<user_name>/<App_name>/local` にあります。ユーザーに正式な権限があることを前提に、ユーザーが特定の App 用にオブジェクトを保存すると、以下のいずれかの処理をしない限り、保存されたオブジェクトは、そのユーザーが該当 App を使用している間のみ利用可能です。

- 該当 App へのアクセス権があるすべてのユーザーが利用できるようなオブジェクトを利用可能にする
- オブジェクトを特定の役割またはユーザーに限定する(該当 App のコンテキスト範囲内)
- すべての App(およびユーザー)が利用できるオブジェクトを指定する(役割またはユーザーで制限する場合は除く)

Splunk ナレッジの利用と共有

ユーザーは、Permissions ダイアログを使って他のユーザーと Splunk ナレッジオブジェクトを共有できます。つまり、App の読み込み権限を持つユーザーは、共有オブジェクトの閲覧と使用が可能になります。例えば、ユーザーが保存済み検索を共有する場合、他のユーザーは、その検索を実行した App を使用する場合に限り保存済み検索を閲覧できます。つまり、App Fflanda で保存済み検索を作成して共有した場合、App Fflanda を使う他のユーザーで読み込み権限を持つユーザーが、その保存済み検索を閲覧できます。

一部のユーザーには、オブジェクトを App レベルで利用可能にする権限があります。つまり、オブジェクトは、そのユーザーのディレクトリからその App ディレクトリにコピーされます。

コピー元

```
$SPLUNK_HOME/etc/users/<user_name>/<App_name>/local/
```

コピー先

```
$SPLUNK_HOME/etc/apps/<App_name>/local/
```

この操作は、App の書き込み権限を持つユーザーのみが実行可能です。

Splunk ナレッジを誰でも使えるようにする

最終的に、ユーザーはオブジェクトをグローバルな利用、つまりすべての App で閲覧可能にするかどうかを決定できます。この場合、ユーザーには元の App に書き込みできる権限が必要です。

適用されるオブジェクト

この場合のオブジェクトは、アクセス制御に関わるオブジェクトに制限されます。これらのオブジェクトは、App レベルオブジェクトとも呼ばれ、Splunk Manager の **App Configuration** タブで設定できます。このページは、作成および共有したオブジェクトを管理するすべてのユーザが利用可能です。

含まれるオブジェクト

- 保存済み検索およびレポート
- イベントタイプ
- ビューおよびダッシュボード
- フィールドの抽出

システムレベルのオブジェクトは、**管理**で管理します。これは、管理者権限(または、オブジェクトに対して読み込み/書き込み権限のあるユーザー)のあるユーザーが行えます。

含まれるオブジェクト

- ユーザ
- 役割
- 認証
- 分散検索
- 入力
- 出力
- デプロイメント
- ライセンス
- サーバー設定(例：ホスト名、ポート等)

重要：入力を追加した場合、Splunk はその時使用している App に属する `inputs.conf` のコピーにその入力を追加します。つまり、Splunk Manager にナビゲートした場合、入力は、Launcher から直接、以下に追加されます。

```
$SPLUNK_HOME/etc/apps/launcher/local/inputs.conf.
```

App の設定とナレッジの優先順位

Splunk にナレッジを追加すると、その時に使用している App のコンテキストに追加されます。Splunk が設定およびナレッジを評価するとき、特定の優先順位に従って評価を行い、コンテキストに対して使用するナレッジ定義および設定を制御できます。Splunk が使用する設定ファイルおよび優先順位についての詳しい情報は、「設定ファイルについて」をご覧ください。

App オブジェクトの管理

App オブジェクトの管理

Splunk ユーザーが App を作成すると、その App を構成するオブジェクトのコレクションが作成されます。これらのオブ

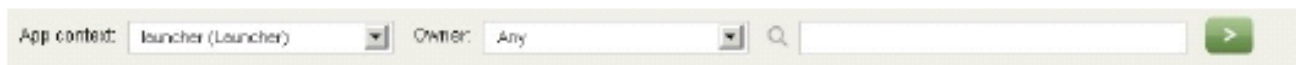
ジェクトには、ビュー、コマンド、ナビゲーション項目、イベントタイプ、保存済み検索、レポートなどを含みます。各オブジェクトには、閲覧および変更可能なユーザーを決定する許可があります。デフォルトでは、管理者ユーザーに、Splunk システムのすべてのオブジェクトを変更できる権限が与えられています。

- App の概要は、本書の「App とは」をご覧ください。
- App 権限についての詳しい情報は、本書の「App アーキテクチャとオブジェクトの所有権」をご覧ください。
- 独自の App を作成する方法については、デベロッパーマニュアルをご覧ください。

Manager による App オブジェクトの管理

システムにあるすべての App オブジェクトを閲覧および制御するには、Splunk Web の Splunk Manager を使用します。

- **Sorting arrows** を使用して、すべてのページのオブジェクトを閲覧および操作します。
- ビューをフィルタリングして、特定のユーザーが所有する App、または **App コンテキストバー** を使用して特定の文字列を含む特定の App のオブジェクトのみを閲覧します。



App オブジェクトの閲覧

Manager を使用して、Splunk デプロイメントで App オブジェクトを閲覧するには以下の方法を使います。

- システム上のすべての App のすべてのオブジェクトを一度に閲覧する：**管理 > すべての設定**
- すべての保存済み検索およびレポートオブジェクトを閲覧する：**管理 > 保存済み検索とレポート**
- すべてのイベントタイプを閲覧する：**管理 > イベントタイプ**
- すべてのフィールド抽出を閲覧する：**管理 > フィールド抽出**
- すべての Python 検索コマンドスクリプトを閲覧する：**管理 > 検索コマンド**

注意：検索コマンドページの各種検索コマンドについての情報は、検索リファレンスをご覧ください。

設定の前に

設定方法

設定方法

Splunk は以下の方法で設定できます。

- 設定ファイルを編集する
- Splunk Web の Splunk Manager を使用する
- Splunk CLI を使用する

この 3 つの方法は、以下に説明する設定ファイルの内容を最終的に変更します。

設定ファイル

Splunk の設定情報の多くは、.conf ファイルに保存されています。これらのファイルは、`/etc/system` の下にある Splunk インストールディレクトリ(通常、説明書では`$SPLUNK_HOME`を指す)に保管されています。標準のテキストエディタを使用して、これらのファイルを変更できます。設定ファイルの編集を始める前に、「設定ファイルについて」と呼ばれるトピックの内容をお読みください。

Splunk Manager

通常の設定タスクの多くは、Splunk に装備されている Web UI、Splunk Web の Splunk Manager から行うことができます。Splunk Web は、デフォルトで、インストールされたホストのポート 8000 上で実行します。

- ローカルマシンで Splunk を実行している場合、Splunk Web にアクセスする URL は、`http://localhost:8000`
- 別のマシンで Splunk を実行している場合、Splunk Web にアクセスする URL は、`http://<hostname>:8000`

`<hostname>`には Splunk を実行しているマシンの名前を入れます。

Splunk Manager にアクセスするには、Splunk Web にログインして、右上端の**管理**をクリックします。

Splunk CLI

CLI を使って多くの設定オプションが利用できます。これらのオプションについては、説明書の各トピックを参照してください。また、`help` コマンドを使用して、完全な CLI ヘルプレファレンスを入手できます。Splunk 実行中に、以下をコマンドラインに入力して、デフォルトの CLI ヘルプページにアクセスします。

```
./splunk help
```

CLI についての詳細は、本書の「CLI について」をご覧ください。

設定変更後の再起動

多くの設定ファイルの変更には、Splunk の再起動が必要です。設定ファイルおよび説明書の参照トピックを見て、変更 Splunk の再起動が必要かどうか確認してください。

Manager で変更すると、再起動が必要かどうかお知らせします。

以下の変更は、有効になる前に追加または異なるアクションが必要です。

- ◆ Transforms.conf への設定変更を有効にするには、Splunk Web で以下の検索を入力します。

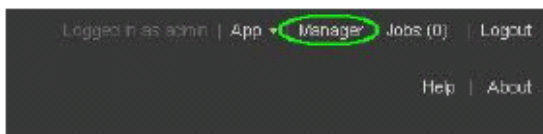
```
|extract reload= ]
```

- Splunk Web の **管理** > **認証セクション** から authentication.conf をバウンスします。

Splunk Manager について

Splunk Manager について

Splunk Web で Splunk を設定するオプションは Splunk Manager に装備されています。Splunk Manager にアクセスするには、Splunk Web にログインして、右上の **管理** をクリックします。



管理者権限を持つユーザーは、Manager の全エリアにアクセスできます。

システム設定

システム設定のエリアでは、以下を管理します。

- システム設定：*Splunk Web ポート、ホスト名、インデックスパスを含む Splunk インスタンスに関する一般設定を管理します。
- サーバー管理：Splunk を再起動します。
- ライセンス：ライセンスの使用統計を閲覧し、新しいライセンスを適用します。
- 分散検索：複数の Splunk インスタンスを通して分散検索を設定します。
- システムロギング：Splunk のログ出力の調整をします。
- メールアラートの設定：メールサーバーやフォーマットなどのメールアラートの設定を指定します。
- インデックス：新しいインデックスを作成し、インデックスサイズの設定管理をします。
- データ入力：スクリプト、ファイル、ディレクトリ、およびネットワークポートから Splunk へデータを追加します。
- 転送と受信：データを送信、受信するためにホストを設定します。
- 認証方法：認証方法を指定します(Splunk、または LDAP)。
- ユーザー：パスワード、メールアドレスなどのユーザ設定を管理します。
- 役割：役割の管理、各種権限設定や検索時の制限の設定をします。

App とナレッジ

- App：インストール済みの App の権限を編集します。新しい App の作成、またはコミュニティが作成した App の Splunkbase を閲覧します。

- 検索とレポート：検索およびレポートの権限を閲覧、編集、設定します。アラートおよびサマリーインデックスを設定します。
- イベントタイプ：イベントタイプの権限を閲覧、編集、設定します。
- フィールドの抽出：フィールドの抽出の権限を閲覧、編集、設定します。
- ビュー：App 用の UI ページを作成および編集します。
- ナビゲーションメニュー：App 用のナビゲーションメニューを作成および編集します。
- 検索コマンド：python ベースの検索コマンドの権限を設定します。
- 全ての設定：全ての App の設定を閲覧します。

設定ファイルについて

設定ファイルについて

Splunk の設定ファイルの情報は、.conf ファイルに保存されています。これらのファイルは、/etc/system の下にある Splunk インストールディレクトリ(通常、説明書では \$SPLUNK_HOME として明記)にあります。以下は、\$SPLUNK_HOME/etc/system の下に存在する関連ディレクトリ構造です。

- \$SPLUNK_HOME/etc/system/default
 - ◆ あらかじめ設定された設定ファイルを含みます。デフォルトのファイルは変更しないでください。
- \$SPLUNK_HOME/etc/system/local
 - ◆ default/ の設定の上書きを含む、すべてのカスタム編集を行います。
- \$SPLUNK_HOME/etc/system/README
 - ◆ サポート資料。このディレクトリには、ユーザー作成の設定ファイルを作成するときに参照するサンプルおよび仕様設定ファイルを含みます。ほとんどの設定ファイル用に、.spec と .example の 2 つのリファレンスファイルがあります。例えば inputs.conf.spec や inputs.conf.example. です。.spec ファイルは、属性および変数が利用可能な構文の仕様です。.example ファイルは、実際の使用に役に立つ例です。これらのファイルはすべて、\$SPLUNK_HOME/etc/system/README ディレクトリに保管されています。
- \$SPLUNK_HOME/etc/system/spec

注意： 特定の設定ファイルで利用できる設定の最も正確で最新のリストは、その設定ファイルの .spec ファイルにあります。本書の「設定ファイルリファレンス」または、\$SPLUNK_HOME/etc/system/README に、最新の .spec および .example ファイルがあります。

UTF-8 が利用されない OS で設定ファイルを作成および編集

Splunk の設定ファイルは通常 ASCII/UTF-8 にあります。UTF-8 が利用されない OS で設定ファイルを作成および編集する場合、使用しているエディタが、ASCII/UTF-8 で保存するように設定する必要があります。

設定ファイルの優先順位

Splunk 設定ファイルは、それぞれが折り重なっています。同じ設定ファイル、およびそのファイル内のスタンザ(異なる設定値を持つ)のマルチプルインスタンスは、デプロイメントに存在します。Splunk は、ディレクトリ構造内の場所をも

とに、使用するファイルおよびスタンザを決定します。

設定ファイルは以下の順序で評価されます。

- ローカル：ローカルの変更および設定が最初に評価されます
- ユーザー作成ディレクトリ：アルファベット順に評価されます。
- デフォルト：Splunk のデフォルト設定が最後に評価されます。

優先順位

ディレクトリは以下の順序で評価されます。

```
$SPLUNK_HOME/etc/users/*
$SPLUNK_HOME/etc/system/local/*
...
$SPLUNK_HOME/etc/system/local/*
$SPLUNK_HOME/etc/apps/A/local/*
...
$SPLUNK_HOME/etc/apps/Z/local/*
$SPLUNK_HOME/etc/apps/A/default/*
...
$SPLUNK_HOME/etc/apps/Z/default/*
$SPLUNK_HOME/etc/system/default/*
```

以下の点に注意してください。

- /etc/users/は、ユーザーがログインしている検索時間に考慮されます。
- /etc/users/は、バックエンド処理では無視されます。ラインマーキング、日付抽出、およびインデックスタイム、正規化時では、ユーザコンテキストは存在しません。
- /etc/apps/は、特定の App(props、入力、transforms など)のコンテキスト内で考慮されます。
- /etc/system/は、グローバル属性(インデックス、サーバー、権限、認証など)として考慮されます。

数字のついたディレクトリは以下の順序で評価されます。

```
$SPLUNK_HOME/etc/apps/myapp1
$SPLUNK_HOME/etc/apps/myapp10
$SPLUNK_HOME/etc/apps/myapp2
$SPLUNK_HOME/etc/apps/myapp20
...
```

属性優先順位

優先順位は属性別に適用されます。つまり、props.conf ファイルが、/etc/system/default と、/etc/system/local に存在する場合、/etc/system/default 内の props.conf ファイルは、props.conf ファイル全体を単に上書きま

たは交換するものではありません。同じ属性/値のペアが、`/system props.conf` および `/local props.conf` の両方に存在する場合は、ローカル `props.conf` がその属性のみを上書きします。

例えば、`$SPLUNK_HOME/etc/system/local/props.conf` に以下のスタンザが含まれ、

```
[source::/opt/Locke/Logs/error*]
sourcetype = t2rss-error
```

さらに、`$SPLUNK_HOME/etc/apps/t2rss/props.conf` に以下のスタンザが含まれる場合、

```
[source::/opt/Locke/Logs/error*]
SHOULD_LINEMERGE = True
BREAK_ONLY_BEFORE_DATE = True
```

`local` のソースタイプ割り当ておよび、`t2rss` のラインマーキング属性の両方が適用します。ただし、`local` と `t2rss` 両方に、`source::/opt/Locke/Logs/error*` のソースタイプ割り当てがある場合、`local` の割り当ては、`t2rss` を上書きします。

同じターゲットに影響するスタンザのセット内の優先順位

2 つ以上のスタンザが同じ項目に影響を与える動作を指定すると、項目は、ASCII の順序で評価されます。例えば、`props.conf` に以下を指定します。

```
[source::.../bar/baz]
attr = val1

[source::.../bar/*]
attr = val2
```

Attr に対する最初のスタンザの値は、2 番目のスタンザのパスの値が ASCII 順位が高く、優先順位が高いため使用されません。

複数の属性が割り当てられているイベントの優先順位

上述の優先順位のルール他に、ホスト、ソース、またはソースタイプ(イベントタイプの場合もある)別に個々のイベントを処理する属性の `props.conf` セットのみに作用する追加の優先順位があります。そのため、1 つのイベントで、ホスト、ソース、ソースタイプの各デフォルトフィールドに同じ属性セットを持つことが可能です。優先順位は以下の通りです。

- ソース
- ホスト
- ソースタイプ

デフォルトの `props.conf` 設定を上書きしたい場合があります。例えば、デフォルトで、`sourcetype = xml_file` という名前の `mylogfile.xml` を追跡しているとします。この設定は、プロパティがソース別に設定されているため、他のソースタイプを手動で指定しても、変更のたびに全てのファイルを再インデックスします。これを上書きするには、ソース別に明白な設定を追加します。

```
[source::/var/log/mylogfile.xml]
```

```
CHECK_METHOD = endpoint_md5
```

設定ファイルの場所

設定ファイルの場所

Splunk をインストールすると、一連のデフォルト設定ファイル(.conf の拡張子)が、`$SPLUNK_HOME/etc/system/default/`に作成されます。各設定ファイルの例および仕様(.spec および.example の拡張子)は、`$SPLUNK_HOME/etc/system/README/`に含まれています。

設定ファイルの使い方については、本書の「設定ファイルについて」をご覧ください。

一部の設定ファイルはデフォルトで作成されたものではありません。それらのファイルが管理する機能を有効にするには、ユーザーが設定ファイルを作成しなければなりません。これらの設定ファイルには、閲覧用に、.spec および.example が付いています。

設定ファイルと App

App を使用中に設定ファイルに書き込まれる変更をした場合、その変更は、その App システムのローカルディレクトリ `$SPLUNK_HOME/etc/apps/<App_name>/local/<configurationfile>.conf` の関連設定ファイルのコピーに書き込まれます。特定の App にのみ変更を適用するよう設定ファイルを編集したい場合、その App のローカルディレクトリにそのファイルのコピーを作成して変更を行います。

設定ファイルのリストとその内容

以下は、各 conf ファイルに関連した利用可能な spec および example ファイルの最新リストです。一部の conf ファイルには、spec または example ファイルがありません。spec または example ファイルのない conf ファイルを編集する前には、サポートに連絡してください。

重要： `$SPLUNK_HOME/etc/system/default/`にある conf ファイルのデフォルトコピーは編集しないでください。`$SPLUNK_HOME/etc/system/local/`のファイルをコピーし、そのコピーを編集してください。Splunk は、設定の検索時にまず `$SPLUNK_HOME/etc/system/local/`を探し、見つけた場合には編集されたファイルを使用します。

ファイル	目的
alert_actions/conf	Splunk のグローバルアラートのアクションをカスタマイズ
app.conf	カスタム App 用のフィールドを設定
audit.conf	監査およびイベントハッシングを設定
authentication.conf	Splunk のビルトイン認証または LDAP の切り替え。LDAP の設定。
authorize.conf	多様なアクセス制御を含む、役割の設定
commands.conf	検索コマンドを、カスタム検索スクリプトに接続
deploymentclient.conf	デプロイメントサーバーのクライアント用の動作を指定
distsearch.conf	分散検索用の動作を指定
eventdiscoverer.conf	type learner (event discovery)で無視する条件を設定

eventtypes.conf	イベントタイプの定義を作成
fields.conf	複数値フィールドの作成およびインデックスフィールド用の検索能力の追加
indexes.conf	インデックス設定の管理と設定
inputs.conf	データ入力の設定
limits.conf	検索コマンド用の様々な制限(最大結果サイズなど)を設定
literals.conf	Splunk Web に表示されるテキストのカスタマイズ
macros.conf	検索言語マクロの定義
multikv.conf	表形式イベント(ps、netstat、ls)のフィールドルールの設定
outputs.conf	フォワーディング、ルーティング、クローニングおよびデータバランシングの設定
procmon-filters.conf	Windows の処理データの監視
props.conf	タイムゾーンオフセットおよびカスタムソースタイプルールを含む、インデックスプロパティ設定の設定。イベントプロパティ変換のマッピング。
pubsub.conf	デプロイメントサーバーのカスタムクライアントの定義
regmonfilters.conf	ウィンドウズレジストリー監視用のフィルタの作成
restmap.conf	REST エンドポイントの設定
savedsearches.conf	保存済み検索およびそれに関連したスケジュールとアラートの定義
segmenters.conf	インデックスイベント用のセグメンテーションルールのカスタマイズ
server.conf	Splunk のバックエンド用の SSL の有効化および証明書情報の保存場所の指定
serverclass.conf	デプロイメントサーバーで使用するデプロイメントサーバークラスの定義
sourceclassifier.conf	ソースタイプ作成時に無視できる条件(秘密データなど)
sourcetypes.conf	ソースタイプトレーニングで作成されたソースタイプ学習ルールを保存する自動生成ファイル
sysmon.conf	Windows レジストリー監視の設定
tags.conf	フィールド用タグの設定
tenants.conf	マルチテナント環境でのデプロイメントの設定
times.conf	Search App で使用するカスタム時間範囲の定義
transactiontypes.conf	トランザクション検索用のトランザクションタイプの追加
transforms.conf	データ入力で実行するための正規表現変換の設定。props.conf を伴うタンデムで使用
user_seed.conf	デフォルトユーザーおよびパスワードの設定
web.conf	Splunk Web の設定、HTTPs の有効化
wmi.conf	WMI(Windows Management Instrumentation)入力の設定

データの追加と入力の設定

Splunk の監視対象

Splunk の監視対象

Splunk でネットワークまたはローカルソースを監視するために特別なプラグインは必要ありません。Splunk は、あらゆるソースから IT データをリアルタイムにインデックスできます。これを、**ユニバーサルインデキシング**と呼びます。

本当にあらゆるデータソースに対応しています。

柔軟性のあるさまざまな入力方法により、すべてのアプリケーション、サーバー、およびネットワークデバイスのログ、設定、トラップおよびアラート、メッセージ、スクリプト、コードおよびパフォーマンスデータをインデックスできます。ファイルシステムを監視して、スクリプトおよび設定の変更、アーカイブファイルの保存、ライブアプリケーションログの検索および追跡、syslog、SNMP およびその他のネットワークベースツールを受信するためのポートへの接続を行います。

Splunk は、指定するどのデータも取り込みます。データをインデックスする前に、必ずデータソースを入力に追加する必要があります。また、ソースは、Splunk のデフォルトフィールドの 1 つとして(ファイル、ディレクトリ、またはネットワークポートに関係なく)リストアップされます。

重要：入力を追加すると、Splunk は追加時に使用している app に属する `inputs.conf` のコピーに入力を追加します。これは、Splunk Manager を使用する場合、Launcher から直接、入力は、`$SPLUNK_HOME/etc/apps/launcher/local/inputs.conf` に追加されることを意味します。

注意：Splunk は、最後に再起動してから 24 時間ごとに監視をするよう設定されている入力を検索します。これは、まだ存在しないディレクトリまたはファイルを監視するためにスタンプを追加した場合、Splunk がその内容のインデックスを始めるまでに最長で 24 時間かかることを意味しています。入力が即座に認識およびインデックスされるように、Splunk Web を使用する、または CLI の `add` コマンドを使用して入力を追加します。

Splunk にデータを取り込む方法

以下の方法で、データ入力を指定します。

- Splunk Web
- Splunk の CLI
- `inputs.conf` 設定ファイル
- 他のシステムからのフォーワーディング

ほとんどのデータソースは Splunk Web を使用して追加できます。より詳細な設定オプションには、`inputs.conf` を使用します。Splunk Web または Splunk CLI を使用して行った変更は、`$SPLUNK_HOME/etc/system/local/inputs.conf` に書き込まれます。

ソース

Splunk は、さまざまな方法でデータ入力を受け入れます。オプションの基本的な概要は以下のとおりです。

ファイルとディレクトリ

データ入力の多くは、ファイルおよびディレクトリから直接取り込まれます。そのほとんどが、Splunk のファイルとディレクトリ入力プロセッサを使って、ファイルとディレクトリのデータをインデックスできます。

また、Splunk のファイルシステム変更監視を設定して、ファイルシステムの変更を監視できます。ただし、監視とファイルシステム変更監視の両方を使用して同じディレクトリまたはファイルを追跡してはいけません。ディレクトリの変更を見る場合は、ファイルシステム変更監視を使用します。ディレクトリの新しいイベントをインデックスする場合には、監視を使用します。

ファイルとディレクトリを監視するには、「ファイルとディレクトリの監視」をご覧ください。

ファイルシステム変更監視を有効にして設定するには、「ファイルシステムへの変更の監視」をご覧ください。

TCP/ネットワークポート

TCP は、信頼性の高い接続型プロトコルで、UDP の代わりに、可能な限りデータの送受信に使用するものです。エンタープライズライセンス版の Splunk は、あらゆる TCP ポートからデータを受信できるため、Splunk で syslog-ng および TCP 経由で転送するその他のアプリケーションからリモートデータを受信できるようにします。

TCP 経由のデータを監視するには、「ネットワークポートの監視」をご覧ください。

Splunk は、UDP 上の監視をサポートしていますが、可能な限り TCP を使用することをお勧めします。UDP は、以下の理由で、通常の転送には適していません。

- 配信を強制しない
- 暗号化されていない
- 失われたデータグラムの補償がない

UDP を使用する必要がある場合は、必ず Splunk Community Wiki の「UDP 接続で作業する」を参照してください。

Windows ソース

Windows 用の Splunk には、Windows アプリケーションが含まれています。このアプリケーションは、Windows 専用の入力およびコンテンツを提供します。また、Windows 用の Splunk は、以下にリストアップされている Windows 専用の入力タイプを定義するページを Splunk Manager に提供します。以下のリンク先トピックの説明に沿って `inputs.conf` を編集すると、あらゆるプラットフォームで Windows の入力を追加できます。監視できる項目は以下の通りです。

- Windows イベントログデータ
- Windows レジストリーデータ
- WMI データ
- アクティブディレクトリデータ

探しているイベントが見つからない場合

Splunk に入力を追加すると、その入力は、使用中の App に関連付けて追加されます。Splunk に付属する *nix や Windows App のような一部の App は、入力データを特定のインデックスに書き込みます(*Nix や Windows の場合、「os」インデックス)。Splunk にあることが確かなデータを見つけられない場合は、正しいインデックスを探しているかどうか確認してください。使用している役割用のデフォルトインデックスのリストに「os」インデックスを追加してみてください。役割に関する詳細は、本書の役割に関するトピックをご覧ください。

ファイルとディレクトリの監視

ファイルとディレクトリの監視

Splunk をファイルまたはディレクトリに合わせます。ディレクトリを指定した場合、Splunk は、そのディレクトリ内のすべてを取り込みます。Splunk には、2 つの異なるファイル入力プロセッサ、**監視**と**アップロード**があります。ほとんどの場合、**監視**を使用してファイルとディレクトリからすべてのデータソースを入力します。履歴ファイルのアーカイブのアップロードなどのワンタイムのデータ入力を行う場合にのみ、**アップロード**を使用します。詳細は以下をお読みください。

監視

ファイルまたはディレクトリへのパスを指定すると、Splunk の監視プロセッサは、あらゆる新しい入力を取り込みます。このようにして J2EE や .Net アプリケーション、Web アクセスログなどのライブアプリケーションログを監視します。Splunk は、入力があるごとにそのファイルまたはディレクトリのデータにインデックスを付け続けます。また、Splunk サーバーがディレクトリから読み込める限り、ネットワークファイルシステムを含めたマウントまたは共有ディレクトリを指定できます。指定されたディレクトリにサブディレクトリが含まれている場合、Splunk は、新しいファイル用にそれを再帰的に点検します。

Splunk は、Splunk サーバーの起動および再起動で、監視設定に指定したファイルまたはディレクトリを点検します。起動時に、指定したファイルまたはディレクトリが存在しない場合、Splunk は、最後に再起動したときから 24 時間の間隔で再点検します。監視されるディレクトリのサブディレクトリは、継続的にスキャンされます。Splunk を再起動することなく新しい入力を追加するには、Splunk Web または、コマンドラインインタフェースを使用します。Splunk に、自動的に潜在的な新しい入力を検索させたい場合は、クローラを使用します。

監視の注意事項

- ほとんどのファイルシステムで、書き込み中でもファイルを読み込むことができます。ただし、Windows ファイルシステムには、書き込み中にファイルの読み込みを回避する機能があり、Windows のプログラムにはこのモードを使用するものがあります。ほとんどのプログラムはこれを使用しません。
- ファイルまたはディレクトリは、ホワイトリストおよびブラックリストを使って含めたり除外したりできます。
- 再起動すると、Splunk は、前回の続きからファイル処理を継続します。
- Splunk は、インデックスする前にファイルを圧縮します。取り扱い可能なファイルタイプは、.tar、.gz、.bz2、.tar.bz2、.zip です。
- Splunk はログファイルのローテーションを検知し、既にインデックスされている名前を変更されたファイルは

処理しません(.tar および.gz ファイルは例外。詳しい情報は、本書の「ログローテーション」をご覧ください)。

- dir/filename のフルパスは、1024 文字を超えてはいけません。
- ディレクトリ用のソースタイプを Automatic に設定します。ディレクトリに、形式の異なるファイルが複数含まれている場合に、手動でソースタイプの値を設定しないでください。手動でソースタイプを設定すると、そのディレクトリのすべてのファイルに1つのソースタイプを強制します。
- 入力の削除をしても、入力ファイルのインデックスは停止しません。代わりに、その入力の再確認が停止します。ただし、最初の内容すべてがインデックスされます。処理中のデータすべてを停止するには、Splunk サーバーを再起動する必要があります。

注意: 現在、同じディレクトリまたはファイルの追跡に、監視およびファイルシステム変更監視の両方は使用できません。ディレクトリの変更を見る場合は、ファイルシステム変更監視を使用します。ディレクトリの新しいイベントをインデックスする場合には、監視を使用します。

注意: 入力スタンプの監視は重複しません。/a/path/subdir を監視しながら/a/path を監視すると、正しい結果が得られません。同様に、同じディレクトリを異なるホワイトリスト、ブラックリスト、およびワイルドカードコンポーネントで監視する入力スタンプの監視は、サポートされていません。

アップロード

ローカルファイルをアップロード、または Splunk サーバー上にあるファイルをインデックスのオプションを使用して、一回スタティックファイルをインデックスします。ファイルは、継続的に監視されません。

バッチ

バッチ入力タイプを inputs.conf で使用して、一度ファイルをローディングし、削除します。デフォルトで、Splunk のバッチプロセスは、\$SPLUNK_HOME/var/spool/splunk にあります。ファイルをこのディレクトリに移動すると、Splunk はインデックスをしてから、削除します。

注意: ファイルのローディングを適切に実行するには、「サイズの異なるアーカイブをインデックスする方法」をご覧ください。

Splunk Web のファイルとディレクトリを監視する

Splunk Web でファイルとディレクトリによる入力を追加します。

1. Splunk Web の右上端にある**管理**をクリックします。
2. システム設定の下にある**データ入力**をクリックします
3. ファイルとディレクトリをクリックします。
4. **新規**をクリックして、入力を追加します。
5. 以下から希望するラジオボタンを選択します。
 - **ファイルまたはディレクトリをモニタ**は、継続的な入力を設定します。そのファイルまたはディレクトリにデータが追加されるたびに Splunk がインデックスします。
 - **ローカルファイルをアップロード**は、ローカルマシンから Splunk へアップロードします。

- Splunk サーバー上でファイルをインデックスは、バッチディレクトリを通じて Splunk にサーバー上のファイルをコピーします。

6. ファイルまたはディレクトリへのパスを指定します。Upload a local file を選択した場合は、Browse... ボタンを使用します。

共有ネットワークドライブを監視するには、以下を入力します。

<myhost><mypath> (または、Windows 上では¥¥<myhost>¥¥<mypath>)。Splunk に、監視するファイルと同様にマウントドライブにも読み込みアクセスがあるか確認してください。

7. ホストヘディングの下で、ホスト名を選択します。モニタまたはバッチ法を使用している場合、いくつかの選択肢があります。ホスト値の設定について詳しくお読みください。

注意：ホストは、Splunk のホストフィールドのみを設定します。ネットワーク上の特定のホストを監視するよう Splunk に指示しません。

8. ここで、ソースタイプを設定します。ソースタイプは、イベントに追加されたデフォルトフィールドです。ソースタイプは、タイムスタンプやイベント境界などの処理特性を決定するのに使用します。

9. ソース、ホスト、およびソースタイプを指定してから、実行をクリックします。

CLI

Splunk のコマンドラインインタフェース (CLI) からファイルとディレクトリを監視します。Splunk の CLI を使用するには、\$SPLUNK_HOME/bin/ディレクトリに移動し、UNIX または Windows コマンドプロンプトから、./splunk コマンドを使用します。

不明な点がある場合に備えて、Splunk の CLI にはヘルプが内蔵されています。splunk help と入力すると、CLI ヘルプのメインページにアクセスします。各コマンドにもそれぞれのヘルプページがあります。Splunk help <command> と入力します。

CLI から入力設定が可能なコマンドを以下に示します。

コマンド	コマンド構文	動作
追加	add monitor \$SOURCE [-parameter value] ...	\$SOURCE から入力を追加。
編集	edit monitor \$SOURCE [-parameter value] ...	以前に \$SOURCE 用に追加された入力の編集。
削除	remove monitor \$SOURCE	以前に追加された \$SOURCE の削除。
リスト	list monitor	現在設定されているモニタを一覧表示。
スプール	spool source	シンクホールディレクトリからファイルを Splunk にコピー。

追加のパラメータを設定して、各データ入力タイプの設定を変更します。パラメータは構文 -parameter value で設定します。

注意：コマンド別に、1 つの -hostname、-hostregex、または -hostsegmentnum を設定できます。

パラメータ	必要性	説明
source	必須	新しい入力の監視のためのファイルまたはディレクトリへのパス
sourcetype	任意	入力ソースからイベント用のソースタイプフィールド値を指定
index	任意	入力ソースからイベント用の宛先インデックスを指定。
hostname	任意	入力ソースからイベント用のホストフィールド値として設定するホスト名を指定
hostregex	任意	入力ソースからイベント用のホストフィールド値として設定するソースファイルパスの正規表現を指定
hostsegmentnum	任意	入力ソースからイベント用のホストフィールド値として設定するソースファイルパスのセグメント数を指定
follow-only	任意	(T/F) True または False。デフォルトでは False。True に設定すると、Splunk は、ソースの終わりから読み込む("tail -f" UNIX コマンド同様)。

例：CLI を使用して、/var/log/を監視する

以下の例では、/var/log/にあるファイルを監視する方法をお見せします。

データ入力として、/var/log/を追加します。

```
./splunk add monitor /var/log/
```

例：CLI を使用して、windowsupdate.log を監視する

以下の例では、Windows アップデートログ(Windows ログの自動アップデート)を監視する方法をお見せします。

データ入力として、C:\Windows\windowsupdate.log を追加します。

```
./splunk add monitor C:\Windows\windowsupdate.log
```

例：CLI を使用して、IIS ログを監視する

この例では、Windows IIS ログのデフォルトの保存場所を監視する方法をお見せします。データ入力として、C:\windows\system32\LogFiles\W3SVC を追加します。

```
./splunk add monitor c:\windows\system32\LogFiles\W3SVC
```

Inputs.conf

入力を追加するには、そのスタanzasを、\$SPLUNK_HOME/etc/system/local/にある inputs.conf に追加する、または \$SPLUNK_HOME/etc/apps/にあるユーザーのカスタムアプリケーションディレクトリに追加します。Splunk の設定ファイルで作業経験の無い方は、始める前に本書の「設定ファイルについて」をお読みください。

入力タイプに関する属性および値の数を設定できます。1つ以上の属性の値を指定しない場合、Splunk は、\$SPLUNK_HOME/etc/system/default/に予めセットされているデフォルトを使用します (以下参照)。

モニタ

```
[monitor://<path>]
```

<attributel> = <val1>

<attribute2> = <val2>

...

このタイプの入力スタンプ(モニタ)は、<path>にあるすべてのファイル(<path>が1つのファイルの場合は指定ファイルのみ)を監視するよう Splunk を設定します。入力タイプおよびパスの指定は必須のため、root(ルート)から始める場合は、パスに3つのスラッシュが入ります。詳しくは、以下の項目「ワイルドカード」をご覧ください。

注意：既存のファイルに新しいコンテンツをコピーするときは、新しいイベントが確実にインデックスされるため、ソース用の props.conf にある CHECK_METHOD = modtime を設定してください。これは、ファイルの modtime を確認し、変更があると再インデックスします。このとき、ファイル全体がインデックスされると、イベントが重複しますのでご注意ください。

host = <文字列>

- 固定値に入力のホスト値を設定します。
- host =は、ショートカットを使用すると自動的に値の先頭に追加されます。
- データ元であるホストの FQDN(完全修飾ドメイン名)の IP アドレスがデフォルトです。

index = <文字列>

- この入力によるイベントが保存されるインデックスを設定します。
- index =は、ショートカットを使用すると自動的に値の先頭に追加されます。
- main がデフォルトです(または、デフォルトインデックスを設定したときはそれがデフォルトとなる)。
- インデックスフィールドに関する詳細は、本書の「インデキシングの機能」をご覧ください。

sourcetype = <文字列>

- この入力によるイベントのソースタイプ名を設定します。
- sourcetype=は、ショートカットを使用すると自動的に値の先頭に追加されます。
- Splunk は、データの様々な要素に基づいて自動的にソースタイプを選びます。ハードコードされるデフォルトはありません。
- ソースタイプフィールドに関する詳細は、ナレッジマネージャマニュアルの「ソースタイプについて」をご覧ください。

source = <文字列>

- この入力によるイベントのソース名を設定します。
- ファイルパスがデフォルトです。
- source=は、ショートカットを使用すると自動的に値の先頭に追加されます。

queue = <文字列> (parsingQueue、indexQueue など)

- 入力プロセッサが読み込んだイベントを保存する場所を指定します。
- パイプラインにある有効な既存のキューを使用できます
- parsingQueue がデフォルトです。

host_regex = <正規表現>

- 指定すると、正規表現は各入力のファイル名からホストを抽出します。
- 具体的に、正規表現の最初のグループをホストとして使用します。
- 正規表現が一致に失敗すると、デフォルト host=属性をデフォルトにします。

host_segment = <整数>

- 指定すると、パスの「/」で別れたセグメントがホストとして設定されます。
- 値が整数でない、または 1 以下の場合、デフォルト host::属性をデフォルトにします。

crcSalt = <文字列>

- 設定すると、この文字列を CRC に追加します。
- この設定を使って、Splunk が一致する CRC を持つファイルを取り込むよう強制します。
- crcSalt = <SOURCE>が設定してある場合(注意: この設定は、大文字と小文字を区別する)、ソースパス全体が CRC に追加されます。

followTail = 0|1

- 1 に設定されている場合、ファイルの終わりから監視が始まります(tail -f 同様)。
- これは、ファイルが最初に選択されたときにのみ適応されます。
- その後は、Splunk の内部ファイル位置記録がファイルを追跡します。

_whitelist = <正規表現>

- 設定すると、指定正規表現に一致する場合に限り、このパスのファイルが監視されます。

_blacklist = <正規表現>

- 設定すると、指定正規表現に一致する場合、このパスのファイルを監視しません。

ワイルドカード

ワイルドカードを使用して、監視される入力の入力パスを指定できます。パスには、...を、ファイルには*を使用します。

- は、一致が見つかるまでディレクトリを再帰します。つまり、/foo/.../bar は、foo/bar、foo/1/bar、foo/1/2/bar などと一致します。ただし、**バーがファイルである場合に限ります。**
 - ◆ サブディレクトリを再帰する場合には、もう一つの...を使用します。例: /foo/.../bar/...
- *は、特定のパスセグメントにある全てを一致します。これは、ディレクトリパスの途中には使用できません。パスの最後のセグメントで使用しなければいけません。例えば、/foo/*.log は、/foo/bar.log に一致しますが、/foo/bar.txt や /foo/bar/test.log には一致しません。
- より具体的な一致は、*および...を組み合わせます。
 - ◆ foo/.../bar/*は、指定パス内にある bar ディレクトリにあるファイル全てと一致します。

注意: Windows では、二つのバックスラッシュ\\を使ってワイルドカードを回避しなければいけません。バックスラッシュのついた正規表現は、現在 Windows の _whitelist および _blacklist でサポートされていません。

ワイルドカードを指定すると、スタンザに対して間接的な`_whitelist`が作られます。最も長い完全に確認されたパスが、監視スタンザとして使用され、ワイルドカードは、以下のマップを使って正規表現をコード変換します。

ワイルドカード	正規表現	意味
*	[^/]*	/以外のすべて
...	.*	すべて(最長マッチ)
.	\.	リテラル(直定数)

さらに、パス全体を一致させるため、ファイルパスの右側最後に変換された表現が固定されます。

例えば、以下のように指定した場合、

```
[monitor:///foo/bar*.log]
```

Splunk は、それを以下のようにコード変換します。

```
[monitor:///foo/]
_whitelist = bar[^/]*¥.log$
```

結果として、同じディレクトリのファイル用のワイルドカードを伴う複数スタンザを持ってません。

また、ワイルドカードと共に、`_whitelist` 宣言を使用できません。

例えば、

```
[monitor:///foo/bar_baz*]
[monitor:///foo/bar_qux*]
```

これは、ディレクトリ`/foo/`をインデックスするスタンザと重複します。Splunk は最初の1つのみを使用するため、`/foo/bar_baz`で始まるファイルのみをインデックスします。両ソースを含めるには、「or」の正規表現構文を使って手動で`_whitelist`を指定します。

```
[monitor:///foo]
_whitelist = (bar_baz[^/]*|bar_qux[^/]*)$
```

注意：異なる属性を持つ複数のホワイトリスト/ブラックリスト入力の追加属性(ソースタイプなど)を設定するには、`props.conf`を使用します。

例

`/apache/foo/logs` または `/apache/bar/logs` 等にある全てをロードするには、

```
[monitor:///apache/.../logs]
```

`.log` で終わる `/apache` にある全てをロードするには、

```
[monitor:///apache/*.log]
```

バッチ

```
[batch://<path>]
move_policy = sinkhole
<attributel> = <val1>
<attribute2> = <val2>
...
```

バッチを使用して、ソースから取り込み後に削除される 1 回使用のデータの入力を設定します。取り込み後に削除されない継続的な入力には、**モニタ**を使用します。処理が終わるとファイルが**削除される**ことを忘れないでください。

注意: `move_policy = sinkhole` は必ず設定してください。これは、ファイルをロードした後に削除します。削除を希望しないファイルにはこの入力タイプを使用しないでください。

`host = <文字列>`

- 固定値に入力のホスト値を設定します。
- `host =`は、ショートカットを使用すると自動的に値の先頭に追加されます。
- データ元であるホストの FQDN(完全修飾ドメイン名)の IP アドレスをデフォルトにします。

`index = <文字列>`

- 入力によりイベントが保存されるインデックスを設定します。
- `index =`は、ショートカットを使用すると自動的に値の先頭に追加されます。
- `main` がデフォルトです(または、デフォルトインデックスを設定した場合はそれがデフォルトとなる)。
- インデックスフィールドに関する詳細は、本書の「インデキシングの機能」をご覧ください。

`sourcetype = <文字列>`

- この入力によるイベントのソースタイプ名を設定します。
- `sourcetype=`は、ショートカットを使用すると自動的に値の先頭に追加されます。
- Splunk は、データの様々な要素に基づいて自動的にソースタイプを選びます。ハードコードされるデフォルトはありません。
- ソースタイプフィールドに関する詳細は、ソースタイプの項をご覧ください。

`source = <文字列>`

- この入力によるイベントのソース名を設定します。
- ファイルパスがデフォルトです。
- `source=`は、ショートカットを使用すると自動的に値の先頭に追加されます。

`queue = <文字列> (parsingQueue、indexQueue など)`

- 入力プロセッサが読み込んだイベントを保存する場所を指定します。
- パイプラインにある有効な既存のキューが使用できます
- `parsingQueue` がデフォルトです。

host_regex = <正規表現>

- 指定すると、正規表現は各入力のファイル名からホストを抽出します。
- 具体的に、正規表現の最初のグループをホストとして使用します。
- 正規表現が一致に失敗すると、デフォルト host=属性がデフォルトになります。

host_segment = <整数>

- 指定すると、パスの「/」で別れたセグメントがホストとして設定されます。
- 値が整数でない、または 1 以下であると、デフォルト host::属性がデフォルトとなります。

注意 : source = <文字列>および<KEY> = <文字列>はバッチでは使用されません。

例

この例では、バッチが、ディレクトリ/system/flight815/から全てのファイルをロードします。

```
[batch://system/flight815/*]  
move_policy = sinkhole
```

ネットワークポートの監視

ネットワークポートの監視

Splunk が、TCP または UDP ポート上の入力を認めるよう設定できます。Splunk は、これらのポート上に送られたデータ全てを取り入れます。Syslog(デフォルトポートは UDP514)用にこの方法を使用または、ネットキャットを設定し、ポートにバインドしてください。

TCP は、Splunk のデータ分布の根底になるプロトコルであり、リモートマシンから Splunk サーバーへデータを送信する際に推奨される方法です。Splunk を実行するユーザーには、ポートへのアクセスが必要です。UNIX システムで、1024 の下でポートにアクセスするにはルートとして実行しなければいけません。

Splunk Web を使用してネットワーク入力を追加

Splunk Web を使用してネットワーク入力を追加します。

1. Splunk Web の右上端にある**管理**をクリックします。
2. システムコンフィグレーションのデータ入力をクリックします。
3. **TCP** または **UDP** を選択します。
4. **新規**をクリックして、入力を追加します。
5. ポートナンバーを入力します。
6. 選択したポートが、全てのホストまたは一つのホストからの接続を受け入れるかどうかを指定します。一つのホストを指定する場合は、ホストの IP アドレスを入力します。
7. ここで、**ソースタイプ**を設定します。

ソースタイプは、イベントに追加されたデフォルトフィールドです。ソースタイプは、タイムスタンプやイベント境界な

どの処理特性を決定するために使用します。以下から選択します。

- リストから
 - ◆ ドロップダウンリストから、予め定義されたソースタイプの一つを選択します。
- 手動
 - ◆ テキストボックスに、独自のソースタイプを名付けます。

8. ソース、ホスト、ソースタイプを指定したら、**実行**をクリックします。

CLI を使用してネットワーク入力を追加

Splunk のコマンドラインインタフェース(CLI)を使ってファイルとディレクトリを監視します。Splunk の CLI を使用するには、\$SPLUNK_HOME/bin ディレクトリに移動して、./splunk コマンドを使用します。

不明な点がある場合に備えて、Splunk の CLI にはヘルプが内蔵されています。CLI ヘルプのメインページにアクセスするには、splunk help と入力します。各コマンドにもそれぞれのヘルプページがあります。Splunk help <command> と入力します。

以下のコマンドで、CLI による入力設定が可能です。

コマンド	コマンド構文	動作
追加	add tcp udp \$SOURCE [-parameter value] ...	\$SOURCE から入力を追加
編集	edit tcp udp \$SOURCE [-parameter value] ...	以前に \$SOURCE 用に追加された入力を編集
削除	remove tcp udp \$SOURCE	以前に追加された \$SOURCE を削除
リスト	list tcp udp	現在設定されているモニタを一覧表示

追加のパラメータを設定して、各データ入力タイプの設定を変更します。パラメータは -parameter value 構文で設定します。

パラメータ	必要性	説明
\$SOURCE	必須	インデックするデータを引くためのポート番号
sourcetype	任意	入力ソースのイベント用のソースタイプフィールド値を指定
index	任意	入力ソースのイベント用の宛先インデックスの指定
hostname	任意	入力ソースのイベント用のホストフィールド値として設定するホスト名を指定。
remotehost	任意	独占的にデータを認める IP アドレス元を指定
follow-only	任意	(T/F) True または False。デフォルトでは False。True に設定して DNS を使い、入力ソースのイベント用のホストフィールド値を設定。

例

ネットワーク入力を設定してから、ソースタイプを設定します。

- UDP 入力を設定して、ポート 514 を監視し、ソースタイプを「syslog」に設定します。

ベストプラクティス Wiki で、Syslog 入力設定時に UDP を使う最善の方法についての情報を確認してください。

```
./splunk add udp 514 -sourcetype syslog
```

- DNS から UDP 入力のホスト値を設定します。ユーザー名とパスワードに `auth` を使用します。

```
./splunk edit udp 514 -resolvehost true -auth admin:changeme
```

注意：ポートを 1024 で監視するには、Splunk をルートとして実行しなければいけません。

inputs.conf を使用してネットワーク入力を追加

入力を追加するには、そのスタanzas を、`$SPLUNK_HOME/etc/system/local/`にある `inputs.conf` に追加する、または `$SPLUNK_HOME/etc/apps/`にあるユーザーのカスタムアプリケーションディレクトリに追加します。Splunk の設定ファイルで作業経験の無い方は、始める前に本書の「設定ファイルについて」をお読みください。

入力タイプに関する属性および値の数を設定できます。1 つまたはそれ以上の属性の値を指定しない場合、Splunk は、`$SPLUNK_HOME/etc/system/default/`に予めセットされているデフォルトを使用します(以下参照)。

TCP

```
[tcp://<remote server>:<port>]
<attributel> = <val1>
<attribute2> = <val2>
...
```

入力スタanzas のこのタイプは、Splunk に、`<port>`上の `<remote server>`をリッスンするよう指示します。`<remote server>`が空白の場合、Splunk は、指定されたポート上の全ての接続をリッスンします。

`host = <文字列>`

- 固定値に入力のホスト値を設定します。
- `host ::`は、ショートカットを使用すると自動的に値の先頭に追加されます。
- データ元であるホストの FQDN(完全修飾ドメイン名)の IP アドレスをデフォルトにします。

`index = <文字列>`

- 入力によりイベントが保存されるインデックスを設定します。
- `index::`は、ショートカットを使用すると自動的に値の先頭に追加されます。
- `main` がデフォルトです(または、デフォルトインデックスを設定した場合はそれがデフォルトとなる)。
- インデックスフィールドに関する詳細は、本書の「インデキシングの機能」をご覧ください。

`sourcetype = <文字列>`

- 入力によるイベントのソースタイプ名を設定します。
- `sourcetype::`は、ショートカットを使用すると自動的に値の先頭に追加されます。
- Splunk は、データの様々な要素に基づいて自動的にソースタイプを選びます。ハードコードされるデフォルトはありません。
- ソースタイプフィールドに関する詳細は、ナレッジマネージャマニュアルのソースタイプについてをご覧ください。

source = <文字列>

- 入力かによるイベントのソース名を設定します。
- ファイルパスがデフォルトです。
- source::は、ショートカットを使用すると自動的に値の先頭に追加されます。

queue = <文字列> (parsingQueue、indexQueue など)

- 入力プロセッサが読み込んだイベントを保存する場所を指定します。
- パイプラインにある有効な既存のキューを使用できます。
- parsingQueue がデフォルトです。

connection_host = [ip | dns]

- ip:を設定すると、TCP 入力プロセッサが、リモートサーバーの ip アドレスを持つホストを書き換えます。
- dns:に設定すると、リモートサーバーの DNS エントリーのホストは書き換えられます。
- ip がデフォルトです。

UDP

重要: バージョン 3.3.3. の Splunk では、UDP によるデフォルト syslog 処理は、改行を正確に扱いません。この問題を回避するには、\$SPLUNK_HOME/etc/system/loca/inputs.conf にある UDP スタンザに、_linebreaker = _linebreaker を追加します。

```
[udp://<port>]
```

```
<attribute1> = <val1>
```

```
<attribute2> = <val2>
```

```
...
```

入力スタンザのこのタイプは、UDP ポート上でリッスンすることを除き TCP タイプと同様です。

host = <文字列>

- 固定値に入力のホスト値を設定します。
- host =は、ショートカットを使用すると自動的に値の先頭に追加されます。
- データ元であるホストの FQDN(完全修飾ドメイン名)の IP アドレスをデフォルトにします。

index = <文字列>

- 入力によりイベントが保存されるインデックスを設定します。
- index =は、ショートカットを使用すると自動的に値の先頭に追加されます。
- main がデフォルトです(または、デフォルトインデックスを設定した場合はそれがデフォルトとなる)。
- インデックスフィールドに関する詳細は、本書の「インデキシングの機能」をご覧ください。

sourcetype = <文字列>

- 入力によるイベントのソースタイプ名を設定します。

- `sourcetype` =は、ショートカットを使用すると自動的に値の先頭に追加されます。
- Splunk は、データの様々な要素に基づいて自動的にソースタイプを選びます。ハードコードされるデフォルトはありません。
- ソースタイプフィールドに関する詳細は、ナレッジマネージャマニュアルのソースタイプについてをご覧ください。

`source` = <文字列>

- 入力によるイベントのソース名を設定します。
- ファイルパスがデフォルトです。
- `source` =は、ショートカットを使用すると自動的に値の先頭に追加されます。

`queue` = <文字列> (`parsingQueue`、`indexQueue` など)

- 入力プロセッサが読み込んだイベントを保存する場所を指定します。
- パイプラインにある有効な既存のキューを使用できます
- `parsingQueue` がデフォルトです。

`_rcvbuf` = <整数>

- UDP ポート用の受信バッファを指定します。
- 値が 0 または負の場合は、無視されます。
- Splunk のデフォルトの値は 1MB です (OS によりデフォルトが異なる)。

`No_priority_stripping` = `true` | `false`

- 属性が `True` に設定されると、Splunk は、受信したイベントから `<priority> syslog` フィールドをストリップしません。
- それ以外の場合、Splunk は、イベントから `syslog` 優先順位をストリップします。

`No_appending_timestamp` = `true`

- この属性が `True` に設定されると、Splunk は、受信したイベントにタイムスタンプおよびホストを追加しません。
- **注意:** 受信したイベントにタイムスタンプおよびホストを追加したい場合には、このキーを含めないでください。

Windows イベントログデータを監視

Windows イベントログデータを監視

このトピックでは、Windows のイベントログ、レジストリー、および WMI データを監視するための Splunk の設定の仕方について説明します。

注意: 別のログチャンネルを追加してローカルホストを監視するには、既存の入力を編集します。リモートマシンを監視するには、新しい入力を追加します。

これは、Splunk Web または設定ファイルを使って設定できます。

Splunk Web で監視する Windows イベントログの設定

1. Splunk Web の右上隅にある**管理**をクリックします。
2. システムコンフィギュレーションの**データ入力**をクリックします。
3. **イベントログの収集**をクリックします。
4. **新規**をクリックして、入力を追加します。
5. この収集に対する固有の名前を入力します。
6. ログを引き出すホストのホスト名または IP アドレスを指定し、**ログの検索...**をクリックして、選択するログのリストを入手します。

注意 : Windows Vista には、たくさんのチャンネルがあります。Splunk で利用可能な CPU により、全てまたは多くのチャンネルを選ぶことにより負荷の度合いが異なります。

1. 任意で、データを引く追加のサーバーをコンマ区切りのリストを指定します。
2. **保存**をクリックします。

入力が追加され、使用可能になりました。

設定ファイルを使用して監視する Windows イベントログの設定

1. inputs.conf を \$SPLUNK_HOME \ etc \ system \ default \ から etc \ system \ local にコピーします。
2. 「読み込み専用」を解除します。
3. 以下の詳細を使用して、Windows イベントログ入力を開き、有効化します。
4. Splunk を再起動します。

Inputs.conf 詳細事項で Windows イベントログを監視

Windows イベントログは、バイナリフォーマット*.evt ファイルのため、フラットファイルのように監視できません。インデックスするイベントログを指定する設定は、inputs.conf の以下のスタンザで行います。

```
# Windows platform specific input processor.
```

```
[WinEventLog:Application]
```

```
disabled = 0
```

```
[WinEventLog:Security]
```

```
disabled = 0
```

```
[WinEventLog:System]
```

```
disabled = 0
```

Splunk にデフォルトではない Windows イベントログを読み込むよう指定できます。ただし、これは最初に Windows イベントビューアーにインポートしてから、inputs.conf のローカルコピー(通常、\$SPLUNK_HOME \ etc \ system \ local \ inputs.conf)に、以下のように追加する必要があります。

```
[WinEventLog:DNS Server]
```

```
disabled = 0
```

```
[WinEventLog:Directory Service]
```

```
disabled = 0
```

```
[WinEventLog:File Replication Service]
```

```
disabled = 0
```

イベントログのインデキシングを無効にするには、`$SPLUNK_HOME\etc\system\local\inputs.conf` のスタンザのリストの下に、`disabled = 1` を追加します。

インデックスの開始点(イベントの最初または最新)の指定

この設定は、イベントをインデックスする年代順(古い順または新しい順)を指定します。また、既存のすべてのイベントまたは新しいイベントのみをインデックスするかどうかも指定します。

```
start_from = oldest
```

```
current_only = 1
```

- `start_from` : デフォルトで、Splunk は最も古いデータからインデックスします。この設定を変更すると非常に非効率的なインデキシング処理につながるためお勧めしません。
- `current_only` : このオプションを使うと、Splunk が起動した瞬間から新しいイベントのみをインデックスできます。ファイルの尾部のように機能します。

エクスポートした Windows イベントログ(.evt または .evtx)ファイルのインデックス

エクスポートした Windows イベントログファイルをインデックスするには、ファイルとディレクトリの監視の方法を使用します。

警告

- ローカルファイルのアップロード機能を使用しないでください。現在この機能は、このファイルタイプを取り扱いません。
- ファイルは、Splunk のローカルとしてアクセスが可能でなければいけません。
- 書き込み中の .evt または .evtx ファイルの監視を試みないでください。windows は、ファイルのロックを解除しません。最善の方法は、ファイルが保存されているディレクトリをモニタすることです。それにより、新規ファイルが自動的にインデックスされます。

Windows レジストリデータの監視

Windows レジストリデータの監視

Splunk は、Windows レジストリ設定のキャプチャーをサポートしているため、レジストリの変更を監視できます。レジストリにエントリーが追加、更新、削除された時間が分かります。レジストリのエントリーが変更されると、Splunk は、変更をしたプロセス名および、変更されたエントリーへのハイブからのキーパスをキャプチャーします。

Windows レジストリ入力モニタアプリケーションは、`splunk-regmon.exe` と呼ばれるプロセスとして実行します。

警告 : `splunk-regmon.exe` を手動で停止または中止しないでください。システムが不安定になります。処理を停止するには、Windows タスクマネージャまたは、Splunk Web から Splunk サーバー処理を停止します。

Splunk Web でレジストリ監視を有効にする

Windows の Splunk では、レジストリ監視が設定されていますが、デフォルトでは無効になっています。1 回実行のベースラインインデックスを行い、マシンまたはユーザーキーの継続的な監視を別に有効化できます。これは、以下の手順で行います。

1. Splunk Web で右上隅にある**管理**をクリックします。
- 2.
3. データ入力>レジストリモニタリングをクリックします。
4. マシンキーまたはユーザーキーを選択して、希望するベースラインおよび継続的監視を有効化します。
5. 保存をクリックします。

しくみの詳細

Windows レジストリは極度に動的であることがあります(そのため大量イベントを作成します)。Splunk では、2 階層の設定を提供して供給されるレジストリイベントデータに適応されるフィルタの微調整を行います。

Splunk の Windows レジストリモニタリングは、2 つの設定ファイルを使用して、システム、`$SPLUNK_HOME \ etc \ system \ local \` の `sysmon.conf` と `regmon-filters.conf` に対して監視する項目を決定します。この設定ファイルは、階層構造で機能します。

- `sysmon.conf` は、監視するイベントタイプ(追加、削除、名前の変更など)、使用する `regmon-filters.conf` ファイルの正規表現フィルタ、および Windows レジストリイベントを監視するかどうかを決めるグローバル設定を含んでいます。
- `regmon-filters.conf` は、Splunk で監視するハイブキープスを微調整およびフィルタするために作成する具体的な正規表現を含んでいます

`sysmon.conf` は、以下を指定するスタanzas を 1 つ含みます。

- `event_types`: 監視するレジストリイベントタイプのスーパーセット(`delete`, `set`, `create`, `rename`, `open`, `close`, `query` など)
- `active_filters`: Splunk で監視する処理およびハイブキープスを正確に指定する `regmon-filters.conf` で定義した正規表現フィルタのリスト。これは、`regmon-filters.conf` のスタanzas をコンマ区切りしたリストです。命名規則に基づいて関連または類似するフィルタのグループに名前を付けて呼び出したい場合に便利なワールドカードが使用できます。このリストのフィルタに名前が付いていない場合は、そのフィルタが `regmon-filters.conf` に存在していても使用されません。つまり、さまざまなフィルタおよびフィルタグループの監視のオン/オフが自由に行えます。
- `disabled`: レジストリ設定変更を監視するかどうかを決定。これを 0 に設定すると、Windows レジストリモニタリングが完全に無効にします。

`regmon-filters.conf` の各スタanzas は、以下の定義を持つ特定のフィルタを示します。

- `proc`: 監視するプロセス(1 つまたは複数)へのパスを含む正規表現
- `hive`: 監視するエントリ(1 つまたは複数)へのハイブパスを含む正規表現。Splunk は、Windows で予め定

義されたルートキー値マッピングをサポートしています。

- ◆ \\REGISTRY\USER\ は、HKEY_USERS または HKU に割り当て
 - ◆ \\REGISTRY\USER\ は、HKEY_CURRENT または HKCU に割り当て
 - ◆ \\REGISTRY\USER_Classes は、HKEY_CLASSES_ROOT または HKCR に割り当て
 - ◆ \\REGISTRY\MACHINE\ は、HKEY_LOCAL_MACHINE または {HKLM に割り当て
 - ◆ \\REGISTRY\MACHINE\SOFTWARE\Classes は、HKEY_CLASSES_ROOT または HKCR に割り当て
 - ◆ \\REGISTRY\MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current は、HKEY_CURRENT_CONFIG または HKCC に割り当て
- type : 監視するイベントタイプのサブセット。delete、set、create、rename、open、close、query が該当します。この値は、sysmon.conf で設定する event_types の値のサブセットでなければいけません。
 - baseline : 特定のハイブパスに対するベースラインスナップショットをキャプチャーするかどうか決定。0 = 非実行、1 = 実行。
 - baseline interval : Splunk が次のスナップショットを撮るまでの停止時間を秒単位で指定。デフォルト値は 24 時間。

ベースラインスナップショットを撮影

レジストリモニタリングを有効にすると、次回 Splunk を起動する時に、レジストリハイブのベースラインスナップショットを記録するオプションが与えられます。デフォルトでは、スナップショットは、ユーザーキーおよびマシンキーハイブ全体を記録します。また、スナップショットを再撮影する時期を設定します。Splunk が、前回のチェックポイントから 24 時間以上停止している場合、ベースラインスナップショットを再撮影します。regmon-filters.conf にある各フィルタの値を、baseline interval の値を設定すると、カスタマイズできます。

注意 : splunk clean all -f を実行すると、その時点のベースラインスナップショットを削除します。

考慮事項

Windows マシンに Splunk をインストールして、レジストリモニタリングを有効にするときは、主要なハイブパスであるどのキーユーザー(HKEY)および/またはキーローカルマシン(HKLM)を監視するか指定します。そのマシンで予測されるレジストリの動的度により、両方を指定することにより、Splunk で監視するデータ量が膨大になる場合があります。多くのレジストリイベントを予測する場合は、Splunk をインストールしてレジストリイベントモニタリングを有効にした後で、Splunk を起動する前に、regmon-filters.conf にあるフィルタの一部を指定して即座に監視の範囲を狭めてください。

同様に、Splunk を初めて起動するとき、および指定の時間が過ぎるごとに、Windows のレジストリの状態をベースラインスナップショットとしてキャプチャーするオプションがあります。ベースライン処理は、プロセッサ集中型であるため数分かかります。regmon-filters.conf を編集し、特に Splunk で監視するレジストリエントリーの範囲を狭めるまで、ベースラインのスナップショットの撮影を先送りにすることができます。

Windows レジストリ入力の設定

inputs.conf の Windows レジストリ入力のデフォルト値を確認してください。値は以下にも掲載されています。デフ

オルト値を変更する場合は、`$SPLUNK_HOME\etc\system\local\`の `inputs.conf` のコピーを編集します。スタンザに変更するパラメータの値のみを入力する必要があります。Splunk 設定ファイルの使用方法についての詳細は、「設定ファイルについて」をご覧ください。

```
[script://$SPLUNK_HOME\bin\scripts\splunk-regmon.py]
interval = 60
sourcetype = WinRegistry
source = WinRegistry
disabled = 0
```

- `source` : レジストリのイベントにラベルを付けます。
- `sourcetype` : これらのイベントをレジストリイベントとして割り当てます。
- `interval` : レジストリの変更をポーリングする頻度を秒単位で指定します。
- `disabled` : 機能が有効であるかを表示します。1 に設定するとこの機能が無効になります。

注意 : Splunk レジストリ入力モニタリングスクリプト(`splunk-regmon.py`)は、スクリプト入力として設定されています。この値は変更しないでください。

注意 : `inputs.conf` にあるスタンザ名のワイルドカードを回避するには、必ず 2 つのバックスラッシュ(`\\`)を使用してください。正規表現にバックスラッシュを使ってファイルパスを指定することはできません。

WMI データの監視

WMI データの監視

Splunk は、Windows の性能データおよびイベントログにエージェントなしでアクセスする WMI(Windows 管理インタフェース)データ入力をサポートしています。つまり、使用環境に何もインストールする必要なく、すべての Windows サーバーおよびデスクトップからイベントログを引き出すことができます。

Splunk の WMI データ入力は、複数の WMI プロバイダに接続し、データを引き出すことができます。WMI データ入力は、Splunk サーバー上で異なる処理(`splunk-wmi.exe`)として実行します。`$SPLUNK_HOME\etc\system\default\inputs.conf` のスクリプト入力として設定されています。このファイルは編集しないでください。

注意 : この機能は、Windows 用 Splunk でのみ利用可能であり、デフォルトでは無効になっています。有効にするには、以下のラインを `$SPLUNK_HOME\etc\system\default\inputs.conf` に追加してください。

`$SPLUNK_HOME\etc\system\local\inputs.conf`:

```
[script://$SPLUNK_HOME\bin\scripts\splunk-wmi.py]
disabled = 0
```

セキュリティおよびリモートアクセスにおける考慮事項

Splunk は、WMI、イベントログ、およびレジストリを含む多くの Windows のデータソースをインデックスするために特権のあるアクセスが必要です。これには、ボックスに接続する能力および、接続後に適切なデータを読み込む権限の両方が含まれます。WMI データにアクセスするには、Splunk をリモート WMI 接続可能な権限を持つユーザーで実行する必要があります。このユーザー名は、アクティブディレクトリドメインのメンバーであり、WMI をクエリする適切な権限を持つ必要があります。クエリを作成する Splunk サーバーおよびクエリされるターゲットシステムの両方が、アクティブディレクトリドメインの一部でなければいけません。

注意： Splunk を LOCAL SYSTEM ユーザーとしてインストールした場合、WMI リモート認証は機能しません。そのユーザーは認証を無効にするため、Windows サーバーは通常そのような接続を認めません。

以下の点を考慮してください。

- WMI によるリモートデータ収集には、ポーリングする WMI リソースにアクセスするのに十分な OS 権限を持つユーザーとして、Splunk サービスを実行しなければいけません。少なくとも、Splunk には、ポーリングするすべてのマシンに対して以下のアクセス権限が必要です。
 - ◆ システムパフォーマンス統計
 - ◆ ネットワーク経由のコンピュータアクセス
 - ◆ Splunk にこれらのリソースへのアクセス権を確実に与える最も簡単な方法は、Splunk のユーザーを、Performance Log Users および Distributed COM Users のドメイングループに追加することです。追加しても十分な権限が得られない場合は、Splunk のユーザーを、リモートマシンの管理者グループに追加します。
- リモートマシンアクセス用に DCOM を有効にして、Splunk ユーザーのアクセスを有効にします。詳細は、リモート WMI 接続の確保について掲載されている Microsoft のトピックスをご覧ください。Splunk のユーザーを **Distributed COM Users** のローカルグループに追加する方法が、この権限を有効にする最速の方法です。追加しても十分な権限が与えられない場合は、Splunk のユーザーを、リモートマシンの管理者グループに追加します。
- Splunk がアクセスする WMI のネームスペース(最も一般的な `root \ cimv2`)には、適切な一連の権限が必要があります。Splunk ユーザーのルートにある WMI ツリー上で以下の権限を有効にします。
 - ◆ 実行方法、アカウントの有効化、リモートの有効化、およびセキュリティの読み取り。
 - ◆ 詳細は、Microsoft how-to HOW TO: Set WMI Namespace Security in Windows Server 2003 をご覧ください。
- ファイアウォールが有効の場合は、必ず WMI へのアクセスを有効にしてください。Windows ファイアウォールを使用している場合は、例外リストに WMI が明確に載っています。この例外設定はメインおよびリモートマシンの両方で行う必要があります。詳しくは、Vista で遠隔起動した WMI の接続についての Microsoft トピックスをご覧ください。

WMI のアクセステスト

以下の手順に従って、Splunk サーバーおよびリモートマシンの設定をテストします。

1. Splunk がユーザーとして実行しているマシンにログインします。
2. **スタート->ファイル名を指定して実行**をクリックし、wbemtest と入力します。wbemtest アプリケーションが起動します。

3. **接続**をクリックし、¥¥<server>¥root¥cimv2(< server>にリモートサーバーの名前を入力)と入力します。**接続**をクリックします。接続できない場合は、マシン間の認証に問題があります。
4. 接続できたら、**クエリ**をクリックして、`select * from win32_service`と入力します。**適用**をクリックします。少し待つと、実行中のサービスのリストをご覧いただけます。これが機能しない場合、認証に問題ありませんが、Splunk を実行しているユーザーにその操作を実行するのに十分な権限がありません。

WMI 入力の設定

WMI の入力、Splunk Web を使う、または設定ファイルを編集して設定できます。設定ファイルのオプションを使用すると、より多くのオプションを利用できます。

Splunk Web による WMI の設定

1. Splunk Web の右上隅にある**管理**をクリックします。
2. システムコンフィギュレーションの下の**データ入力**をクリックします。
3. **WMI データ収集**をクリックします。
4. **新規**をクリックして、入力を追加します。
5. この収集用に固有の名前を入力します。
6. ターゲットホストを入力する**クエリ...**をクリックして、利用可能な選択するプロパティのクラスのリストを入手します。
7. 任意で、データを読み出す追加サーバーのリストをコンマ区切りで指定します。
8. ポーリングの間隔を秒単位で指定します。
9. **有効化?**ラジオボタンが**はい**に設定されていることを確認してから、**保存**をクリックします。

入力は追加され有効化されました。

設定ファイルによる WMI の設定

wmi.conf から WMI 入力のデフォルト値を確認してください。デフォルト値を変更する場合は、`$SPLUNK_HOME \ etc \ system \ local \`にある `wmi.conf` のコピーを編集します。データ入力のタイプに対して変更する属性の値のみ設定します。Splunk で設定ファイルを使用する方法についての詳細は、「設定ファイルについて」をご覧ください。

```
[settings]
initial_backoff = 5
max_backoff = 20
max_retries_at_max_backoff = 2
result_queue_size = 1000
checkpoint_sync_interval = 2
heartbeat_interval = 500
[WMI:AppAndSys]
server = foo, bar
interval = 10
```

```
event_log_file = Application, System, Directory Service
disabled = 0
[WMI:LocalSplunkWmiProcess]
interval = 5
wql = select * from Win32_PerfFormattedData_PerfProc_Process where Name = "splunk-wmi"
disabled = 0
```

[settings]スタanzasは、ランタイムパラメータを指定します。そのすべてのスタanzasおよび各パラメータはオプションです。スタanzasが無い場合、Splunk は、システムデフォルトを採用します。

- 以下の属性は、エラー発生時にエージェントが指定された WMI プロバイダに再接続する方法を制御します。全ての時間は秒単位です。
 - ◆ `initial_backoff`: 最初のエラー発生後、再接続を試みるまでの待ち時間。その後、エラーが再発すると、待ち時間は、`max_backoff` に到達するまで 2 倍になります。
 - ◆ `max_backoff`: `max_retries_at_max_backoff` を呼び出すまでの最大待ち時間
 - ◆ `max_retries_at_max_backoff`: 待ち時間が `max_backoff` に到達した場合、待ち時間にこれを何回も試みます。エラーが再発すると、Splunk サービスが再起動されるまで、問題のある WMI プロバイダに再接続しません。
- `result_queue_size`: データが出力に書き込まれているのを待つ間に WMI プロバイダがブロックしないようにするキューのサイズ。WMI プロバイダから受け取る結果は、このキューに記録されます。
- `checkpoint_sync_interval`: ディスクに状態データ (イベントログチェックポイント) が書き込まれる最短待ち時間。秒単位。
- `heartbeat_interval`: WMI プロバイダへの接続を管理するスレッドが等間隔で実行。ミリ秒単位。

データ入力には、イベントログと未加工 WQL (WMI クエリ言語) の 2 種類が指定できます。イベントログ入力スタanzasには、`event_log_file` パラメータがあり、WQL 入力スタanzasには、`wql` があります。

以下のパラメータは両タイプで共通です。

- `server`: データを引き出すサーバーの一覧(コンマ区切り)。このパラメータが無い場合、Splunk は、ローカルマシンを想定します。
- `interval`: 新しいデータをポーリングする頻度 (秒単位)。必須。
- `disabled`: この機能の有効または無効を表示。このパラメータを 1 に設定すると、Splunk への WMI 入力を無効にします。

WQL 専用パラメータ:

- `namespace`: WMI プロバイダへのパスを指定します。ローカルマシンは、委任認証を使用してリモートマシンに接続できる必要があります。この属性はオプションです。リモートマシンへのパスを指定しない場合、Splunk は、クエリを行うほとんどのプロバイダがある、デフォルトローカルネームスペース (`\root\cimv2`) に接続します。Microsoft は、Windows XP とそれ以降の Windows バージョンのネームスペースのリストを提供します。
- `wql`: WQL クエリを提供します。上記の例は、`splunkd` と名づけられた実行処理に関するデータを 5 秒ごとに

ポーリングします。

イベントログ専用パラメータ : `event_log_file` : ログファイルのコンマ区切りリストを指定して、`event_log_file` パラメータでポーリングします。例で示されている通り、空白を含むファイル名をサポートしています。

WMI データのフィールド

WMI から受信した全てのイベントに、wmi に設定されたソースがあります。

- イベントログデータの場合、ソースタイプは、WinEventLog:<name of log file>に設定されています(例 : WinEventLog:Application)。
- WQL データの場合、ソースタイプは、設定スタンザの名前に設定されています(例 : [WMI:LocalSplunkdProcess] と名づけられたスタンザには、フィールドは、WMI:LocalSplunkProcess に設定されています)。

ホストは、受信したデータから自動的に特定されます。

アクティブディレクトリの監査

アクティブディレクトリの監査

アクティブディレクトリ監査を入力として設定し、AD フォレストの一部または全てへの変更を監視し、ユーザーおよびマシンメタデータを収集します。

この機能を有効にして Splunk を再起動すると、AD データおよび AD スキーマのベースラインスナップショットを撮影します。このデータを使用して、監視項目に対する開始点を得ます。この処理はスロットルのため、リモート AD インスタンスを監査していても接続に負担をかけませんが、完了までに少し時間がかかります。

知っておくべき事項

- この機能は、Windows のプラットフォームでのみ利用可能なため、機能するには Windows のアプリケーションを有効にする必要があります。
- Splunk を実行しているマシンは、監視するドメインに属している必要があります。
- Splunk を実行しているユーザーも、ドメインの一部である必要があります。おそらく、ローカルシステムユーザー以外を指定する必要があります。
- AD 監査を設定する際、Splunk は、実行しているユーザー名を使用して指定された AD インスタンスに移動してクエリします。このしくみを利用して、Splunk のアクセスレベルおよび閲覧可能事項を制御します。
- Windows のユーザー権利ポリシーにより、アクティブディレクトリのスキーマを完全にインデックスできないことがあります。

inputs.conf および admon.conf による AD 監査の設定

この機能は Windows アプリケーションに含まれているため、そのアプリケーションのディレクトリ構造内の関連ファイルを設定する必要があります。そのため、正しい位置にあるファイルを編集していることを確認してください。

1. `$SPLUNK_HOME \ etc \ apps \ windows \ default \ inputs.conf` のコピーを作成し、`$SPLUNK_HOME \ etc \ apps \`

windows \ local \ inputs.conf に保存します。

2. コピーを編集し、disabled の値を 0 に設定して、スクリプト入力[script://\$SPLUNK_HOME \ bin \ scripts \ splunk-admon.py]を有効にします。
3. 次に、\$SPLUNK_HOME \ etc \ apps \ windows \ default \ admon.conf と類似のコピーを作成し、\$SPLUNK_HOME \ etc \ apps \ windows \ local \ admon.conf に保存します。
4. このトピックの後半にある説明に従って編集します。デフォルトで有効な場合は、Splunk を実行しているユーザーが属しているデフォルトドメインコントローラをインデックスします。問題がない場合は、それ以上の設定は必要なく、正常に動作します。

admon.conf の設定

monitorSubtree = 0 は、Splunk にターゲットコンテナのみインデックスするよう指示します。1 (デフォルト) の値は、アクセスのある全てのコンテナおよびドメインを列挙するよう指示します。

targetDC = には、監視するドメインコントローラホストの固有の名前を指定します。以下の場合に固有の名前を指定します。

- 大量の AD があり、特定のブランチ(ou)、サブドメインの情報だけを監査する。
- ツリーの特定のサブのメインのみに範囲を限定する。
- 高いセキュリティ環境でその目的用に提供された特定の(読み込み専用)ドメインコントローラがある。
- 信用設定に複数のドメインフォレストがある場合、これを使用して、Splunk がある場所とは異なるツリーをターゲットにできます。

複数の DC をターゲットにしたい場合は、そのツリーのターゲット用に別の [<uniquename>TargetDC] スタンザを追加します。

startingNode = には、Splunk がインデクシングを始める完全に確認された LDAP 名(例:

"LDAP://OU=Computers,DC=ad,DC=splunk,DC=com") を指定します。Splunk はそこから始め、上述の

monitorSubtree の設定に従って、サブコンテナへ列挙します。何も指定しないと、アクセスできるツリーの最も高いルートドメインから始めます。

startingNode は、ターゲットにしている DC の範囲内でないと成功しません。

AD モニタリング設定の例

ターゲットにする OU よりも高いルートレベルのターゲット DC を監視できます。例えば、

OU = には eng.ad.aplunk.com サブドメインのコンピュータを指定します。

DC が ad.splunk.com のコントローラの 1 つとなるようにターゲットします。これを行うのは、サブドメインだけでなくツリー全体にスキーマを得たい場合です。開始ノードを、eng.ad.splunk.com の OU に設定し、その OU で追加および削除されるマシンを監査します。

```
[default]
```

```
monitorSubtree = 1
```

```
disabled = 0
DefaultTargetDC]
targetDC = pri01.ad.splunk.com
startingNode = OU=Computers,DC=eng,DC=ad,DC=splunk,DC=com
```

AD データを使う強力なルックアップ

この機能とダイナミックリストルックアップと組み合わせて使用すると、AD で利用可能な情報でイベントを修飾また変更できます。Splunk Community Wiki にあるこのトピックの概要をお読みください。

ファイルシステムへの変更を監視

ファイルシステムへの変更を監視

Splunk のファイルシステム変更モニタは、ファイルシステムの変更の追跡に便利です。ファイルシステム変更モニタは、指定したディレクトリを監視し、ディレクトリに変更がある場合には、(Splunk に)イベントを生成します。これは、完全に設定可能で、システム上のどのファイルが編集、削除、または追加されても検知します(Splunk 専用ファイルではありません)。例えば、ファイルシステム変更モニタで `/etc/sysconfig/` を監視し、システム設定が変更されるとアラートを送るよう指示できます。

`inputs.conf` でファイルシステム変更モニタを設定します。

注意: Windows が読み込むファイルを検査する場合は、Splunk Community ベストプラクティス Wiki にあるこのトピックをご確認ください。ユーザーの中には、Windows 用の監査ツールを使う方が簡単だと思う方もいます。

ファイルシステム変更モニタの機能

ファイルシステム変更モニタは、以下を利用して変更を検知します。

- 変更 日/時
- グループ ID
- ユーザ ID
- ファイルモード(読み込み/書き込み属性など)
- ファイル内容のオプションの SHA256 ハッシュ

ファイルシステム変更モニタの以下の機能を設定できます。

- 正規表現を使用するホワイトリスト
 - ◆ 必ず確認されるファイルを指定
- 正規表現を使用するブラックリスト
 - ◆ スキップするファイルを指定
- ディレクトリ再帰
 - ◆ シンボリックリンクトラバーサルを含む
 - ◆ それぞれのポーリング頻度がある複数のディレクトリのスキャン

- 暗号化署名
 - ◆ ファイルシステム変更の分散観察証跡を作成
- 追加/変更で、ファイル全体をイベントとしてインデックス
 - ◆ ファイル全体の送信および/またはハッシングのサイズカットオフ
- Splunk によりインデックスされ検索可能なすべての変更イベント

ファイルシステム変更モニタの設定

デフォルトで、ファイルシステム変更モニタは、`$SPLUNK_HOME/etc/`の内容が変更、削除、または追加されるたびにイベントを生成します。初めて Splunk を起動すると、`$SPLUNK_HOME/etc/`ディレクトリおよびサブディレクトリの各ファイルに対して add 監査イベントが生成されます。その後、すべての設定変更は(起源に関係なく)、影響のあるファイルに対して監査イベントを生成します。監査イベントは、監査インデックス(`index=_audit`)にインデックスされます。

スタanzas を `inputs.conf` に追加することにより、ファイルシステム変更モニタを使用して、あらゆるディレクトリが監視できます。

`$SPLUNK_HOME/etc/system/local` に独自の `inputs.conf` を作成します。`$SPLUNK_HOME/etc/system/local` または `$SPLUNK_HOME/etc/apps` のカスタムディレクトリにあるこのファイルを編集します。設定ファイルについての全般的詳細は、設定ファイルの機能をご覧ください。

[`fschange`]スタanzas を編集して、ファイルシステム変更モニタを設定します。スタanzas 名 `fschange:<directory or file to monitor>`を除き、全ての設定がオプションです。

注意: [`fschange`]スタanzas を変更したら、必ず Splunk を再起動させてください。

```
[fschange:<directory or file to monitor>]
index=<indexname>
recurse=<true | false>
followLinks=<true | false>
pollPeriod=N
hashMaxSize=N
fullEvent=<true | false>
sendEventMaxSize=N
signedaudit=<true | false>
filters=<filter1>,<filter2>,...<filterN>
```

可能性のある属性/値のペア

```
[fschange:<directory or file to monitor>]
```

- システムは、ディレクトリおよびサブディレクトリに対するすべての追加/更新/削除を監視します。
- いかなる変更も、Splunk によりインデックスされるイベントを生成します。
- `$SPLUNK_HOME/etc` がデフォルトです。

```
index=<indexname>
```

- 生成された全てのイベントを保存するインデックス。
- `_audit` がデフォルト。

`recurse=<true | false>`

- `True` の場合、`[fschange]`で指定されたディレクトリ内のディレクトリを再帰的に読み込みます。
- `True` がデフォルトです。

`followLinks=<true | false>`

- `True` の場合、ファイルシステム変更モニタは、シンボリックリンクを追跡します。
- `False` がデフォルトです。

注意：`followLinks` は慎重に設定しないと、ファイルシステムループが起こることがあります。

`pollPeriod=N`

- `N` 秒ごとにこのディレクトリに変更がないか確認します。
- `3600` がデフォルトです。
 - ◆ 変更すると、ファイルシステム監査イベントが生成され、監査検索で利用できるようになるまで 1 秒から 3600 秒かかります。

`pollPeriod=N`

- バイトサイズが `N` 以下または同等のファイル全ての `SHA1` ハッシュを計算します。
- このハッシュは、ファイル/ディレクトリの変更を検知するための追加方法として使用できます。
- `-1` がデフォルトです(変更検知にハッシングを使わない)。

`signedaudit=<true | false>`

- 暗号化署名で追加/更新/削除したイベントを送信します。
- `False` がデフォルトです。
- `True` に設定すると、`_audit` インデックスにイベントを生成します。

インデックスの設定をご希望の場合、故意に `False` に設定するべきです。

注意：`signedaudit` を `True` に設定するとき、監査は、`audit.conf` で有効になっていることを確認してください。

`fullEvent=<true | false>`

- 追加または更新の変更が検知された場合に完全なイベントを送信します。
- `sendEventMaxSize` 属性により更に確認されます。
- `False` がデフォルトです。

`sendEventMaxSize=N`

- イベントのサイズが `N` バイトより小さいかまたは同等の場合に完全なイベントのみ送信します。
- インデックスされるファイルデータを制限します。
- デフォルトは無制限を示す `-1` です。

sourcetype = <文字列>

- この入力によるイベントのソースタイプを設定します。
- "sourcetype="は、自動的に<string>の先頭に追加されます。
- デフォルトは、sourcetype = fs_notification です。

filesPerDelay = <整数>

- <integer>ファイル処理後、'delayInMills' によって指定された遅れを投入します。
- CPU を消費しすぎないように、ファイルシステムモニタリングのボトルネックに使用されます。

delayInMills = <整数>

- 'filesPerDelay' で指定された全ての<integer>ファイルの処理後に使用する遅れ(ミリ秒)です。
- CPU を消費しすぎないように、ファイルシステムモニタリングのボトルネックに使用されます。

filters=<フィルタ 1>,<フィルタ 2>,...<フィルタ N>

各フィルタは、モニタリングサイクルで検索された各ファイルまたはディレクトリに対して左から右へ適用します。

フィルタを定義するには、以下の通り、[filter...]を追加します。

```
[filter:blacklist:backups]
```

```
regex1 = .*bak
```

```
regex2 = .*bk
```

```
[filter:blacklist:code]
```

```
regex1 = .*\.c
```

```
regex2 = .*\.h
```

```
[fschange:/etc]
```

```
filters = backups,code
```

Fschange ホワイトリスト/ブラックリストロジックは、典型的なファイアウォールと同様に渡されます。イベントは、最初の一致に到達するまでフィルタのリストを渡します。イベントを一致させる最初のフィルタがホワイトリストである場合、イベントがインデックスされます。イベントを一致させる最初のフィルタがブラックリストである場合、イベントはインデックスされません。イベントが、一致しないままチェーンの最後まで到達した場合、そのイベントがインデックスされます。インデックスしないデフォルトを作成するには、全てのイベントのブラックリストチェーンを終わらせます。

例えば、

...

```
filters = <フィルタ 1>, <フィルタ 2>, ... terminal-blacklist
```

```
[filter:blacklist:terminal-blacklist]
```

```
regex1 = .
```

クローラで監視対象事項を詳細検索

クローラで監視対象事項を詳細検索

クローラを使用して、インデックスに追加する新規のデータソースのファイルシステムを検索します。crawl.conf に 1 以上のクローラのタイプを設定し、結果に含める(または除外する)データソースのタイプを定義します。

設定

\$SPLUNK_HOME/etc/system/local/crawl.conf を編集して、crawl コマンド実行時にデータソースをブラウズする 1 以上のクローラを設定します。各クローラ属性の値を指定して、各クローラを定義します。クローラを crawlers_list に追加して有効にします。

クローラロギング

crawl コマンドは、\$SPLUNK_HOME/var/log/splunk/crawl.log に保存されているクローラの活動ログを作成します。ロギングレベルを、crawl.conf の[default]スタンザにある logging キーで設定します。

```
[default]
logging = <warn | error | info | debug>
```

クローラを有効化

[crawlers]スタンザの crawlers_list キーにあるクローラ仕様スタンザ名をリストアップして、クローラを有効にします。

コンマ区切りのリストを使用して、複数のクローラを指定します。

スタンザ[file_crawler]、[port_crawler]、および[db_crawler]で定義されたクローラを有効にします。

```
[crawlers]
crawlers_list = file_crawler, port_crawler, db_crawler
```

クローラの定義

crawl.conf に定義スタンザを追加して、クローラを定義します。追加のスタンザを追加して、追加のクローラ定義を追加します。

Crawl.conf のクローラスタンザの例 :

```
[Example_crawler_name]
....
[Another_crawler_name]
....
```

キー/値ペアをクローラ定義スタンザに追加して、クローラの動作を設定します。以下のキーは、file_crawler の定義に利用できます。

引数	説明
bad_directories_list	除外するディレクトリを指定
bad_extensions_list	除外するファイル拡張子を指定
bad_file_matches_list	文字列、またはファイル名が除外される文字列を含むコンマ区切りリストを指定。ワイルドカードの使用可能(例: foo*.*、foo*bar、*baz*)。
packed_extensions_list	含める共有アーカイブファイルタイプの拡張子を指定。Splunk は、読み込む前に圧縮ファイルを解凍します。tar、gz、bz2、tar.gz、tgz、tbz、tbz2、zip、および z ファイルに対応。アーカイブファイルタイプを追加しない場合はこれを空白のままにします。
packed_extensions_list	ソースがディレクトリと見なされるために必要な最小ファイル数を指定。
days_sizek_pairs_list	クローラされるファイルを制約する年代(日数)およびサイズ(kb)がペアのコンマ区切りリストを指定。例えば、days_sizek_pairs_list = 7-0, 30-1000 は、7 日以内に変更されたサイズ 0kb 以上のファイル、または、30 日以内に変更されたサイズ 1000kb 以上のファイルのみをクローラするよう Splunk に指示します。
big_dir_filecount	クローラされるためにディレクトリが持てる最大ファイル数を設定。クローラは、指定した最大数を超えるファイルを含むディレクトリを除外します。
index	クローラされたファイルおよびディレクトリの内容を追加するインデックスの名前を指定。
max_badfiles_per_dir	クローラするファイルのディレクトリの深さを指定。Splunk がディレクトリをクローラし、指定した max_badfiles_per_dir 内で有効なファイルが見つからない場合、Splunk はそのファイルを除外します。
root	クローラがクローラするファイルを指定。

例

simple_file_crawler と呼ばれるクローラの例です。

```
[simple_file_crawler]
bad_directories_list= bin, sbin, boot, mnt, proc, tmp, temp, home, mail, .thumbnails, cache, old
bad_extensions_list= mp3, mpg, jpeg, jpg, m4, mcp, mid
bad_file_matches_list= *example*, *makefile, core.*
packed_extensions_list= gz, tgz, tar, zip
collapse_threshold= 10
days_sizek_pairs_list= 3-0,7-1000, 30-10000
```

```
big_dir_filecount= 100
index=main
max_badfiles_per_dir=100
```

SNMP イベントを Splunk に送信

SNMP イベントを Splunk に送信

SNMP イベントをインデックスする最も効果的な方法は、`snmptrapd` を使用してファイルに書き込むことです。

最初に、ディスク上のファイルに書き込むよう `snmptrapd` を設定します。

```
# touch /var/run/snmp-traps
# snmptrapd -Lf /var/run/snmp-traps
```

次に、Splunk サーバーでファイルを入力として追加するよう設定します。

カスタム(スクリプト)入力の設定

カスタム(スクリプト)入力の設定

Splunk は、ユーザー提供のスクリプトからのイベントを受け入れます。スクリプト入力は、`vmstat`、`iostat`、`netstat`、`top` などのコマンドラインツールを使うと便利です。API からのデータ、および機能試験システムや他のリモートデータインタフェース、メッセージキューを得て、`vmstat`、`iostat` などの `app` 状態コマンドからメトリクスおよび状態データを生成します。Splunk App ストアの多くのアプリケーションも、特定のアプリケーションのスクリプト入力を提供します。App は、**Launcher** の **App** の参照で検索できます。

Splunk Web の Splunk Manager を使う、または `inputs.conf` を編集してカスタムスクリプト入力を設定できます。

注意：Windows のプラットフォームでは、パールやパイソンなどのテキストベーススクリプトを、中間ウィンドウズバッチ(.bat)ファイルで有効にできます。

警告：スクリプト入力起動スクリプトは、Splunk の環境を受け継ぐため、スクリプト操作に影響を及ぼす可能性のある環境変数は必ずクリアしてください。環境変数で問題を起こす可能性が高い唯一のものは、ライブラリパス (`linux/solaris/freebsd` で `LD_LIBRARY_PATH` としてもっと一般的に知られています) です。

Splunk Web にスクリプト入力を追加

Splunk Web にスクリプト入力を追加するには、

1. Splunk Web の右上隅にある**管理**をクリックします。
2. システム設定の下の**データ入力**をクリックします。
3. **スクリプト**をクリックします。
4. **新規**をクリックして、入力を追加します。
5. スクリプトへのパスおよびスクリプトランタイムの間隔を秒単位で入力します。
6. 任意で、**ソースタイプ**を設定します。ソースタイプは、イベントに追加されるデフォルトフィールドです。ソースタ

イプは、タイムスタンプやイベント境界などの処理特性の決定に使用します。これを自動的に設定すると、Splunk は、自動的にソースタイプを分類し割り当てます。不明なソースタイプには、代替の名前が与えられます。

7. 任意で、このソースからデータの宛先インデックスを設定します。これをデフォルトに設定すると、データはメインインデックスに送信されます。

inputs.conf によるスクリプト入力の追加

以下の属性を使用して inputs.conf を設定します。

```
[script://$SCRIPT]
interval = X
index = <index>
sourcetype = <iostat, vmstat, etc> OPTIONAL
source = <iostat, vmstat, etc> OPTIONAL
disabled = <true | false>
```

- script は、スクリプトの場所への完全に確認されたパスです。
 - ◆ 最善の方法として、スクリプトが指定されている inputs.conf に最も近い bin/ディレクトリにスクリプトを入力します。つまり、\$SPLUNK_HOME/etc/system/local/inputs.conf を設定する場合、スクリプトを、\$SPLUNK_HOME/etc/system/bin に保存します。\$SPLUNK_HOME/etc/apps/\$APPLICATION でアプリケーションの作業をしている場合、\$SPLUNK_HOME/etc/apps/\$APPLICATION/bin/にスクリプトを保存します。
- interval は、秒単位です。
 - ◆ Splunk は、インスタンスごとにスクリプトを 1 回呼び出します。間隔はスクリプト完了時を基にしています。そのため、スクリプトを 10 分ごとに実行するよう設定し、各スクリプトの完了に 20 分かかると、次の実行は、最初の実行の 30 分後となります。
 - ◆ 継続したデータストリームには、1 を入力します(またはスクリプトの間隔より小さな値)。
 - ◆ ワンショットデータストリームには、-1 を入力します。
 - ◆ **注意:** 間隔を-1 に設定すると、splunk デーモンが再起動するたびにスクリプトを再実行します。
- index は Splunk インスタンスにある任意のインデックスです。
 - ◆ デフォルトは main です。
- disabled は、入力を無効にする場合に True に設定できるブーリアン値です。
 - ◆ デフォルトは True です
- sourcetype および source は、自由な値です。
 - ◆ 指定する値は、sourcetype=または source=フィールドのスクリプトによるデータに追加されます。
 - ◆ これらはオプションの設定です。

スクリプトを継続的に実行する場合は、終了しないようなスクリプトを書き、短い間隔で設定します。これにより、問題がある場合にはスクリプトが再起動されます。Splunk は、生成したスクリプトを追跡し、終了時にスクリプトを停止します。

inputs.conf を使用した例

この例は、UNIX の top コマンドをデータ入力ソースとして使用します。

- 新規のアプリケーションディレクトリを作成して始めます。この例では、scripts/を使用します。

```
$ mkdir $SPLUNK_HOME/etc/apps/scripts
```

- アプリケーションディレクトリの中にある bin/ディレクトリから全てのスクリプトを実行します。

```
$ mkdir $SPLUNK_HOME/etc/apps/scripts/bin
```

- この例では、小さなシェルスクリプト top.sh を使用します。

```
$ #!/bin/sh
```

```
top -bn 1 # linux only - different OSes have different paramaters
```

- スクリプトが実行可能であることを確かめます。

```
chmod +x $SPLUNK_HOME/etc/apps/scripts/bin/top.sh
```

- シェルでスクリプトを実行して機能することをテストします。

```
$SPLUNK_HOME/etc/apps/scripts/bin/top.sh
```

- スクリプトは、1つの top 出力を送信します。
- スクリプトエントリを \$SPLUNK_HOME/etc/apps/scripts/default/ の inputs.conf に追加します。

```
[script:///opt/splunk/etc/apps/scripts/bin/top.sh]
```

```
interval = 5 # run every 5 seconds
```

```
sourcetype = top # set sourcetype to top
```

```
source = script:///./bin/top.sh # set source to name of script
```

```
props.conf
```

props.conf を変更する必要があることがあります。

- デフォルトで、Splunk は、1つの top エントリを複数のイベントに分けます。
- この問題を解決する最も簡単な方法は、出力される前にのみ分けるよう Splunk サーバーに、指示します。

例えば、以下を \$SPLUNK_HOME/etc/apps/scripts/default/props.conf に追加すると、全てのラインが強制的に 1つのイベントになります。

```
[top]
```

```
BREAK_ONLY_BEFORE = <stuff>
```

トップ出力にタイムスタンプがないため、Splunk に現在の時間を教える必要があります、これには props.conf に以下を設定します。

```
DATETIME_CONFIG = CURRENT
```

ホワイトリストまたはブラックリスト専用の受信データ

ホワイトリストまたはブラックリスト専用の受信データ

ホワイトリストおよびブラックリストルールを使用して、ディレクトリ監視時に取り込むファイルを Splunk に指示しま

す。ホワイトリストを定義すると、Splunk はそのリストのファイルのみインデックスします。逆に、ブラックリストを定義すると、Splunk はそのリストのファイルを見逃し、その他全部を取り込みます。ホワイトリストとブラックリストの両方を定義する必要はありません。これらは独立した設定です。両方設定し、両方に一致するファイルがある場合、そのファイルはインデックスされません。例えば、`_blacklist` は、`_whitelist` を無効にします。

注意: ...または*ワイルドカードを使用して入力を定義すると、間接的なホワイトリストを作成します。その後の'_whitelist' 設定が無視されます。

ホワイトリストおよびブラックリストルールは、正規表現構文を使用して、ファイル名/パスの一致を定義します。また、ルールは、設定スタンザ内に含まれている必要があります。例えば、`[monitor://<path>]`では、このスタンザ以外(グローバルエントリ)は無視されます。

データ入力をホワイトリストまたはブラックリストに記載する代わりに、特定のイベントをフィルタし、異なるキューまたはインデックスに送信できます。異なるキューへのイベントのフィルタリングとルーティングおよびイベントの代替のインデックスへのルーティングについて詳しくお読みください、また、クローラ機能を使用して、ファイルシステムに追加されたときに Splunk に自動的にインデックスするファイルとインデックスしないファイルを予め定義できます。

ホワイトリストおよびブラックリストのエントリを、正確な正規表現構文で定義します。“...”ワイルドカードはサポートされていません。

ホワイトリスト(許可)ファイル

Splunk で独占的にインデックスするファイルを定義するには、以下のラインを、`/local/inputs.conf` ファイルの `monitor` スタンザに追加します。

```
_whitelist = $YOUR_CUSTOM_REGEX
```

例えば、Splunk に `.log` 拡張子のファイルのみ監視させたい場合、

```
[monitor:///mnt/logs]
```

```
_whitelist = ¥.log$
```

`¥`(OR)演算子を使用すると、複数のファイルを 1 行でホワイトリストに掲載できます。例えば、`query.log OR my.log` を含むファイル名をホワイトリストに掲載するには、

```
_whitelist = query¥.log$|my¥.log$
```

または、正確な一致をホワイトリストに掲載するには、

```
_whitelist = /query¥.log$|/my¥.log$
```

注意: “\$”を正規表現に付けてラインの終わりを示します。“|”演算子の前後には空白はありません。

ブラックリスト(無視)ファイル

Splunk のインデキシングで除外するファイルを定義するには、以下のラインを、その入力定義される App の `/local/inputs.conf` ファイルの `monitor` スタンザへ追加します。

```
_blacklist = $YOUR_CUSTOM_REGEX
```

重要：無視する各ファイルに対して `_blacklist` ラインを作成した場合、Splunk は、最後のフィルタのみ実行します。

Splunk に、`.txt` の拡張子を持つファイルのみを、無視させて、監視させない場合は、以下のように記述します。

```
[monitor:///mnt/logs]
_blacklist = ¥.(txt)$
```

Splunk に、拡張子が `txt` または `gz` のいずれかのすべてのファイルは無視させて、監視させない場合(これには“|”を使用します)は、以下のように記述します。

```
[monitor:///mnt/logs]
_blacklist = ¥.(txt|gz)$
```

Splunk に、この例で参照する監視入力の下すべてのディレクトリを無視させる場合は、以下のように記述します。

```
[monitor:///mnt/logs]
_blacklist = (archive|historical|¥.bak$)
```

上記の例は、Splunk に、履歴ディレクトリのアーカイブディレクトリ内の `/mnt/logs/` のファイル全てを無視し、`*.bak` で終わるファイル全てを無視するよう指示しています。

Splunk に、特定の文字列を含むファイルは無視させたい場合は、以下のように記述できます。

```
[monitor:///mnt/logs]
_blacklist = 2009022[89]file¥.txt$
```

上記の例では、`/mnt/logs/` の下にある `webserver20090228file.txt` および `webserver20090229file.txt` を無視します。

リストの確認

ホワイトリストおよびブラックリストルールが適切に設定されているかを確認するには、`$SPLUNK_HOME/bin` ディレクトリにある `listtails` ユーティリティを実行します。`Listtails` は、全てのアプリケーションディレクトリにある `inputs.conf` の設定を読み取り、ディレクトリをスキャンし、再起動時に Splunk が監視するファイルの正確なリストを表示します。

`$SPLUNK_HOME/bin` ディレクトリで、以下を実行します。

```
./splunk cmd listtails
```

ログローテーションの処理方法

ログローテーションの処理方法

Splunk は、監視しているファイル (`/var/log/messages` など) が、ローテーションすると (`/var/log/messages1`)、それを認識し、2 回目にローテーションしたファイルを読み込みません。

注意：Splunk は、ログローテート (`tar` または `gzip` などの) により作られたアーカイブファイルを、圧縮されていないオリジナルと同様であると認識しません。これらのファイルを Splunk が監視していると、データが重複します。アーカイブ

ファイルタイプのブラックリストルールを確実に設定し、Splunk がこれらのファイルを新規のログファイルとして読み込まないようにすることができます。または、ログローテートを設定して、これらのファイルを、Splunk に読み込まないよう指示したディレクトリに移動もできます。

Splunk は、以下のアーカイブファイルタイプを認識します。tar、gz、bz2、tar.gz、tgz、tbz、tbz2、zip、z

ブラックリストルールの設定の詳細は、本書の「ホワイトリストとブラックリスト専用受信データ」をご覧ください。

ログローテーションの機能

モニタリングプロセッサは、新規のファイルを選び、ファイルの最初と最後の 256 バイトを読み込みます。このデータは、始点および終点巡回冗長検査(CRC)にハッシュされます。Splunk は以前に確認したファイルの CRC 全てを含むデータベースに対して新規の CRC を点検します。また、Splunk が最後にファイルで読み込んだ場所も保存されます。

CRC 点検から得られる結果には 3 種類あります。

1. データベースでこのファイルに一致する始点および終点 CRC はありません。これは新規のファイルのため、始点から選択および取り込まれます。Splunk は、ファイルを取り込み中に、新規の CRC および `seekPtrs` を伴ったデータベースを更新します。
2. 始点 CRC および終点 CRC が存在しているが、ファイルサイズが Splunk で保存した `seekPtr` を超えています。これはつまり、Splunk で以前そのファイルを確認した後に情報が追加されていることを示します。Splunk はファイルを開き、前回のファイル終点を探して、そこから読み込みを開始します(こうして、Splunk は新しいデータのみを取り込み、すでにあるデータは取り込みません)。
3. 始点 CRC が存在するが、終点 CRC が一致しません。これは、ファイルが Splunk で最後に読み込まれてから変更され、既に読み込まれた部分が異なることを意味します。つまり、Splunk が以前に読み込んだデータも変更されています。この場合、Splunk は、再びファイル全体を読み込む他に手段はありません。

インデキシングとイベント処理

イベントとは何か

イベントとは何か

イベントは、ソフトウェアアプリケーションにより作成されるログファイルのエントリーです。例えば、Web アクティビティログにあるイベントは、以下のとおりです。

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

この例で、01/Jul/2005:2:05:27 -0700 を含む文字列の部分はタイムスタンプです。

Splunk がイベントを処理する(インデキシングと呼ぶ)と、タイムスタンプを引き出し、識別し(無い場合には追加)、適切にマルチラインイベントを処理し、イベントセグメンテーションを行い、便利な一連の標準フィールド(ホスト、ソース、およびソースのタイプ)を自動抽出します。

別のイベントの表示、タイムスタンプ、および Splunk におけるイベントセグメンテーションの方法に関する概要は、ナリッジマネージャマニュアルの[Documentation:Knowledge:Aboutevents|“About events”]をご参照ください。

Splunk では、関心のあるイベントを検索し、Splunk の豊富な統計およびレポートツールを使って、環境内の問題およびトレンドを特定します。保存済み検索による結果ナリッジを保存および再利用したり、詳細なグラフィックレポートを作成したり、特定のイベントやイベントの区分にタグを付けたりすることで、データセットが膨張しても簡単に検索できます。

インデキシング処理のしくみ、カスタマイズおよび調整可能なインデキシング処理、およびタイムスタンプについての詳細は、この章と次の章をお読みください。

インデキシングのしくみ

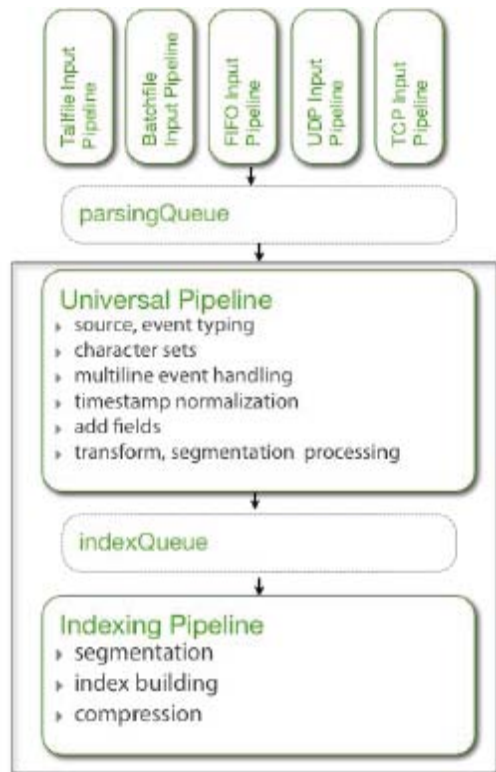
インデキシングのしくみ

インデキシングとは、ユーザーが送信したデータを Splunk が処理する方法のことです。Splunk は、あらゆる種類の時系列データ(タイムスタンプのあるデータ)をインデックスします。データがインデックスされる時、そのタイムスタンプに基づいてイベントに分類されます。

Splunk に送られるすべてのデータは、ユニバーサルパイプラインによりインデックスされます。データは、大きな(10,000 バイト)塊としてユニバーサルパイプラインに送られます。パイプライン処理の一環として、これらの固まりは、イベントに分類されます。まず、改行文字がイベント境界に信号を送ります。次の処理段階で、Splunk は、prps.conf で指定されたラインマーキングルールを適用します。

インデキシングの一環として、イベントは、**セグメント**と呼ばれる区分に分けられます。Splunk は、segmenters.conf に設定可能な分類文字およびその他のルール(セグメント別の最大文字数など)のリストを使用します。

インデキシングは、I/O 集中型プロセスです。多くのデータをインデックスするようシステムを構築している場合、Splunk は考慮が必要です。



Splunk 最適化プロセス

Splunk がデータをインデックスしている間、splunk-optimize プロセスの 1 つ以上のインスタンスを継続的に実行し、インデックスファイルを 1 つにまとめて、データ検索時のパフォーマンスを最適化します。splunk-optimize プロセスは、大量の CPU を使用しますが、使用するのは短時間です。splunk-optimize で同時使用するインスタンスの数は、indexes.conf の maxConcurrentOptimizes の値で変更できますが、通常その必要はありません。

splunk-optimize は、db-hot 上でのみ実行してください。

数値ほど大きい .tsidx ファイル (25 以上) - ./splunk-optimize <directory> を見つけた場合、手動で、warm DB 上で実行できます。

splunk-optimize を頻繁に実行しないと、検索効率に影響します。

インデックスには何があるか

Splunk はインデックスにすべての処理済データを保存します。インデックスは、順々に、\$SPLUNK_HOME/var/lib/splunk のデータベースに保存されます。データベースは、db_<starttime>_<endtime>_<seq_num> と名づけられたディレクトリです。

Splunk には、以下に示す予め設定されたインデックスがあります。

- main : デフォルト Splunk インデックス。指定されない限り、すべての処理済データが個々に保存されます。

- splunklogger : Splunk はこのインデックスの内部ログを追跡します。
- _internal : このインデックスは、Splunk のプロセッサからのメトリクスを含みます。
- sampledata : 少量のサンプルデータが、トレーニング用にここに保存されます。
- _thefishbucket : ファイル処理の内部情報を格納します。
- _audit : ファイルシステム変更モニタ、監査、および全てのユーザ検索履歴からのイベント。

詳細は、本書の「インデックスの管理について」をお読みください。

セグメンテーションによるデータ圧縮の向上

セグメンテーションによるデータ圧縮の向上

ホストフィールドの値の設定方法

ホストフィールドの値の設定方法

イベントの `host` 値は、イベントが作られたネットワークの物理的デバイスの名前です。ホストは特定のデバイスで作られたすべてのデータを最も簡単に検索する方法を提供します。ホストにタグを付けると、共通の機能または設定を持つホストのグループからデータを検索できます。ホストの値は、IP アドレス、ホスト名、または FQDN(完全修飾ドメイン名)です。Splunk は、インデックスするすべてのイベントの `host` 値をインデックスおよび保存します。

デフォルトホストの割り当て

ソースに対して他のホストルールを指定していない場合(ここにある情報および後のトピックスにある情報を使って)、ホストフィールドの値には、特定の Splunk サーバーに送られるすべてのデータに適用するデフォルトが設定されます。デフォルトホスト値は、ネットワークホストのホスト名または IP アドレスです。イベントが発生したサーバーで Splunk を実行すること(これが最も一般的なケースです)は正しく、変更をする必要はありません。データが別のホストから転送された場合、またはアーカイブデータを大量にローディングしている場合、値を変更する必要がある場合があります。

ホストフィールドのデフォルト値を設定するには、Splunk Manager を使う、または `inputs.conf` を編集します。

Splunk Manager を使用したホスト値の設定

Splunk Manager でホストフィールドの値を変更するには、

1. Splunk Web の右上隅にある**管理**をクリックします。
2. **システム設定**をクリックします。
3. **インデックス設定**で、**デフォルトホスト名**を変更します。これにより、他のホスト名を受信しないすべてのイベントに対するホストフィールドの値を設定します。

設定ファイルを使用したホスト値の設定

デフォルトのホスト割り当ては、Splunk をインストールするときに、`inputs.conf` に設定されます。ホストエントリを変更するには、`$SPLUNK_HOME/etc/system/local/inputs.conf` を編集します。

以下は、inputs.conf でホストを割り当てるためのフォーマットです。

```
host = <文字列>
```

* これは `MetaData:Host = <string>` の短縮版です。このイベントにより文字列に指定されるイベントのホストを設定します。"host::" は、ショートカットを使用すると自動的に値の先頭に追加されます。

<文字列>のエントリーを変更して、独自のホスト値を設定します。

他のシステムのデータに対するホスト値を無効にする

セントラルログアーカイブで Splunk を実行している場合、または、その環境で他のホストからコピーしたファイルで作業している場合、デフォルト割り当てを無効にする場合があります。異なるサブディレクトリに各ホストのログアーカイブを隔離するディレクトリ構造になっているときなど、入力のホスト割り当てを、その入力の全データのカスタムホスト値またはソースのパスまたはファイル名の一部の一致に基づいて、定義できます。

イベントデータを使用してホストの値を無効にする

Splunk にイベントを送信する集中ログホストがある場合には、多くのサーバーが関与していることがあります。セントラルログサーバーは、レポートホストと呼ばれます。イベントが発生したシステムは、元となるホスト(または単にホスト)と呼ばれます。この場合、イベント自体の情報に基づいてホストフィールドの値を設定するルールを定義する必要があります。

入力用にホストの値を設定

入力用にホストの値を設定

これらの指示を使用して、特定の設定入力から送られるすべてのデータに対するホスト値を明確に設定します。同じ入力のイベント全てに対してホストを静的に設定、または、ソースの完全なパスを持つ正規表現またはセグメントでホストを動的に設定します。同じ入力で異なるソースまたはソースタイプに異なるホストを割り当てるには、イベントデータに基づいてホストフィールドの値を設定します。

ホストの値を静的に設定

ここで説明する方法は、入力のすべてのイベントに同じホストを割り当てます。

注意：インプットに対するホストフィールドの値を変更すると、その入力に送られる新規のデータにのみ影響します。既にインデックスされたデータに対して Splunk Web に表示されるホストフィールドの値を変更するには、ホストフィールドにタグを付ける必要があります。

Splunk Web を使用してホストの固定値を静的に設定

Splunk Manager のデータ入力セクションでデータ入力を追加するたびにホストフィールドの値を設定します。

1. 新規入力のページのホストセクションで、**継続した値**を選択して、このデータソースから送られる各イベントのホストとして固定値を割り当てます。

2. ホストフィールド値フィールドにホストの値を入力します。

設定ファイルを使用してホストの固定値を設定

inputs.conf を編集して、ホスト値を指定します。\$SPLUNK_HOME/etc/system/local/inputs.conf の適切なスタanzas に host = 属性を含めます。\$SPLUNK_HOME/etc/system/local/または、\$SPLUNK_HOME/etc/apps のカスタムアプリケーションディレクトリの inputs.conf を編集します。設定ファイルについての一般的詳細は、本書の「設定ファイルについて」をご覧ください。

設定

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

入力について詳しくお読みください。

例

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

これは、TCP port 9995 上の 10.1.1.10 から送られるすべてのイベントに対するホストを“webhead-1”として設定します。

ホストの値を動的に設定

この方法は、ソース入力のセグメントからホスト名を抽出する場合に使用します。例えば、インデックスしたいアーカイブディレクトリがあり、そのディレクトリの各ファイルの名前が関連したホスト情報を含んでいる場合、Splunk を使って、その情報を抽出し、それをホストフィールドに割り当てられます。

Splunk Web でホストの値を動的に設定

Splunk Manager のデータ入力セクションでデータ入力を追加するたびにホストフィールドの値を設定します。

1. 新規の入力ページのホストセクションで、以下のいずれかを選択します。

- **パス上の正規表現** : このオプションは、正規表現でホスト名を抽出する場合に選択します。正規表現に、ホストを抽出するための正規表現を入力します。
- **パスのセグメント** : このオプションは、データソースのパスにあるセグメントからホスト名を抽出する場合に選択します。セグメント番号ボックスにセグメント番号を入力します。

設定ファイルでホストの値を動的に設定

inputs.conf 設定時に、動的ホスト抽出ルールを設定できます。以下の属性/値のペアを追加して、ホストフィールドを無効にできます。

host_regex = <正規表現>

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

host_segment = <整数>

- 指定すると、\ / で区切られた指定パスのセグメントが、各入力のホストとして設定されます。
- 値が整数でない場合、または 1 以下の場合、デフォルト host =属性がホストとして設定されます。

例

この例では、ファイルパス上の正規表現を使用して、ホストを設定します。

```
[monitor:///var/log]
```

```
host_regex = /var/log/(¥w+)
```

/var/log/foo.log のイベントに、ホスト名“foo”が与えられます。

この例では、パスのセグメントを使用して、ホストを設定します。

```
[tail://apache/logs/]
```

```
host_segment = 3
```

```
sourcetype = access_common
```

これは、ホスト名を、パス apache/logs の 3 番目のセグメントとして抽出します。

イベントデータに基づいてホストの値を設定

イベントデータに基づいてホストの値を設定

この方法は、イベント自体にあるデータに基づいてイベントに割り当てられたデフォルトホスト名を無効にする場合に使用します。これを行うには、transforms.conf および props.conf を編集しなければいけません。

設定

transforms.conf および props.conf で、動的に抽出されたソースまたはソースタイプのホスト名を設定します。\$SPLUNK_HOME/etc/system/local/、または、\$SPLUNK_HOME/etc/apps/のカスタムアプリケーションディレクトリにあるファイルを編集します。設定ファイルについての一般的詳細は、本書の「設定ファイルについて」をご覧ください。

transforms.conf の編集

カスタムスタンザを、\$SPLUNK_HOME/etc/system/local/transforms.conf に追加します。スタンザを以下のよう

に設定します。

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

データの正しい値を、スタンザ名および正規表現フィールドに入力します。

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

注意: スタンザには固有の識別子を付けてください(そうすることにより、\$SPLUNK_HOME/etc/system/default/transforms.conf にあるスタンザと混乱しません)。

props.conf の編集

\$SPLUNK_HOME/etc/system/local/props.conf にスタンザを作成して、transforms.conf 正規表現を、props.conf のソースタイプに割り当てます。

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

<spec>には、以下が指定できます。

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

\$name は、変換に与える固有の識別子です。

\$UNIQUE_STANZA_NAME は、今 transforms.conf に作成した変換のスタンザ名と一致しなければいけません。

注意: スタンザを定義するときに、props.conf からその他の有効な属性/値のペアを任意に追加します。これは属性を<spec>に割り当てます。例えば、同じ<spec>に設定するカスタム改行ルールがある場合、それらの属性をスタンザに追加します。

例

以下の houseness.log ファイルのイベントは、3 番目の位置にホストを含んでいます。

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

ホスト値を抽出する正規表現を作成して、新しいスタンザを \$SPLUNK_HOME/etc/system/local/transforms.conf: に追加します。

```
[houseness]
DEST_KEY = MetaData:Host
```

```
REGEX = ¥s(¥w*)$
```

```
FORMAT = host:::¥1
```

ここで、変換が呼び出されるよう、transforms.conf スタンザを、\$SPLUNK_HOME/etc/system/local/props.conf にリンクします。オプションで、必要に応じて props.conf から更に属性/値のペアを追加します。

上記の変換は、props.conf の以下のスタンザで有効です。

```
[source::.../housesness.log]
```

```
TRANSFORMS-rhallen=housesness
```

```
SHOULD_LINEMERGE = false
```

上記のスタンザには、追加の属性/値ペア SHOULD_LINEMERGE = false があります。これは、Splunk に、改行で新規イベントを作成するよう指示します。

注意：属性 TRANSFORMS-rhallen の追加の-rhallen は、この変換を他の変換と区別する役目をします。

これでイベントは以下の用に Splunk Web に表示されます。



```
8 6/22/00 4:44:44.000 PM 41602052:17 accepted fflanda
9 6/22/00 4:44:44.000 PM 41602050:29 accepted rhallen
10 6/22/00 4:44:44.000 PM 41602046:53 accepted fflanda
```

イベントにメタデータをダイナミックに割り当てる

イベントにメタデータをダイナミックに割り当てる

複数行イベントのインデックス

複数行イベントのインデックス

多くのイベントログには、1行に1つのイベントを記述するという厳しいフォーマット制限がありますが、ない場合もあります。通常、Splunk はイベント境界を自動認識します。ただし、希望通りにイベント境界認識ができない場合には、props.conf を設定してカスタムルールを設定します。

設定

複数行イベントを設定するには、イベントのフォーマットを調べます。イベントのパターンを割り出し、イベントの始点または終点として設定します。次に、\$SPLUNK_HOME/etc/system/local/props.conf を編集して、データ処理に必要な属性を設定します。

複数行イベントを取り扱う方法は2つあります。

1. イベントストリームをリアルイベントに分けます。これはインデキシングのスピードを著しく速めるため、推奨します。LINE_BREAKER を使用します(下記参照)。

2. イベントストリームをラインに分けて、再び集めます。これは遅いですが、より強力な設定オプションを提供します。

LINE_BREAKER に加えて改行属性を使用します(下記参照)。

以下は、改行ルールを設定する属性の例です。\$SPLUNK_HOME/etc/system/README/props.conf.spec:

TRUNCATE = <non-negative integer>

- デフォルトの最大ライン長さを変更します。
- 切捨てない場合は、0 に設定します(ただし、長すぎるラインは、無駄なデータの兆候)。
- デフォルトは 10000

LINE_BREAKER = <regular expression>

- 設定されていない場合、ローストリームは、\r または \n によって区切られる各ラインのイベントに分けられます。
- 設定されている場合、指定の正規表現でローストリームをイベントに分けます。
- 正規表現は、一致するグループを含んでいなければいけません。
- 正規表現が一致するたびに、最初の一一致したグループの始めの部分は、以前のイベントにないテキストと見なされます。
- 最初に一一致したグループの終わりは、区切り部分の終わりで見なされ、次の文字は、次のイベントの始めと見なされます。
- 例えば、“LINE_BREAKER = ([\r\n]+)”は、デフォルトルールと同等です。
- 最初の一一致グループの内容は、前または次のイベントでは発生しません。
- **注意:** ラインマーキングを使用して個別のラインをイベントに再構築するより、LINE_BREAKER を使用して複数行イベントを区切る方が、処理速度が速くなります。

LINE_BREAKER_LOOKBEHIND = <整数> (100)

- 正規表現ベースの改行コードのデフォルト後方参照を変更します。
- 前回のローチャンクに残りのデータがあるとき、これを見て、正規表現を適用し始めるべき位置 (次のチャンクの連結位置) を判断します。

SHOULD_LINEMERGE = <true/false>

- 正に設定すると、Splunk は、以下の設定属性に基づき、複数の入力ラインを 1 つのイベントに結合します。
- デフォルトは正です。

以下は、SHOULD_LINEMERGE = True のときにのみ使用します。

AUTO_LINEMERGE = <true/false>

- Splunk に、自動学習方法を使用するよう指示し、イベントのどこで改行するかを決定します。
- デフォルトは正です。

BREAK_ONLY_BEFORE_DATE = <true/false>

- 正に設定すると、Splunk は、日付と一致する新規のラインに遭遇した場合のみ、新規のイベントを作成します。

- デフォルトは False です。

BREAK_ONLY_BEFORE = <正規表現>

- 設定した場合、Splunk は、正規表現と一致する新規のラインに遭遇した場合にのみ、新規のイベントを作成します。
- デフォルトは空です。

MUST_BREAK_AFTER = <正規表現>

- 設定した場合、正規表現が現在のラインと一致すると、Splunk が次の入力ラインの新規イベントを作成することが保証されます。
- Splunk は、他のルールが一致すると現在のラインの前で改行する場合があります。

デフォルトは空です。

MUST_NOT_BREAK_AFTER = <正規表現>

- 設定し、現在のラインが正規表現と一致すると、Splunk は、MUST_BREAK_LINE 表現が一致するまでその後のラインを改行しません。
- デフォルトは空です。

MUST_NOT_BREAK_BEFORE = <正規表現>

- 設定し、現在のラインが正規表現と一致すると、Splunk は、現在のラインの前の最後のイベントを分けません。
- デフォルトは空です。

MAX_EVENTS = <整数>

- イベントに追加される入力ラインの最大数を指定します。
- Splunk は、指定された数のラインが読み込まれた後に改行します。
- デフォルトは 256 です。

例

```
[my_custom_sourcetype]
```

```
BREAK_ONLY_BEFORE = ^¥d+¥s*$
```

この例は、ソースタイプが Splunk で *sourcetype::my_custom_sourcetype* となるよう設定または決定されているソースに対して、ファイルにあるイベントを分けるよう Splunk に指示する、または、全ての桁を含むライン全てが新しいイベントの始めであると仮定してストリームするよう Splunk に指示します。

他の例：

以下のログイベントは、同じリクエストの一部である複数のラインを含んでいます。リクエストの区切りは、“Path”です。カスタマーは、ここに示された全てのラインを1つのイベントエントリにすることを希望しています。

```
{{"2006-09-21, 02:57:11.58", 122, 11, "Path=/LoginUser
Query=CrmId=ClientABC&ContentItemId=TotalAccess&{"2006-09-21, 02:57:11.60", 122, 15,
```

```
"UserData:<User CrmId="clientabc" UserId="p12345678"><EntitlementList></{"2006-09-21,
02:57:11.60", 122, 15, "New Cookie: SessionId=3A1785URH117BEA&Ticket=646A1DA4STF896EE
```

この複数ラインイベントを適切にインデックスするには、設定にある Path 区切りを使用します。以下を、`$SPLUNK_HOME/etc/system/local/props.conf` に追加します。

```
[source::source-to-break]
SHOULD_LINEMERGE = True
BREAK_ONLY_BEFORE = Path=
```

このコードは、Splunk に、イベントのラインを連結し、`path=`の前でのみ改行するよう指示します。

キャラクタセットエンコードの設定

キャラクタセットエンコードの設定

sed によるデータの匿名化

sed によるデータの匿名化

このユーティリティは、`sed` スクリプトを使用し、インデックスの時間に、文字列を取り替えまたは代用することで、データを匿名化できるようにします。

ほとんどの UNIX ユーザーは、ファイルを読み込み、入力をコマンドのリストで指定通りに変更する Unix ユーティリティ `sed` を使うことができるはずですが、ここで、`sed` のような構文を使って、`props.conf` のデータを匿名化できます。

注意：`$SPLUNK_HOME/etc/system/local` で、`props.conf` のコピーを編集または作成してください。

`props.conf` で `sed` スクリプトを定義する

`props.conf` スタンザで、`SEDCMD` を使用して `sed script` を記述します。

```
[<stanza_name>]
SEDCMD-<class> = <sed script>
```

`stanza_name` は、匿名化または変換で変更する `host`、`source`、または `sourcetype` に制限されています。

`sed script` は、インデックスタイムの、`_raw` フィールドにのみ適用します。Splunk は現在、取替(s)および文字代用(y)の `sed` コマンドのサブセットをサポートしています。

注意：`props.conf` の変更を有効にするには、Splunk を再起動する必要があります。

文字列を正規表現一致に取り替える

sed 取替の構文

```
SEDCMD-<class> = s/<regex>/<replacement>/flags
```

- `regex` は、PERL 正規表現です。

- replacement は、正規表現を取り替えるための文字列で、後方参照に“\n”を使用します(n は 1 桁)。
- flags は、全ての一致を取り替える“g”、または指定の一致を取り替えるための数のどちらかです。

例

ここでは社会保障番号とクレジットカード番号を含むデータをインデックスするとします。インデックスタイムで以下の値を隠して、イベントでは最後の 4 桁だけが分かるようにします。props.conf スタンザは以下のようになります。

```
[source:.../accounts.log]
SEDCMD-accounts = s/ssn=¥d{5}(¥d{4})/ssn=xxxxx¥1/g
s/cc=(¥d{4}-){3}(¥d{4})/cc=xxxx-xxxx-xxxx-¥2
```

ここで、アカウントイベントの社会保障番号に、ssn=xxxxx6789、クレジットカード番号に、cc=xxxx-xxxx-xxxx-1234 が表示されます。

文字を代用

sed 文字代用の構文は、

```
SEDCMD-<クラス> = y/<文字列 1>/<文字列 2>/
```

文字列 1 で起こる各文字を、文字列 2 の文字に代用します。

例

ここでは、インデックスするファイル abc.log があり、イベント内の小文字“a”、“b”、“c”を大文字の“A”、“B”、“C”に代用するとします。props.conf に以下を追加します。

```
[source:.../abc.log]
SEDCMD-abc = y/abc/ABC/
```

これで、source=“*/abc.log”を検索した場合、データに小文字“a”、“b”、“c”は見つからないはずですが。Splunk は、各“a”を“A”に、各“b”を“B”に、各“c”を“C”に代用しました。

Splunk 構文解析をサポートするカスタムログの設定

Splunk 構文解析をサポートするカスタムログの設定

タイムスタンプ

タイムスタンプのしくみ

タイムスタンプのしくみ

Splunk は、タイムスタンプを使用して、イベントを時間と関連付け、Splunk Web で時系列ヒストグラムを作成し、検索の時間範囲を設定します。タイムスタンプはインデックスタイムのイベントに割り当てられます。ほとんどのイベントは、ローイベントデータの情報に基づいて、タイムスタンプ値が割り当てられます。イベントがタイムスタンプ情報を含んでいない場合、Splunk は、インデックスされるたびにイベントにタイムスタンプ値を割り当てようと試みます。Splunk は、タイムスタンプ値を、_time フィールドに保存します(UTC 時間フォーマット)。

新規データ追加時の考慮事項

Splunk が自動で行う設定以外でデータにタイムスタンプの設定が必要な場合、そのタイムスタンプ抽出を設定したら、再度データをインデックスする必要があります。新しいデータ入力は、生産 Splunk インスタンスに追加する前に、正しいタイムスタンプを得るまで何回かデータを一掃してインデックスし直さなければいけない場合に備えて、「サンドボックス」 Splunk インスタンス(または別のインデックス)でテストすることをお勧めします。

タイムスタンプ割り当ての優先順位ルール

Splunk は、以下の優先順位を使用して、イベントにタイムスタンプを割り当てます。

1. 確実な TIME_FORMAT が提供されている場合、それを使用して、イベント自体にある時間または日付を探します。
ローデータに 1 つ以上のタイムスタンプ値があるイベントには、位置を指定したタイムスタンプを使用します。
2. TIME_FORMAT が提供されていない場合、または一致が見つからない場合、イベント自体の時間または日付を自動的に特定しようと試みます。
ローデータの一つ以上のタイムスタンプ値があるイベントには、位置を指定したタイムスタンプ抽出を使用します。
3. イベントに時間または日付がない場合、同じソースで最近のイベントのタイムスタンプを使用します。
ソースにあるイベントのどれにも日付が無い場合、
4. ソース(またはファイル)名を探します(イベントに時間がなければいけません)。
5. ファイルソースで、ファイル名に時間または日付が特定されない場合には、ファイルの変更時間を使用します。
6. 他のタイムスタンプが見つからない場合、現在のシステム時間(イベントのインデックス時間)にタイムスタンプを設定します。

タイムスタンプの設定

ほとんどのイベントは、特別なタイムスタンプの取り扱いが必要ありません。一部のソースおよび分散デプロイメントでは、タイムスタンプのフォーマットを設定して、イベントからタイムスタンプを抽出する必要があります。Splunk のタイムスタンプ抽出プロセッサは props.conf を編集して設定します。props.conf で利用可能なタイムスタンプの設定についての完全な情報は、この概要をご覧ください。

また、Splunk のタイムスタンプ抽出プロセッサを設定して以下が行えます。

- タイムゾーンオフセットの適用
- ヨーロッパの日付フォーマットの認識
- 複数のタイムスタンプを持つイベントから、正しいタイムスタンプの引き出し
- インデキシング性能の向上

最後に、Splunk に新規のタイムスタンプフォーマットを認識するよう設定します。

タイムスタンプ認識の設定

タイムスタンプ認識の設定

Splunk は、タイムスタンプを使用して、イベントを時間と関連付け、Splunk Web で時系列ヒストグラムを作成し、検索の時間範囲を設定します。タイムスタンプはインデックス時間にイベントに割り当てられます。

ほとんどのイベントは、ローイベントデータの情報に基づいて、タイムスタンプ値が割り当てられます。イベントにタイムスタンプ情報が含まれていない場合、Splunk は、インデックス時にイベントにタイムスタンプ値を割り当てようと試みます。Splunk は、タイムスタンプ値を、`_time` フィールドに保存します(UTC 時間フォーマット)。

ほとんどのイベントには特別なタイムスタンプの処理は必要ありません。設定せずに、Splunk に処理させることができます。

タイムスタンプ割り当ての優先順位ルール

タイムスタンプ割り当ての優先順位ルール

1. 確実な `TIME_FORMAT` が提供されている場合、それを使用して、イベント自体にある時間または日付を探します。ローデータに 1 つ以上のタイムスタンプ値があるイベントには、位置を指定したタイムスタンプ抽出を使用します。
2. `TIME_FORMAT` が提供されていない場合、または一致が見つからない場合、イベント自体の時間または日付を自動的に特定しようと試みます。
3. イベントに時間または日付がない場合、同じソースで最近のイベントからタイムスタンプを使用します。
4. ソースにあるイベントのどれにも日付が無い場合、ソース(またはファイル)名を探します。
5. ファイルソースで、ファイル名に時間または日付が特定されない場合には、ファイルの変更時間を使用します。
6. 他のタイムスタンプが見つからない場合、現在のシステム時間(イベントのインデックス時間)にタイムスタンプを設定します。

タイムスタンプの設定

ほとんどのイベントには特別なタイムスタンプの処理は必要ありません。設定せずに、Splunk に処理させることができます。

一部のソースおよび分散デプロイメントには、タイムスタンプのフォーマットを設定して、イベントからタイムスタンプを抽出する必要があります。Splunk のタイムスタンプ抽出プロセッサは `props.conf` を編集して設定します。

`props.conf` を編集して、Splunk がタイムスタンプを認識する方法を設定します。Splunk は、`strptime()` フォーマ

ットを使用して、イベントのタイムスタンプ値を特定します。Splunk がタイムスタンプとして認識する項目を、`TIME_FORMAT=キー`にある `strptime()` フォーマットを設定して指定します。

注意： イベントに複数のタイムスタンプがある場合は、位置を指定したタイムスタンプ抽出により正しいタイムスタンプを認識するよう Splunk を設定します。

`$(SPLUNK_HOME)/etc/system/README/props.conf.example` を例として使用、または、自作の `props.conf` を作成します。`$(SPLUNK_HOME)/etc/system/local/` または、`$(SPLUNK_HOME)/etc/apps/` にあるカスタムアプリケーションディレクトリにある `props.conf` のコピーに設定変更を行います。

`props.conf` にある以下の属性のいずれかを設定し、Splunk のタイムスタンプ認識を設定します。キーの完全仕様は、`$(SPLUNK_HOME)/etc/system/README/props.conf.example` を参照してください。

[<spec>]

`DATETIME_CONFIG = <filename relative to $(SPLUNK_HOME)>`

`MAX_TIMESTAMP_LOOKAHEAD = <integer>`

`TIME_PREFIX = <regular expression>`

`TIME_FORMAT = <strptime-style format>`

`TZ = <posix timezone string>`

`MAX_DAYS_AGO = <integer>`

`MAX_DAYS_HENCE = <integer>`

[<spec>]

- <spec>は、タイムスタンプ抽出の適用対象を示します。以下のいずれかに適用します。
- <sourcetype>、イベントのソースタイプ
- `host::<host>`、<host>はイベントのホストを示す
- `source::<source>`、<source>はイベントのソースを示す
- イベントが、<spec>の値に一致するデータを含む場合、スタンザで指定されたタイムスタンプルールがそのイベントに適用します。
- 追加のスタンザを追加して、様々なイベントのタイプのタイムスタンプ認識をカスタマイズします。

`DATETIME_CONFIG = <filename relative to $(SPLUNK_HOME)>`

- Splunk のタイムスタンププロセッサの設定に使用するファイルを指定します(デフォルトでは、Splunk は、`$(SPLUNK_HOME)/etc/datetime.xml` を使用)。
- カスタム `datetime.xml` を使用するには、`datetime.xml` を参照する全てのキーにあるカスタムファイルに正しいパスを指定します。
- `DATETIME_CONFIG = NONE` を設定してタイムスタンププロセッサの実行を回避します。
- `DATETIME_CONFIG = CURRENT` を設定してインデックス時に現在のシステムタイムをイベントに割り当てます。

`MAX_TIMESTAMP_LOOKAHEAD = <integer>`

- Splunk がタイムスタンプを検索するイベントの長さ(文字数)を指定します。
- デフォルトは 150 字です。
- 0 に設定すると、イベントのインデックスタイムで現在のシステムタイムを割り当てます。

TIME_PREFIX = <regular expression>

- イベントのタイムスタンプ直前の空白を指す正規表現を使用します。
- 例えば、タイムスタンプが、Time=という句の後にある場合、正規表現は、イベントのこの部分に一致させます。
- タイムスタンププロセッサは、イベントの TIME_PREFIX の後のタイムスタンプのみを検索します。
- デフォルトはなし(空)です。

TIME_FORMAT = <strptime-style format>

- strptime()フォーマット文字列を指定して、日付を抽出します。
- 抽出するタイムスタンプにある要素の順番に一致する順番で strptime() 値を設定します。
- Splunk のタイムスタンププロセッサは、一致する TIME_PREFIX 値の直後に TIME_FORMAT の処理を始めます。
- イベント内タイムゾーンはサポートしません。
- TIME_FORMAT は、一致する TIME_PREFIX の後に読み込みを始めます。
- <strptime-style format>値は、時間、分、月、および日を含んでいる必要があります。
- デフォルトは空です。

TZ = <タイムゾーン文字列>

- zoneinfo TZID データベースの値を使用してタイムゾーン設定を指定します。
- 詳しい内容および例は、タイムゾーンオフセットの設定の仕方を参照します。
- デフォルトは空です。

MAX_DAYS_AGO = <整数>

- 抽出日を有効にする過去最大日数(現在の日付からの)を指定します。
- 例えば、MAX_DAYS_AGO = 10 の場合、10 日前より古い日付は無視されます。
- デフォルトは 1000 です。

注意：データが 1000 日より古い場合は、設定を行う必要があります。

MAX_DAYS_HENCE = <整数>

- 抽出日を有効にする未来最大日数(現在の日付から)を指定します。
- 例えば、MAX_DAYS_HENCE = 3 の場合、3 日より先の日付は無視されます。
- デフォルト値(2)は、明日の日付を許可します。

注意：マシンの日付セットが間違っている場合、または、1 日先のタイムゾーンにある場合、この値を 3 以上に設定してください。

強化された strptime()サポート

props.conf のタイムスタンプ構文解析を、TIME_FORMAT=キーで設定します。Splunk は、追加のフォーマット(ミクロ

秒、ミリ秒、様々な時間幅フォーマット、および互換性に合わせた追加のフォーマットが可能)をサポートする Unix `strptime()` の強化バージョンを導入しています。下記のサポートされている追加の `strptime()` フォーマットのリストの表をご覧ください。

以前のバージョンでは、Splunk は、標準の Linux `strptime()` 変換仕様のみを使用してタイムスタンプの構文解析をしていました。現在、標準の Linux `strptime()` フォーマットに加えて、Splunk の `strptime()` は、以下の日付-時間フォーマットの認識をサポートしています。

<code>%N</code>	GNU 日付-時間ナノ秒用。幅： <code>%3N</code> = ミリ秒、 <code>%6N</code> = ミクロ秒、 <code>%9N</code> = ナノ秒の提供により、秒単位以下の構文解析を指定します。
<code>%Q%q</code>	Apache Tomcat のミリ秒、ミクロ秒用。 <code>%Q</code> および <code>%q</code> は、幅を指定すると、どの時間の変換も可能です。
<code>%l</code>	12 時間時計フォーマットの時間用。 <code>%l</code> が、 <code>%S</code> または <code>%s</code> (“%H:%M:%S.%l” のような) の後に現れる場合、ミリ秒を意味する <code>log4cp</code> を受け入れます。
<code>%</code>	標準の UNIX 日付フォーマットタイムスタンプ用。
<code>%v</code>	BSD および OSX 標準日付フォーマット用。
<code>%z</code> 、 <code>%::z</code> 、 <code>%:::z</code>	GNU libc サポート
<code>%o</code>	AIX タイムスタンプサポート用(<code>%Y</code> の別名として使用される <code>%o</code>)。
<code>%p</code>	AM または PM の地元言語。(注意：ない場合があります。)

strptime()フォーマット表現の例

以下は、処理をする `strptime()` 表現を伴ったサンプルデータフォーマットの一部です。

1998-12-31	<code>%Y-%m-%d</code>
98-12-31	<code>%y-%m-%d</code>
1998 years, 312 days	<code>%Y years, %j days</code>
Jan 24, 2003	<code>%b %d, %Y</code>
January 24, 2003	<code>%B %d, %Y</code>
q 25 Feb '03 = 2003-02-25 q	<code>%d %b '%y = %Y-%m-%d </code>

例

データには、以下のように、簡単に認識できる抽出用タイムスタンプを含んでいる場合があります。

```
...FOR: 04/24/07 PAGE 01...
```

`props.conf` のエントリは、以下のとおりです。

```
[host::foo]
TIME_PREFIX = FOR:
TIME_FORMAT = %m/%d/%y
```

データには、以下の用に Splunk がタイムスタンプとして構文解析するその他の情報を含んでいる場合があります。

```
...1989/12/31 16:00:00 ed May 23 15:40:21 2007...
```

Splunk は、日付を Dec 31, 1989 として抽出しますが、これは便利ではありません。この場合、`props.conf` を設定して、`host::foo:`からのイベントから正しいタイムスタンプを抽出します。

```
[host::foo]
TIME_PREFIX = %d{4}/%d{2}/%d{2} %d{2}:%d{2}:%d{2} %w+%s
TIME_FORMAT = %b %d %H:%M:%S %Y
```

この設定は、`host::foo`のタイムスタンプ全てが同じフォーマットであると想定しています。潜在的なタイムスタンプエラーを回避するために、`props.conf` スタンザができる限り多様になるよう設定します。

また、Splunk のタイムスタンプ抽出プロセッサで以下を行う設定ができます。

- タイムゾーンオフセットの適用
- ヨーロッパの日付フォーマットの認識
- 複数のタイムスタンプを持つイベントから、正しいタイムスタンプの引き出し
- インデキシング性能の向上

最後に、Splunk に新規のタイムスタンプフォーマットを認識するよう設定します。

複数のタイムスタンプを持つイベントのタイムスタンプ抽出を設定

複数のタイムスタンプを持つイベントのタイムスタンプ抽出を設定

イベントが、認識可能なタイムスタンプを複数含んでいる場合、Splunk に特定のタイムスタンプを使用するよう指示できます。これは特に、`syslog` ホストチェーンデータを含むイベントをインデックスするときに便利です。

`props.conf` を編集して、位置を指定したタイムスタンプ抽出を設定します。

`props.conf` で位置を指定したタイムスタンプ抽出の設定

```
TIME_PREFIX =および MAX_TIMESTAMP_LOOKAHEAD =キーを props.conf の [<spec>]スタンザに追加して、タイムスタンプがイベント内のどこにあるかと認識するよう Splunk を設定します。MAX_TIMESTAMP_LOOKAHEAD =の値を設定し、Splunk にイベントのタイムスタンプを検索する長さを指示します。TIME_PREFIX =の値を設定し、Splunk に検索するタイムスタンプの最初の文字パターンを指示します。
```

注意：`$SPLUNK_HOME/etc/system/README/props.conf.example` を例として使用する、または、自作の `props.conf` を作成します。`$SPLUNK_HOME/etc/system/local/`または、`$SPLUNK_HOME/etc/apps/`にあるカスタムアプリケーションディレクトリにある `props.conf` のコピーに設定変更を行います。

```
$SPLUNK_HOME/etc/apps/.
```

例：イベントが以下のような場合

```
1989/12/31 16:00:00 ed May 23 15:40:21 2007 ERROR UserManager - Exception thrown Ignoring
unsupport
```

May 23 15:40:21 2007 のタイムスタンプを特定するには、

props.conf を以下のように設定します。

```
[source::/Applications/splunk/var/spool/splunk]
TIME_PREFIX = %d{4}/%d{2}/%d{2} %d{2}:%d{2}:%d{2} %w+%s
MAX_TIMESTAMP_LOOKAHEAD = 44
```

注意：抽出するタイムスタンプを探すために必要なイベントのみを検索するよう MAX_TIMESTAMP_LOOKAHEAD = の値を設定して、タイムスタンプ抽出の速度を最適化します。この例では、MAX_TIMESTAMP_LOOKAHEAD = は、イベントを 44 文字まで検索するよう最適化されています。

タイムスタンプにタイムゾーンオフセットを適用

タイムスタンプにタイムゾーンオフセットを適用

異なるタイムゾーンからデータをインデックスする場合、タイムゾーンオフセットを使用して、検索時に正確に相互関連するようにします。イベントのホスト、ソースまたはソースタイプに基づいてタイムゾーンオフセットを設定できます。

props.conf でタイムゾーンオフセットを設定します。Splunk は、デフォルトで、以下の順番でルールを使用してタイムゾーンオフセットを適用します。

1. ローイベントデータのタイムゾーンを使用します(PST、-0800 など)。
2. TZ が、props.conf のスタanzas に設定されている場合で、イベントがスタanzas で指定されたホスト、ソース、またはソースタイプに一致する場合に TZ を使用します。
3. イベントをインデックスする Splunk サーバーのタイムゾーンオフセットを使用します。

props.conf でタイムゾーンオフセットを設定

\$SPLUNK_HOME/etc/system/README/props.conf.example を例として使用する、または、自作の props.conf を作成します。\$SPLUNK_HOME/etc/system/local/または、\$SPLUNK_HOME/etc/apps/にあるカスタムアプリケーションディレクトリにある props.conf のコピーで設定変更を行います。

TZ =キーを、props.conf にあるホスト、ソース、またはソースタイプのタイムスタンプ設定スタanzas に追加して、タイムゾーンオフセットを設定します。Splunk の TZ =キーは、zoneinfo TZID を認識します(zoneinfo (TZ)データベースにある全てのタイムゾーン TZ ID をご覧ください。)TZID に対する TZ =値を、希望のホスト、ソース、またはソースタイプのタイムゾーンオフセットに設定します。

例

この例では、正規表現 nyc.* に一致するホスト名のイベントのタイムゾーンオフセットを、東部標準時に設定します。

```
[host::nyc*]
TZ = US/Eastern
```

この例では、パス/mnt/ca/...にあるソースのイベントのタイムゾーンオフセットを、太平洋標準時に設定します。

```
[source::/mnt/ca/...]
```

TZ = US/Pacific

zoneinfo (TZ)データベース

zoneinfo データベースは、公に管理されているタイムゾーン値のデータベースです。

- UNIX バージョンの Splunk は、インストールしている UNIX ディストリビューションに含まれている TZ データベースに依存しています。ほとんどの UNIX ディストリビューションは、`/usr/share/zoneinfo` ディレクトリにデータベースを保存します。
- Solaris バージョンの Splunk は、`/usr/share/lib/zoneinfo` ディレクトリに TZ 情報を保存します。
- Windows バージョンの Splunk は、TZ データベースのコピーに同封されて納品されます。

`props.conf` で `TZ =`として設定できる値については、`zoneinfo (TZ)`データベースを参照してください。

ローカライズされたタイムスタンプフォーマット(ヨーロッパなど)の認識

ローカライズされたタイムスタンプフォーマット(ヨーロッパなど)の認識

デフォルトで、Splunk のタイムスタンプは、ブラウザのロケールに従ってフォーマットされています。ブラウザが US 英語を選ぶよう設定すると、タイムスタンプは、アメリカ式の `MM/DD/YYYY:HH:MM:SS` で表示されます。ブラウザが大陸フランス語またはイギリス英語を選ぶよう設定すると、タイムスタンプは、伝統的なヨーロッパ日付表示形式 `DD/MM/YYYY:HH:MM:SS` を使用して Splunk の設定に表示されます。

ブラウザのロケール設定は、ブラウザ固有です。

ブラウザのロケールを無効にする

特定のセッションに Splunk が使用するロケールは、Splunk のアクセスに使用する url を変更するだけで、変えられます。Splunk 4 URL は、`http://host:port/locale/...`の形式に従います。例えば、Splunk にアクセスしてログインするとき、url は、`http://hostname:8000/en-US/account/login`として現れます。US 英語文字列を、例えば `http://hostname:8000/en-GB/account/login`に変えてロケールを変更できます。このセッションは有効で、その間、タイムスタンプをイギリス英語で受け入れます。

ローカライズされていない Splunk インタフェースに対してロケールを要求すると、`Invalid language Specified` のメッセージが表示されます。

Splunk にタイムスタンプを認識するよう指示

Splunk にタイムスタンプを認識するよう指示

Splunk は、デフォルトでほとんどのタイムスタンプを認識します。詳細は、Splunk のタイムスタンプ抽出方法をお読みください。Splunk が特定のタイムスタンプを認識しない場合、`train dates` コマンドを使用して、Splunk にパターンを教えることができます。`train dates` の出力は、固有のタイムスタンプ抽出を設定する `datetime.xml` および `props.conf` に追加可能な正規表現です。

`train` コマンドを使うと、タイムスタンプ、フィールド、およびソースタイプの新しいパターンを Splunk にインタラクティブに教えることができます。Train および一緒に使用できる異なる引数についての詳細は、`$SPLUNK_HOME/bin` の

train ヘルプページを参照してください。

```
./splunk help train
```

重要：タイムスタンプを `props.conf` で設定できないときに限り `train dates` を使用してください。

train dates でタイムスタンプを設定する手順

Splunk に新しいタイムスタンプパターンを教えるには、以下の手順で行います。

1. タイムスタンプデータのサンプルを、プレーンテキストファイルにコピーします。

Splunk は、このテキストファイルのパターンに基づいてタイムスタンプのパターンを学習します。

2. `train dates` コマンドを実行します。

この機能はインタラクティブです。プロンプトすると、タイムスタンプデータを含むテキストファイルへのパスを提供します。コマンドは、タイムスタンプの正規表現を生成します。

3. カスタム `datetime.xml` を作成します。

`train` コマンドの出力を、`datetime.xml` ファイルのコピーにコピーします。

注意：デフォルト `datetime.xml` ファイルは、`$SPLUNK_HOME/etc/datetime.xml` に保存されています。このファイルは変更しないでください。代わりに、デフォルト `datetime.xml` を、`$SPLUNK_HOME/etc/apps` または `$SPLUNK_HOME/etc/system/local` にあるカスタムアプリケーションディレクトリにコピーしてください。詳細は本書のアプリケーションについてのトピックスを参照してください。

4. ローカル `props.conf` を編集します。

関連のあるスタンザに、カスタム `datetime.xml` ファイルへのパスを含めます。

```
./splunk [command]
```

train dates コマンドを実行

`train` コマンドはインタラクティブな CLI ツールです。Splunk に新しいデータフォーマットを学習させるには、ファイルおよびパターンを確実に提供する必要があります。その後、Splunk は、ユーザーが `datetime.xml` に追加する文字列に戻ります。

1. Splunk に、新しいタイムスタンプを認識させるには、`$SPLUNK_HOME/bin` に移動して、以下を入力します。

```
./splunk train dates
```

Splunk は、処置を指示します。

```
-----  
What operation do you want to perform? (default=learn)  
-----
```

```
Enter choice: [Learn]/Test/Quit >
```

デフォルト処置は、学習です。

2. トレーニング操作を行うには、“L”、“l”、または“learn”と入力してから、入力をクリックします。
Splunk は、トレーニングで使用するサンプルファイルを与えるよう指示します。

```
Enter full filename from which to learn dates > sampling.txt
```

3. Splunk サーバーのファイルのパスを入力します(この手順は、タブコンプリートを許可しません)。
Splunk は、サンプルの最初のラインを表示し、そのタイムスタンプの値を教えるよう訊ねます。

```
-----  
Interactively learning date formats.  
-----  
  
INSTRUCTIONS: If a sample line does not have a timestamp, hit Enter.  
If it does have a timestamp, enter the timestamp separated by commas  
in this order: month, day, year, hour, minute, second, ampm, timezone.  
Use a comma as a placeholder for missing values. For example, for a  
sample line like this "[Jan/1/08 11:56:45 GMT] login", the input  
should be: "Jan, 1, 08, 11, 56, 45, , GMT" (note missing AM/PM).  
Spaces are optional.  
  
SAMPLE LINE 1:  
Tue Jul 10 21:23:06 PDT 2007 Received Trade 330 with detail user: user3456 date: date:  
23:06 action: sell 3583 MNAG @ 42  
  
-----  
Enter timestamp values as: month, day, year, hour, minute, second, ampm, timezone.  
> 7, 10, 2007, 9, 23, 06, pm, PDT
```

4. 月、日、年、時、分、秒、am/pm、タイムゾーンの値を入力します(上述参照)。これは、Splunk に、入力した値を、
タイムスタンプの指定された部分として認識するよう教えます。

値が十分でない場合、Splunk は以下を表示します。

```
Learned pattern.  
-----  
If you are satisfied that the timestamps formats have been learned, hit control-c.  
-----
```

5. control+C を押しながら c を押すと、Splunk は以下を表示します。

```
Patterns Learned.  
  
It is highly recommended that you make changes to a copy of the default datetime.xml file.  
For example, copy "/Applications/splunk/etc/datetime.xml" to "/Applications/splunk/etc/system/local/ In  
that custom file, add the below timestamp definitions, and add the pattern names to timePatterns and  
datePatterns list.
```

For more details, see <http://www.splunk.com/doc/latest/admin/TrainTimestampRecognition>

```
-----
<define name="trainwreck_1_date" extract="day,month,year,">
<text><![CDATA[:\d+\s(w+)\s(\d+)\s(w+)\s(\d+)]]></text>
</define>

<define name="trainwreck_1_time" extract="hour,minute,second,ampm,">
<text><![CDATA[(\d+):(\d+):(\d+)\s(w+)]]></text>
</define>
-----
```

What operation do you want to perform? (default=learn)

Enter choice: [Learn]/Test/Quit > q

6. 出力を確認します。

- 正確な場合、終了します。それから、出力をコピーして、次のセクションを続けます。
- 正確でない場合、学習の選択を入力し、Splunk に教えなおしてください。

カスタム datetime.xml の作成

train 実行後、Splunk は、新しいタイムスタンプパターンを説明する文字列を出力します。

カスタム datetime.xml ファイルで以下を実行します。

7. train から戻された文字列を<timePatterns>および<datePatterns>スタンザの前に貼り付けます。

8. <use name="define name"/>を、<timePatterns>および<datePatterns>の両方に<define name="string"文字列を付けて追加します。

例：

以下の train dates 出力には、

```
<define name="_utceepoch" extract="utceepoch">
<text><![CDATA[(?<=^|[¥s#, "=¥(¥[¥|¥{]) (?:1[01]|9)¥d{8}|^@[¥da-fA-F]{16,24})(?:¥d{3})?(?![
¥</define>
```

変更された datetime.xml ファイルは、以下のようになります。

```
<define name="_utceepoch" extract="utceepoch">
<text><![CDATA[(?<=^|[¥s#, "=¥(¥[¥|¥{]) (?:1[01]|9)¥d{8}|^@[¥da-fA-F]{16,24})(?:¥d{3})?(?![
¥</define>
<timePatterns>
<use name="_time"/>
<use name="_hmttime"/>
<use name="_hmttime"/>
```

```

<use name="_dottime"/>
<use name="_combdatetime"/>
<use name="_utcePOCH"/>
</timePatterns>
<define name="_utcePOCH" extract="utcePOCH">
<text><![CDATA[(?<=^[^\s#, "\=\\(\[\|\}\])?(?:1[01]|9)\d{8}|^@[\da-fA-F]{16,24})(?:\d{3})?(?![
\</define>
<datePatterns>
<use name="_usdate"/>
<use name="_isodate"/>
<use name="_eurodate"/>
<use name="_bareurlitdate"/>
<use name="_orddate"/>
<use name="_combdatetime"/>
<use name="_masheddate"/>
<use name="_masheddate2"/>
<use name="_utcePOCH"/>
</datePatterns>

```

ローカル props.conf の編集

カスタムタイムスタンプを適用するには、Splunk に新しい datetime.xml の場所を知らせる必要があります。

props.conf を以下の手順で変更します。

9. DATETIME_CONFIG キーを、タイムスタンプ設定スタンプに追加します。
10. DATETIME_CONFIG の値を、カスタム datetime.xml のパスに設定します。

注意：スタンプに設定可能なすべてのキーを参照してタイムスタンプ認識を設定してください。

例：

この例では、カスタム datetime.xml を、ホスト "london" のイベントに適用します。

```

[host::london]
DATETIME_CONFIG = /etc/system/local/datetime.xml

```

props.conf を編集して、ホスト、ソース、ソースタイプのカスタムタイムスタンプ抽出パターンを設定できます。

インデキシング性能を向上させるタイムスタンプ抽出の調整

インデキシング性能を向上させるタイムスタンプ抽出の調整

props.conf を編集して Splunk のタイムスタンプ抽出を調整します。Splunk のタイムスタンププロセッサがイベントを

検索する長さを調整する、またはタイムスタンププロセッサをオフにしてインデキシングの速度を速めます。

注意： \$SPLUNK_HOME/etc/system/README/props.conf.example を例として使用する、または、自作の props.conf を作成します。\$SPLUNK_HOME/etc/system/local/または、\$SPLUNK_HOME/etc/apps/にあるカスタムアプリケーションディレクトリにある props.conf のコピーの設定変更を行います。設定ファイルの一般的な情報は、設定ファイルの機能についてをご覧ください。

タイムスタンプ前方参照の調整

タイムスタンプ前方参照は、タイムスタンププロセッサがタイムスタンプを探すために検索するイベントの長さ(文字数)を決定します。タイムスタンプスタンザにある MAX_TIMESTAMP_LOOKAHEAD =キーの値(文字数)を設定して、タイムスタンププロセッサが検索する長さを調整します。

注意： 各タイムスタンプスタンザに対して MAX_TIMESTAMP_LOOKAHEAD =に異なる値に設定できます。

タイムスタンププロセッサがイベントを検索するデフォルト文字数は、150 です。MAX_TIMESTAMP_LOOKAHEAD = に低い値を設定して、イベントがインデックスされる速度を上げます。この操作は、イベントの最初の部分でタイムスタンプが発生する場合に行ってください。

リアルタイムでイベントがインデックスされる場合、タイムスタンプ前方参照を止めて (MAX_TIMESTAMP_LOOKAHEAD = 0 に設定)、Splunk の全体のインデキシング性能を向上させます。これにより、Splunk はイベントのタイムスタンプ検索を行わず、イベントのタイムスタンプにインデキシング時間(現在のシステム時間を使用)を設定します。

例：

この例では、タイムスタンププロセッサに、ソース foo のイベントを 20 文字検索するよう指示します。

```
[source::foo]
MAX_TIMESTAMP_LOOKAHEAD = 20
...
```

タイムスタンププロセッサを止める

タイムスタンププロセッサを完全に止めて、インデキシング性能を大幅に向上させます。DATETIME_CONFIG =キーをスタンザに追加し、値を NONE に設定することで、タイムスタンプスタンザで指定されたホスト、ソース、ソースタイプに一致するイベントのタイムスタンプ処理を止めます。タイムスタンプ処理が止めると、Splunk はイベントデータから抽出するタイムスタンプを検索しません。Splunk は、代わりにイベントのタイムスタンプにインデキシング時間(現在のシステム時間を使用)を設定します。

例：

この例では、ソース foo から送られるイベントのタイムスタンプ抽出を止めます。

```
[source::foo]
DATETIME_CONFIG = NONE
...
```

ユーザーと役割について

ユーザーと役割について

Splunk をエンタープライズライセンスで実行している場合、パスワードを持つユーザーを作成し、作成した役割をユーザーに割り当てられます。無料ライセンスの Splunk は、ユーザー認証をサポートしていません。

Splunk は、1つのデフォルトユーザー、**管理者ユーザー**が付属しています。管理者ユーザーのデフォルトパスワードは、**changeme**です。パスワードが意味するように、Splunk をインストールしたら即座にこのパスワードを変更してください。

役割について

役割とは、例えば、入力を追加する、保存済み検索を編集する許可を誰かに与える(または与えない)を定義する能力のセットです。さまざまな能力が以下、および`$SPLUNK_HOME/etc/system/README/authorize.conf.spec` にリストアップされています。役割が存在すると、その役割をユーザーに割り当てることができます。

また、ユーザーを作成するたびに、そのユーザーの役割が自動作成されます。これは、特定のユーザーとオブジェクト(保存済み検索やレポートなど)の共有をサポートするために行います。その理由は、オブジェクト所有権が役割システムの一部であるからです。

デフォルトで、Splunk には、以下の役割が予め定義されています。

- Admin – この役割には、利用可能なすべての能力が割り当てられています。
- Power – この役割は、全ての共有オブジェクト(保存済み検索等)やアラート、イベントのタグ、およびその他の類似タスクを編集できます。
- User – この役割は、自作の保存済み検索の作成・編集、検索の実行、管理事項の編集、イベントタイプやその他の類似タスクの編集ができます。

許可されていない文字

ユーザー名および役割には、空白、コロン、フォーワードスラッシュは使えません。

ユーザーの追加と役割の割り当て

ユーザーの追加と役割の割り当て

このトピックスでは、新規ユーザーの作成方法と、既存のユーザーのプロパティ(パスワードなど)の変更方法について説明します。

Splunk Web によるユーザーの追加と編集

- Splunk Web で、**管理**をクリックします
- **ユーザー**をクリックします
- **新規**をクリックまたは既存のユーザーを編集します。
- そのユーザーの新しいまたは変更する情報を指定します。
- このユーザーに、既存の役割または複数の役割を割り当て、**保存**をクリックします。

ユーザーを作成すると、Splunk は、そのユーザーの役割を自動作成します。次に、役割を編集して、そのユーザーに与える Splunk のアクセス権を指定します。

CLI を使用したユーザーの追加と編集

- パスワード changeme2 を持つ新規の管理者ユーザーを追加する
 - ◆ `./splunk add user admin2 -password changeme2 -role administrator -auth admin:changeme`
- 既存のユーザーパスワードを fllanda に変更する
 - ◆ `./splunk edit user admin -password fllanda -role administrator auth admin:changeme`

Splunk Web を使用した役割の追加と編集

- Splunk Web の管理をクリックします
- 役割をクリックします。
- 新規をクリック、または既存の役割を編集します。
- この役割の新しいまたは変更する情報を指定します。特に以下ができます。
 - ◆ 検索フィルタでこの役割が検索できるデータを制限できます(詳細は、以下の「フィルタフォーマットの検索」をご覧ください)。
 - ◆ この役割で検索可能な時間のウィンドウを制限できます。
 - ◆ この役割が他の役割から能力を継承できるかどうかを指定できます。
 - ◆ この役割に対して個別の能力を選択できます。
 - ◆ デフォルトでこの役割が検索するインデックスを指定できます。
 - ◆ この役割が特定のインデックスに制限されるかどうかを指定できます。
- 保存をクリックします。

注意：複数の役割を持つメンバーは、最も弱い権限を持つ役割から能力を継承します。

authorize.conf を使用した役割の追加と編集

authorize.conf を編集して役割を設定します。役割は能力のリストにより定義されます。また、役割を使用して、各役割に対して検索フィルタを設定することにより細かいアクセス制御を作成できます。

警告： \$SPLUNK_HOME/etc/system/default/authorize.conf の役割は編集または削除しないでください。削除すると、管理能力が破壊されます。\$SPLUNK_HOME/etc/system/local/、または \$SPLUNK_HOME/etc/apps/ のカスタムアプリケーションディレクトリにあるファイルの設定変更を行います。設定ファイルの一般的な情報は、設定ファイルについてをご覧ください。

以下の属性/値ペアを、\$SPLUNK_HOME/etc/system/default/authorize.conf に追加します。

```
[role_$ROLE_NAME]
$CAPABILITY1 = enabled
$CAPABILITY2 = enabled
```

...

```
importRoles = $OTHER_ROLE
```

```
srchFilter = $SEARCH_STRING
```

- role_ \$ROLE_NAME:
 - ◆ 役割に与えり名前。例えば、security、compliance、ninja など。
- \$CAPABILITY1:
 - ◆ 下表に示される能力。役割には複数の能力が指定できます。
- importRoles = <役割>:
 - ◆ 設定すると、現在の役割は、<role>からすべての能力を継承します。
- srchFilter = <検索>:
 - ◆ 詳細なアクセス制御にこのフィールドを使用。この役割の検索は、この表現によりフィルタされます。
- srchTimeWin = <文字列>
 - ◆ この役割が実行する検索の最大時間範囲。
- srchDiskQuota = <整数>
 - ◆ この役割に属すユーザーの検索ジョブに使用するディスクの最大空き容量 (MB)。
- srchJobsQuota = <整数>
 - ◆ この役割のメンバーが同時に実行する検索の最大数。

注意: authorize.conf に変更を行った後は、Splunk を再起動する必要があります。そうしないと、新しい役割は、役割リストに現れません。

フィルタフォーマットの検索

srchFilter/検索フィルタフィールドには、以下の検索用語を含めることができます。

- source=
- host=およびホストタグ
- eventtype=およびイベントタイプタグ
- sourcetype=
- 検索フィールド
- ワイルドカード
- 複数の用語を使用する場合は OR、検索をより限定する場合は AND を使用します。

注意: 複数の役割を持つメンバーは、最もゆるい権限を持つ役割から能力を継承します。検索フィルタの場合、ユーザーに異なる検索フィルタを持つ役割が割り当てられると、すべてのフィルタが適用されます。

検索用語に以下を含めることはできません。

- インデックス
- 保存済み検索
- 時間演算子
- 正規表現

- Splunk Web が上書きできるフィールドまたは修飾子

Splunk Web で役割をユーザーに割り当て

authorize.conf で役割を作成すると、Splunk Web でユーザー(複数可能)にその役割を割り当てます。

- 右上端にある**管理**リンクをクリックします。
- 次に、**ユーザー**リンクをクリックします。
- 既存のユーザーを編集、または新規のユーザーを作成します。
- **役割**リストから割り当てる役割を選択します。
 - ◆ authorize.conf で作成したすべてのカスタム役割がここに一覧表示されます。

重要 : Splunk Web 内で既存のユーザー/グループ役割 LDAP マッピングを変更(および保存)する場合、Splunk Web に現在ログインしているすべてのユーザーは、即座に強制ログアウトされるため、作業を進めるには再びログインしなければなりません。これは、役割マッピングを変更することによりアクセスを持たなくなったユーザーが確実にアクセスできないようにするためです。

authoriza.conf で役割を作成する例

この例では、デフォルトの役割ユーザーおよび全員から能力を継承する役割 Ninja を作成します。Ninja は、アラートを作成できない(保存済み検索のみ)ことを除いて Power とほぼ同じ能力を持ちます。また、Ninja は、host=fflanda の検索が制限されています。

```
[role_Ninja]
edit_save_search = enabled
schedule_search = enabled
edit_eventtype = enabled
edit_role_search = enabled
edit_local_search = enabled
savesearch_tab = enabled
edit_tags = enabled
importRoles = User;Everybody
srchFilter = host=fflanda
```

利用可能な能力一覧

このリストは役割に対して利用可能な能力を示しています。このリストの最新バージョンは、authorize.conf で確認してください。管理者は、このリストにあるすべての能力を持っています。

```
[role_Admin]
edit_user           = CLI/UI のユーザー情報を変更
edit_search_server = ユーザーに $SPLUNK_HOME/etc の xml config ファイルに書き込む能力を与える
delete_user        = UI/CLI のユーザー削除
```

change_authentication	= 認証設定を保存できる
bounce_authentication	= UI/CLI に認証をリロードする
delete_by_keyword	= 検索削除演算子にアクセス
license_tab	= ライセンスタブにアクセス
edit_alert_action	= アラート対応の変更
edit_roles	= 役割に対するユーザーマッピングの変更
edit_deployment_server	= デプロイメントサーバー設定の変更
edit_deployment_client	= デプロイメントクライアント設定の変更
indexes_edit	= インデックス設定の変更
edit_input_defaults	= デフォルト入力設定の変更
edit_monitor	= 入力監視設定の変更
edit_scripted	= スクリプト入力設定の変更
edit_splunktcp	= tcp 経由の分散データ設定の設定
edit_splunktcp_ssl	= tcp ssl 設定の設定
edit_tcp	= tcp 入力設定の変更
edit_udp	= udp 入力設定の変更
edit_server	= server.conf のサーバー設定の変更
edit_web_settings	= web.conf 設定の変更
edit_forwarders	= フォワーディング側の設定変更
use_file_operator	= ファイルシステムを検索するファイル演算子の使用
request_auth_token	= 他のユーザーに対する認証トークンの取得
rest_apps_management	= REST エンドポイント経由でアプリケーション管理
rest_properties_get	= REST サービス/プロパティの読み込み
rest_properties_set	= REST サービス/プロパティの書き込み
admin_all_objects	= システムのすべてのオブジェクト(ユーザーオブジェクト、検索)を管理する能力
importRoles	= この役割がインポートする能力によるその他の役割 (パワーおよびユーザー役割からインポートする管理者ユーザー)
srchFilter	= この役割に対して Splunk が表示および変更でき制限

Splunk によるユーザー認証の設定

Splunk によるユーザー認証の設定

エンタープライズライセンスの Splunk では、デフォルトで Splunk のビルトイン認証が有効化されています。Splunk の認証は、組織の必要性に応じて、ユーザーの追加、ユーザーの役割の割り当て、およびこれらの役割へのカスタム権限の授与を許可します。

組織が LDAP を使用する場合、そこで定義したユーザーおよびグループを Splunk 認証の代わりに使用する場合は、そのようにできます。

LDAP によるユーザー認証の設定

LDAP によるユーザー認証の設定

Splunk は、その内部認証サービスまたは既存の LDAP サーバーによる認証に対応サポートしています。

Splunk は、LDAP v2 および LDAP v3 をサポートしていますが、LDAP 参照はサポートしていません。LDAP v3 は、使用されるデフォルトプロトコルです。参照を返す LDAP サーバーに対する認証方法についての情報は、Splunk Community Wiki を確認してください。

処理の概要

このトピックスでは、以下を行う手順を説明します。いずれの手順も Splunk で LDAP 認証の使用が必要です。

- Splunk で LDAP 認証を使用するための設定
- 既存の LDAP グループを Splunk 役割に割り当てる
- 任意で、LDAP サーバーが SSL 使用を要求する、または、アクティブディレクトリ(AD)を使用する場合に、CA をインポートする
- 続行する前にこのトピックスの最後に記述される「Splunk と LDAP について知っておくべき事項」をお読みください。

ユーザー管理

Splunk で LDAP 認証を使用するように変更すると、Splunk 内でユーザー管理は行われません。したがって、LDAP サーバー内でユーザーを管理しなければいけません。例えば、

- LDAP ユーザーを Splunk 役割に追加するには、LDAP サーバーの LDAP グループにユーザーを追加します。その後、Splunk Web で、**管理 > 認証方法**を選択して、**認証設定のリロード**をクリックします。
- ユーザーの役割メンバーシップを変更するには、LDAP サーバー上でそのユーザーがメンバーである LDAP グループを変更します。その後、Splunk で、**管理 > 認証方法**を選択して、**認証設定のリロード**をクリックします。
- Splunk 役割からユーザーを削除するには、LDAP サーバー上で LDAP グループからユーザーを削除します。その後、Splunk で、**管理 > 認証方法**を選択して、**認証設定のリロード**をクリックします。

重要 : Splunk が、あらゆる LDAP メンバーシップの変更も認識するには、認識設定をリロードする必要があります。これには、ユーザーの追加または削除も含まれます。これは、Splunk 起動時に、Splunk が LDAP ユーザー情報をキャッシュするためです。

LDAP の設定

このトピックスでは、Splunk Web で LDAP を設定する方法について説明します。authentication.conf を編集して設定する場合、ここで authentication.conf の完全な例をご覧になれます。また、他の設定例は、Splunk Community Wiki のトピックス、「参照を返す LDAP サーバーに対する認証」でも見ることができます。

設定ファイルで認証を設定し、デフォルトの Splunk 認証に返す場合、最も簡単な方法は、既存の authentication.conf を移動し(*.disabled に名前を変更するのも良い)、Splunk を再起動することです。こうすると、後で戻る場合に以前の設定

が変わらずに維持されます。

ユーザーおよびグループベース DN の決定

Splunk に LDAP 設定を割り当てる前に、ユーザーおよびグループベース DN、または独自の名前を見つけ出します。DN は、認証情報が保存されているディレクトリの位置です。すべての情報が、各ユーザーのエントリーに保存されている場合、その DN もまた同じである必要があります。ユーザーのグループメンバーシップ情報が異なるエントリーに保管されている場合、グループ情報が保存されているディレクトリのサブツリーを特定する異なる DN を入力します。

この情報を入手できない場合は、LDAP 管理者に連絡して相談してください。

Splunk Web による LDAP の設定

最初に、認証ストラテジとして LDAP を設定します。

1. 右上隅にある**管理**をクリックします。
2. システムコンフィギュレーションで、**認証方法**をクリックします
3. Splunk で LDAP を使う設定をクリックします。

次に、LDAP 設定を入力します。

4. 設定用の LDAP ストラテジ名を定義します。名前には LDAP および空白は使用できません。
5. LDAP 設定を保存すると、ストラテジ名が認証ストラテジを設定ドロップダウンに追加されます。
6. LDAP サーバーのホスト名を指定します。Splunk サーバーがそのホスト名を解決できるよう確認してください。
7. Splunk が LDAP サーバーとの接続に使用するポートを指定します。

- デフォルトで、LDAP は TCP ポート 389 をリッスンします。
- LDAPS(SSL を伴う LDAP)のデフォルトは、ポート 636 です。

8. **SSL 有効化**を選択して、SSL を有効にします。

- **注意** : LDAP サーバーの SSL も必ず有効化してください。

9. **バインド DN** を入力します。

- これは、LDAP サーバーを結びつける独特の名前です。
- これは通常、管理者または管理ユーザです。
- このユーザーには、Splunk に追加するすべての LDAP ユーザーに対するアクセスが必要です。

10. **バインディングユーザーのバインド DN パスワード**を入力し、確認します。

11. **ユーザーベース DN** を指定します。複数のユーザーベース DN エントリーは、セミコロンで区切って指定します。

- Splunk は、この属性を使用してユーザー情報を位置付けています。
- **注意** : この属性を設定しないと、認証は機能しません。

12. ユーザーをフィルタするオブジェクトクラスの**ユーザーベースフィルタ**を指定します。

- デフォルト値は `objectclass=*` で、ほとんどの設定で機能します。
13. **グループベース DN** を指定します。複数のユーザーベース DN エントリーは、セミコロンで区切って指定します。
- LDAP のユーザーグループの位置
14. **グループベースフィルタ** を入力します。
- この属性はグループ名を定義します。
 - デフォルト値は `objectclass=*` で、ほとんどの設定で機能します。
 - Splunk は、グループベースフィルタとして `GID` も受け入れ可能です。
15. **ユーザー名を定義するユーザー名属性** を入力します。
- **注意**：ユーザー名属性にホワイトスペースは使用できません。ユーザー名は大文字と小文字を区別します。
 - アクティブディレクトリでは、これは `sAMAccountName` です。
 - 値 `uid` は、ほとんどの設定で機能します。
16. **ユーザーのリアル名属性**(共有名とも呼ぶ)を指定します。
- 値 `displayName` または `name` は、ほとんどの設定で機能します。
17. **グループ名属性** を入力します。
- ユーザーまたはグループが同じツリーで定義されている場合のみ設定します。
 - これは通常 `cn` です。
18. **グループメンバー属性** を指定します。
- これは通常、`member` または `memberOF` です。メンバーシップがグループエントリーまたはユーザーエントリーにあるかで異なります。
19. **グループマッピング属性** を入力します。
- メンバーエントリーに `dn` 文字列が含まれない場合のみこの値を設定します。ただし、通常はこのフィールドを空白のままにしておくことができます。
 - このフィールドに入力する場合、通常 `dn` を入力します。
20. **pageSize** の値を入力します。
- これは、一度に返すレコードの数を決定します。
 - 0 を入力するとページングを無効化し、LDAPv2 に戻ります。Sun LDAP に接続するためには、必ず `pageSize` に 0 を設定します。
21. **Failsafe ユーザー名** を指定します。
- これで、LDAP サーバーに到達不可能な場合、Splunk への認証が可能になります。
 - **注意**：このユーザーは、Splunk 内で管理者権限があります。
22. **failsafe ユーザーの Failsafe パスワード** を入力し、確認します。

既存の LDAP グループに Splunk 役割を割り当てる

LDAP サーバー経由で認証するよう Splunk を設定したら、既存の LDAP グループを作成した役割に割り当てます。グループを使用しない場合は、ユーザーを個別に割り当てることができます。

注意：ユーザーまたはグループのいずれかを割り当てられますが、両方の割り当てはできません。グループを使用している場合、Splunk にアクセスするすべてのユーザーが適切なグループのメンバーである必要があります。グループは、メンバーの中で最高レベルの役割から能力を継承します。

すべてのユーザー/グループは、Splunk Manager の **ユーザー** ページで表示されます。適切なユーザーまたはグループをクリックして、ユーザー役割を定義します。

重要： Splunk Web 内で既存のユーザー/グループ役割 LDAP マッピングを変更(および保存)する場合、Splunk Web に現在ログインしているすべてのユーザーは、即座に強制ログアウトされるため、作業を進めるには再びログインしなければなりません。

これは、役割マッピングを変更することによりアクセスを持たなくなったユーザーが確実にアクセスできないようにするためです。

LDAP 設定のテスト

Splunk インストールが LDAP サーバーに接続できない場合は、以下のトラブルシューティングの手順を試してください。

23. userBaseFilter および groupBaseFilter 用に追加したすべてのカスタム値を削除します。

24. \$SPLUNK_HOME/var/log/splunk/splunkd.log に認証エラーがないか確認します。

25. ldapsearch を実行して、指定の変数が機能するかテストします。

```
ldapsearch -h "<host>" -p "<port>" -b "<userBaseDN>" -x -D "<bindDN>" -W"
```

```
ldapsearch -h "<host>" -p "<port>" -b "<groupBaseDN>" -x -D "<bindDN>" -W"
```

注意： Solaris では、検索にフィルタを追加する必要があります。

```
ldapsearch -h "<host>" -p "<port>" -b "<groupBaseDN>" -x -D "<bindDN>" "<groupBaseFilter>" -W"
```

CA のインポート

注意： セキュリティのある LDAP 経由で AD に接続するときが、必ず CA を追加してください。

Splunk の LDAP に自作の CA を使うよう設定するには、以下の操作を行います。

1. ルート CA cert を、**BASE-64 encoded X.509** フォーマットでエクスポートします。
2. 以下のラインを \$SPLUNK_HOME/etc/openldap/ldap.conf に追加します。
TLS_CACERT \$SPLUNK_HOME/etc/openldap/certs/\$YOUR_CERT_NAME
TLS_CACERTDIR \$SPLUNK_HOME/etc/openldap/certs
3. ディレクトリ \$SPLUNK_HOME/etc/openldap/certs を作成します。
4. \$SPLUNK_HOME/etc/openldap/certs/\$YOUR_CERT_NAME にエクスポートした CA cert を移動します。
5. Splunk を再起動します。

6. Splunk Web で、**管理 > 認証方法**を選択します。

- ページの下にある**認証設定のリロード**をクリックします。

7. これで、指定した AD グループを Splunk の対応する役割を割り当てることができます。

例

この例では、LDIF を取得し、`authentication.conf` を設定する手順を説明します。また、上述の説明に従って、これらの設定を Splunk Web に入力できます。

注意 : LDAP サーバーの詳細は異なります。LDAP サーバー設定を確認し、`authentication.conf` 属性を環境に適応させてください。

完全な例は、[\[\[Documentation:Admin:authentication.conf|authentication.confhere\]\]](#)をご覧ください。また、その他の設定例は、Splunk Community Wiki トピックスの「[参照を返す LDAP サーバーに対する認証](#)」をご覧ください。

LDIF の取得

`authentication.conf` の設定には、ユーザーLDIF およびグループ LDIF の両方が必要です。

ユーザーLDIF

注意 : Windows システムでは、AD サーバーから `ldifde` コマンドで `ldifs` を抽出できます。

```
ldifde -f output.ldif
```

`ldifde` コマンドは、AD のすべてのエントリーをエクスポートします。その後、シンプルテキストエディタでファイルを開き、適切なエントリーを検索します。

以下のコマンド(自作の `ou` および `dc` を使用)を実行して、ユーザーLDIF を取得します。

```
# ldapsearch -h ldaphost -p 389 -x -b "ou=People,dc=splunk,dc=com" -D  
"cn=bind_user" -W
```

Solaris の場合は、以下のように記述します。

```
# ldapsearch -h ldaphost -p 389 -x -b "ou=People,dc=splunk,dc=com" -D  
"cn=bind_user" -W
```

これは以下を返します。

```
# splunkadmin, People, splunk.com  
dn: uid=splunkadmin,ou=People, dc=splunk,dc=com  
uid: splunkadmin  
givenName: Splunk  
objectClass: top  
objectClass: person  
objectClass: organizationalPerson
```

```
objectClass: inetorgperson
sn: Admin
cn: Splunk Admin
```

グループ LDIF

以下のコマンド(自作の ou および dc を使用)を実行して、グループ LDIF を取得します。

```
# ldapsearch -h ldaphost -p 389 -x -b "ou=groups,dc=splunk,dc=com" -D
"cn=bind_user" -W
```

これは以下を返します。

```
# SplunkAdmins, Groups, splunk.com
dn: cn=SplunkAdmins,ou=Groups, dc=splunk,dc=com
description: Splunk Admins
objectClass: top
objectClass: groupofuniqueNames
cn: SplunkAdmins
uniqueMember: uid=splunkadmin,ou=People, dc=splunk,dc=com
```

authentication.conf の設定

以下の指示に従って、authentication.conf を設定します。\$SPLUNK_HOME/etc/system/local/、または \$SPLUNK_HOME/etc/apps/にあるカスタムアプリケーションディレクトリでこのファイルを編集します。設定ファイルの一般的な情報は、「設定ファイルのしくみ」をご覧ください。

Splunk Web d ையை設定するには、上記の指示をご覧ください。

認証タイプの設定

デフォルトで、Splunk は、Splunk の認証タイプを使用します。それは、[authentication] スタンザで変更します。

```
[authentication]
authType = LDAP
authSettings = ldaphost
```

- authType = LDAP を設定して、LDAP を有効にします。
- authSettings を、LDAP 設定スタンザに割り当てます(以下)。

LDAP のサーバーエントリに割り当てる

ここで、LDIF を authentication.conf の属性/値に割り当てます。

```
[ldaphost]
host = ldaphost.domain.com
pageSize = 0
```

```
port = 389
SSLEnabled = 0
failsafeLogin = admin
failsafePassword = admin_password
bindDN = cn=bind user
bindDNpassword = bind_user_password
groupBaseDN = ou=Groups,dc=splunk,dc=com;
groupBaseFilter = (objectclass=*)
groupMappingAttribute = dn
groupMemberAttribute = uniqueMember
groupNameAttribute = cn
realNameAttribute = displayName
userBaseDN = ou=People,dc=splunk,dc=com;
userBaseFilter = (objectclass=*)
userNameAttribute = uid
```

役割の割り当て

スタンプを設定して、`authorize.conf` で作成したカスタム役割を、`authentication.conf` で Splunk アクセス用に有効化した LDAP グループに割り当てることができます。

```
[roleMap]
Admin = SplunkAdmins;
ITUsers = ITAdmins;
```

ユーザーディレクトリの割り当て

ユーザーディレクトリを Splunk 役割に割り当てる必要がある場合、`groupBaseDN = userBaseDN` を設定して実行できます。例えば、

```
[supportLDAP]
SSLEnabled = 0
bindDN = cn=Directory Manager
bindDNpassword = #####
failsafeLogin = failsafe
failsafePassword = #####
groupBaseDN = ou=People,dc=splunksupport,dc=com;
groupBaseFilter = (objectclass=*)
groupMappingAttribute = dn
groupMemberAttribute = uniqueMember
groupNameAttribute = cn
```

```
host = supportldap.splunksupport.com
pageSize = 0
port = 389
realNameAttribute = cn
userBaseDN = ou=People,dc=splunksupport,dc=com;
userBaseFilter = (objectclass=*)
userNameAttribute = uid
[roleMap]
Admin = Gina Lee;
```

保存済み検索を LDAP に変換

既に作成した保存済み検索を新しい LDAP で使えるように変換する場合は、以下の操作を行います。

1. 以下を入力して、Splunk CLI でユーザーID を特定します。

```
./splunk list user
```

2. 次に、`$SPLUNK_HOME/etc/system/local/savedsearches.conf` を変更し、各スタanzas の `userid=` フィールドを `ldap userid` に切り替えます。
3. これが機能するかどうかをテストするには、LDAP `userid` フォーマットを確認できる 1 つの保存済み検索を LDAP ユーザーとして作成し、次に、既存の保存済み検索に変更を加えます。
4. `savedsearches.conf` を変更したら、必ず Splunk を再起動してください。

Splunk と LDAP について知っておくべき事項

Splunk で LDAP との作業を設定するときは、以下に注意してください。

- Splunk Web および `authentication.conf` のエントリは、大文字と小文字を区別します。
- Splunk はスクローリングをサポートしません。SUN/iPlanet ディレクトリサーバー(バージョン 5.x および 6.x) のようなスクローリングを使用する LDAP サーバーは、`pageSize` を 0 に設定してページングを無効にします。
- Splunk は、一度に 1 つの LDAP サーバーでのみ作業します。
- Splunk は、(エンドユーザ)匿名バインドをサポートしません。このため、最低限の権限を持つユーザーの作成が必要な場合があります。
- Splunk Web は、最大 499 の LDAP グループを表示できます。
 - ◆ Splunk が不要なグループをリストアップしないように、`groupBaseFilter` を使用します。例 : `groupBaseFilter = ((cn=SplunkAdmins)(cn=SplunkPowerUsers)(cn=Help Desk))`
 - ◆ 499 グループ以上の役割割り当てが必要な場合は、`authentication.conf` を手動で編集できます。

```
[roleMap]
Admin = SplunkAdmins;
Power = SplunkPowerUsers;
User = Help Desk;
```

- LDAP ストラテジ名に[LDAP]は使用できません。
- LDAP から Splunk 認証に変更した後は、ログインするために再起動が必要です。
- ユーザーとグループが同じベースに存在している場合、userBaseDN の値を groupBaseDN と同じ値にできません。改善策として、groupBaseDN(またはもう片方から)レベルを 1 つ削除します。例：userBaseDN = cn = Users, dc=domain, dc=com の場合、groupBaseDN = dc =domain, dc=com と設定します。
- Splunk の認証モジュールは、ドミノ LDAP または Apache ディレクトリで機能しません。
-

Splunk で PAM または RADIUS 認証を使用するための設定

Splunk で PAM または RADIUS 認証を使用するための設定

Splunk は、3 つの認証システム、Splunk のビルトインシステム、LDAP、およびスクリプト認証 API のサポートが標準装備されています。スクリプト認証システムで、PAM や RADIUS など、既に搭載されている認証システムを持つインタフェースを Splunk で使えるように設定できます。authentication.conf を使用して認証を設定します。

スクリプト認証の最新情報は、`$SPLUNK_HOME/etc/share/splunk/authScriptSamples/`の README ファイルをご覧ください。このディレクトリには、PAM および RADIUS 用のサンプル、および各認証システム用の authentication.conf のサンプルがあります。

注意：これらのスクリプトはサンプルのため、使用する環境で機能させるには編集が必要です。

スクリプト認証に関する既知の問題

- 現在スクリプト認証は、分散検索で機能しません。
- ユーザーレベルの権限はすべてのユーザーに与えられます。Splunk Web の管理セクションを使用して、ユーザーに適切な Splunk 役割を割り当てます。
 - ◆ `$SPLUNK_HOME/etc/share/splunk/authScriptSamples/`にもユーザーマッピングスクリプトのサンプルがあります。これは、カスタマイズしないと機能しないようデザインされているため、使用するには、スクリプトを環境に適応させる必要があります。

設定

authentication.conf でスクリプト管理を設定します。PSM を使用している場合は、“etc/pam.d/pamauth”にあるシステムの pamauth ファイルも編集する必要があります。

authentication.conf の編集

以下の設定を `$SPLUNK_HOME/etc/system/local/`(またはカスタムアプリケーションディレクトリ)の authentication.conf に追加して、特定のスクリプトを有効にします。また、`$SPLUNK_HOME/etc/share/splunk/authScriptSamples/`から authentication.conf のサンプルをコピーできます。

[authentication]スタanzaヘッディングの下で、scripted を認証タイプに指定します。

```
[authentication]
authType = Scripted
authSettings = script
```

[script] スタンザヘッディングの下で、スクリプト変数を設定します。

```
[script]
scriptPath = $SPLUNK_HOME/bin/python
$SPLUNK_HOME/share/splunk/authScriptSamples/<scriptname>
scriptSearchFilters = 1
```

ユーザーに割り当てられた役割の検索フィルタを有効にする場合は、`scriptSearchFilters` を 1 に設定します。0 で無効化します。

スクリプトで必要な場合は、任意で、`[cacheTiming]` スタンザを追加します。これらの設定を使用して、Splunk がアプリケーションを呼び出す頻度を調整します。各呼び出しは、秒単位で特定されたタイムアウトがあります。指定すると、キャッシングは発生しません。

```
[cacheTiming]
userLoginTTL = 1
searchFilterTTL = 1
getUserInfoTTL = 1
getUserTypeTTL = 1
getUsersTTL = 1
```

スクリプトコマンド

スクリプト認証は、スクリプトで使用するための以下のコマンドを含みます。以下は、入力および出力を含むこれらのコマンドの記述的リストです。

- `userlogin`: ユーザー名/パスワードペアでログイン
 - ◆ 入力: `username=<username> --password=<password>` (stdin で渡す)
 - ◆ 出力: `status=<status_bit> --search_filter=<search_filter>` (オプション)
`--authToken=<tok>` (オプション) 成功 (または失敗)
- `getUserType`: このコマンドは、Splunk 内の役割に対応 (管理者、パワー、ユーザーなど)
 - ◆ 入力: `--username=<username> --authToken=<tok>` (オプション)
 - ◆ 出力: `--status=<status_bit> --role=<role>` (管理者など)
- `getUserInfo`: ユーザ情報の取得
 - ◆ 入力: `--username=<username> --authToken=<tok>` (オプション)
 - ◆ 出力: `--status=<status_bit>`
`--userInfo=<userId>;<username>;<realname>;<role>`

補足の呼び出し

- getUsers
 - ◆ 入力: --authToken=<tok> (オプション)
 - ◆ 出力: --status=<status_bit>
--userInfo=<userId>;<username>;<realname>;<role>
--userInfo=<userId>;<username>;<realname>;<role>....

詳細呼び出し

- checkSession
 - ◆ 入力: --authToken=<tok> (オプション)
 - ◆ 出力: --status=<status_bit>
- getSearchFilter = <role>
 - ◆ このコマンドは、Splunk 内の役割に対応(管理者、パワー、ユーザーなど)
 - ◆ 入力: --username=<username> --authToken=<tok> (オプション)
 - ◆ 出力: --status=<status_bit> --search_filter=<filter> (複数指定可能--検索フィルタ)

すべての出力は、以下のいずれかの<status_bit>で始まります。

- success
 - ◆ コマンドは正確に成功しました。
- tmp_fail
 - ◆ 認証プラグインの一時的失敗。処理を続行します。
- auth_fail
 - ◆ 認証失敗。ユーザーセッションを終了します。

PAM 認証

PAM を使用する場合で README の手順を実行後に認証できない場合は、システムが pamauth config をサポートするためのエントリを追加しているか確認します。/etc/pam.d/pamauth を編集して以下のラインを入力します。

```
auth sufficient pam_unix.so
```

CLI を使用したユーザアカウントの削除

CLI を使用したユーザアカウントの削除

userdata 引数の後に ./splunk clean を入力して、Splunk インストールからすべてのユーザーデータ(ユーザーアカウント)を削除します。これは、Splunk に含まれるデフォルトユーザー(管理、パワー、ユーザ)を除くすべてのユーザーを削除します。

警告: ユーザーデータの削除は元に戻せません。誤ってユーザーデータを削除した場合は、手動でアカウントを再追加する必要があります。

システムのすべてのユーザーアカウントを削除する場合

```
./splunk clean userdata
```

システムのユーザーアカウントを削除し、Splunk で確認プロンプトを表示しないよう強制する場合

```
./splunk clean userdata -f
```

インデックスの管理

インデックスの管理について

インデックスの管理について

データを Splunk に追加すると、Splunk はそれを処理してインデックスに保存します。デフォルトで、Splunk に供給するデータは、メインインデックスに保存されますが、Splunk が異なるデータ入力を使用するように別のインデックスを作成および指定することもできます。

インデックスは、`$SPLUNK_HOME/var/lib/splunk` にあるデータベースに保存されます。データベースは、`db_<starttime>_<endtime>_<sez_num>` という名前のディレクトリです。インデックスは、データベースディレクトリの集まりです。

メインインデックスのほかに、Splunk には数々の内部インデックスが予め設定されています。内部インデックスは、アンダーライン(_)で始まる名前が付けられています。内部インデックスは、監査、インデキシングボリューム、Splunk ロギングおよびその他のデータを保存します。Splunk Web のインデックスの完全リストは、Splunk Web の右上にある **管理** リンクをクリックし、**インデックス** をクリックすると表示されます。

- **main** : デフォルト Splunk インデックス。指定しない限り、すべての処理済データがここに保存されます。
- **splunklogger** : Splunk は、このインデックでその内部ログを追跡します。
- **_internal** : このインデックスは、Splunk のプロセッサからのメトリクスを含みます。
- **sampledata** : 少量のサンプルデータがトレーニング用にここに保存されます。
- **_thefishbucket** : ファイル処理の内部情報
- **_audit** : ファイルシステム変更モニタ、監査、およびすべてのユーザー検索履歴のイベント

このセクションで、以下についての情報をお読みください。

- 複数インデックスの設定、インデックスの移動、インデックスの削除
- 大量のデータセットでより早くレポートするためのサマリーインデキシングの使用
- インデックスサイズの限定またはセグメンテーションの設定による、ディスク使用量の管理

インデキシング処理に興味がある方は、

以下を参照してください。

- 本書のインデキシングの機能のセクション
- Community Wiki のバケツの理解についてのトピックス
- Community Wiki の検索性能についてのトピックス

複数インデックスの設定

複数インデックスの設定

Splunk は、デフォルトですべてのイベントを保存するメイン(main)と呼ばれるインデックスが標準装備されています。デフォルトでは、Splunk は、内部システムおよびサマリーインデックスおよびイベント監査などの Splunk の追加機能で使用する数々のその他のインデックスを作成および使用します。

エンタープライズライセンスの Splunk では、無限数のインデックスを追加することができます。main インデックスは、インデックスを指定しない入力および検索コマンドのデフォルト入力として機能しますが、デフォルトを変更することもできます。Splunk Web、Splunk の CLI、または `indexes.conf` を使用してインデックスを追加できます。

Splunk は、指定しない限り、デフォルトインデックス(デフォルトでは main)を自動的に検索します。新しいインデックスを作成した場合、またはデフォルトではないインデックスを検索したい場合、以下のように検索でインデックスを指定できます。

```
index=hatch userid=henry.gale
```

これは、hatch インデックスの `userid=henry.gale` を検索します。

役割を作成または編集するとき、その特定の役割の代替のデフォルトインデックスを指定して、検索できます。

Splunk Web を使用したインデックスの作成

1. Splunk Web で、**管理 > インデックス**を選択して、**新規**をクリックします。
2. 新しいインデックスを作成するには、以下を入力します。
 - インデックスの名前。インデックスの名前は、数字、文字、ピリオド、アンダーライン、ダッシュで構成します。
 - パスロケーション(すべてオプション、デフォルトは `$SPLUN_DB/INDEX_NAME/`)
 - ◆ ホームパス：デフォルトの `$SPLUN_DB/INDEX_NAME/db` 用に空白を残す
 - ◆ コールド db パス：デフォルトの `$SPLUN_DB/INDEX_NAME/colddb` 用に空白を残す
 - ◆ Thawed/resurrected db パス：デフォルトの `$SPLUN_DB/INDEX_NAME/thaweddb` 用に空白を残す
 - インデックス全体の最大サイズ(MB、オプション)。デフォルトは 500000MB。
 - このインデックスのホット(現在検索されている)部分の最大サイズ(MB、オプション)。

注意：最大サイズ(`maxDataSize`)を設定するとき、大容量のインデックスには `auto_high_volume` を使用します。それ以外は、`auto` を使用します。

3. 必要な値を設定したら、**保存**をクリックします。インデックスが作成されます。新しくインデックスを作成した、または既存インデックスのプロパティを編集したときには、必ず Splunk を再起動します。

Splunk Web の**管理**の**インデックス**セクションにあるインデックス名をクリックして、インデックスを編集できます。既存インデックスのプロパティを編集した場合は、必ず Splunk を再起動してください。

変更できないプロパティは、灰色になっています。このプロパティを変更するには、`indexex.conf` を使用します。

注意：追加のインデックスプロパティは、`indexex.conf` でインデックスを作成・編集した場合に設定可能です。完全な

リストは、`indexes.conf` をご覧ください。

Splunk の CLI を使用したインデックスの作成と編集

Splunk の CLI を使用するには、`$SPLUNK_HOME/bin/` ディレクトリに移動して、`./splunk` コマンドを使用します。

重要：既存のインデックスのプロパティを編集する場合、最初に Splunk を停止しなければいけません。新規インデックスを作成する場合には、最初に Splunk を停止する必要はありません。新しくインデックスを作成した、または既存のインデックスを編集した後は、必ず Splunk を再起動してください。

CLI を使用して“fflanda”と呼ばれる新規のインデックスを追加または編集する場合

```
./splunk [add|edit] index fflanda
```

また、`indexex.conf` にある任意のオプションに対する値は、flag (`-dir` など) を `<code>[add|edit] index <name>` コマンドに渡して指定できます。

新しくインデックスを作成した、または既存インデックスを編集した後は、必ず Splunk を再起動してください。

Indexes.conf によるインデックスの作成

`$SPLUNK_HOME/etc/system/local` の `indexes.conf` にスタanzasを追加します。`indexex.conf.spec` の設定詳細および例をご覧ください。

注意：特定の設定ファイルに利用可能な設定の最も正確な最新リストは、その設定ファイルの `.spec` ファイルにあります。本書の設定ファイルレファレンス、または `$SPLUNK_HOME/etc/system/README` に最新バージョンの `.spec` および `.example` ファイルがあります。

インデックスの無効化

Splunk Web でインデックスの使用を無効化できます。これを行うには、**管理 > インデックス** を選択して、無効にするインデックスの右側にある **無効** をクリックします。

インデックスの削除

不要なインデックスを削除するには、`indexes.conf` を編集します。Splunk Web または CLI を使用してインデックスを削除できません。

重要：インデックスを削除する前に Splunk を停止して、削除後に再起動する必要があります。

indexex.conf のインデックスを削除

`indexes.conf` のインデックススタanzasを削除します。カスタムインデックスは、`$SPLUNK_HOME/etc/system/local` に、カスタムアプリケーションディレクトリは、`$SPLUNK_HOME/etc/system/apps` にあります。

重要：インデックスを削除する前に Splunk を停止して、削除後に再起動する必要があります。

サマリーインデキシングによるレポート効率の向上

サマリーインデキシングによるレポート効率の向上

ディスク使用量の制限を設定

ディスク使用量の制限を設定

Splunk が使用するディスク容量を制御する方法はいくつかあります。ほとんどのディスク容量は、Splunk のインデックスおよび圧縮ログファイル(合わせてデータベースと呼ぶ)に使用します。ディスク容量が足りなくなると、Splunk はインデキシングを停止します。最低空き容量の制限を設定し、インデキシングが停止する前に利用可能な最低空き容量を制御します。インデキシングは、空き容量が最低容量よりも大きくなると再開します。

最低ディスク空き容量の設定

Splunk Web の設定を使用して、インデックスされたデータが保存されるディスク上の最低空き容量を設定します。制限に到達すると、空き容量が大きくなるまで、サーバーは、データのインデキシングを停止します。

注意：

- Splunk サーバーは、この方法では、ディスク容量のクリーンアップをしません。より大きな容量ができるまで、そのまま待ちます。
- 一時停止の期間にファイルに書き込まれていないイベントは失われることがあります。

Splunk Web の操作

- Splunk Web の右上隅にある**管理**をクリックします。
- **システム設定**をクリックします。
- インデックス設定セクションの下で、インデキシングを一時停止する**最低空き容量(MB)**を見つけます。

index settings

Default host name (optional)

Sets the host field value for all events coming from this server.

Path to indexes

Pause indexing if free disk space (in MB) falls below (optional)

Cancel Save

- 希望する最低ディスク空き容量をメガバイト単位で入力します。
- **保存**をクリックします。

Splunk を再起動して変更を有効にします。

コマンドラインインタフェース(CLI)の操作

Splunk の CLI で最低空き容量(メガバイト)を設定できます。CLI を使用するには、\$SPULNK_HOME/bin/ディレクトリに移動して、./splunk コマンドを使用します。

```
# splunk set minfreemb 20000 # set minfree to 20GB
# splunk restart
```

データベースサイズの設定

インデックスの制御は、indexes.conf で行います。合計インデックスサイズ、データベース内のデータのページ、およびエージングポリシーを制御して、ディスクの使用量を制御できます。制限のいずれかに到達すると、データは削除されます。Splunk に内蔵されるアーカイブスクリプトのいずれかを使う、または自作のスクリプトを作成して、データをアーカイブできます。\$SPLUNK_HOME/etc/system/local/、または、\$SPLUNK_HOME/etc/apps/のカスタムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイルについての一般的詳細は、本書の設定ファイルの機能をご覧ください。default のコピーは編集しないでください。

以下の indexes.conf を設定します。

```
maxTotalDataSizeMB = (500000)
```

* インデックスの最大サイズ。インデックスがこのサイズより大きくなると、古いデータが無視され、新しい値(メガバイト)が設定されます。

例:

```
[main]
maxTotalDataSizeMB = 2500000
```

Splunk を再起動して変更を有効にします。Splunk が、新しいポリシーに従うためにイベントをインデックスから取り出すため、CPU 使用量が高くなり、処理には 30 分から 40 分ほどの時間がかかるかもしれません。

ディスク使用を管理するセグメンテーションの設定

ディスク使用を管理するセグメンテーションの設定

セグメンテーションとは、Splunk がインデキシング中にイベントをトークンと呼ばれる使用可能な塊に分ける方法のことです。トークンは、エラーコードやユーザーID などのイベント内に含まれる個々の情報です。選択するセグメンテーションのレベルにより、トークンのサイズを調整できます。

セグメンテーションは、インデキシング、検索速度、さらにはディスク使用量に影響を与えます。通常その必要はありませんが、セグメンテーションのレベルを変更して、インデキシングや検索速度を向上することができます。

セグメンテーションルールを調整して、インデックスの圧縮率を改善したり、特定のデータソースの利用度を向上したりできます。Splunk のデフォルトのセグメンテーション動作を変更する場合は、segmenters.conf を編集します。

segmenters.conf にルールを設定すると、props.conf の特定のソース、ホスト、またはソースタイプをルールと結び

付けます。内部セグメンテーションまたは完全セグメンテーション以外のモードはお勧めしません。

`$SPLUNK_HOME/etc/system/local/`または、`$SPLUNK_HOME/etc/apps/`のカスタムアプリケーションディレクトリにあるすべての設定ファイルを編集します。

注意：この方法で、異なるホスト、ソース、および/またはソースタイプに適用されるセグメンテーションルールをいくつでも有効化できます。

`segmenters.conf`を設定する方法はさまざまですが、自分のデータに適した方法をご利用ください。`props.conf`およびセグメンテーションを使用して、特定のホスト、ソース、ソースタイプに使用するセグメンテーションルールを指定します。以下に設定変更可能な一般的な例を示します。

完全セグメンテーション

Splunk は、デフォルトで、完全セグメンテーションを使用するよう設定されています。完全セグメンテーションは、内部およびアウターセグメンテーションを組み合わせたセグメンテーションです。

内部セグメンテーション

内部セグメンテーションは、検索およびインデキシングの両方でほとんどの検索機能を保持しながらできる、最も効率の良いセグメンテーション設定です。ただし、先行入力の理解力が多少衰えます。インデックスタイムで内部セグメンテーションに交換しても、検索動作には全く変化はありません。

インデックスタイムで内部セグメンテーションを設定するには、`props.conf` のソース、ソースタイプ、またはホストを `SEGMENTATION = inner` に設定します。この設定にすると、Splunk は、より小さなトークンでデータをインデックスします。例えば、`user.id=foo` は、`user id foo` としてインデックスされます。

アウターセグメンテーション

アウターセグメンテーションは、内部セグメンテーションの逆です。個別に小さなトークンをインデックスする代わりに、すべての用語をインデックスし、大きなトークンを少数生成します。例えば、`"10.1.2.5"` は、`"10.1.2.5"` としてインデックスされるため、フレーズを個別に検索できません。ただし、ワイルドカードは使用できるため、フレーズを個別に検索できます。例えば、`"10/1"` を検索して、IP アドレスが `"10/1"` から始まるイベントすべてを取得できます。また、アウターセグメンテーションは、IP アドレス `48.15.16.23` の `48.15` セグメントのような検索結果の異なるセグメントをクリックする能力を無効にします。アウターセグメンテーションは、完全セグメンテーションに比べ多少効率が良く、内部セグメンテーションはかなり効率が良いとされています。

インデックスタイムでアウターセグメンテーションを有効にするには、`props.conf` のソース、ソースタイプ、またはホストを `SEGMENTATION = outer` に設定します。また、検索が適切に機能するには、

`$SPLUNK_HOME/etc/system/local/segmenters.conf` に以下のラインを追加し、検索システムがより大きなトークンを検索できるようにします。

```
[search]
```

```
MAJOR = [ ] < > ( ) { } | ! ; , ' " * ¥n ¥r ¥s ¥t & ? + %21 %26 %2526 %3B %7C %20 %2B %3D -- %2520
```

```
MINOR =
```

セグメンテーション無し

最も適切なセグメンテーション設定は、セグメンテーションを完全に無効化することです。ただし、セグメンテーションは検索と密接に関係しています。例えば、Splunk にセグメンテーションなしにインデックスするよう設定すると、時間、ソース、ホスト、およびソースタイプの検索が制限されます。したがって、詳細検索能力が全く必要のない場合にのみこの設定を使用してください。

この設定を有効化するには、`props.conf` のソース、ソースタイプ、またはホストに `SEGMENTATION = none` を設定します。このソース、ソースタイプ、またはホストのキーワード検索は、結果を返しません。ただし、インデックスされたフィールドの検索はできます。

セグメンテーション無しは、最も容量効率のいい設定ですが、検索を非常に難しくします。更に結果を制限するには検索コマンドで検索を絞る必要があります。このタイプの設定は、検索能力よりも保存効率が重要な場合に便利です。

Splunk Web のセグメンテーション

Splunk Web にもセグメンテーションの設定があります。これは、インデキシングセグメンテーションとは関係ありません。Splunk Web のセグメンテーションは、ブラウザとの相互関係に影響を与え、検索結果のスピードを速める場合があります。

ホスト、ソース、ソースタイプに対するカスタムセグメンテーションの設定

ホスト、ソース、ソースタイプに対するカスタムセグメンテーションの設定

デフォルトで、Splunk は、イベントを完全に区分し、最も柔軟性のある検索を可能にします。セグメンテーション全般については、このページのセグメンテーションに関する説明を参照してください。

特定のホスト、ソース、またはソースタイプの検索、またはイベント処理を行う方法が決まっている場合、その特定のタイプのイベントに対してカスタムセグメンテーションを設定できます。特定のホスト、ソース、またはソースタイプのカスタムセグメンテーションを設定することにより、インデキシングおよび検索性能を向上させ、インデックスサイズ(ディスク上)を減らすことができます。

`props.conf` によるカスタムセグメンテーションの設定

`props.conf` のホスト、ソース、またはソースタイプスタanzas に `SEGMENTATION` および `SEGMENTATION-<segment selection>` 属性を追加して、ホスト、ソース、またはソースタイプのイベントのカスタムセグメンテーションを設定します。インデックスタイムおよび `segmenters.conf` で定義された検索タイム(Splunk Web) セグメンテーションのルールを使用して、属性に値を追加します。

`$SPLUNK_HOME/etc/system/local/props.conf` にスタanzas を追加します。以下の属性/値ペアを指定します。

```
[<spec>]
```

```
SEGMENTATION = $SEG_RULE
```

```
SEGMENTATION-<segment selection> = $SEG_RULE
```

[<spec>]には、以下が指定できます。

- <sourcetype>: イベントデータにあるソースタイプ
- host::<host>: イベントデータにあるホスト値
- source::<source> イベントデータのソース

SEGMENTATION = \$SEG_RULE

- セグメンテーションを指定して、インデックスタイムで使します。
- \$SEG_RULE を、内部、アウター、なし、または完全に設定します。

SEGMENTATION-<segment selection> = \$SEG_RULE

- セグメンテーションを指定して、検索タイムで使します。
- Splunk Web の表示のみに適用します。
- <segment selection>は、Splunk Web 設定画面のラジオボタンを参照します。これらのラジオボタンを、カスタム\$SEG_RULE に割り当てます。
- <segment selection>は、完全、内部、アウター、ローのいずれかです。

\$SEG_RULE

- segmenters.conf で定義されているセグメンテーションルール。
- デフォルトは、内部、アウター、なし、完全です。
- \$SPLUNK_HOME/etc/system/local/segmenters.conf を編集してカスタムルールを作成します。
- segmenters.conf の設定についての詳細は、このページをご覧ください。

例

以下の例では、検索性能(Splunk Web)を向上させ、syslog イベントのインデックスサイズを減らします。

以下を、props.conf の[syslog]ソースタイプスタanzaに追加します。

```
[syslog]
SEGMENTATION = inner
SEGMENTATION-all = inner
```

この例は、sourcetype=syslog を持つすべてのイベントのセグメンテーションを、インデックスタイム (SEGMENTATION 属性を使用) および Splunk Web (SEGMENTATION-<segment selection>属性を使用) の内部セグメンテーションに変更します。

注意: 必ず Splunk を再起動して Splunk Web セグメンテーションへの変更を適用させ、データを再インデックスしてインデックスタイムセグメンテーションへの変更を適用してください。

インデックスの移動

インデックスの移動

Splunk のインデックスをある位置から別の位置に移動します。

警告： インデックスの一部を手動で分けたり、移動したりしないでください。既存のインデックスを再分する必要がある場合は、Splunk サポートにご連絡ください。

設定

1. ターゲットファイルシステムに十分な空き容量があることを確認してください。インデックスを予定しているローダーの合計の 1.2 倍以上の容量が必要です。
2. ターゲットディレクトリが、正しい権限を持っていて、splunkd 処理がファイルへの書き込みを行えることを確認してください。

```
# mkdir /foo/bar
# chown splunk /foo/bar/
# chmod 755 /foo/bar/
```

3. 新規インデックスホームの準備ができたら、Splunk の CLI でサーバーを停止します(実行している場合)。
Splunk の CLI を使用するには、\$SPLUNK_HOME/bin/ディレクトリに移動して、./splunk コマンドを使用します。

```
# ./splunk stop
```

4. 既存のインデックスファイルシステムを新規のホームへコピーします。

```
# cp -r $SPLUNK_DB/* /foo/bar/
```

5. /etc/splunk-launch.conf を編集して新規のインデックスディレクトリを反映させます。
6. /etc/splunk-launch.conf の中で、SPLUNK_DB の変数を、新規のインデックスディレクトリを指すように変更します。

```
SPLUNK_DB=/foo/bar
```

注意： パス \$SPLUNK_HOME/var/lib/splunk/searches が存在することを確認してください。Splunk は、ここに少量のインデックスを保存するため、存在しないと、インデックスが消えたように見えることがあります。

7. サーバーを起動します。

```
# ./splunk start
```

Splunk サーバーは、前回停止したところから始め、古いインデックスファイルシステムのコピーから読み込み、書き込みます。

インデックスデータに別のパーティションを使用

インデックスデータに別のパーティションを使用

Splunk は、インデックスデータ用に別々のディスクおよびパーティションを使用します。そのマウントと DB ローリングが適切に設定されている限り、Splunk が、インデックスや warm/cold に基づいて多くのディスク/パーティション/ファイルシステムを使用するよう設定することが可能です。ただし、1 つの高性能ファイルシステムを使用し、Splunk インデックスデータを保存することをお勧めします。

Splunk は、以下の 4 つのステージでロールをインデックスします。

- **Hot** – 書き込みを受け入れます。複数の hot バケツがあります。検索可能。
- **Warm** – hot からローテーションしたデータ。多くの warm のバケツがあります。検索可能。
- **Cold** – warm からローテーションしたデータ。多くの cold のバケツがあります。検索がこれらのファイルに含まれている時間の範囲を指定したときのみ検索可能。
- **Fronzen** – frozen 状態に入ったバケツは即座に削除されます。

別のパーティションを使用する場合、Splunk のインデックスデータを編成する最も一般的な方法は、ローカルマシンに hot および warm バケツを保管し、異なる配列またはディスクに cold バケツを保存します(長期保管用)。書き込み・読み込み速度が速いパーティション付きのマシンで hot および warm バケツを実行することをお勧めします(ほとんどの検索操作を、hot および warm で行うため)。Cold は、ディスクの確かな配列に配置してください。

バケツの流れ

- 指定したサイズ(maxDataSize)に到達すると、1つの hot のバケツが warm へローテーションします。
- warm バケツの数が、設定した最大数(maxWarmDBCount)を超えると、バケツは warm から cold にローテーションします。
- アーカイブ用に選択されるまでバケツは cold(または warm)に留まります。

別パーティションの設定

パーティションは、通常オペレーティングシステムで設定するときと同じように設定します。ディスク/パーティションをマウントし、Splunk が indexes.conf で正しいパスを指しているよう確認します。

最初に、\$SPLUNK_HOME/etc/system/local/indexes.conf に正しいパスを追加します。[\$INDEX]エントリーでインデックス毎にパスを設定します。

homePath = <サーバーのパス>

- インデックスの hot および warm データベースとフィールドを含むパス。
- warm データベースには、splunkd に常時開いているハンドルがあります。
- 警告：パスは、書き込み可能でなければいけません。

coldPath = <サーバーのパス>

- インデックスの cold データベースを含むパス。
- Cold データベースは、検索時必要に応じて開きます。
- 警告：パスは、書き込み可能でなければいけません。

thawedPath = <サーバーのパス>

- インデックスの解凍(復活)データベースを含むパス。

Splunk からインデックスされたデータを削除

Splunk からインデックスされたデータを削除

Splunk は、特別な演算子、delete を提供します。delete 演算子を使用する前に、このセクションを注意してお読みください

い。

警告：ユーザーデータを削除すると元に戻せません。検索から削除するイベントを選択するとき、または Splunk インストールから削除するデータを選択するときには注意してください。データを元に戻したい場合は、対象のデータソースを再インデックスします。

削除できる人？

`delete` 演算子は、"`delete_by_keyword`"能力を持つユーザーのみがアクセスできます。Splunk には、デフォルトでこの能力を持つ特殊役割"`can_delete`"が設定されています(他はなし)。管理者には、この機能がデフォルトで設定されていません。Splunk は、インデックスデータを削除する場合のみに使用する専用のユーザーログインを作成するよう推奨しています。詳しくは、本書の「ユーザーの追加と役割の割り当て」を参照してください。

削除の仕方

`delete` 演算子を使用するには、削除したいイベントを返す検索を実行します。その検索が、削除したいイベントのみを返し、他のイベントを返さないことを確認してください。

例えば、未来の検索で出てこないよう無効化/削除した `/fflanda/incoming/cheese.log` と呼ばれずソースから既にインデックスしたイベントを削除したい場合、以下を行います。

最初に、そのソースを検索します。

```
source="/fflanda/incoming/cheese.log
```

これが削除したいデータであることを確認したら、その検索を `delete` とパイプします。

```
source="/fflanda/incoming/cheese.log | delete
```

更なる例は、検索参照マニュアルの削除演算子についてのページをご覧ください。

検索を `delete` 演算子とパイプすることにより、その検索が返すすべてのイベントに印しを付け、今後の検索で返らないようにします。Splunk で検索時、誰も(管理権限を持つユーザーも含む)このデータを見ることはできません。

現在、`delete` とパイプしてもディスクの空き容量を取り戻しませんが、Splunk の将来のリリースでは、ディスクの空き容量を取り戻すユーティリティをお届けする予定です。これは、`delete` 演算子による印し付きのすべてのイベントを検索し、永久に削除します。また、`delete` 演算子は、イベントのメタデータを更新しません。そのためどのメタデータ検索も、検索可能でないイベントを含んでしまいます。メインの全てのインデックスされたデータダッシュボードは、削除されたソース、ホスト、またはソースタイプのイベント数を表示します。

'clean'コマンドで CLI からイベントデータを削除

ディスクからイベントデータを完全に消去するには、CLI の `clean` コマンドを使用して、インデックスからデータを完全に消去します。通常、すべてのデータを再インデックスする前にこの操作を行います。

`-f` パラメータを追加して、`clean` が、確認プロンプトをスキップするよう強制します。

Splunk CLI で、`./splunk help clean` を入力して、`clean` のヘルプページにアクセスします。

eventdata 引数の後に `./splunk clean` を入力して、Splunk インストールのインデックスからイベントデータを永久に削除します。index を指定して、指定されたインデックスからイベントデータを削除します。index を指定しないと、Splunk はすべてのインデックスからすべてのイベントデータを削除します。

例

注意：これらのコマンドを実行する前に、必ず先に Splunk を停止してください。

```
./splunk stop
```

この例では、Splunk にすべてのインデックスのデータを削除するよう指示します(index 引数が指定されていないため)。

```
./splunk clean eventdata
```

この例では、_internal インデックスからインデックスされたイベントデータを削除し、Splunk に、確認プロンプトをスキップするよう強制します。

```
./splunk clean eventdata _internal -f
```

アラートの定義

アラートのしくみ

アラートのしくみ

アラートは、予定の時刻に実行し、結果を送信するよう設定した検索です。アラートを使用して、監視しているデータ、ネットワーク構造基盤、ファイルシステム、またはその他のデバイスでの変更を知らせるようにします。アラートは、Eメールまたは RSS を通じて送信され、シェルスクリプトの実行トリガができます。いかなる保存済み検索もアラートに変えられます。

アラートは、以下で構成されています。

- 検索を行うスケジュール
- アラートトリガの条件
- トリガ条件が一致したときに行うアクション

アラートの有効化

保存済み検索を作成するときにアラートを設定、または、編集の権限を持つ既存の保存済み検索のアラートを有効化します。アラートは以下を使って設定します。

- Splunk Web
- savedsearches.conf

アラート用の全体的 E メール設定を指定

メールホスト、メールフォーマット、メールの件名、送信者、およびアラートの結果がインラインに含まれるべきかべきでないかを指定するには、以下を行います。

- Splunk Web で、**管理 > メールアラート設定**をクリックして、選択を指定します。
- **保存**をクリックします。

これですべてのアラートがこの設定を使用します。

スクリプトアラート

アラートでシェルスクリプトをトリガできます。アラートを設定するとき、記述したスクリプトを指定します。この機能を使って、他のアプリケーションにアラートを送信できます。スクリプトアラートの設定について詳しく説明をお読みください。

スクリプトアラートを使って、syslog イベント、または SNMP トラップを送信できます。

アラートのカスタマイズ

alert_actions/conf ファイルを使用して、アラート設定をカスタマイズします。例えば、メール設定(メールサーバー、件名

ライン等)を変更します。アラートオプションのカスタマイズについて詳しく説明をお読みください。

考慮事項

アラートを設定するときは、次の点に注意してください。

- 一度に多くのアラート/保存済み検索を実行すると、システムが遅くなることがあります。ハードウェアにより異なりますが、一度に 20 から 30 のアラートの実行なら許容範囲です。アラートが基準にする検索が複雑な場合は、間隔を長く設定し、検索範囲を広げます。
- 矛盾のないアラートのタイムフレームを設定します。検索の実行に 4 - 5 分以上かかる場合は、5 分毎に実行するように設定しないでください。
- Splunk サーバーが接続できる LAN 上で実行しているメールサーバーが必要です。Splunk は、メールサーバーに対して認証しません。
- ここにある Splunk Community Wiki のアラート設定の最良実施例について詳しくお読みください。

savedsearches.conf によるアラートの設定

savedsearches.conf によるアラートの設定

savedsearches.conf でアラートを設定します。\$SPLUNK_HOME/etc/system/README/savedsearches.conf.example を例として使用する、または自作の savedsearches.conf を作成します。\$SPLUNK_HOME/etc/system/local/、または \$SPLUNK_HOME/etc/apps/のカスタムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイル一般に関する詳しい情報は、設定ファイルの機能をご覧ください。

以下の手順に従います。

1. 保存済み検索を作成します。
2. 検索のスケジュールを組みます。
3. アラート条件を定義します。
4. アラートアクションを設定します。

アラートは保存済み検索を作成するときに設定できます。または、後で保存済み検索スタンプにアラート設定を追加します。

注意：アラートが送信されるよう Splunk サーバーでメールを有効化する必要があります。また、Splunk サーバーが、メールサーバーに連絡できる必要があります。管理でメール設定をします。

保存済み検索の作成

最初に、保存済み検索を設定します。savedsearches.conf でも保存済み検索を設定できます。

検索のスケジュール

次に、検索のスケジュールを組みます。つまり、検索が指定されたスケジュールに沿って実行されるように指定します。例えば、Splunk が、検索を毎時間または真夜中に実行します。検索がアラート条件に一致すると、Splunk がアラートを

送ります。

以下の属性/値ペアを、保存済み検索スタンザに追加して、スケジュールに沿って検索を実行します。

userid = <整数>

- この保存済み検索を作成したユーザの Userld
 - ◆ Splunk は、Splunk Web で検索を実行し、編集能力を作成した人物をログするためにこの情報が必要です。
- 可能性のある値：Splunk user ID。
- User ID は、\$SPLUNK_HOME/etc/passwd にあります。
 - ◆ 各ラインの username 直前の最初の番号を探します。
 - ◆ 例：2:penelope...

enableSched = < 0 | 1 >

- 1 を設定して検索スケジュールを有効にします。
- デフォルトは 0 です。

schedule = <文字列>

- クローンスタイルスケジュール。
- 例：*/12****)

execDelay = <整数>

- 最近のイベントからスケジュールされた検索クエリの実行までの時間(秒単位)。
- デフォルトは 0 です。

アラート条件

ここで、アラート条件を定義します。アラート条件は、アラートを送信するかどうかを Splunk に指示します。結果にイベント、ソース、またはホストのしきい値の数を入力します。アラート条件が一致すると、Splunk は、メールまたは RSS フィードで通知し、シェルスクリプトの実行をトリガします。

counttype = <文字列>

- アラートのカウントタイプを設定します。
- 可能性のある値：イベントの数、ホストの数、ソースの数、ソースタイプの数。

relation = <文字列>

- カウントタイプに対する比較方法。
- 可能性のある値：より大きい、未満の、同等の、減少する、上昇する。

quantity = <整数>

- 特定のカウントタイプに対して比較する数

ここで、以下を指定します。

```
counttype = number of events
relation = rises by
quantity = 25
```

Splunk は、検索結果が、前回検索が実行されてから 25 上昇するとアラートを送信します。

アラートアクションを設定

アラートがトリガされたときの Splunk の動作を指示します。次のいずれかが可能です。

RSS の有効化

```
action_rss = < 0 | 1 >
```

- RSS リンクの作成を切り替えます。
- 1 で送信、0 で無効化。

メールを送信

```
action_email = <文字列>
```

- アラートを送信する先をコンマ区切りにしたメールアドレスのリスト。

```
sendresults = < 0 | 1 >
```

- メール/シエルスクリプトで結果を送信するかどうかを指定。
- 1 で送信、0 で無効化。

例

この例では、スケジュールに沿って、用語“sudo”を含むイベントの検索を実行し、RSS フィードを通じて結果を送信します。

```
[sudoalert]
action_rss = 1
counttype = number of events
enableSched = 1
quantity = 10
search = sudo
relation = greater than
schedule = */12 * * * *
sendresults = 0
role = Admin
```

スクリプトアラートの設定

スクリプトアラートの設定

savedsearches.conf でアラートを設定します。

`$SPLUNK_HOME/etc/system/README/savedsearches.conf.example` を例として使用する、または、自作の `savedsearches.conf` を作成します。`$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` にあるカスタムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイル一般に関する詳しい情報は、設定ファイルの機能をご覧ください。

スクリプトオプション

アラートでシェルスクリプトを実行できます。このスクリプトは、必ず `$SPLUNK_HOME/bin/scripts` に保存します。以下の属性/値ペアを使用します。

```
action_script = <string>
```

- 検索でシェルスクリプトの実行をトリガできます。
- 実行するシェルスクリプトの名前を指定します。
- スクリプトを、`$SPLUNK_HOME/bin/scripts` に保存します。
- このスクリプトに以下のコマンドライン引数を渡します。
- `$0` = スクリプト名
- `$1` = 返すイベントの数
- `$2` = 検索用語
- `$3` = 完全に確認されたクエリ文字列
- `$4` = 保存済み splunk の名前
- `$5` = トリガの理由(例、イベントの数が 1 より大きい)
- `$6` = 保存済み検索を閲覧するブラウザ URL
- `$7` = この保存済み検索に属するタグのリスト
- `$8` = この検索の結果が保存されているファイル(未加工結果を含む)

注意: 保存済みタグがない場合、`$7` は、検索結果を含むファイルの名前(`$8`)になります。

異なる言語で記述されたスクリプト(PERL、Python、VBScript など)を実行する場合は、`#!`に続いて、スクリプトの最初のラインに Splunk に使用するインタプリタを指定する必要があります。

例:

PERL スクリプトを実行する場合

```
----- myscript.pl -----  
#!/path/to/perl  
.....  
.....
```

Python を使用して、スクリプトファイルを解釈する場合

```
----- myscript.py -----  
#!/path/to/python  
  
.....  
.....
```

アラートで作業できるようにスクリプトを設定する方法の例は、SNMP トラップの送信をご覧ください。

例

Splunk を設定して、syslog にアラートを送信できます。これは、既にアラートを他のアプリケーションに送信するように syslog を設定してあり、Splunk のアラートも含めたい場合に便利です。

Syslog 入力設定時に UDP を使用する最善の方法についての情報は、Splunk Wiki をご確認ください。

logger(または syslog に書き込む他のプログラム)を呼び出すスクリプトを書き込みます。スクリプトは、アラートが返す変数のどの数も呼び出すことができます。

以下のスクリプトを作成して、実行可能にします。

```
logger $5
```

スクリプトを \$SPLUNK_HOME/bin/scripts に記述します。

ここで、スクリプトを呼び出すアラートを書き込みます。アラート設定に情報は、アラートの設定をご覧ください。アラートのシェルスクリプトの実行トリガにパスを指定して、スクリプトを呼び出すようアラートを設定します。

保存済み検索を編集して、スクリプトを呼び出します。スクリプトが \$SPLUNK_HOME/bin/scripts にある場合は、完全なパスを指定する必要はありません。



このログは、syslog に理由をトリガします。

```
Aug 15 15:01:40 localhost logger: Saved Search [j_myadmin]: The number of events(65) was greater
```

SNMP トラップを他のシステムに送信

SNMP トラップを他のシステムに送信

Splunk を、監視ツールとして使用して、Network Systems Management コンソールのような他のシステムに SNMP アラートを送信できます。

Windows で SNMP トラップの送信に興味がある場合は、この Community Wiki トピックスをご覧ください。

設定

必要条件

- Perl は、以下のスクリプトを実行するのに必要です。
- `/usr/bin/snmptrap` コマンドの使用には、Net-SNMP が必要です。シェルスクリプトで SNMP トラップを送信する他の方法がある場合は、必要に応じて変更します。
- Splunk インストールの `$SPLUNK_HOME/bin/scripts` ディレクトリへの管理アクセス。
- 安全上の理由から、スクリプトは、必ず `$SPLUNK_HOME/bin/scripts` に保存します。

シェルスクリプトの作成

- `$SPLUNK_HOME/bin/scripts` ディレクトリに、`trasphsts.pl` スクリプトを作成します。
 - ◆ 安全上の理由から、スクリプトは必ずこのディレクトリに保存します。ない場合は、ディレクトリを作成します。
 - ◆ 以下のコードを、`sendsnmptrap.pl` にコピーします。
- `chmod +x sendsnmptrap.pl` で実行可能にします。
- SNMP トラップハンドラの `host:Port`、外部コマンド `splunk` および `snmptrap` へのパス、およびユーザ/パスワードを、必要に応じて変更します。
- perl スクリプトは、Perl を持つ MS Windows システムで機能します。ただし、Windows システムの中には、perl がインストールされていないか、perl スクリプトが Splunk を通じて直接実行できるように設定されていないことがあります。このような場合には、Windows CMD スクリプトを使用して SNMP トラップを送信する方が簡単な場合があります。

```
#!/usr/bin/perl
#
# sendsnmptrap.pl: A script to for Splunk alerts to send an SNMP trap.
#
# Modify the following as necessary for your local environment
#
$hostPortSNMP = "qa-tml:162"; # Host:Port of snmpd or other SNMP trap handler
$snmpTrapCmd = "/usr/bin/snmptrap"; # Path to snmptrap, from http://www.net-snmp.org
$OID = "1.3.6.1.4.1.27389.1.1"; # Object IDentifier for an alert, Splunk Enterprise OID is 27389
# Parameters passed in from the alert.
# $1-$9 is the positional parameter list. $ARGV[0] starts at $1 in Perl.
$searchCount = $ARGV[0]; # $1 - Number of events returned
$searchTerms = $ARGV[1]; # $2 - Search terms
$searchQuery = $ARGV[2]; # $3 - Fully qualified query string
$searchName = $ARGV[3]; # $4 - Name of saved search
$searchReason = $ARGV[4]; # $5 - Reason saved search triggered
$searchURL = $ARGV[5]; # $6 - URL/Permalink of saved search
if ( $ARGV[7] ) { # We received tags
$searchTags = $ARGV[6]; # $7 - Tags, if any, otherwise $7 is $8
```

```

$searchPath = $ARGV[7]; # $8 - Path to raw saved results in Splunk instance (advanced)
} else { # We didn't receive tags
$searchPath = $ARGV[6]; # $7 - Path to raw saved results in Splunk instance (advanced)
}
# Send trap, with the the parameter list above mapping down into the OID.
if ( $ARGV[7] ) { # We received tags
$cmd = qq/$snmpTrapCmd -v 1 -c public $hostPortSNMP $OID ' 1 0 '
$OID.1 i $searchCount $OID.2 s "$searchTerms" $OID.3 s "$searchQuery" $OID.4 s
"$searchName" $OID.5 s "$searchReason" $OID.6 s "$searchURL" $OID.7 s
"$searchTags" $OID.8 s "$searchPath"/;
system($cmd);
} else { # We didn't receive tags
$cmd = qq/$snmpTrapCmd -v 1 -c public $hostPortSNMP $OID ' 1 0 '
$OID.1 i $searchCount $OID.2 s "$searchTerms" $OID.3 s "$searchQuery" $OID.4 s
"$searchName" $OID.5 s "$searchReason" $OID.6 s "$searchURL" $OID.8 s
"$searchPath"/;
system($cmd);
}

```

アラートを設定してシェルスクリプトを呼び出す

- 保存済み検索を作成します。詳細は保存済み検索の設定についてをお読みください。
- 保存済み検索をアラートに変換します。詳細は、アラートの設定についてをお読みください。
- \$SPLUNK_HOME/bin/scriptsにあるスクリプトの名前を指定することによりシェルスクリプトを呼び出すようにアラートを設定します。



これは、実行中のスクリプトの例です(返し値も含む)。

```

[root@qa-tml ~]# snmptrapd -f -Lo
2007-08-13 16:13:07 NET-SNMP version 5.2.1.2 Started.
2007-08-13 16:14:03 qa-el4.splunk.com [172.16.0.121] (via UDP: [172.16.0.121]:32883) TRAP, SNMP
SNMPv2-SMI::enterprises.27389.1 Warm Start Trap (0) Uptime: 96 days, 20:45:08.35
SNMPv2-SMI::enterprises.27389.1.1 = INTEGER: 7 SNMPv2-
SMI::enterprises.27389.1.2 = STRING: "sourcetype::syslog" SNMPv2-
SMI::enterprises.27389.1.3 = STRING: "search sourcetype::syslog starttime:12/31
/1969:16:00:00 endtime::08/13/2007:16:14:01" SNMPv2-SMI::enterprises.27389.1.4
= STRING: "SyslogEventsLast24" SNMPv2-SMI::enterprises.27389.1.5 = STRING:

```

```
"Saved Search [SyslogEventsLast24]: The number of hosts(7) was greater than 1"
SNMPv2-SMI::enterprises.27389.1.6 = STRING: "http://qa-el4:18000/?q=sourcetype
%3a%3asyslog%20starttimeu%3a%3a0%20endtimeu%3a%3a1187046841" SNMPv2-
SMI::enterprises.27389.1.7 = STRING: "/home/tet/inst/splunk/var/run/splunk
/SyslogEventsLast24"
2007-08-13 16:14:15 NET-SNMP version 5.2.1.2 Stopped.
```

高度な条件付アラート

高度な条件付アラート

アラート条件の基礎を保存検索に置いた場合、アラートのトリガの具体的な条件を作成し、誤認ポジティブアラートの数を減らすことができます。また、アラートの内容を変更することなく結果を複雑に計算したアラートをトリガできます。

保存済み検索のアラート条件の定義方法

注意：このトピックスでは、既に保存済み検索があるものと想定しています。

保存済み検索に基づいてアラート条件を定義する場合

1. `savedserches.conf` で、アラートをトリガする条件を定義します。

保存済み検索のスタンザに、次のラインを追加します。

```
alert_condition = <文字列>
```

`string` は、保存済み検索の結果を評価する検索です。検索が結果を生成すると、この条件は、アラートアクションをトリガします。

2. `savedsearches.conf` で、トリガするアラートアクションのタイプを指定し、アクションに関連した引数を定義します。

同じ保存済み検索スタンザ内で、条件が一致するときにトリガするアラートアクションを指定します。アクションは、メールの送信、RSS の有効化および公開、およびスクリプトの実行などを含みます。複数のアクションを指定できます。

a. メールアラートを送信したい場合、以下のラインを追加します。

```
action_email = <電子メールアドレスの一覧>
```

これは、アラートを受信するメールアドレスのコンマで区切りリストを指定します。

b. アラート用に RSS を有効化する場合、以下のラインを追加します。

```
action_rss = 1
```

これは、RSS リンクを作成します。デフォルトは 0 (RSS リンクを作成しない) です。

c. シェルスクリプトをトリガするには、次のラインを追加します。

```
action_script = <文字列>
```

これは、アラート条件が一致するとき実行するシェルスクリプトの名前を指定します。script は、`$SPLUNK_HOME/bin/scripts` に保存されています。

3. `alert_actions.conf` で、アラートアクションを定義します。

a. メールアラートの場合、予め `savedsearches.conf` にアラートを受信するメールアドレスが指定されています。ここでは、メールの送信者、件名とフォーマット、およびメール送信時に使用する SMTP メールサーバーを定義できます。

b. RSS アラートでは、保存済み RSS フィードの数を指定します。デフォルトの数は 30 です。

4. Splunk を再起動して設定ファイルの変更を実行します。

条件付メールアラートの例

`access_combined_error` と呼ぶ保存済み検索があると仮定します。これは、以下を検索します。

```
sourcetype=access_combined error
```

あらゆるホストマシンで 5 回以上このエラーが発生する度にアラートの受信を希望します。

このアラート条件を設定するには、最初に `savedsearches.conf` を編集します。スタanzas は、以下のように記述します。

```
[access_combined_error]
search = sourcetype=access_combined error
role = Admin
alert_condition = | stats count by host | where count>5
action_email = me@myhost.com
```

次に、`alert_actions.conf` を編集して、メールアラートをフォーマットします。スタanzas は以下のように記述します。

```
[email]
from = alert@mysplunk.com
subject = access combined error
format = plain
```

最後に、Splunk を再起動して設定変更を実行します。

バックアップと保存方針の設定

バックアップの対象

バックアップの対象

Splunk データは、3 つの主なカテゴリーに分類されます。

- インデックスされたイベントデータ
- ユーザーデータ
- 設定データ

必要な空き容量

必要な空き容量

このトピックスでは、Splunk インデックスおよび関連データのサイズを推定する方法を説明し、必要ディスク容量を計画できるようにします。

Splunk がデータをインデックスするとき、結果データは 2 つの基本的カテゴリーに分類されます。保存されるローデータおよびそのデータを指すインデックスです。少し使ってみれば、必要なディスク容量を推定できます。

通常、Splunk がデータ入力から抽出する圧縮された保存されるデータは、Splunk に送られるローデータの 10% を占めます。このデータにアクセスするために作成されたインデックスは、送られるデータの 10% から 110% を占めます。この値は、データにいくつの独自の用語が発生するかが強く影響します。データの特性により、セグメンテーション設定を調整してください。セグメンテーションの機能およびインデックスサイズへの影響については、Splunk の一流開発者のひとりが作成したセグメンテーションのビデオもご覧ください。

インデックスサイズを予測する最善の方法は、どこかに Splunk のコピーをインストールし、データの代表サンプルをインデックスしてから、結果として生じるディレクトリ `defaultdb` のサイズを確認します。

以下にその手順を示します。

サンプルをインデックスしたら、

1. `$SPLUNK_HOME/var/lib/splunk/defaultdb/db/db-hot` に移動します。
2. `du -sh rawdata` を実行して、圧縮および保存されるローデータの大きさを決定します。
3. `du -ch *.tsidx` を実行して、最後の `total` ラインを見て、インデックスのサイズを確かめます。

これは、インデックスにあるアイテムが指す保存されるデータです。通常、このファイルのサイズは、インデックスしたサンプルデータセットのサイズの約 10% です。

4. 集めた値を一緒に加えます。

これは、インデックスしたサンプルのインデックスおよび関連データの合計サイズです。これを使って、長期の Splunk インデックスおよび `rawdata` ディレクトリの必要サイズを推定できます。

インデックスされたデータのバックアップ

インデックスされたデータのバックアップ

このトピックでは、Splunk のインデックスされたデータのバックアップについて説明します。最初に、インデックスされたデータが Splunk の操作でどう移動するかの概要を説明し、次に一般またはデフォルト Splunk インデックス設定に基づいた基本バックアップストラテジを説明します。最後に、Splunk インデックスデータの保存ポリシーの設定または変更のオプションを提供します。

このトピックで説明するデフォルト値およびポリシーは、`indexes.conf` に設定されています。より複雑なインデックス設定がある、または異例の量のデータがある場合には、それを参照して、詳細情報とオプションを探します。設定ファイルを変更する前に、設定ファイルについてお読みください。

Splunk によりデータが移動するしくみ

Splunk がインデックスするとき、データは、定義するポリシーに基づいて一連の段階を踏んで移動します。高レベルにおけるデフォルトの動作は以下の通りです。

データが最初にインデックスされる時、データは“hot”データベースに入れられます。

データは、“warm”データとして再分類されるポリシー条件と一致するまで、hot db に保管されます。これを、データを warm への「ローテーション」と呼びます。デフォルトで、これは、特定の hot db が、指定されたサイズまたは年齢に到達したときに起こります。

hot db がローテーションすると、そのディレクトリは、warm db で「バケツ」になるよう再び名前が付けられます。この時点で、warm db バケツを安全にバックアップできます。

次に、指定数の warm バケツに到達すると(デフォルト値は 300 バケツ)、バケツは、300 の warm バケツを維持するために cold バケツに名前を変更します。(cold db が他のファイルシェアに位置している場合、warm バケツはそこに移動し、warm db ディレクトリから削除されます。)

最後に、データバケツが、定義されたポリシーの必要事項に一致すると、“frozen”になります。このデフォルトの動作は、削除することです。データをアーカイブまたは保存する必要がある場合、削除する前にバケツに強制的な処置ができるスクリプトを提供できます。

まとめ

- hot dbs : その時点で増加しない変更を書き込みます。これをバックアップしないでください、代わりに warm db をバックアップします。
- warm db : 増分を追加します。安全にバックアップできます。複数の warm「バケツ」で構成されています。
- cold db : ポリシーに基づいて(デフォルトは 300 バケツ)、その数のバケツに到達すると、バケツは改名するか(hot の時と同様)、または cold へコピーされ(そして warm ディレクトリから削除され)ます(異なるファイルシステムの場合もある)。
- frozen : デフォルトポリシーは削除。

バックアップストラテジの選択

通常、希望する増分バックアップユーティリティを使用して、定期的に warm db バケツのバックアップをスケジュールします。

Hot データベースは、ファイルのスナップショット、VSS(Windows/NTFS)、ZFS スナップショット(ZFS)、またはストレージサブシステムが提供するスナップショット設備などを使用してのみバックアップできます。このような設備が利用できない場合、hot データベース内のデータは、warm db にローテーションした後に限りバックアップできます。

Splunk は、定義するポリシーに基づいて hot db を warm db にローテーションします。デフォルトでは、メインインデックスは、特定のサイズに到達すること、またはデータが 86400 秒(一日)追加されていない場合、どちらかが先に起こったときに、hot db をローテーションします。

インデックスのサイズまたはインデックスにあるデータの年齢を制御して、ローテーションおよびアーカイブポリシーを設定できます。Splunk インデックスは、以下の 4 つの段階で行われます。

- **Hot** – 書き込みを受け入れます。複数の hot バケツがあります。検索可能。
- **Warm** – hot からローテーションしたデータ。多くの warm のバケツがあります。検索可能。
- **Cold** – warm からローテーションしたデータ。多くの cold のバケツがあります。検索がこれらのファイルに含まれている時間の範囲を指定したときのみ検索されます。
- **Fronzen** – frozen 状態に入ったバケツは即座に削除されます。

警告：全てのインデックスの位置が書き込み可能である必要があります。

これらのファイルのサイズ、位置、年齢は、`indexes.conf` により制御されています。設定ファイルを変更する前に、変更ファイルについてお読みください。

リカバリの推奨

致命的でないディスク不具合が起きた場合(データの一部が残っているのに、Splunk が実行しないなど)、Splunk は、インデックスディレクトリを処理せず、一部破損したデータストアの上に復元するよりも、バックアップから復元することを推奨しています。Splunk は、必要に応じてスタートアップに hot ディレクトリを自動作成して、インデックスを再開します。監視されたファイルおよびディレクトリは、バックアップ時の状態から始めます。

データ保存ポリシー

インデックスにあるインデックスのサイズまたはデータの年齢を制御して、保存およびアーカイブポリシーを設定できます。

これらのファイルのサイズ、位置、年齢は、`indexes.conf` により制御されています。設定ファイルを変更する前に、変更ファイルについてお読みください。

特定のサイズを超えたファイルの削除

`indexes.conf` にあるこのエントリを検索し、それを新規値(メガバイト単位)に設定します。

```
maxTotalDataSizeMB = <non-negative number> (500000)
```

* The maximum size of an index. If an index grows bigger than this the oldest data is frozen

例 :

```
[main]
maxTotalDataSizeMB = 2500000
```

新しい設定を有効にするためにはサーバーを再起動させる必要があります。Splunk が、新しいポリシーに従うためにイベントをインデックスから取り出すため CPU 使用量が多くなり、最長で 40 分ほどの時間がかかる場合があります。

注意 : 値が正しい単位であることを確認してください。基本単位変換を Google で行えます。

Google で、“50000 megabytes in gigabytes”を検索します。

特定の年齢を超えたファイルの削除

Splunk は、バケツでデータを熟成します。特に、特定のバケツで最近のデータが設定された年齢に到達すると、バケツ全体がローテーションします。大量のイベントをインデックスしている場合、バケツサイズがすぐにいっぱいになってしまうため、バケツサイズは、ローテーションポリシーを考慮しません。index.conf で maxDataSize の値を小さくしてバケツのサイズを調整し、ローテーションを速くすることができます。ただし、小さなバケツが多い状態は、大きなバケツが少ない状態よりも検索に時間がかかります。ほしい結果を得るには、経験を積んで適切なサイズを指定するしかありません。インデックスの構造の理由上、時間とデータサイズの間には直接の因果関係はありません。

インデックスされたデータが消去されるまでの時間を秒数で、indexes.conf の変数 frozenTimePeriodInSecs に設定します。以下の例では、Splunk がインデックスから 180 日以上経つ古いイベントを処分するよう設定します。デフォルトの値は、およそ 6 年です。

```
[main]
frozenTimePeriodInSecs = 15552000
```

新しい設定を有効にするには、サーバーを再起動する必要があります。

注意 : 値が正しい単位であることを確認してください。基本単位変換を Google で行えます。

Google で、“15552000 megabytes in gigabytes”を検索します。

手動によるローテーション

指定されたインデックスのバケツを hot から warm へローテーションするには、以下のコマンドを使って、<index_name> をローテーションするインデックスの名前に置き換えます。

CLI の場合 : `./splunk search "| debug cmd=roll index=<index_name>"`

検索バーの場合 : `| debug cmd=roll index=<index_name>`

設定情報のバックアップ

設定情報のバックアップ

設定をバックアップするには、`$SPLUNK_HOME/etc/`(`$SPLUNK_HOME` は Splunk をインストールしたディレクトリで、デフォルトで `/opt/splunk`) のアーカイブまたはコピーを作成します。このディレクトリは、Splunk インストールのデフォルトおよびすべてのカスタム設定、および、保存済み検索、ユーザアカウント、タグ、カスタムソースタイプ名および設定ファイルを含むすべてのアプリケーションを含んでいます。

このディレクトリを、新しい Splunk インスタンスにコピーして回復します。これを行うために Splunk を停止する必要はありません。

ローテーションとアーカイブポリシーの設定

ローテーションとアーカイブポリシーの設定

インデックスにあるインデックスのサイズまたはデータの年齢を制御して、ローテーションとアーカイブポリシーを設定します。

Splunk データをバックアップする最善の方法については、Deployment Wiki の「バックアップの最善方法」をご覧ください。「バケツ」に関する関連事項および Splunk のバケツの使用法は、Deployment Wiki の「バケツを理解する」をご覧ください。

警告： データローテーションおよびアーカイブポリシー設定を変更するたびに、Splunk はユーザーに通知せずに古いデータを削除します。

注意： すべてのインデックスの位置は、データ設定のために書き込み可能でなければいけません。

Splunk インデックスは、ローテーションの 4 つのステージで行われます。インデックスが frozen ステージに到達すると、Splunk はデフォルトですべての frozen データを削除します。`$SPLUNK_HOME/etc/system/local/indexes.conf` にある有効な `coldToFrozenScript` (または `$SPLUNK_HOME/etc/apps` にあるカスタム app ディレクトリ) を指定して、データの紛失を回避する必要があります。

ローテーションステージ	説明	検索可能
Hot	書き込みを受け入れます。各インデックスに一つのみ。	可能
Warm	Hot からローテーションするデータ。多くの warm インデックスがあります。	可能
Cold	warm からローテーションしたデータ。多くの cold のバケツがあります。	検索の時間の範囲が Cold ステージでデータに適用したときにのみ。
Frozen	Cold からローテーションしたデータ。削除の対象	デフォルトで Splunk は frozen データを削除します。

Splunk は、`indexes.conf` にあるインデックスのサイズ、場所、および年齢を定義します。

注意： `$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` にあるカスタムアプリケーションディレクトリにある `indexex.conf` を編集します。設定ファイル一般に関する詳しい情報は、設定ファイルの機能をご覧ください。default にあるコピーは編集しないでください。

指定サイズを超えるファイルの削除

インデックスが、指定された最大サイズを超えた場合、最も古いデータは、frozen にアーカイブされます。最大サイズを設定するには、以下のラインをカスタム `indexes.conf` に追加します。

```
maxTotalDataSizeMB = <non-negative number> (500000)
```

例：

```
[main]
maxTotalDataSizeMB = 2500000
```

新しい設定を有効化するためには Splunk を再起動させる必要があります。Splunk が、新しいポリシーに従うためにイベントをインデックスから取り出すため CPU 使用量が高くなり、最長で 40 分ほどの時間がかかる場合があります。

注意： `maxTotalDataSize` =用に指定するデータサイズが、メガバイト単位で表現されているよう確かめてください。

指定年齢を超えたファイルの削除

Splunk は、バケツでデータを熟成します。特に、特定のバケツで最近のデータが設定された年齢に到達すると、バケツ全体がローテーションします。大量のイベントをインデックスしている場合、バケツサイズがすぐにいっぱいになってしまうため、バケツサイズは、ローテーションポリシーを考慮しません。`indexex.conf` で `maxDataSize` の値を小さくしてバケツのサイズを調整し、ローテーションを速くするようにできます。ただし、小さなバケツが多い状態は、大きなバケツが少ない状態よりも検索に時間がかかります。ほしい結果を得るには、経験を積んで適切なサイズを指定するしかありません。インデックスの構造の理由上、時間とデータサイズの間には直接の因果関係はありません。

インデックスされたデータが消去されるまでの時間を秒数で、`indexes.conf` の変数 `frozenTimeperiodInSecs` に設定します。以下の例では、Splunk がインデックスから 180 日以上経つ古いイベントを処分するよう設定します。デフォルトの値は、およそ 6 年です。

```
[main]
frozenTimePeriodInSecs = 15552000
```

新しい設定を有効化するためには Splunk を再起動させる必要があります。

注意： `frozenTimePeriodInSecs` =用に指定する時間が、秒単位で表現されているよう確かめてください。

アーカイブの自動化

アーカイブの自動化

Splunk が、古くなったデータを自動的にアーカイブするよう設定します。これを行うには、`indexes.conf` を設定し、

`$SPLUNK_HOME/bin` に位置するアーカイブスクリプトを呼び出します。`$SPLUNK_HOME/etc/system/local/`、または `$SPLUNK_HOME/etc/apps/` にあるカスタムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイル一般に関する詳しい情報は、設定ファイルの機能をご覧ください。`default` にあるコピーは編集しないでください。

注意：デフォルトでは、Splunk は、すべての frozen データを消去します。データの紛失を避けるために、`$SPLUNK_HOME/etc/system/local/indexes.conf` にある有効な `coldToFrozenScript` (または、`$SPLUNK_HOME/etc/apps` にあるカスタム app ディレクトリ) を指定しなければいけません。

アーカイブを署名

Splunk は、アーカイブ署名をサポートします。これを設定すると、アーカイブを修復するときにインテグリティを確認できます。

Splunk のインデックスエージングポリシーを使用したアーカイブ

Splunk は、データローテーションポリシーに基づいてインデックスから古いデータをローテーションします。データは、ファイルディレクトリの位置に一致するいくつかのステージを移動します。データは、hot データベース `$SPLUNK_HOME/va/lib/splunk/defaultdb/db/db_hot` から始まります。そして、データは、**warm** データベース `$SPLUNK_HOME/va/lib/splunk/defaultdb/db` を移動します。最終的にデータは、**cold** データベース `$SPLUNK_HOME/va/lib/splunk/defaultdb/colddb` で熟成されます。

最後に、データは **frozen** 状態に到達します。Splunk は、frozen インデックスデータが、`indexex.conf` の `frozenTimePeriodinSecs` より古くなると削除します。`coldToFrozenScript` (また `indexes.conf` で指定されている) は、frozen データが削除される前に実行します。デフォルトスクリプトは単に、例えば `/opt/splunk/var/lib/splunk/defaultdb/colddb` のようなローテーションしたディレクトリの名前を、ログファイル `$SPLUNK_HOME/var/log/splunk/splunkd_stdout.log` に書き込みます。

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

```
[<index>]
```

```
coldToFrozenScript = <script>
```

- [`<index>`]
 - ◆ アーカイブするインデックスを指定します。
- `coldToFrozenScript = <script>`
 - ◆ `<script>` を変更して、使用するアーカイブスクリプトを指定します。
 - ◆ `splunk/bin` に関連した `<$script>` パスを定義します。
 - ◆ Splunk には、使用できる 2 つのデフォルトアーカイブスクリプトが付属しています。
 - ◆ **注意：**これらのスクリプトを改名し、その後変更して、インストールの場所を設定します。デフォルトでは、場所は、`opt/tmp/myarchive` に設定されています。
 - ◆ `compressedExport.sh`: gz として圧縮された `tsidx` ファイルでエクスポート。
 - ◆ `flatfileExport.sh`: フラットテキストファイルをしてエクスポート。

注意：使用するスクリプトを改名、または他の場所へ移動させて(さらに、`indexes.conf` でその場所を指定)、Splunk

をアップグレードするときに、変更が上書きされるのを回避します。

- Windows ユーザーは、次の表記を使用します。coldToFrozenScript = <script> "\$DIR"
 - ◆ <script>は以下のいずれか。
 - ◆ compressdExport.bat (ここでスクリプトをダウンロード)。
 - ◆ flatfileExport.bat (ここでスクリプトをダウンロード)。

注意：使用するスクリプトを改名、または他の場所へ移動させて(さらに、indexes.conf でその場所を指定)、Splunk をアップグレードするときに、変更が上書きされるのを回避します。

アーカイブデータの修復

アーカイブデータの修復

アーカイブデータは、アーカイブを解凍ディレクトリ/var/lib/splunk/defaultdb/thaweddb に移動して修復されます。プラットフォームの種類に関係なく、アーカイブを Splunk サーバーに修復できます。thaweddb のデータは、サーバーのインデックスエージング計画(hot> warm> cold> frozen)の対象になりません。必要な限り、古いアーカイブデータを thawed に保管できます。データが不要になった場合、単純に削除する、または thawed の外へ移動させます。

アーカイブデータ修復方法の詳細は、アーカイブ方法により異なります。

注意：Splunk のどのインデックスまたはインスタンスにもアーカイブデータを修復できます。アーカイブデータを、アーカイブされる前の位置に修復する必要はありません。

復旧コマンドで修復

復旧コマンドは、Splunk の CLI を使用し、アーカイブからイベントを抜粋して修復できます。アーカイブ位置、修復イベントを保管するインデックス、および修復の時間範囲を指定します。

コマンドの構文は、以下のとおりです。

```
resurrect archive_directory index from_time end_time
```

注意：thaweddb で追加または削除するときには、サーバーを停止して起動する必要はありません。

Splunk の CLI を使用するには、\$SPLUNK_HOME/bin ディレクトリに移動して、./splunk コマンドを使用します。

例：

```
./splunk resurrect /tmp/myarchive oldstuff 01/01/2000:00:00:00 01/01/2001:00:00:00
```

このコマンドは、/tmp/myarchive にあるアーカイブで見つかる西暦 2000 年からのイベントを修復します。このイベントは、oldstuff インデックスに置かれます。圧縮されたインデックスでアーカイブした場合、Splunk はインデックスを解凍します。インデックスなしでアーカイブ下場合、Splunk は、インデックスを再構築します。

アーカイブデータの使用が終わったら、unresurrect で削除できます。Unresurrect は、修復アーカイブからイベントの一部を削除するのにも使用できます。

例：、

```
./splunk unresurrect oldstuff 07/01/2000:00:00:00 08/01/2000:00:00:00
```

インデックス oldstuff から7月からのイベントを削除します。

コピーされたインデックスアーカイブの修復

以前保存したアーカイブを thawed ぬにコピーまたは移動できます。db ファイル全体を移動する場合は、時間とインデックスを指定する代わりに cp を使用します。

```
# cp -r db_1181756465_1162600547_0 $SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb
```

データセキュリティの設定

Splunk で何が保護できるか

Splunk で何が保護できるか

Splunk Server が保護できます。監査設定により、暗号署名およびイベントハッシングを含むデータセキュリティが可能になります。

認証

認証には、SSL や HTTPS、ユーザーベースのアクセスコントロール(役割と呼ぶ)および LDAP があります。

SSL/HTTPS

SSL は、Splunk のバックエンド (ブラウザと会話する `splunkd`) およびフロントエンド (Splunk Web にログインするときの HTTPS) の両方で設定できます。Splunk のバックエンド用にを設定するには、この指示に従います。

Splunk で HTTPS を有効にするには、以下の指示に従います。

役割の設定

管理者、パワー、またはユーザーなど、Splunk のデフォルト役割を使用する必要はありません。これらの役割が Splunk に作られると、機能リストに独自の役割を定義できます。`authorize.conf` を使って Splunk ユーザー用に柔軟性のある役割りを作成します。

役割の設定について詳しくお読みください。

LDAP

Splunk は、内部認証サーバーまたは既存の LDAP サーバーによる認証をサポートしています。

LDAP の設定について詳しくお読みください。

スクリプト認証

スクリプト認証を使用して Splunk の認証を外部の認証システム(RADIUS、PAM など)を関連付けます。

スクリプト認証について詳しくお読みください。

監査

Splunk には、ユーザーがデータの信頼性を追跡できる監査機能が含まれています。ファイルシステム変更モニターでファイルやフォルダを観察する、監査イベントで Splunk の監視動作 (検索または設定変更など) を監視する、監査イベント署名で監査イベントに暗号署名する、IT データ署名で Splunk インデックスに入力されるデータの署名をブロックします。

ファイルシステム変更モニタ

Splunk プレビューのファイルシステム変更モニタを使用して、ディレクトリまたはファイルを観察します。Splunk は、ファイルシステムに変更が施されたとき、または観察ファイルを誰かが編集したときにイベントをインデックスします。ファイルシステム変更モニタの動作は、`inputs.conf` で完全に設定可能です。

ファイルシステム変更モニタを設定する方法について詳しくお読みください。

監査イベント

監査イベントを監視して Splunk のインスタンスを観察します。監査イベントは、誰かが Splunk のインスタンス(検索、設定変更、管理操作など) にアクセスすると生成されます。各監査イベントには、変更の内容、箇所、時期、および変更した人物を示す情報が含まれます。監査イベントは、分散 Splunk の設定で Splunk サーバーの設定およびアクセスコントロールの変更を検出するときに特に便利です。

監査イベントのしくみについて詳しくお読みください。

監査イベントの署名

Splunk をエンタープライズライセンスで使用する場合、監査イベントに暗号署名を付ける設定ができます。監査イベントの署名を行うと、連番 (データ間のギャップを検出して不正を明らかにするため) を付加し、暗号化されたハッシュ署名を各監査イベントに付加します。

`audit.conf` および `inputs.conf` にスタanzasを設定して監査を続行します。

監査イベントの署名について詳しくお読みください。

IT データの署名

Splunk をエンタープライズライセンスで使用する場合、Splunk でインデックス時に IT データの整合性を検証するよう設定できます。IT データの署名を有効にすると、Splunk は署名を作成してインデックス時のデータをブロックします。署名を使うと、データ間のギャップまたは不正なデータを検出できます。

IT データの署名について詳しくお読みください。

HTTPS で Splunk に安全にアクセスする

HTTPS で Splunk に安全にアクセスする

SSL で Splunk サーバーに安全にアクセスする

SSL で Splunk サーバーに安全にアクセスする

検索ピアに対する証明書配布

検索ピアに対する証明書の配布

Splunk インスタンスで分散検索を有効にすると、(再起動した後)、`$SPLUNK_HOME/etc/auth/distServerKeys/` にキーが生成されます。

サーバーからサーバーに `$SPLUNK_HOME/etc/auth/distServerKeys/trusted.pem` および `private.pem` ファイルを配布すると、分散検索が可能になります。

複数の Splunk インスタンスによる異なるキーのサポート

Splunk インスタンスは、別のインスタンスに保管されている承認用の独自の固有証明書を持つことができます。インスタンスは、`$SPLUNK_HOME/etc/auth/distSearchKeys/<peer_name>/<trusted|private>.pem` にキーを保管します。

例：A および B という Splunk インスタンスがあるとき、それぞれが異なるキーを持ち、Splunk インスタンス C を検索する場合は、以下を行います。

- ピア C で、`create $SPLUNK_HOME/etc/auth/distSearchKeys/A/` および `etc/auth/distSearchKeys/B/` を作成します。
- 次に、キー A を `$SPLUNK_HOME/etc/auth/distSearchKeys/A/` にコピーして、キー B を `$SPLUNK_HOME/etc/auth/distSearchKeys/B/<code/>` にコピーします。
- 最後に C を再起動します。

ファイルシステムに対する変更の監視

ファイルシステムに対する変更の監視

Splunk のファイルシステム変更モニタは、ファイルシステムの変更を追跡するときに便利です。ファイルシステム変更モニタは、指定したディレクトリを観察して、そのディレクトリに変更があると Splunk にイベントを生成します。これは、詳細に設定することができ、システムのファイル(Splunk 専用ファイルに限らない)を編集、削除、追加すると検出されます。例えば、ユーザーは、ファイルシステム変更モニタに `/etc/sysconfig/` を観察して、システムの設定が変更されたら警告するよう指示できます。

ファイルシステム変更モニタは、`inputs.conf` で設定します。

注記：Windows のファイル読み込みを監査する場合は、Splunk Community ベストプラクティス Wiki のこのトピックを参照してください。Windows に標準装備される監査ツールの方が使いやすいと感じるユーザーがいる場合もあります。

ファイルシステム変更モニタのしくみ

ファイルシステム変更モニタは、以下を使用して変更を検出します。

- 変更の日付と時間
- グループ ID
- ユーザー ID
- ファイルモード(読み込み/書き込み属性など)
- ファイルコンテンツの SHA256 ハッシュ(オプション)

ファイルシステム変更モニタの以下の機能が設定できます。

- 正規表現を使用したホワイトリスト
 - ◆ 必ずチェックするファイルを指定
- 正規表現を使用したブラックリスト
 - ◆ スキップするファイルを指定
- ディレクトリ再帰
 - ◆ シンボリックリンクトラバーサルを含む
 - ◆ 複数ディレクトリの検索(独自のポーリング頻度で実施)
- 暗号署名
 - ◆ ファイルシステム変更の分散監査追跡を作成
- 追加/変更でファイル全体をインデックス
 - ◆ ファイル全体を送信するためサイズをカットオフしてハッシュ
- Splunk でインデックスされ検索可能なすべてのイベント

ファイルシステム変更モニタの設定

デフォルトで、ファイルシステム変更モニタは、\$SPLUNK_HOME/etc/ のコンテンツが変更、削除、追加されるとイベントを生成します。初めて Splunk を起動するとき、\$SPLUNK_HOME/etc/ ディレクトリおよびすべてのサブディレクトリの各ファイルに監査イベントが追加されます。その後は、設定(変更者に関係なく)に変更があると、影響のあるファイルに対して監査イベントを生成します。監査イベントは、監査インデックス (index=_audit) にインデックスされます。

ファイルシステム変更モニタを使用し、スタanzasを inputs.conf に追加してディレクトリを監視できます。

自分専用の inputs.conf を \$SPLUNK_HOME/etc/system/local/ に作成します。

\$SPLUNK_HOME/etc/system/local/、または \$SPLUNK_HOME/etc/apps/ の独自のカスタムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイルの全般的な内容については、設定ファイルのしくみを参照してください。

[fschange] スタanzasを編集して、ファイルシステム変更モニタを設定します。スタanzas名 schange:<directory or file to monitor> を除いて、すべての設定は任意です。

注記 : [fschange] スタanzasに変更を加えた場合は、必ず Splunk を再起動してください。

[fschange:<監視するディレクトリまたはファイル>]

index=<インデックス名>

recurse=<true | false>

followLinks=<true | false>

pollPeriod=N

hashMaxSize=N

fullEvent=<true | false>

sendEventMaxSize=N

signedaudit=<true | false>

filters=<フィルタ 1>,<フィルタ 2>,...<フィルタ N>

可能な属性/値ペア

[fschange:<監視するディレクトリまたはファイル>]

- システムは、このディレクトリおよびサブディレクトリに対するすべての追加/更新/削除を監視します。
- あらゆる変更により Splunk がインデックスしたイベントを生成します。
- デフォルトは \$SPLUNK_HOME/etc/

index=<インデックス名>

- 生成されたすべてのイベントを保管するためのイベント
- デフォルトは `_audit`

recurse=<true | false>

- `true` の場合、[fschange] で指定したディレクトリ内のディレクトリを再帰します。
- デフォルトは `true`

followLinks=<true | false>

- `true` の場合、ファイルシステム変更モニタはシンボリックリンクを使います。
- デフォルトは `false`

注意: followLinks の設定は慎重に行わないと、システムループが発生する場合があります。

pollPeriod=N

- このディレクトリで変更を N 秒毎にチェックします。
- デフォルトは 3600
 - ◆ 変更すると、ファイルシステム監査イベントが、1~3600 秒で生成され、監査検索で利用可能になります。

hashMaxSize=N

- N サイズ(バイト)以下の各ファイルに対して SHA1 ハッシュを算出します。
- このハッシュは、ファイル/ディレクトリの変更を検出するための追加方法として利用可能です。
- デフォルトは -1 (変更検出にハッシュは使わない)

signedaudit=<true | false>

- 暗号署名の付いた追加/更新/削除イベントを送信します。
- デフォルトは `false`
- `true` を設定すると、`_audit` インデックスにイベントを生成します。
- インデックスを設定する場合は、これを `false` に設定します。

注記: signedaudit を `true` に設定する場合は、`audit.conf` で監査が有効になっていることを確認してください。

fullEvent=<true | false>

- 追加または変更が検出された場合にフルイベントを送信します。

- `sendEventMaxSize` 属性で確認されます。
- デフォルトは `false`

`sendEventMaxSize=N`

- イベントのサイズが `N` バイト以下の場合に限りフルイベントを送信します。
- これは、インデックスされたファイルデータのサイズを制限します。
- デフォルトは `-1` (無制限)

`sourcetype = <文字列>`

- この入力に対するソースタイプを設定します。
- `"sourcetype="` は、`<文字列>` の前に自動付加されます。
- デフォルトは `sourcetype = fs_notification`

`filesPerDelay = <整数>`

- `<整数>` ファイルの処理後に `'delayInMills'` で指定された遅延を注入します。
- これによりファイルシステム監視を絞り込み、CPU の消費を軽減します。

`delayInMills = <整数>`

- 各 `<整数>` ファイルの処理後に `'filesPerDelay'` で指定されたミリ秒の遅延を使用します。
- これによりファイルシステム監視を絞り込み、CPU の消費を軽減します。

`filters=<フィルタ 1>,<フィルタ 2>,...<フィルタ N>`

各フィルタは、監視ポーリングサイクルで発見された各ファイルまたはディレクトリに対して左から右に適用されます。

フィルタを定義するには、以下のとおりに `[filter...]` スタンザを追加します。

```
[filter:blacklist:backups]
```

```
regex1 = .*bak
```

```
regex2 = .*bk
```

```
[filter:blacklist:code]
```

```
regex1 = .*¥.c
```

```
regex2 = .*¥.h
```

```
[fschange:/etc]
```

```
filters = backups,code
```

Fschange ホワイトリスト/ブラックリストロジックは、通常のファイアウォールと同様に渡されます。イベントは、フィルタリストを最初の一一致に到達するまで渡します。イベントと一致する最初のフィルタがホワイトリストの場合は、イベントをインデックスします。イベントと一致する最初のフィルタがブラックリストの場合は、イベントはインデックスされません。イベントが一一致せずにチェーンの最後に到達すると、それがインデックスされます。インデックスなしデフォルトを作成するには、すべてのイベントに対してチェーンをブラックリストで終了します。例：

```
filters = <フィルタ 1>, <フィルタ 2>, ... terminal-blacklist
```

```
[filter:blacklist:terminal-blacklist]
```

```
regex1 = .
```

アーカイブ署名の設定

アーカイブ署名の設定

アーカイブ署名を使用して、Splunk データをアーカイブ (colddb から frozen に移動) として署名します。こうすると、アーカイブをリストアするときに識別しやすくなります。自動アーカイブポリシーを設定して、スライスのサイズを設定します。

アーカイブ署名のしくみ

データは以下の場合に colddb から frozen にアーカイブされます。

- colddb のサイズが指定した最大値に到達した
- colddb のデータが特定の年齢に到達した

自動アーカイブポリシーを指定して、データをアーカイブする方法を定義します。

Splunk には 2 つの標準スクリプト標準装備されていますが、作成することもできます。データは、() で指定する coldToFrozen スクリプトで colddb から frozen にアーカイブされます。coldToFrozen は、データのフォーマット (gz、raw など) とアーカイブ先を Splunk に指示します。アーカイブ署名は、coldToFrozen スクリプトがデータをアーカイブ形式にフォーマットした後に行われます。その後、データは、アーカイブポリシーに従って指定されたアーカイブの場所に移されます。

アーカイブ署名は、データスライスにあるすべてのデータのハッシュ署名です。

アーカイブ署名を呼び出すには、スタンドアロンの signtool ユーティリティを使います。signtool -s <path_of_archive> を coldToFrozen スクリプトのデータフォーマットラインの後でデータをアーカイブにコピーするラインの前に追加します。coldToFrozen スクリプトの設定については、後述のセクションを参照してください。

アーカイブデータ署名の検証

Splunk は、リストア時にアーカイブデータ署名を自動検証します。signtool -v <path_to_archive> を使用して手動で署名を検証することもできます。

coldToFrozen スクリプトの設定

signtool ユーティリティでラインを追加して coldToFrozen スクリプトを設定します。

注記：Splunk 標準アーカイブスクリプトを使用する場合は、スクリプトの名前を変更する、または別の場所 (indexes.conf にその場所を指定) に保存するのいずれかを行って、Splunk をアップグレードするときに上書きされないようにしてください。

Splunk 標準アーカイブスクリプト

Splunk に標準装備されている 2 つの標準アーカイブスクリプトを、アーカイブ署名と共に下に示します。

Splunk のアーカイブスクリプト

compressedExport.sh

このスクリプトは、tsidx ファイルを gz に圧縮してエクスポートします。

```
#!/bin/sh
gzip $1/*.tsidx
signtool -s <アーカイブのパス> # ここに署名するアーカイブのパスを記述
cp -r $1 /opt/tmp/myarchive # このにアーカイブディレクトリを記述
```

flatfileExport.sh

このスクリプトは、フラットテキストファイルをエクスポートします。

```
#!/bin/sh
exporttool $1${1}/index.export meta::all
rm -rf ${1}/*.data
rm -rf ${1}/rawdata
rm -rf ${1}/*.tsidx
signtool -s <アーカイブのパス> # ここに署名するアーカイブのパスを記述
cp -r $1 /opt/tmp/myarchive # このにアーカイブディレクトリを記述
```

カスタムスクリプト

カスタマイズしたスクリプトを使用して、データを cold から frozen に移動できます。

データスライスの署名または検証

`$(SPLUNK_HOME)/etc/bin}}` にある `signtool` を使用して、データスライスがアーカイブされたとき、またはアーカイブの整合性を検証するときに署名します。

構文

署名する場合：

```
signtool [-s | --sign] archive_path
```

検証する場合：

```
signtool [-v | --verify] archive_path
```

暗号署名監査イベント

暗号署名監査イベント

Splunk は、監査を有効にすると、監査イベントを作成および署名して監査追跡情報を作成します。監査イベント署名は、Splunk をエンタープライズライセンスで実行する場合にのみ利用可能です。

監査イベント署名のしくみ

監査プロセッサは、連番 ID をイベントに適用し、その連番 ID およびイベントのタイムスタンプからハッシュ署名を作成して監査イベントに署名します。

連番とギャップ検出

連番 ID を使うと、システムの不正を識別可能なデータ内のギャップを検出できます。監査イベントを検索して、ギャップが検出されたか判別できます。

```
index=_audit | audit
```

イベントのステータスを持つフィールドは「正当性」と呼びます。この値には以下があります。

- VALIDATED - このイベントの前にギャップはなく、イベント署名が一致する
- TAMPERED - イベント署名が一致しない
- NO SIGNATURE - 署名が見つからない

ギャップステータスを持つフィールドは、「ギャップ」と呼びます。この値には以下があります。

- TRUE - ギャップが発見された
- FALSE - ギャップは発見されなかった
- N/A - ID が見つからない

ハッシュ暗号化

処理された監査イベントの場合、Splunk の監査プロセッサが、全データの SHA256 ハッシュを計算します。その後、プロセッサは、ハッシュ値を暗号化して、Base64 エンコーディングに適用します。その後、Splunk はこの値を `audit.conf` で指定した何らかのキー(プライベートキー、またはデフォルトキー)と比較します。

監査イベント署名の設定

`audit.conf` で Splunk の監査機能に対する以下の設定を行います。

- 監査イベント署名のオン/オフの切り替え
- デフォルトのパブリック/プライベートキーの設定

`audit.conf` の設定

自作の `audit.conf` を作成します。`$SPLUNK_HOME/etc/system/local/` または `$SPLUNK_HOME/etc/apps/` のカスタム

タムアプリケーションディレクトリにあるこのファイルを編集します。設定ファイルの全般的な内容については、設定ファイルのしくみを参照してください。

`$SPLUNK_HOME/etc/apps/` の `genAuditKeys.py` を使用して自分のキーを生成します。

```
# python genAuditKeys.py
```

注記： "source setSplunkEnv" を実行して環境変数を設定する必要がある場合があります。

これは、プライベートおよびパブリックキー、`$SPLUNK_HOME/etc/auth/audit/private.pem` および `$SPLUNK_HOME/etc/auth/audit/public.pem` を作成します。

これらのキーを使用するには、`$SPLUNK_HOME/etc/system/local/audit.conf` にある `privateKey` および `publicKey` をキーのパスに設定します。

```
[auditTrail]
privateKey = $PATH_TO_PRIVATE_KEY
publicKey = $PATH_TO_PUBLIC_KEY
```

注記： `[auditTrail]` スタンザがない場合は、監査イベントは生成されますが、署名されません。 `publicKey` または `privateKey` に値がない場合、監査イベントは生成されますが、署名されません。

Splunk 動作の監査

Splunk 動作の監査

監査を有効にすると、Splunk は特定のイベントを監査インデックス(`index=_audit`)にログします。Splunk とのすべての相互関係(検索、設定変更など)は、監査イベントを生成します。ファイル変更監視で監視されるディレクトリも監査イベントを作成します。このトピックでは、監査イベントの構成と生成について説明します。

注記： `punct` フィールドは、生成時に PKI を使って書名されている、`_audit` インデックスのイベントには使えません。

監査イベントとは何か？

- タイムスタンプ：
 - ◆ イベントの日付と時刻
- ユーザー情報：
 - ◆ イベントを生成したユーザー。
 - ◆ イベントにユーザー情報がない場合は、Splunk が現在ログインしているユーザーを設定します。
- 補助情報：
 - ◆ 利用可能なイベント詳細 - ファイル名、成功/失敗など
- ID(監査イベント署名がオンの場合のみ)
 - ◆ データのギャップを検出するためのイベントに割り当てられる連番
- ハッシュ署名：
 - ◆ PKI で暗号化された SHA256 ハッシュ署名(タイムスタンプ、ID を含む)

- イベントのタイプに特定される追加の属性/値ペア

例

以下は、署名付き監査ログのエントリー例です。

```
11-01-2007 09:23:59.581 INFO AuditLogger - Audit:[timestamp=Thu Nov 1 09:23:59 2007, id=1, user=
```

最初の([]) に記述される情報は、ハッシュされた署名データ。2 つ目の文字列セットは、ハッシュ署名。

監査イベントを生成する動作とは何か？

監査イベントは、検出により生成されます。

- の設定ディレクトリ (\$SPLUNK_HOME/etc/*) にあるすべてのファイル
 - ◆ ファイルシステム変更モニタを使用するファイルの追加/変更/削除は監視されます。
- システムの開始と停止。
- ユーザーのログインとログアウト。
- 新規ユーザーの追加と削除
- ユーザー情報(パスワード、役割など)の変更
- システムにある機能は例外です。
 - ◆ に記載される機能

イベントストレージの監査

Splunk は監査イベントをローカルの監査インデックス(index=_audit)に保管します。監査イベントはログファイル (\$SPLUNK_HOME/var/log/splunk/audit.log)にログされます。

Splunk でフォワーダを分散設定に設定すると、監査イベントは別のイベントと同様に転送転送されます。署名はフォワーダまたは Splunk インスタンスの受信で行うことができます。

監査イベントの処理

audit.conf ファイルは、監査イベントを暗号化するかどうかを監査プロセッサに指示します。監査イベントが生成されると、Splunk の監査プロセッサは、連番をイベントに割り当て、イベント情報を SQLite データベースに保管します。イベント生成時にユーザー情報が指定されていない場合は、Splunk は現在のユーザー情報を使用します。最後に、監査イベント署名が設定されている場合、Splunk はイベントをハッシュして暗号化します。

監査イベントの検索

Splunk Web または Splunk の CLI で監査イベントを検索します。これは、検索を監査コマンドにパイプさせて行います。監査検索コマンドは、監査イベント署名が設定されている場合に便利です。

ただし、監査イベント署名が設定されていない (または整合性検証をスキップする) すべての監査イベントを検索する場合、監査インデックス全体を検索する場合があります。

- すべての監査イベントを検索するには、`_audit` インデックスを指定します。

```
index=_audit
```

この検索は、すべての監査イベントを返します。

- 検索を監査コマンドにパイプします。

```
index=_audit | audit
```

この検索は、監査インデックス全体を返し、`audit` コマンドで見つけた監査イベントを処理します。

監査コマンドをパイプする前に検索を絞り込みます。ただし、時間範囲による絞り込み、または単一ホストによる制限しか行えません。これは、各ホストに個別の ID 列があるためです。

連続 ID の存在により監査イベントのギャップ検出を有効にするため、複数ホストを渡る検索を絞り込むと、不正なギャップ検出の原因となる場合があります。

イベントのステータスを持つフィールドは「正当性」と呼びます。この値には以下があります。

- `VALIDATED` - このイベントの前にギャップはなく、イベント署名が一致する
- `TAMPERED` - イベント署名が一致しない
- `NO SIGNATURE` - 署名が見つからない

ギャップステータスを持つフィールドは、「ギャップ」と呼びます。この値には以下があります。

- `TRUE` - ギャップが発見された
- `FALSE` - ギャップは発見されなかった
- `N/A` - ID が見つからない

フォワーディングと受信の設定

フォワーディングと受信について

フォワーディングと受信について

環境にわたって多くのソースが分散している場合、各ソースマシンに Splunk の簡易バージョンをインストールし、そのマシンからデータを 1 つまたは複数の中央 Splunk インデクスインスタンスにフォワーディングするという方法があります。

サポート対象の OS プラットフォーム上で実行される Splunk サーバーであれば、データを(他のシステムに転送するのと同様に)他の Splunk インスタンスにリアルタイムで転送することができます。これによって、特定の環境における 1 つの Splunk ホストで収集されたデータを他の Splunk インスタンスに送り、インデックス作成および検索を行うことが可能になります。

また、Splunk インスタンスを設定し、データを他の Splunk インスタンスのグループに転送して、クラスタ化されたインデックス作成により水平スケーリングが可能になります。Splunk インスタンスはまた、データのコピーを作成して他の Splunk インスタンスの複数のグループに分け、高い可用性を有する環境においてデータの冗長性を高めることもできます。

フォワーディングと受信は全ての設定情報を含みます。設定情報では、1 つの Splunk インスタンス(フォワーダ)はデータを 1 つまたは複数の Splunk インデックスマシン(受信ホスト)に送信してからインデックスを作成します。また、フォワーダはローカルでもデータをインデックス化することができます。

古いフォワーダのバージョン

3.3.x のフォワーダは Splunk の 4.x バージョンで動作します。このことは、特に大規模なフォワーダを装備している場合に有効です。新しい展開サーバーの設定に慣れてから、フォワーダを 4.x に移動するのがよいでしょう。

フォワーディング

フォワーディングは、**フォワーディングと受信**の最も簡単なセットアップです。フォワーディングは、データを他のサーバーに送信してインデックスを作成するための Splunk インスタンスを参照します。

Splunk フォワーダには、フォワーダとライトフォワーダの 2 種類あります。それら 2 つのフォワーダの主な違いは、フォワーダのプロセス、つまりフォワーディング前のデータの加工方法であり、ライトフォワーダは、未処理のデータ、すなわちローデータを受信ホストに送信します。

Splunk フォワーダに関する詳細は、**フォワーディング設定**を参照してください。



ルーティング

ルーティングを有効にすると、フォワーダはイベント自身のパターンを基に条件を検索し、複数のイベントを選択して1つのSplunk インスタンスに送信し、他のイベントを他のインスタンスに送信します。



クローニング

フォワーディングおよび受信におけるクローニングとは、すべてのイベントを2つ以上のSplunk インスタンスに送信し、データ冗長性を構築することを特に意味します。このとき、クローニングは、2つ以上の全く同一のインデックスの作成を保証しないことに注意が必要です。受信ホストの1つが利用できない場合、データは利用できる受信ホストにのみ送信されます。その結果、同一でないインデックスが作成されることになります。



自動ロードバランシングとラウンドロビン・データバランシング

どのフォワーダからのデータを送信するかを指定することができます。指定は、自動ロードバランシングまたはデータバ

ランシングを使用して、Splunk インデックスマシンのグループを定義することに基づいて行います。これらの設定は、大容量のデータの取り扱いを可能にします。そのためには、定義したルールに基づいてフォワーダがデータを送信する先のインデクシング受信ホストのターゲットグループを作成する必要があります。フォワーダのこれらインデックスマシンの選択方法、および利用可能な受信ホストのリストの管理方法を指定することができます(各フォワーダ上のスタティックリスト、または DNS 記録を利用)。



ロードバランシング中のバッファリング

ロードバランシング中にサーバーにアクセス不能となった場合、Splunk はすべてのアクセス可能なサーバーにイベントを送信し続けます。

最終的に、Splunk は、応答のないサーバーへの送信を停止し、サーバーのオフライン状態を記録します。サーバーがすべてアクセス不能な場合、Splunk はフォワーダ側のバッファに書き込みます。

ターゲットグループ

データを単一の受信ホストに送信することが可能なことに加え、フォワーダは**ターゲットグループ**ごとのインデックスマシンに送信することができます。ターゲットグループは 1 つまたは複数の受信インデックスマシンから構成されます。既存の受信ホストは、複数のターゲットグループの一部とすることができます。

クローニングはすべてのイベントをすべてのターゲットグループに送信するのに対し、**ルーティング**は特定のイベントを 1 つのターゲットグループに送信し、異なるイベントを他のターゲットグループに送信します。また、デフォルトグループを設定することもできます。これは、ターゲットグループに送信されないデータのすべてを受信します。複数のグループが指定された場合、Splunk はすべてのリストアップされたデフォルトグループに対してイベントをクローニングします。

```
defaultGroup=<グループ名 1>,<グループ名 2>...
```

サードパーティシステムへの送信(syslog または HTTP を使用)

デフォルトでは、データは、標準の Splunk フォワーダから**処理済データ**として送信されます。すなわち、イベントはすでにインデックス化されており、*Splunk* 受信ホストに到達したときには検索が可能となります。ただし、*Splunk* フォワーダが**未処理データ**、すなわち**ローデータ**を送信できるように設定することができます。これによって、サードパーティシステムが正しくデータを処理できるようになります。特に、データを syslog アグリゲータまたは HTTP ホストに送信するように指定することができます。



セキュリティ

Splunk フォワーダは、TCP を通じて、クリアテキストまたは SSL により、受信データの一部またはすべてをリアルタイムで、他の Splunk サーバおよび他のシステムに送信することができます。

受信の設定

受信の設定

フォワーダを設定する前に、1 つ以上の受信ホストを設定しなければなりません。受信ホストを設定すると、フォワーダの設定を行うことができます。

重要：受信ホストは、フォワーダと同じかそれ以降のバージョンの Splunk を実行している必要があります。例えば、バージョン 4.0 の受信ホストは、それ以前のバージョンを実行しているフォワーダからのトラフィックを受信することができます。バージョン 3.4 の受信ホストは 4.0 のフォワーダからの接続を受け入れることはできません。

作業を開始する前に、質問事項に対する回答を次にあげます。

データはどこからやってくるか？

次のステップで選択するオプションはいくつかの事項に左右されます。その 1 つに次のものがあります。このデータはどこからフォワーディングされたのか？ 一般的に、フォワーディングされたデータは、他の Splunk インスタンス(フォワーダまたはライトフォワーダ)または Splunk でないソースから送られます。

受信するデータのフォーマットは何か？

- データが Splunk フォワーダから受信したものであれば、TCP であり**処理済**です。つまり、データはすでにフォワーダにより処理されています。
- データが Splunk ライトフォワーダから受信したものであれば、TCP であり**一部処理済**です。つまり、受信ホストはデータを処理しなければなりません。
- データが Splunk でないソース(log4j や syslog など)、TCP または UDP であり、**未処理**です。つまり、受信ホストはデータを処理しなければなりません。

Splunk Web による受信ホストの設定

Splunk Web で受信を有効にします。

- データを受信してインデックスを作成するサーバー上の Splunk Web にログインします。フォワーディングおよび受信ホストの設定を許可されたユーザーとしてログインする必要があります。ほとんどの場合、管理ユーザーとしてログインします。
- Splunk Web の右上隅にある**管理**リンクをクリックします。
- フォワーディングと受信 > フォワーダからのデータ受信を選択します。
- **新規**をクリックして、この受信ホストがリスンするポートを指定します。例えば、9997 であれば TCP ポート 9997 番でデータを受信します。
- **保存**をクリックします。プロセスを完了するには Splunk を再起動する必要があります。

重要 : Splunk インスタンスの受信は、フォワーダのバージョンと同じまたはそれ以降のバージョンの Splunk を実行している必要があります。Splunk の 4.0 インスタンスは、3.x のフォワーダからのデータを受信できますが、その逆はできません。

SplunkCLI を使用して受信ホストを設定

Splunk の CLI からの受信を有効にします。Splunk の CLI を使用するには、`$SPLUNK_HOME/bin/`ディレクトリに移動して、`./splunk` コマンドを使用します。

ログインするには次のようにします。

```
./splunk login
Splunk username: admin
Password:
```

受信ホストを有効にするには次のようにします。

```
# ./splunk enable listen 42099 -auth admin:changeme
Listening for Splunk data on TCP port 42099.
```

受信ホストを無効にするには次のようにします。

```
# ./splunk disable listen -auth admin:changeme
No longer listening for Splunk TCP data.
You must restart the Splunk Server for your changes to take effect.
```

重要 : Splunk インスタンスの受信は、フォワーダのバージョンと同じまたはそれ以降のバージョンの Splunk を実行している必要があります。

フォワーディングの設定

フォワーディングの設定

このトピックでは、Splunk フォワーダを設定する際のオプションを説明します。次のような事前設定フォワーダを選択す

ることができます。

- Splunk フォワーダ
- Splunk ライトフォワーダ

Splunk フォワーダを有効にする場合は(ただし、ライトフォワーダは有効にしない)、インデックス化されたデータのローカルコピーをフォワーディングするホストに保存するかどうかのオプションもあります。

Splunk フォワーダまたはライトフォワーダを有効にする前に次の事項をお読みください。

- フォワーディングと受信を設定する場合、Splunk 受信ホストは、フォワーダと同じかそれ以降のバージョンの Splunk を実行している必要があります。4.x の受信ホストは 3.3.x のフォワーダからのデータを受信することができますが、3.3.x の受信ホストは 4.x フォワーダからのデータを受信することはできません。
- **ライトフォワーダ**と同時にラウンドロビン・データバランシングを使用することはできません。その理由は、データが送信前にパース(処理)されていないからです。イベントが受信ホストに到達する前にバラバラに分割され、部分イベントが生じる可能性があります。ただし、自動ロードバランシングを使用することはできます。
- Splunk Web は、フォワーディングホスト上での Splunk のフットプリントを軽減するために、**ライトフォワーダではオフ**にされます。したがって、フォワーディング Splunk インスタンスを設定するために Splunk Web を使用する必要がある場合、フォワーダアプリケーションを有効にする前に、これを実行してください。フォワーダアプリケーションを有効にした後は、Splunk CLI でしかフォワーダを設定することができません。
- 分散検索を通じて検索を分散したホスト上でライトフォワーダを有効にする場合、検索をそのホスト上で実施することができなくなり、インデックスがデータを含んでいても、その**インデックスは無効化**されます。
- フォワーディングを設定する前に、受信ホストを設定する必要があります。このように、Splunk 受信ホストは、データフォワードのために準備がなされます。その次に、フォワーダを設定します。

異なる OS 間のフォワーディング

ある OS から異なる OS にフォワーディングする場合(例えば、Windows から Linux のインデックスマシンへ)、インデックスマシン上のフォワーダの OS にその OS 専用のアプリケーションをインストールする必要があります。例えば、Windows から Linux へフォワーディングする場合、Linux 受信ホストに Windows アプリケーション用の Splunk をインストールする必要があります。関連の OS 専用アプリケーションをダウンロードした後、そのアプリケーション用の `inputs.conf` を移動または削除します(Windows アプリケーションの場合は、`$SPLUNK_HOME/etc/apps/windows/default/inputs.conf`。アプリケーションを有効にする前に、そのデフォルト入力がインデックスマシンに追加されていないことを確認します)。

Splunk フォワーダとは？

Splunk フォワーダは、中央 Splunk インデックスマシンまたはインデックスマシンのグループにデータを送信することを可能にする Splunk のひとつバージョンです。以下のモジュールは (`$SPLUNK_HOME/etc/apps/SplunkForwarder/default/app.conf`)で、Splunk フォワーダについて無効化されます。

```
[modules]
```

```
distributedDeployment = disabled
```

```
distributedSearch = disabled
```

```
input/FIFO = disabled
```

これらのモジュールは、デプロイメントサーバー(デプロイメントクライアントではない)、分散検索、および名前をつけられたパイプ/FIFO です。

他のすべての機能およびモジュールは有効化されたままにします。

正確な設定の詳細は、\$SPLUNK_HOME/etc/apps/SplunkForwarder/default(SPLUNK_HOME は、Splunk をインストールするディレクトリ)にある SplunkForwarder アプリケーションのための setup.conf ファイルを参照してください。

Splunk ライトフォワードとは？

Splunk ライトフォワードは Splunk のライトウェイトバージョンであり、ローカルのログファイルとディレクトリを監視し、Windows イベントログを収集して、スクリプトインプットを使用することができます(Windows 上のローカル WMI とレジストリデータを含む)。オーバーヘッドを減らすために、標準 Splunk サーバーの他の多くの機能が無効にされます。

特に、Splunk ライトフォワードは以下を行います。

- イベントの署名と、ディスクがいっぱいかどうかのチェックを無効にする
(/\$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/default-mode.conf)
- splunkd への内部データ入力を制限し、ログのみを計測し、これらがフォワーディングされていることを確実にする (\$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/inputs.conf)
- すべてのインデックス作業を無効化する
(/\$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/indexes.conf)
- データをパースしない。したがって、ライトフォワードと受信インスタンスの両方で inputs.conf を含むアプリケーションをインストールする必要があります。
- Splunk Web インタフェースを無効化する (\$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/web.conf)
- 監視、実行および Windows イベントログ入力について、スループットを 256KBps に制限する
(/etc/apps/SplunkLightForwarder/default/limits.conf および/etc/apps/SplunkLightForwarder/config/input/*の下にある設定)
- (\$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default/app.conf)の中にある次のモジュールを無効化する:

```
[modules]
```

```
distributedDeployment = disabled
```

```
distributedSearch = disabled
```

```
input/FIFO = disabled
```

```
input/UDP = disabled
```

```
input/tcp = disabled
```

```
input/syslogFIFO = disabled
```

```
input/syslogUDP = disabled
```

これらのモジュールは、デプロイメントサーバー(デプロイメントクライアントではない)、分散検索であり、名前をつけられたパイプ/FIFO からのものであり、また、ネットワークポートから直接入力されます。

正確な設定の詳細は、\$SPLUNK_HOME/etc/apps/SplunkForwarder/default(SPLUNK_HOME は、Splunk をインストールするディレクトリ)にある SplunkLightForwarder アプリケーションのための setup.conf ファイルを参照してください。

ライトフォワーダ設定の変更

Splunk ライトフォワーダの設定を変更するには(例えば、特定の入カタイプを追加/削除するなど)、\$SPLUNK_HOME/etc/apps/SplunkLightForwarder/default にある SplunkLightForwarder の setup.conf のコピーを作成し、/default と同じレベルの新規/ローカルディレクトリにそれを保存し、必要に応じて変更します。ネットワーク帯域制限を変更するには、そのローカルディレクトリ(デフォルトは変更しない)の中に、新規[thruput]スタanzasと共に新規 limits.conf を作成して必要な制限を設定します。

Splunk Web におけるフォワーディングの設定と有効化

Splunk Web でフォワーディング(ライトウェイトフォワーディングではない)を有効にする場合、各受信ホストにフォワードすると同時に、インデックス化されたデータをローカルにコピーを保存するかどうかを選択することができます。それには次を実行します。

1. Splunk Web にログインします。
2. 右上隅にある**管理**をクリックします。
3. フォワーディングと受信 > フォワーディングデフォルトをクリックします。
4. **はい**を選択して、このホストでインデックス化されたデータのローカルコピーを保存します。この設定は、ライトウェイトフォワーディングを有効にすると、無効になります。

Splunk フォワーダまたはライトフォワーダを Splunk Web で有効にするには次のようにします。

5. Splunk Web にログインします。
6. 右上隅にある**管理**をクリックします。
7. フォワーディングと受信をクリックします。
8. 以下のフォワーディングオプションを選択します。
 - ◆ **処理済**(プリプロセス済、インデックス済)データを転送するには、**ホストへのフォワーディングを設定**をクリックし、フォーマット host:port または IP:port を使用して、1つ以上のホストの、カンマで区切られたリストを提出します。Splunk はすべてのイベントのコピーを、指定された各ホストに送信します。転送先の各ホストにおける受信を設定しておく必要があります。
 - ◆ **部処理済**(部分的にプリプロセスしただけの)データを転送するには、**ライトウェイトフォワーディングの有効化**をクリックします。フォワーディング用このホストをさらに設定するには CLI を使用する必要があります。したがって、行っている作業内容をよく理解する必要があります。

重要 : Splunk ライトフォワーダを有効にすると、Splunk Web はアクセス不能になります。CLI でのみ、この Splunk インスタンスの設定にアクセスできます。

CLI による有効化

Splunk フォワーダまたはライトフォワーダを CLI で有効にするには次のようにします。

```
./splunk enable app [SplunkForwarder|SplunkLightForwarder] -auth <ユーザ名>:<パスワード>
```

注意 : Splunk を無料ライセンスで実行している場合は、ユーザー名とパスワードを入力する必要はありません。

CLIによる有効化

Splunk フォワーダまたはライトフォワーダを CLI で有効にするには次のようにします。

```
./splunk disable app [SplunkForwarder|SplunkLightForwarder] -auth <ユーザ名>:<パスワード>
```

注意 : Splunk を無料ライセンスで実行している場合は、ユーザー名とパスワードを入力する必要はありません。

フォワーダと受信ホストとの間で SSL 暗号化を使用

フォワーダと受信ホストとの間で SSL 暗号化を使用

フォワーダと受信ホスト間のコミュニケーションで SSL 認証および暗号化、または単に暗号化だけを使用するように指定できます。

これを設定するには、各受信ホスト上の `inputs.conf` のコピーと各フォワーダ上の `outputs.conf` のコピーを編集します。数多くの Splunk フォワーダと受信ホストにこの設定を行う場合は、適当な設定ファイルのコピーを作成し、それを編集して必要な設定を反映させ、それからそれを設定を行う各システム上の `$SPLUNK_HOME/etc/system/local` ディレクトリにコピーすることができます。

フォワーダで SSL を設定するには、次の属性/値ペアを設定します。認証のために SSL を使用する場合は、認証する必要のある各受信ホストについてスタンザを追加します。

フォワーダでの SSL の設定

フォワーダで SSL を設定するには、`$SPLUNK_HOME/etc/system/local/outputs.conf` を編集します。認証のために SSL を使用する場合は、認証する必要のある各受信ホストについてスタンザを追加します。

```
[tcpout-server://$IP:$PORT]
```

```
sslCertPath=<クライアント証明書へのフルパス>
```

```
sslPassword=<認証用パスワード>
```

```
sslRootCAPath=<ルート証明書権限ファイルへのオプションパス>
```

```
sslVerifyServerCert=<true|false>
```

```
sslCommonNameToCheck=<サーバーの共通名、sslVerifyServerCert が true に設定されている場合にのみ設定>
```

```
altCommonNameToCheck=<サーバーの代替名、sslVerifyServerCert が true に設定されている場合にのみ設定>
```

`sslCertPath` キー/値ペアを、サーバー証明書ファイルへのフルパスを指定するために使用します。

- `sslRootCAPath`
 - ◆ キー/値ペアを、ルート証明書権限ファイルへのローカルパスを指定するために使用します。オプション。ルート CA がローカルの場合に設定します。

- sslPassword

- ◆ 証明書のためのパスワード。デフォルトは sslPassword=パスワード。

true に設定する場合、sslVerifyServerCert は、接続しているサーバーが有効なものであるかどうかを確認します (認定)。次に、サーバーの共通名および代替名が一致するかどうかチェックされます。デフォルトは false。

sslCommonNameToCheck は、この名前に対してサーバー証明書の共通名をチェックします。一致しない場合、このサーバーに対して認定されていないと考えられます。'sslVerifyServerCert' が true の場合、このキー/値ペアを指定する必要があります。

altCommonNameToCheck は、この名前に対してサーバー証明書の代替名をチェックします。一致しない場合、このサーバーに対して認定されていないと考えられます。'sslVerifyServerCert' が true の場合、このキー/値ペアを指定する必要があります。

暗号化のみの設定

暗号化のみで送信する場合、\$SPLUNK_HOME/etc/system/local/outputs.conf にある SSL スタンザを次のように設定します。

```
[tcpout-server://$IP:$PORT]
sslCertPath=/home/myhome/certs/foo.pem
sslPassword=パスワード
sslRootCAPath=/home/myhome/certs/root.pem
sslVerifyServerCert=<false>
```

注意: スタンザは、特定の [tcpout-server://\$IP:\$PORT] またはサーバーグループまたはデフォルトグループのいずれかで設定できます。

暗号化および認証の設定

暗号化と同様に認証のための SSL を設定するには、\$SPLUNK_HOME/etc/system/local/outputs.conf にある SSL スタンザを以下のように設定します。

```
[tcpout-server://$IP:$PORT]
sslCertPath=<クライアント証明書へのフルパス>
sslRootCAPath=<ルート証明書権限ファイルへのオプションパス>
sslVerifyServerCert=<true|false>
sslCommonNameToCheck=<サーバーの共通名、sslVerifyServerCert が true に設定されている場合にのみ設定>
altCommonNameToCheck=<サーバーの代替名、sslVerifyServerCert が true に設定されている場合にのみ設定>
```

注意: スタンザは、SSL で認定する各固有の送信用接続について記述する必要があります。

受信ホストでの SSL の設定

受信用に SSL を使用するには、2 つの事を行う必要があります。

- \$SPLUNK_HOME/etc/system/local/inputs.conf に [SSL] という名のスタンザを含めます。

[SSL]

```
serverCert=<サーバー証明書へのフルパス>  
password=<サーバー証明書パスワード(存在する場合)>  
rootCA=<証明書権限リスト(ルートファイル)>  
dhfile=<dhfile.pem へのオプションパス>  
requireClientCert=<true|false> - 認証を設定する場合、true に設定
```

serverCert キー/値ペアを、サーバー証明書ファイルへのパスを指定するために使用します。

password は、証明書がパスワードを使用する場合に使用します。オプション。

rootCA キー/値ペアを、ルート証明書権限ファイルへのパスを指定するために使用します。

接続を確立するために、クライアントからの有効な証明書をシステムが要求するようにする場合は、requireClientCert を 'true' に設定します。その他の場合には 'false' に設定します。

必要に応じて、異なるポートについて異なる証明書を使用することができます。それによって、クライアントの異なるセットが異なるポートに接続することが可能になります。

リスナースタanzasを \$SPLUNK_HOME/etc/system/local/inputs.conf に追加します。

```
[splunktcp-ssl:9996]  
queue=indexQueue
```

上記のスタanzasは、ポート 9996 上で、他の Splunk サーバーの暗号化された処理済データに対するリスナーを開始します。

```
[tcp-ssl:9995]  
queue=parsingQueue
```

上記のスタanzasは、ポート 9995 上のロー暗号化データに対するリスナーを開始します。

自動ロードバランシングの設定

自動ロードバランシングの設定

フォワーダがデータをインデックスマシンに送信する際、複数の Splunk インデックスマシンに渡って負荷が自動的にバランスするように Splunk を設定することができます。つまり、インデックスマシンがダウンしたり利用できなくなった場合(ネットワークや電源問題など)、データを送信する必要のあるフォワーダが利用できる他のインデックスマシンに送信するようにします。

各フォワーダ上の outputs.conf にある利用可能なインデックスマシンのスタティックリストを定義する、または各フォワーダ上の outputs.conf で 1 つのインデックスマシン名を指定し、DNS でその名前に応答するインデックスマシンのリストを定義することができます。各オプションにはそれぞれ次のような利点があります。

- DNS の記録に利用可能なインデックスマシンのリストを保持することにより、インデキシングバンド幅の変更が必要になったとき、インデックスマシンを追加または削除することが容易になります。
- 各フォワーダ上でインデキシングホストの固定リストを作成することにより、各インデックスマシンについて異

なるポートを指定することが可能になります。つまり、1つのホスト上で複数の Splunk インデックスマシンを実行している場合、インスタンス毎に異なるポートを指定することにより、それらの間でフォワーディング負荷を分散することが可能になります。

動作のしくみ

フォワーダの出力設定は `outputs.conf` で設定します。`outputs.conf` のデフォルトコピーはないため、すべてのフォワーダに適用する標準 `outputs.conf` を作成し、次に、それを各フォワーディングインスタンスについて `$SPLUNK_HOME/etc/system/local/` にコピーすることができます。`outputs.conf.spec` のコピーを開始ポイントとして使用することができますが、実行ニーズに合うように、十分にそれを検討し編集する必要があります。`outputs.conf.spec` 内の情報を見れば、どのようなデフォルト値がオプションとしてあるかを知ることができます。

フォワーダを指定して自動ロードバランシングを設定するには、`outputs.conf` でグローバル設定スタanzasを設定し(後述)、次に 1つまたは複数のインデックスマシンのターゲットグループを定義します。グローバルスタanzasで行なった設定は、個別のインデックスマシンまたはインデックスマシンのターゲットグループについて異なる設定を指定しない限り、すべての受信インデックスマシンに適用されます。特定のインデックスマシンまたはインデックスマシングループについて定義する設定は、より高いレベルで定義したものを上書きします。次に、グローバルスタanzasまたは各ターゲットグループベースで、自動ロードバランシングを有効にし設定することができます。

なぜ複数のターゲットグループなのか？ 複数のインデックスマシンのターゲットグループがある場合、指定フォワーダはデータのコピーを両方のターゲットグループに送信します。これは、2つの異なる Splunk インデックスマシンがある場合に有利になることがあります。例えば、Splunk デプロイメントを新しいバージョンにアップグレードし、既存のデプロイメントと平行してその新規バージョンをテストしたい場合、この設定により、同じデータを両方のインデックスマシンに送信することが可能になります。あるいは、単にインデックス化されデータのバックアップインスタンスが必要な場合もあります。

注意: `outputs.conf` のコピーを `$SPLUNK_HOME/etc/system/local/` に置き、スタanzasのヘディング以外になにも指定しない場合、Splunk はデフォルト値をそのファイルに書き込みます。次の例にある値は、これらのデフォルトを必ずしも反映しません。

グローバル設定

グローバル設定は`[tcpout]`スタanzasに設定されます。グローバル設定は、特定のターゲットグループまたは個別のサーバーの設定で上書きされない限り有効です。

```
[tcpout]
defaultGroup = my_indexers
disabled = false
prependKeyToRaw = <キー>
```

注意点：

- このスタanzasのみを含むフォワーダ上の `$SPLUNK_HOME/etc/system/local/` に `outputs.conf` をコピーし、このフォワーダがデータを送信する先の `my_indexers` と呼ばれるインデックスマシンのデフォルトターゲットグループ

プを定義してある場合、次に説明するとおり、そのターゲットグループについての自動ロードバランシングその他の動作を定義することができます。

- `prependKeyToRaw` はオプションです。ここでキーを設定する場合、Splunk がそのキーをすべてのイベントを参照します。イベントがこのキーを含む場合、その値は、インデキシングサーバーに送信されるローデータの先頭に追加されます。これは、`sendCookedData = false` の場合にのみ有効になります。キー/値ペアおよびその引き出し方は、`props.conf` と `transforms.conf` で設定します。syslog ファイルを監視し、それを syslog サーバーに送信することにより得た syslog イベントに `<priority>` を追加するのにこれを使用する必要がある場合があります。このようにイベントを特定する特別な必要性がない場合、これを含めないでください。

ターゲットグループ設定

ターゲットグループについて名付けられたスタンザにおいて、ターゲットグループに特定の詳細を設定するには次のようになります。

```
[tcpout:my_indexers]
disabled = false
autoLB = true
autoLBFrequency = 10
server = fflanda-lb.fflanda.com:9995
```

- `autoLB` - これを `true` に設定すると、自動ロードバランシングが有効になります。
- `autoLBFrequency` - 新規インデックスマシンが、ランダムに利用可能なインデックスマシンのリストから選択された後のインターバルを秒で設定します。
- `server` - 単独のインデックスマシン、またはインデックスマシンのリスト。
- Splunk は、完全に確認されたドメイン名をここで指定することを推奨します。Splunk は IP アドレスに対する名前を検索し、それらの間の負荷バランスと共に記憶します。

DNS レコードとはどのようなものか

完全に確認されたドメイン名を使用し、データを分散するためのフォワーダについて 3 つのインデックスマシンがある場合、DNS レコードは次のようになります。

```
fflanda-lb    A      10.10.10.1
fflanda-lb    A      10.10.10.2
fflanda-lb    A      10.10.10.3
```

また、`nslookup` 出力は次のようになります。

```
$ nslookup fflanda-lb
Server: 127.0.0.1
Address: 127.0.0.1#53
Name: fflanda-lb.fflanda.com
Address: 10.10.10.2
Name: fflanda-lb.fflanda.com
```

Address: 10.10.10.3

Name: fflanda-lb.fflanda.com

Address: 10.10.10.1

また、autoLBFrequency で指定された時間が過ぎると、nslookup 出力は次のようになります(順番が変わる)。

```
$ nslookup fflanda-lb
```

```
Server: 127.0.0.1
```

```
Address: 127.0.0.1#53
```

```
Name: fflanda-lb.fflanda.com
```

```
Address: 10.10.10.3
```

```
Name: fflanda-lb.fflanda.com
```

```
Address: 10.10.10.1
```

```
Name: fflanda-lb.fflanda.com
```

```
Address: 10.10.10.2
```

インデックスマシンへの接続が中断または利用不能になった場合

- あるインデックスマシンが利用できない場合、Splunk はそのインデックスマシンを監視し続けます。利用可能となると、利用可能インデックスマシンリストに再び追加されます。
- インデックスマシンを使用中に接続が中断した場合、その動作は次の状況によります：ひとかたまりのデータがすでに部分的にインデックスマシンに送信されている場合、Splunk はそのデータの残りを同じインデックスマシンに送信しようとします。接続が中断し、5 秒以内に復旧した場合、残りのデータは復旧した接続を通じて送信されます。接続が 5 秒以上利用できない場合、残りのデータは新たに選択された利用可能インデックスマシンに送信されます。

ターゲットグループについてのオプション属性

これらの属性はオプションです。これらの属性は、グローバルまたは各ターゲットグループベースで設定することができます。

- sendCookedData=true/false
 - ◆ true の場合、イベントは処理されます(Splunk により処理され、ローの状態ではない)。
 - ◆ false の場合、イベントはローの状態で、送信前には何の処理もされません。
 - ◆ デフォルトは true。
- heartbeatFrequency=60
 - ◆ ハートビートパケットを受信ホストに送信する頻度を秒で指定します。
 - ◆ ハートビートは、sendCookedData=true の場合にのみ送信されます。
 - ◆ デフォルトは 30 秒。

ターゲットグループについてのキュー設定

フォワーダがデータを送信する際、データはキューに入ってから、フォワーダから送信されます。データを受信するイン

デックスマシンがない場合、このスタンプは、キューに入ったデータに対する処理を判断します。これらの属性は、グローバルまたは各ターゲットグループベースで設定することができます。

- `maxQueueSize=20000`
 - ◆ キューに入ったイベントの最大数(キューサイズ)
 - ◆ デフォルトは 1000。
- `usePersistentQueue=false`
 - ◆ `true` に設定され、キューがいっぱいの場合、イベントをディスクに書き出します。
 - ◆ ディレクトリは `persistentQueuePath` で設定されます。
 - ◆ デフォルトは `false`。
- `maxPersistentQueueSizeInMegs=1000`
 - ◆ 持続キューがイベントを保存するディスクファイルの最大サイズをメガバイトで指定します。
 - ◆ デフォルトは 1000。
- `dropEventsOnQueueFull=10`
 - ◆ キューにスペースができるまで、すべての新規イベントを捨てる前に $N * 5$ 秒待ちます。
 - ◆ これを-1 または 0 に設定すると、キューがいっぱいになるとブロックします。これにより、プロセスチェーンを遮断します。
 - ◆ ターゲットグループキューにブロックされたものがあると、データはそれ以上他のターゲットグループに到達しなくなります。
 - ◆ ロードバランシンググループの利用は、この状態を軽減するための最良の方法です。これは、複数の受信ホストがダウンしない限り、キューブロッキングが生じないためです。
 - ◆ デフォルトは-1(イベントをドロップしない)。

ターゲットグループについてのバックオフ設定

あるターゲットグループ内のインデックスマシンがアクセス不能になった場合、フォワーダが再接続を試みるように設定することができます。接続を再度試みる必要がある場合は、フォワーダは `backoffAtStartup` または `initialBackoff` を使用して、秒単位で待ちます。この時間を越えると、`maxBackoff` に到達するまで、フォワーダはその秒数を 2 倍することを繰り返します。その時間に到達すると、フォワーダは、再試行の間に、その秒数を 2 倍することを停止し、同じ `maxBackoff` 秒を使用します。`maxNumberOfRetriesAtHighestBackoff` の頻度で試みを繰り返すか、または、その値が-1 の場合は永久に試みを繰り返します。

- `backoffAtStartup=N`
 - ◆ 必要な最初の再試行までの待ち時間を秒数で定義します。
 - ◆ デフォルトは 5 秒。
- `initialBackoff=N`
 - ◆ 必要な最初の再試行以外の、毎回の再試行までの待ち時間を秒数で定義します。
 - ◆ デフォルトは 2 秒。
- `maxBackoff=N`
 - ◆ 最大バックオフ頻度に到達するまでの秒数を指定します。

- ◆ デフォルトは 20。
- `maxNumberOfRetriesAtHighestBackoff=N`
 - ◆ 最大バックオフ期間に到達した後、完全に停止する前に、システムが再試行する回数を指定します。
 - ◆ -1 は永久に再試行します。
 - ◆ これは、デフォルトから変更しないことを推奨します。変更すると、フォワーダは、ダウンした URI への転送をある時点で完全に停止します。
 - ◆ デフォルトは-1(永久)。

ラウンドロビン・データバランシングの設定

ラウンドロビン・データバランシングの設定

コンテンツに基づいて異なる場所にデータをルーティング

コンテンツに基づいて異なる場所にデータをルーティング

1 つの Splunk インスタンスから他のインスタンスへ、コンテンツに基づいて、データフォワードのルーティングを有効にします。例えば、ソースタイプ、インデックスタイムで抽出したフィールド、またはローイベントの内容に基づいて、データとシステムにルーティングすることができます。特にルーティングにより、イベントをシステムに分散することが可能になります。

ルーティングの設定

ルーティングを設定するには次のようにします：

- まず、どのイベントをどのサーバーにルーティングするかを決定します。
- 次に、フォワーディングサーバーにある `props.conf`、`transforms.conf`、および `outputs.conf` を編集します。

`props.conf` の編集

`$SPLUNK_HOME/etc/system/local/props.conf` を編集し、`TRANSFORMS-routing=` の属性を設定します。

[<spec>]

`TRANSFORMS-routing=$UNIQUE_STANZA_NAME`

<spec> には、以下が指定できます。

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

`transforms.conf` でエントリーを作成する際は、`$UNIQUE_STANZA_NAME` を使用します(後述)。

`transforms.conf` の編集

`$SPLUNK_HOME/etc/system/local/transforms.conf` を編集し、`props.conf` スタンザを一致させるルールを設定します。

```
[ $UNIQUE_STANZA_NAME ]
REGEX=$YOUR_REGEX
DEST_KEY=_TCP_ROUTING
FORMAT=$UNIQUE_GROUP_NAME
```

- \$UNIQUE_STANZA_NAME は、props.conf で作成した名前と一致する必要があります。
- \$YOUR_REGEX で正規表現のルールを入力し、条件でルーティングするイベントを決定します。
- DEST_KEY は、イベントを TCP 経由で送信するために、_TCP_ROUTING に設定する必要があります。
- FORMAT を \$UNIQUE_GROUP_NAME に設定します。これは、outputs.conf で作成したグループ名に一致する必要があります。

outputs.conf の編集

\$SPLUNK_HOME/etc/system/local/outputs.conf を編集し、各サーバーまたはグループに割り当てる tcpout 出力を設定します。

```
[ tcpout:$UNIQUE_GROUP_NAME ]
server=$IP:$PORT
```

- transforms.conf で作成した名前と一致するように、\$UNIQUE_GROUP_NAME を設定します。
- 受信サーバーと一致するように IP アドレスとポートを設定します。

基本的なルーティングの例

次の例では、sourcetype="syslog"ですべてのイベントを1つのターゲットグループに送信すると共に、error という語を含むイベントのすべてを他のターゲットグループに、その他すべてのイベントを第三のターゲットグループに送信します。

1. \$SPLUNK_HOME/etc/system/local/props.conf を編集して、TRANSFORMS-routing= の属性を設定します。

```
[default]
TRANSFORMS-routing=errorRouting
[syslog]
TRANSFORMS-routing=syslogRouting
```

2. \$SPLUNK_HOME/etc/system/local/transforms.conf を編集して、errorRouting と syslogRouting のルールを設定します：

```
[errorRouting]
REGEX=error
DEST_KEY=_TCP_ROUTING
FORMAT=errorGroup
[syslogRouting]
REGEX=.
DEST_KEY=_TCP_ROUTING
FORMAT=syslogGroup
```

3. \$SPLUNK_HOME/etc/system/local/outputs.conf を編集して、各サーバーまたはグループに割り当てる tcpout 出力を設

定めます。

```
[tcpout]
defaultGroup=everythingElseGroup
[tcpout:syslogGroup]
server=10.1.1.197:9997
[tcpout:errorGroup]
server=10.1.1.200:9999
[tcpout:everythingElseGroup]
server=10.1.1.250:6666
```

高度な例

この例は、ルーティング、データバランシング、およびターゲットグループ別パラメータを組み合わせます。この outputs.conf では、sourcetype="syslog"ですべてのイベントを1つのバランシングされたターゲットグループに送信すると共に、error という語を含むイベントのすべてを異なるターゲットグループに送信し、その他のすべてを2つのターゲットグループにクローニングします。syslogGroup は/tmp ディレクトリにある persistent queue を使用し、最大ディスクサイズ 100MB に制限されます。ターゲットグループのハートビート頻度はすべて 10 秒に調整されます。

注意： ステップ 1 と 2 は、props.conf と transforms.conf は上記の例と同様です。

3. \$SPLUNK_HOME/etc/system/local/outputs.conf を編集して、各サーバーまたはグループに割り当てる tcpout 出力を設定します。

```
[tcpout]
defaultGroup=everythingElseGroup1, everythingElseGroup2
heartbeatFrequency=10
[tcpout:syslogGroup]
server=10.1.1.197:9997, 10.1.1.198:7777
usePersistentQueue=true
blockOnQueueFull=true
persistentQueuePath=/tmp
maxPersistentQueueSizeInMegs=100
[tcpout:errorGroup]
server=10.1.1.200:9999
[tcpout:everythingElseGroup1]
server=10.1.1.240:6666
[tcpout:everythingElseGroup2]
server=10.1.1.245:5555
```

イベントの特定キューへのルーティング

イベントの特定キューへのルーティング

データの転送および受信について Splunk を設定する際、特定のデータを異なるキューに送信して、さらに処理を行うことができます。このトピックでは、データのフィルタリングの方法と、それを nullQueue または Splunk の /dev/null ディレクトリへ送信する方法について説明します。

データをインデックス化する前にイベントをフィルタリング(およびその保存)するには、以下の説明に従ってそれらのイベントを nullQueue に送信します。

特定のイベントをデフォルトでないインデックスに送信したい場合は、特定のインデックスにイベントをフィルタリングおよびルーティングする方法を参照してください。

重要：データのフィルタリングの選択時期は、分散セットアップにより異なります。ただし、フィルタリングは、データをパースする Splunk インスタンスにおいて実行する必要があります。つまり、インデックスマシンまたはフォワーダインスタンスのどちらかで実行することができます。

特定のキューへのイベントをルーティングするための設定

特定のイベントをフィルタリングするには次のようにします。

1. イベントを他から分類するためのイベントの属性を特定します。
2. ソース、ソースタイプまたはホストの props.conf でエントリを作成し、TRANSFORMS クラスと TRANSFORMS 名を指定します。クラス名は、transforms.conf が置かれる正規表現スタanzasを参照します。
3. 特定した属性に一致する正規表現で transforms.conf にエントリを作成し(ステップ 1 から)、キューの対象となる DEST_KEY と特定キューの対象となる FORMAT キーを設定します。

\$SPLUNK_HOME/etc/system/README/props.conf.example および ../transforms.conf.example を例として使用する、または自作の props.conf と transforms.conf を作成します。 \$SPLUNK_HOME/etc/system/local/または \$SPLUNK_HOME/etc/apps/にある自分のカスタムアプリケーションディレクトリで変更します。設定ファイル一般に関する詳細は、「設定ファイルの動作」を参照してください。

props.conf の編集

\$SPLUNK_HOME/etc/system/local/props.conf で、次のスタanzasを追加します。

```
[<spec>]
```

```
TRANSFORMS-$name=$UNIQUE_STANZA_NAME
```

<spec> には、以下が指定できます。

- <sourcetype>、イベントのソースタイプ
- host::<host>、<host>はイベントに対するホスト
- source::<source>、<source>はイベントに対するソース

\$NAME は、transform に付ける個別の識別子です。

\$UNIQUE_STANZA_NAME は、transforms.conf で作成した transform のスタンザ名に一致する必要があります。

transforms.conf の編集

\$SPLUNK_HOME/etc/system/local/transforms.conf で、次のスタンザを追加します。

```
[$UNIQUE_STANZA_NAME]
REGEX = $YOUR_CUSTOM_REGEX
DEST_KEY = queue
FORMAT = nullQueue
```

props.conf で指定した名前に一致するように、\$UNIQUE_STANZA_NAME でスタンザに名前をつけます。特定した属性に基づいて\$YOUR_CUSTOM_REGEXを追加します。これは、削除したいイベントと特定するキーとなる語を指定する必要があります。

DEST_KEY および FORMAT を上記の値のままにし、特定されたイベントを nullQueue に送信します(例えば、インデキシングの前にそれらを削除)。

一致したイベントの nullQueue への送信

この例は、すべての sshd イベントを/var/log/messages から nullQueue へ送信します。

```
props.conf 内：
[source::/var/log/messages]
TRANSFORMS-null= setnull

transforms.conf 内：
[setnull]
REGEX = ¥[sshd¥]
DEST_KEY = queue
FORMAT = nullQueue
```

一致した WMI イベントの nullQueue への送信

WMI を使用して Windows からイベントをキャプチャする場合、構文はソース上の props.conf で特定のものになります。この例により、2つの異なるイベントコード(592 or 593)を、正規表現の"or"ステートメントを使用してフィルタリングすることが可能になります。

```
props.conf 内：
[wmi]
TRANSFORMS-foo=wminull

transforms.conf 内：
[wminull]
REGEX=(?m)^EventCode=(592|593)
DEST_KEY=queue
```

```
FORMAT=nullQueue
```

一致したイベントを `indexQueue` に送信し、その他のすべてを `nullQueue` に送信する

この例は、前の例の逆です。 `/var/log/messages` からの `sshd` イベントだけを保持し、その他のすべてを `nullQueue` に送信します。この場合、2 つの `transform` を定義する必要があります。

`props.conf` 内：

```
[source:/var/log/messages]
```

```
TRANSFORMS-set= setnull,setparsing
```

`transforms.conf` 内

```
[setnull]
```

```
REGEX = .
```

```
DEST_KEY = queue
```

```
FORMAT = nullQueue
```

```
[setparsing]
```

```
REGEX = ¥[sshd¥]
```

```
DEST_KEY = queue
```

```
FORMAT = indexQueue
```

イベントを特定のインデックスにルーティング

イベントを特定のインデックスにルーティング

デフォルトでは、Splunk はすべてのイベントを `main` と呼ばれるインデックスに送ります。ただし、特定のイベントを他のインデックスに送る必要がある場合があります。例えば、データをセグメント化したり、雑多なデータを含むソースからのイベントデータを、受信専用のインデックスに送りたい場合があります。データをローカルでルーティングしたり、リモートソースまたは Splunk インスタンスから受信したデータをルーティングしたりすることができます。

注意：データを別のインデックスに置き、そのインデックスを検索する場合は、`index=`コマンドで検索のインデックスを指定する必要があります。

```
index=foo
```

データ入力からのイベントのすべてを特定のインデックスに送る

特定のデータ入力からのイベントのすべてを別のインデックスにルーティングするための設定には、次の記述を `inputs.conf` にある適当なスタンザに追加します。

```
index = myindex
```

例

次の例の `inputs.conf` エントリーはデータを `index = fflanda` にルーティングします。

```
[monitor:///var/log]
disabled = false
index = fflanda
```

フォワーダで異なるインデックスを指定する場合、そのイベントがインデキシングインスタンスに達すると、それらは名付けられたインデックスにルーティングされます。ただし、そのインデックスはすでに存在している必要があります。

特定のイベントを異なるインデックスにルーティングする

あるイベントを別のインデックスにルーティングするための設定には、ローカル Splunk インスタンスにある `props.conf` と `transforms.conf` を編集します。

1. イベントを他から分類するための、イベントの属性を特定します。
2. ソース、ソースタイプまたはホストの `props.conf` でエントリーを作成し、`TRANSFORMS` クラスと `TRANSFORMS` 名を指定します。クラス名は、`transforms.conf` が置かれる正規表現スタanzaを参照します。

この例では、`TRANSFORMS` クラス名がインデックスで、`TRANSFORMS` 名が `AppRedirect` となります。

3. 特定された属性に一致するようにエントリーを `transforms.conf` 内に正規表現で作成し(ステップ 1 から)、別のインデックス名(この例では `Verbose`)を `FORMAT` キーに書き出し、`DEST_KEY` を設定してインデックス属性である `_MetaData:Index` を指定します。

`props.conf` の編集

次のスタanzaを `$SPLUNK_HOME/etc/system/local/props.conf` に追加します。

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

<spec> には、以下が指定できます。

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

`$NAME` は、`transform` に付ける固有の識別子です。

`transforms.conf` の編集

`$SPLUNK_HOME/etc/system/local/transforms.conf` に次のスタanzaを追加します。

```
[$UNIQUE_STANZA_NAME]
REGEX = $YOUR_CUSTOM_REGEX
DEST_KEY = _MetaData:Index
FORMAT = Verbose
```

`props.conf` で指定した名前に一致するように、`$UNIQUE_STANZA_NAME` でスタanzaに名前を付けます。

`$YOUR_CUSTOM_REGEX` を、指定した属性に基づいて追加します。

例

属性を特定します。

```
web1.example.com MSWinEventLog 1 Application 721 Wed Sep 06 17:05:31 2006
4156 MSDTC Unknown User N/A Information WEB1 Printers String
message: Session idle timeout over, tearing down the session. 179
web1.example.com MSWinEventLog 1 Security 722 Wed Sep 06 17:59:08 2006
576 Security SYSTEM User Success Audit WEB1 Privilege Use
Special privileges assigned to new logon: User Name: Domain: Logon
ID: (0x0,0x4F3C5880) Assigned: SeBackupPrivilege SeRestorePrivilege
SeDebugPrivilege SeChangeNotifyPrivilege SeAssignPrimaryTokenPrivilege 525
```

この例では、Application フィールドをトリガとして使用します。ソースタイプ {windows_snare_log} からのイベントの 'Application' に一致すると、transforms スタンザの AppRedirect で値が割り当てられます。1 つの割り当ては、インデックス名 Verbose です。

例： props.conf の編集

次のスタンザを \$SPLUNK_HOME/etc/system/local/props.conf に追加します。

```
[windows_snare_syslog]
TRANSFORMS-index = AppRedirect
```

例： transforms.conf の編集

\$SPLUNK_HOME/etc/system/local/transforms.conf に次のスタンザを追加します。

```
[AppRedirect]
REGEX = Application
DEST_KEY = _MetaData:Index
FORMAT = Verbose
```

データをサードパーティシステムにルーティング

データをサードパーティシステムにルーティング

Splunk は、データを Splunk でないシステムにルーティングするように設定できます。それには、Splunk サーバーを設定し、TCP 上のローデータを outputs.conf を介してサーバーとポートに送信するようにします。受信サーバーは、そのポートでデータストリームを受信します。

また、他社システムを経由するデータを具体的に指定して props.conf と transforms.conf で条件付きルーティングができます。

設定

データルーティングを設定するには、`props.conf`、`transforms.conf`、および `outputs.conf` を編集する必要があります。

これらのファイルは、Splunk サーバーの `$SPLUNK_HOME/etc/system/local/` にあります。

注意: これらのファイルが `$SPLUNK_HOME/etc/system/local/` にない場合は、それを `$SPLUNK_HOME/etc/system/default/` からコピーします。

`props.conf` で、ホスト、ソース、またはデータストリームのソースタイプを指定します。入力を実行するために `transform` を指定します。

`transforms.conf` で、`transforms` を定義し、`TCP_ROUTING` を指定して適用します。また、入力についてより詳細に選択したい場合は、`REGEX` を使用することもできます。

`outputs.conf` で次のようにします。

- データを受信するターゲットグループを定義します。
- IP アドレスと TCP ポート `$IP:$PORT` を指定します。これは、他社システムでデータを受信するためです。
- `sendCookedData` を `false` に設定し、Splunk サーバーがローデータを転送できるようにします。

注意: ターゲットグループまたはデフォルトグループの一部として 1 つのサーバーをリストしてデータを送信します。`outputs.conf` におけるターゲットグループの設定を参照してください。

例

データサブセットの送信

この例は、データのサブセットを Splunk から転送する方法を示します。

まず、`props.conf` と `transforms.conf` を編集し、Splunk 以外のシステムに送信するデータを指定します。

1. `props.conf` で、`bigmoney transform` を `nyc` で始まるすべてのホストネームに適用します。

```
[host::nyc*]
```

```
TRANSFORMS-nyc = bigmoney
```

`transforms.conf` で、TCP ルーティングを、デフォルト TCP グループと Splunk 以外のサーバーグループがあるグループに設定します。

```
[bigmoney]
```

```
DEST_KEY=_TCP_ROUTING
```

```
FORMAT=bigmoneyreader
```

2. 次に、`outputs.conf` のターゲットグループを定義します。

```
[tcpout]
```

```
defaultGroup = default-clone-group-192_168_1_104_9997
```

```
[tcpout:default-clone-group-192_168_1_104_9997]
disabled = false
server = 192.168.1.104:9997
[tcpout:bigmoneyreader]
disabled = false
server=10.1.1.197:7999
sendCookedData=false
```

全データの送信

この例は、すべてのデータを Splunk から転送する方法を示します。

すべてのデータを送信するため、`outputs.conf` で、すべてのデータを Splunk 以外のシステムに送信するよう指定するだけです。

```
[tcpout]
defaultGroup = fastlane
disabled = false
indexAndForward = true
[tcpout:fastlane]
disabled = false
server = 10.1.1.35:6996
sendCookedData = false
```

クローニングされたデータを複数の受信ホストに転送

クローニングされたデータを複数の受信ホストに転送

クローニングが有効な場合、Splunk フォワーダはそのデータを複数の他の Splunk インスタンスに送信します。

重要：これは、全く同一のインデックスが複数作成されることを保証するものではありません。受信ホストの1つが利用できない場合、データは利用できる受信ホストにのみ送信されます。その結果、同一でないインデックスが作成されることがあります。

フォワーディングサーバーにある Splunk 管理または `outputs.conf` でクローニングを設定します。フォワーダがすべてのデータを送信する先の受信サーバーのターゲットグループを設定します。

フォワーディングサーバー上で、次を `$$SPLUNK_HOME/etc/system/local/outputs.conf` に追加します。

```
[tcpout]
defaultGroup = indexer1, indexer2
heartbeatFrequency=10
maxQueueSize=10000
[tcpout:indexer1]
```

```
server=10.1.1.197:9997
```

```
[tcpout:indexer2]
```

```
server=10.1.1.200:9999
```

この設定により、すべてのイベントが、10.1.1.197:9997 および 10.1.1.200:9999 の両方に送信されます。クローニングしたデータを送信する先のすべてのサーバで受信ができることを確認してください。

syslog または HTTP フォーマットでの転送

syslog または HTTP フォーマットでの転送

このトピックでは、標準 syslog または HTTP フォーマットでデータを送信するための Splunk フォワーダの設定方法について説明します。

このように設定すると、Splunk はデータを別の出力プロセッサを通じて送信します。Splunk は RFC 3164 準拠のイベントを 1 つのプラットフォームから TCP/UDP ベースのサーバおよびポートに転送します。これにより、それに準拠していないペイロードを RFC 3164 準拠にすることができます。次を指定することができます：

- TCP 優先順位(facility と severity の組み合わせ)
- 正規表現の指定および正規表現に一致するデータのみを転送するオプション
- ソースタイプにより送信されるデータその他のメタデータのフィルタリング
- データを 1024 に強制短縮(RFC 3164 に準拠させるため)

これを設定するには、まず、Splunk フォワーダが syslog または HTTP フォーマットデータを送信する先のシステムを特定し、それらをフォワーダの \$SPLUNK_HOME/etc/system/local/outputs.conf で定義するターゲットグループに追加する必要があります。

注意： syslog データで複数イベントタイプを定義すると、そのイベントタイプはすべて 'syslog' という文字を含んでいる必要があります。そうでないと、正しく動作しません。

```
#####
```

```
#----Syslog 出力---
```

```
#####
```

```
# 次の設定は syslog は syslog を使用して出力を送信するために使います。
```

```
[syslog:$TARGET_GROUP]
```

```
attribute1 = val1
```

```
attribute2 = val2
```

```
...
```

```
#----必要な設定----
```

```
# syslog 出力に必要な設定：
```

```
server = ip/servername:<port>
```

```
syslog サーバーが実行されている IP またはサーバ名
```

```
syslog サーバーがリスニングしているポート
```

```
デフォルト値はなし。ポートを指定する必要があります。syslog はデフォルトで 514 を使用。
```

```

#----オプション設定----
#       syslog 出力に必要なオプション設定 :
type = tcp | udp
*       使用プロトコル。 タイプが指定されない場合、デフォルトは udp。
priority = <ddd>
*       ddd は、syslog ヘッダで<ddd>として表示される値。
*       ユーザーは ddd を (<facility> * 8) + <severity>で計算すること
*       facility が 4(セキュリティ/認証メッセージ)、severity が 2(Critical:危険状態)、優先値は 34 = (4 * 8) + 2
となります。
*       TODO: default = ?
syslogSourceType = <string>
*       string は syslog のソースタイプを示す
*       この属性がない場合、 'sourcetype::syslog' は syslog メッセージのソースタイプとなります。
timestampformat = <%b %e %H:%M:%S>
*       これが指定されると、そのフォーマットは、タイムスタンプをヘッダに追加する際に使用されます。
*       TODO: default = ?
#####
#----HTTP 出力----
#####
#       次の設定は出力を HTTP を通じて送信するために使用します。
[httpoutput:$TARGET_GROUP]
attribute1 = val1
attribute2 = val2
...
#----必要な設定----
#       HTTP 出力に必要な設定 :
username = <ユーザー名>
*       ユーザー名は Splunk インデックスマシンに対する認証で使用されます。
password = <パスワード>
*       パスワードは Splunk インデックスマシンに対する認証で使用されます。
server = ip/servername:port
*       Splunk 受信ホストの ip/servername
*       port は Splunk 受信ホストがリスニングするポート
#----オプション設定----
#       HTTP 出力に必要なオプション設定 :
ssl = true | false
*       HTTP 出力のための SSL を設定します。
*       デフォルトは true。

```

他の Splunk インスタンスへの展開

デプロイメントサーバーについて

デプロイメントサーバーについて

このセクションでは、Splunk デプロイメントサーバーとその機能について説明します。個別の組織における Splunk デプロイメントの別の設計方法に関する概要は、コミュニティ Wiki のデプロイメント情報を参照してください。

デプロイメントサーバーは、集中設定マネジャーとして機能する Splunk インスタンスです。これは、1つのグループにまとめ、Splunk インスタンスを一括して管理します。Splunk インスタンスは、1つのインデキシングデータがローカルである場合でも、デプロイメントサーバーとして機能します。デプロイメントサーバーがリモートで設定した Splunk インスタンスはデプロイメントクライアントと呼ばれます。Splunk インスタンスは、デプロイメントサーバーとクライアントに同時にすることができます。

1つまたは複数の Splunk デプロイメントクライアントの設定を変更するには、指定のサーバークラスの更新済設定情報を、デプロイメントサーバーから、そのサーバークラスが属するデプロイメントクライアントに送信します。

サーバークラスは、定義済の Splunk デプロイメントクライアント設定です。クライアント設定を管理するには、Splunk デプロイメントクライアントを1つまたは複数のサーバークラスに割り当てます。そうすると、指定のサーバークラスに属するすべての Splunk デプロイメントクライアントを1つのユニットとして管理することができます。デプロイメントクライアントは、同時に複数のサーバークラスのメンバーになることができます。インデックス化するグループクライアントを、アプリケーション、OS、データタイプでグループ化することができ、また、Splunk デプロイメントの他の機能でグループ化することができます。

デプロイメントサーバーとクライアントとの間のコミュニケーション

デプロイメントクライアントは定期的に、デプロイメントサーバーをポーリングし、自身を特定します。デプロイメントサーバーは次にその設定にある情報をレビューし、その特定のクライアントに送信すべき新しい又は更新された情報がないかどうかをチェックします。指定のデプロイメントクライアントに展開する新しいコンテンツがある場合、デプロイメントサーバーはクライアントに対し、どれを取得すべきかを伝え、そのクライアントに、クライアントに提供されている REST エンドポイントを使うように伝えます。デプロイメントクライアントはその新しいコンテンツを取得し、所属するサーバークラスに特定の指示に従ってそれを処理します。例えば、再起動、スクリプト実行、またはその他の命令を待ちます。

デプロイメントの計画

デプロイメントの計画

Splunk インスタンスが多くの人に利用される場合、その設定は、その用途より異なります。例えば、ヘルプデスクサービスのための Splunk インスタンスで、特定のアプリケーションをインストールし、Windows デスクトップの問題についてのトラブルシューティングを支援する場合があります。あるいは、運用スタッフが使用する Splunk インスタンスの

グループで、複数の異なるアプリケーションを設定し、ネットワーク問題、セキュリティ、Eメールトラフィック管理を行う場合があるかもしれません。Splunk インスタンスの 3 つ目のグループは、運用チーム内のウェブホスティンググループを支援する、などです。

これらさまざまな Splunk インスタンスを 1 つずつ管理する必要はなく、その利用方法によってそれらをグループに分け、各グループで必要なさまざまな設定とアプリケーションを特定し、デプロイメントサーバーを使用して、必要に応じてそれらさまざまなアプリケーションと設定の更新を管理することができます。

ここでの例は、デプロイメントサーバーを使用する際により容易な管理ができるように、Splunk インスタンスをグループ分けする 1 つの例を示すものです。Splunk インスタンスを、単に OS、ハードウェアタイプ、バージョン、地理的位置またはタイムゾーンでグループ分けすることもできます。デプロイメントサーバーは、さまざまなトポロジーをサポートする柔軟性を持っています。

基本的なデプロイメントサーバートポロジー

最も基本的なポロジでは、1 つのデプロイメントサーバーと 1 つ以上のデプロイメントクライアントがあります。クライアントはそのデプロイメントサーバーをポーリングし、コンテンツをそれらクライアントに提供します。

階層デプロイメントサーバートポロジー

1 つのデプロイメントサーバーが複数のクライアントを持つ形態に加えて、そのデプロイメントサーバー自身がデプロイメントクライアントになるもなることができます。

マルチテナント・デプロイメントサーバートポロジー

このポロジでは、1 つの Splunk インスタンスが複数のデプロイメントサーバーをホストし、そのそれぞれが自身のデプロイメントクライアントを持ちます。

サーバークラスの定義

サーバークラスの定義

1 つのサーバークラスで、デプロイメントサーバーが 1 つまたは複数のデプロイメントクライアントに展開する 1 セットのコンテンツを定義します。このコンテンツは、アプリケーション、システム設定、およびスクリプト、画像、サポートリングマテリアルなどその他の関連コンテンツから構成することができます。

(デプロイメントクライアントは、自身の設定を持ち、`deploymentclient.conf` で定義されます。

`deploymentclient.conf` はそのデプロイメントクライアントに、それが属するサーバークラスが与える、そのコンテンツをどこで取得するかの情報を与えます。)

異なるサーバークラスを定義し、異なる要件、OS およびデプロイメントクライアントの目的をを反映させることができます。

デプロイメントサーバーにある `serverclass.conf` でサーバークラスを定義します。デフォルトでは `serverclass.conf` ファイルはありません。したがって、`$SPLUNK_HOME/etc/system/local` にそれを作成し、その中でサーバークラスを定義しま

す。

複数のサーバークラスがある場合、デフォルトですべてのデプロイメントクライアントに適用するグローバルまたは一般サーバークラスインスタンスを定義し、より個別でより特定のサーバークラスで必要に応じてさまざまなアスペクトを上書きします。例えば、一般的にそれを指定、つまりすべてのデプロイメントクライアントがそれらのコンテンツを一箇所から取得し、なおかつ、それらの1つのサブセットが異なる場所から取得するように指定することができます。

重要: すべての設定情報は、設定ファイルでその数値が評価され、次に文字が評価されます(まず 0-9、次に a-z)。従って、その命名が重要です。

サーバークラスで定義できるもの

グローバルサーバークラスおよび個別サーバークラスは、ここに記述される設定を指定することができます。グローバルサーバークラスを定義するには、[global]と呼ばれるスタanzasを作成します。より具体的にサーバークラスを定義するには、サーバークラス別に命名されたスタanzas[serverClass:<serverClassName>]を作成します。ここで、serverClassName は、サーバークラスに与える名です。

注意: 所与の設定ファイルで利用できる最も正確かつ最新の設定ファイルは、各設定ファイルの.spec ファイルにあります。.spec ファイルおよび.example ファイルの最新バージョンは、本書の設定ファイルレファレンスまたは \$SPLUNK_HOME/etc/system/README を参照してください。

repositoryLocation

これは、このサーバークラスでコンテンツが展開されるデプロイメントサーバー上の場所であり、またデフォルトでは、クライアントが展開される際に終了する場所でもあります。deploymentclient.conf でこれを上書きすることができます。デフォルトでは、これは \$SPLUNK_HOME/etc/apps です。

continueMatching

これを false に設定すると、デプロイメントサーバーは、この設定ファイルにあるサーバークラスのリストを読み、最初の設定と一致すると停止します。これを true に設定すると、マッチングを継続します。このオプションは、複数の階層化されたサーバークラスのセットを定義するために提供されています。デフォルト値は true です。

endpoint

これは、デプロイメントサーバーがデプロイメントクライアントに伝える、コンテンツ取得を開始する HTTP 位置です。デプロイメントサーバーは、自身を変数置換で書き込みます。同じ変数を使用する限り、ここで URI を入力することができます。デフォルト値は

\$deploymentServerUri\$/services/streams/deployment?name=\$serverClassName\$: \$appName\$ です。

filterType

これは、デプロイメントサーバーに対し、グローバルサーバークラス定義における設定情報の層のフィルタリング方法を指定します。これは、個別定義されたサーバークラスの情報とは別に行います。評価項目をホワイトリストまたはブラックリストのエントリーとして選択することができます。ブラックリストを選択した場合、例えば、デプロイメントサーバ

ーは、ブラックリストエントリーフィルタで指定しないものに一致します。

この値はサーバークラスレベルで上書き可能であることに注意してください。したがって、グローバルレベルでホワイトリストを使用し、個別サーバークラスでブラックリストを使用すると、設定が完全に上書きされます。そのサーバークラス定義で別のフィルタを提供し、上書きしたものを置き換える必要があります。

重要：すべての設定情報は、設定ファイルで先頭から最後に向かって評価されます。したがって、順番が重要です。

例： `filterPreference` がホワイトリストの場合：

```
whitelist.0=*.fflanda.com
blacklist.0=printer.fflanda.com
blacklist.1=scanner.fflanda.com
```

これによって、'printer'と'scanner'を除いて、fflanda.com にあるすべてのホストがこのサーバークラスと一致します。

`filterPreference` がブラックリストの場合：

```
blacklist.0=*
whitelist.0=*.web.fflanda.com
whitelist.1=*.linux.fflanda.com
```

これにより、'web'および'linux'ホストのみがサーバークラスとマッチします。他のホストは一致しません。

デフォルト値はホワイトリストです。

startBuild and endBuild

この設定により、Splunk ビルド番号の範囲を指定し、指定のサーバークラスがその範囲でのみビルドを許可するかどうかを制限することができます。

enabledOnClient

この設定により、app を受信するデプロイメントクライアントが、インストールされたアプリケーションを有効にするか無効にするかを指定することができます。

machineTypes

この設定により、デプロイメントクライアントのハードウェアタイプをフィルタとして使用できるようになります。このフィルタは、クライアントがホワイトリスト/ブラックリストフィルタを使用して一致できない場合にのみ使用されます。

ここでの値は、ハードウェアプラットフォームが指定した特定の文字列です。クライアント上でこの文字列を見つける方法は、プラットフォームにより異なります。しかし、デプロイメントクライアントがすでにデプロイメントサーバーに接続されている場合、デプロイメントサーバー上のこの Splunk CLI コマンドを使用して、この文字列が何であるかを決定することができます。 `./splunk list deploy-clients`

これは、`machineTypes` を指定するのに使うことができる `utsname` の値を返します。

この設定は、カンマで区切られたリストでマシンタイプと一致します。一般的に使用されるマシンタイプは、linux-x86_64、

windows-intel、linux-i686、freebsd-i386、darwin-i386、sunos-sun4u、linux-x86_64、sunos-i86pc、freebsd-amd64
です。

restartSplunkWeb

<True または False>

restartSplunkd

<True または False>

デプロイメントクライアントの設定

サーバークラスの定義

このトピックでは、Splunk デプロイメントサーバー機能を使用する場合の、デプロイメントクライアント設定用のオプションについて説明します。デプロイメントクライアントは、1 つまたは複数のサーバークラスに属します。

- サーバークラスは、指定のデプロイメントクライアントでダウンロードする必要のあるコンテンツを定義します。
- デプロイメントクライアントは `deploymentclient.conf` の自身のコピーにある情報を使用し、どのデプロイメントサーバーにコンタクトしてサーバークラスが命じるコンテンツを取得するのか、およびダウンロード後にどこに置くかを検索します。

デプロイメントクライアントで定義できるもの

Splunk インスタンスをデプロイメントクライアントとして有効にするには、`$SPLUNK_HOME/etc/system/local` に `deploymentclient.conf` を作成し、`[deployment-client]` および `[target broker]` スタンザをそこに追加します。次の設定を使用して、デプロイメントクライアントが、それが属するサーバークラスが指定するコンテンツのどこから取得するかを定義します。

注意： 指定の設定ファイルで利用できる最も正確かつ最新の設定ファイルは、各設定ファイルの `.spec` ファイルにあります。`.spec` ファイルおよび `.example` ファイルの最新バージョンは、本書の設定ファイルリファレンスまたは `$SPLUNK_HOME/etc/system/README` を参照してください。

disabled

<false または true>

- デフォルトは `false`。
- デプロイメントクライアントを有効/無効にします。

clientName

`deploymentClient`

- また、`'tag'`ともいう。デフォルトは `'deploymentClient'`。
- デプロイメントサーバーがフィルタリングに利用することができる名前であり、`hostnames` に優先します。

workingDir

`$(SPLUNK_HOME)/var/run/deploy-client`

- `deploymentClient` が使用する一時フォルダであり、サーバークラスとアプリケーションをダウンロードします。

repositoryLocation

`$(SPLUNK_HOME)/etc/apps`

- アプリケーションがデプロイメントサーバーからダウンロードされた後にインストールされる場所。
- ダウンロードされる各アプリケーションについて、デプロイメントサーバーが、それをインストールするための `repositoryLocation` を指定する場合があります。注意してください。
- デプロイメントクライアントは、以下に定義する '`serverRepositoryLocationPolicy`' を使用し、どの場所を使用するかを決定します。

serverRepositoryLocationPolicy

<`acceptSplunkHome`、`acceptAlways`、`rejectAlways` のいずれか>

- デフォルトは `acceptSplunkHome`。
- `acceptSplunkHome` - `deploymentServer` が供給した `repositoryLocation` を受け入れます。ただし、`$(SPLUNK_HOME)` によりルーティングされた場合のみ。
- `acceptAlways` - `deploymentServer` が供給した `repositoryLocation` を常に受け入れます。
- `rejectAlways` - 供給されたプロパティ拒否し、この設定で指定された `repositoryLocation` のみを使用します。

endpoint

`$(deploymentServerUri)/services/streams/deployment?name=$(serviceClassName):$(appName)`

- サーバークラス(アプリケーション)がダウンロードされる元の HTTP エンドポイント。`deploymentClient` により決定されると、いくつかのサービスクラスをリフレッシュする必要があります。
- ダウンロードする必要のある各アプリケーションについて、`deploymentServer` はダウンロードのエンドポイントを指定する場合があります。注意してください。
- `deploymentClient` は、下記で定義する '`serverEndpointPolicy`' を使用して、そのプロパティを使用するかを決定します。
- `$(deploymentServerUri)` は、下記の '`target-broker`' スタンザで定義される '`targetUri`' プロパティを決定します。
- `$(serviceClassName)` と `$(appName)` は、それぞれが何を提供するかを意味します。

serverEndpointPolicy

<`acceptAlways`、`rejectAlways` のいずれか>

- デフォルトは `acceptAlways`。
- `acceptAlways` - サーバーが提供するエンドポイントを常に受け入れます。
- `rejectAlways` - サーバーが提供するエンドポイントを拒絶します。常に上記の '`endpoint`' 定義を使用します。

phoneHomeIntervalInSecs

<N>

- デフォルトは 30。

maxRetries

<N>

- アプリケーションのダウンロードの再試行回数。

このクライアントが使用するデプロイメントサーバーを指定する

[target-broker:deploymentServer]スタanzasを deploymentClient.conf に追加し、このデプロイメントクライアントが targetUri と使用するデプロイメントサーバーおよびポートを指定します。

targetUri

<deploymentServer>:<mgmtPort>

- デプロイメントサーバーの URI。

マルチテナント環境への展開

マルチテナント環境への展開

マルチテナントデプロイメントサーバポートポリシーとは、同じ Splunk インスタンス上で実行しているデプロイメントサーバーが複数あり、各デプロイメントサーバーが、デプロイメントクライアントの自身のセットに対してコンテンツを供給していることを意味します。(また、2 つの Splunk インスタンス使用し、それぞれが自身の設定を有する場合に同じ効果を達成することができます。)

1 つの Splunk インスタンスで複数のデプロイメントサーバーを設定するには、次のようにします。

- ホワイトリストまたはブラックリストを含む tenants.conf を作成します。それらは、使用するデプロイメントサーバーインスタンスをデプロイメントクライアントに提供します。
- 各デプロイメントサーバーにおいて serviceclass.conf のインスタンスを別に作成します。インスタンスはそのデプロイメントサーバーについて命名します。例えば次のようにします。 <tenantName>-serviceclass.conf.
- 各デプロイメントクライアントについて、deploymentclient.conf を定義し、デプロイメントサーバーが 1 つしかない場合にどうするかを記述します。

tenants.conf で定義できるもの

tenants.conf で異なるデプロイメントサーバーを"tenants"として特定します。tenants.conf は、これらのデプロイメントサーバーをホスティングする Splunk インスタンスにあります。デフォルトでは tenants.conf ファイルはありません。したがって、\$SPLUNK_HOME/etc/system/local にそれを作成し、その中に tenants を定義します。

各 tenant について、ヘッディングを [tenant:<tenantName>]とするスタanzasを作成し、ホワイトリスト/ブラックリスト

filterType を指定します。これで、デプロイメントクライアントは、これが使用すべきデプロイメントサーバーインスタンスかどうかを決定するのに使用します。

filterType

<whitelist または blacklist>

- デフォルトは whitelist。
- whitelist.<n> = <ipAddress または hostname または clientName>
- blacklist.<n> = < clientName の hostname の ipAddress>
 - ◆ 'n'は 0 から開始し、1 ずつ増分します。'n'がブレイクするとフィルターを見るのを停止します。
 - ◆ デプロイメントクライアントの ipAddress を 10.1.1.*のようにワイルドカードを使用することもできます。
 - ◆ デプロイメントクライアントの hostname。 *.splunk.com のようにワイルドカードを使用することもできます。
 - ◆ clientName- deploymentclient.conf の各デプロイメントクライアントに割り当てることができる論理又は'tag'名。clientName は、クライアントがフィルタにマッチしたとき、(ip/hostname)に対して優先します。

Apps および設定の展開または更新

Apps および設定の展開または更新

新規又は変更した App または設定をデプロイメントクライアントに展開する準備ができたなら、Splunk CLI を使用して、デプロイメントサーバーに対し、そのクライアントにとって新規コンテンツがあることを伝えます。そうすれば、次回、クライアントがそのデプロイメントサーバーをポーリングし、新規コンテンツを取り上げたことを認識します。

Splunk の CLI を使用するには、\$SPLUNK_HOME/bin/ディレクトリに移動し、./splunk コマンドを使用します。(Windows では、あらかじめ./を含める必要ありません。)

```
./splunk reload deploy-server
```

このコマンドは、すべてのサーバークラスについて変更をチェックし、クライアントに通知します。

-または

```
./splunk reload deploy-server -class <サーバークラス>
```

このコマンドは、指定したサーバークラスのみに通知します。例えば、

```
./splunk reload deploy-server -class www
```

このコマンドは、www サーバークラスのメンバーであるクライアントのみに通知します。

クライアントが設定を受信すると、splunkd を再開し、serviceclass.conf に設定されている場合は、関連 Apps を有効にします。

デプロイメント更新の確認

すべてのクライアントが設定を正しく受信したかどうかを確認するには、次を使ってデプロイメントサーバーからチェックします。

```
./splunk list deploy-clients
```

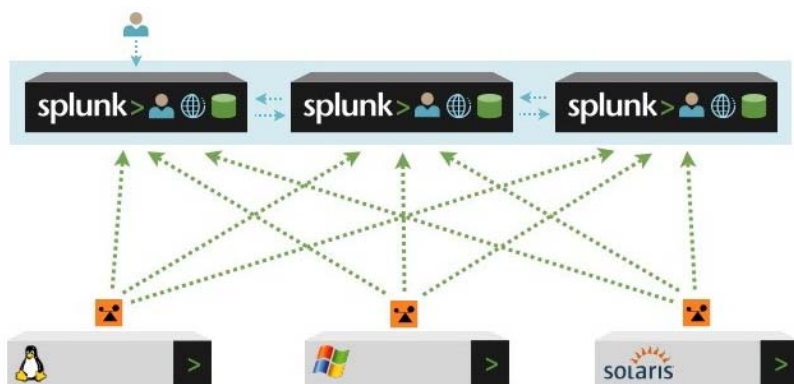
これは、すべてのデプロイメントクライアントと、前回の同時時間をリストアップします。

分散検索の設定

分散検索とは何か？

分散検索とは何か？

Splunk サーバーは検索要求を他の Splunk サーバーに分散し、その結果をマージしてユーザーに返すことができます。分散検索はバランスインデキシングと統合し、水平スケーリングを提供します。これにより、1日に何百ギガバイトまたはテラバイトの検索およびインデックス化が可能になります。さらに、分散検索により、ことなるデータサイロにわたってデータを関連づけることを可能にします。



分散デプロイメント用ライセンス

Splunk が分散環境において複数のホスト上で実行されている場合、各ホストについて、別々の固有ライセンスキーが必要です。すべてのホストで1つのライセンスキーを使用してきた場合は、この方法はバージョン4.0以降では使用できません。サポートに連絡して追加キーの生成をしてもらうか、ライセンスを複数キーに分けてもらってください。

分散検索の設定

分散検索の設定

このトピックは、分散検索の設定方法を説明します。方法は、Splunk Web または Splunk CLI を使用して行うもの、または設定ファイルを編集するものがあります。分散検索の設定後、そのキーを1つのインスタンスから他のインスタンスに分散する必要があります。

分散検索は、すべての Splunk ホストで、デフォルトで有効です。これは、以下のように他の設定を変更しない場合、Splunk の動作には影響しません。

Splunk Web を使用して分散検索を有効化する

Splunk Web を使用して分散検索を有効化するには、次を行います。

1. Splunk Web にログインし、**管理**をクリックします。

2. **分散検索の設定**をクリックします。
3. この分散検索ペアに必要な設定を指定し、**保存**をクリックします。

CLI で分散検索を有効にする

Splunk の CLI を使用して分散検索を有効にするには、この指示に従ってください。

Splunk の CLI を使用するには、`$SPLUNK_HOME/bin/`ディレクトリに移動し、`./splunk` コマンドを使用します。

分散検索を有効にする

```
splunk enable dist-search -auth admin:changeme
Distributed search enabled.
You need to restart the Splunk Server for your changes to take effect.
```

auto-discovery を有効にする

```
splunk enable discoverable -auth admin:changeme
Discoverable mode is now enabled.
You need to restart the Splunk Server for your changes to take effect.
```

検索サーバーの追加

```
splunk add search-server -host 10.10.10.10 -port 8888 -auth admin:changeme
Success.
You need to restart the Splunk Server for your changes to take effect.
```

distsearch.conf を編集して分散検索を有効にする

分散のための最も高度な指定は、`distsearch.conf` で利用できます。このファイルを `$SPLUNK_HOME/etc/system/local/` または `$SPLUNK_HOME/etc/apps/` にあるユーザーのカスタムアプリケーションディレクトリで編集します。設定ファイル一般に関する詳細は、「設定ファイルについて」を参照してください。

```
[distributedSearch]
```

- 分散検索設定オプションをこのスタンザ名で設定します。
- このスタンザ名に、次の属性/値のペアを続けてください。
- 属性を何も設定しない場合、Splunk はデフォルト値を使用します(リストにある場合)。

```
disabled = <true | false>
```

- 分散検索のオン/オフを切り替えます。
- デフォルトは `false`(分散検索スタンザはデフォルトで有効です)。

```
heartbeatFrequency = <秒>
```

- ハートビートを秒で指定。

- 0 はすべてのハートビートを無効にします。
- ハートビートが無効の場合、他の Splunk サーバーもこのインスタンスを自動検出できません。
- デフォルトは 2。

heartbeatMcastAddr = <IP アドレス>

- マルチキャストアドレスを設定します。
- デフォルトは 255.0.0.37。

heartbeatPort = <ポート>

- ハートビートポートを設定します。
- デフォルトは 60。

serverTimeout = <秒>

- サーバー接続の待ち時間。
- 接続された場合、検索はこの値の 10 倍でタイムアウトします。
 - ◆ 例えば、10 秒に設定する場合、最大検索は 100 になります。
- この設定は'removeTimedOutPeers'と平行して動作します。
- デフォルトは 10。

statusTimeout = <秒>

- サーバーがステータスを返す待ち時間。
- ピアされたサーバーが遅い場合、または、サーバー名が SplunkWeb ウィジェットから消えた場合、この数値を上げます。

removedTimedOutServers = <true | false>

- true の場合、'serverTimeout'内で確立できないサーバー接続を削除します。
- false の場合、そのサーバーに対するすべての呼び出しが接続を試みます。
 - ◆ 注意：これにより、ユーザーインターフェースが遅くなることがあります。

checkTimedOutServersFrequency = <秒>

- このタグは、'removeTimedOutServers'が true に設定されている場合にのみ意味があります。
- 'removeTimedOutServers'が false の場合、この属性は無視されます。
- この頻度(秒)でサーバーを再チェックします。
- これが 0 に設定されていると、再チェックされません。
- デフォルトは 60。

autoAddServers = [True | False]

- このタグが'true'に設定されている場合、このノードは自動的に発見されたすべてのサーバーを追加します。
- skipOurselves = [True | False]
- これが'true'に設定されている場合、このサーバーはサーバーとしてその検索その他の呼び出しにも参加しません。

- これは、他のサーバーからの結果をマージする以外にはなにも行わないノードを構築するのに使用されます。
- デフォルトは'false'。

`allowDescent = [True | False]`

`ttl = <整数>`

- 有効期間。
- この数値を増やすと、UDP マルチキャストパケットが現在のサブネットを超え、指定されたホップ数まで広がります。
- **注意:** これは、すべてのルーターが UDP マルチキャストパケットを通過させるように設定されている場合にのみ有効となります。
- デフォルトは 1(このサブネット)

`servers =`

- サーバーの初期リスト。
- 'autoAddServers'モードで完全に動作している場合(すべてのサーバーを検出)、サーバーをここに入力する必要はありません。

`blacklistNames =`

- ピアリングしたくないサーバー名のリスト。
- サーバー名は、スタートアップ時に作成された'server name'です。

`blacklistURLs =`

- ブラックリストに載せられた発見されたサーバーのコンマ区切りリスト。
- サーバー名(上記)またはサーバーURI((x.x.x.x:port))上でブラックリストに載せることができます。

例

`[distributedSearch]`

`heartbeatFrequency = 10`

`servers = 192.168.1.1:8059,192.168.1.2:8059`

`blacklistNames = the-others,them`

`blacklistURLs = 192.168.1.3:8059,192.168.1.4:8059`

このエントリーは検索を 192.168.1.1:8059,192.168.1.2:8059 に分散します。

サーバーはハートビートを 10 秒毎に送信します。

4 つのブラックリストに載せられたインスタンスがあり、`blacklistNames` および `blacklistURLs` に渡ってリストに記載されます。

属性はここでは設定されません。`distsearch.conf.spec` にリストアップされたデフォルトを使用します。

キーファイルの分散

Splunk インスタンスで分散検索(および再起動)を有効にした後、キーが `$SPLUNK_HOME/etc/auth/distServerKeys/` に作成されます。

`$SPLUNK_HOME/etc/auth/distServerKeys/trusted.pem` および `private.pem` を 1 つのホストから、分散検索に参加する他のホストに分散します。

複数 Splunk インスタンスから異なるキーをサポートする

Splunk インスタンスは、認証のために他のインスタンス上に保存される証明書を持つことができます。インスタンスはキーを `$SPLUNK_HOME/etc/auth/distSearchKeys/<peername>/<trusted|private>.pem` に保存することができます。

例： Splunk インスタンス A および B があり、両方が異なるキーを持ち、Splunk インスタンス C を検索したい場合、次のようにします。

- ピア C で `$SPLUNK_HOME/etc/auth/distSearchKeys/A` と `etc/auth/distSearchKeys/B` を作成します。
- 次に、A のキーを `$SPLUNK_HOME/etc/auth/distSearchKeys/A` にコピーし、B のキーを `$SPLUNK_HOME/etc/auth/distSearchKeys/B` にコピーします。
- 最後に、C を再起動します。

特定の Splunk サーバを分散検索から除外する

特定の Splunk サーバを分散検索から除外する

特定の Splunk サーバーを分散検索から除外するには(ブラックリストイングともいう)、サーバーのコンマ区切りリストを `distsearch.conf` に追加します。IP アドレスまたは完全に記述したドメイン名をスタンザに追加します。

```
blacklistNames = the-others,them
```

```
blacklistURLs = 192.168.1.3:8059,192.168.1.4:8059
```

```
blacklistNames
```

- 分散検索サーバーのために Splunk で定義した名前。

```
blacklistURLs
```

- 分散検索サーバーへの URL フルパス。

検索ジョブの管理

ジョブおよびジョブ管理について

ジョブおよびジョブ管理について

ユーザーが検索を Splunk で実装すると、それは"ジョブ"としてシステムに作成されます。このジョブはまた、与えられた検索が返す検索結果(検索結果など)を含んでいます。ユーザーはジョブマネージャで自分のジョブを停止および復帰することができます。管理者は、システム内のすべてのユーザーのジョブを管理することができます。

ジョブマネージャにアクセスするには、Splunk Web の右上にある、**ジョブ**をクリックします。



注意：ジョブリンクの横にある括弧内に表示されるジョブの数は、ログインしているユーザーが現在実行しているジョブの数であり、システム全体で実行しているジョブの数ではありません。管理者としてログインしていても、同様です。

また、OS のコマンドラインからでもジョブを管理することができます。

ユーザーが実行できるジョブを制限する

ユーザーが実行できるジョブの数およびジョブの結果のためのスペースを制限するための方法は、それらの制限で役割を定義し、それらをそれに割り当てることです。これは非常に高いレベルの精度で行うことができます。システム内の各ユーザーは自身の役割を持つことができます。

`$SPLUNK_HOME/etc/system/local` にある `authorize.conf` のコピーで `capability` を作成し、それに次の適当な値を与えます。

- `srchDiskQuota`: この役割に属するユーザーの検索ジョブに与えられるディスクスペースの最大量(MB)。
- `srchJobsQuota`: この役割のメンバーが持つことのできる同時に実行している検索の最大数。

詳細は、本書の役割作成に関するトピックを参照してください。

Splunk Web でのジョブの管理

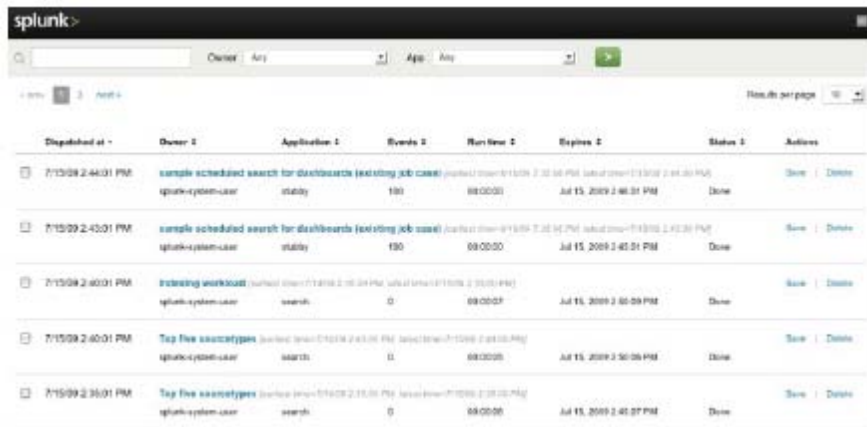
Splunk Web でのジョブの管理

管理者は、システム上の他のすべてのユーザーのジョブ実行を管理することができます。これらのジョブには、Splunk Web のジョブマネージャからアクセスできます。



注意：ジョブリンクの横にある括弧内に表示されるジョブの数は、ログインしているユーザーが現在実行しているジョブの数であり、システム全体で実行しているジョブの数ではありません。管理者としてログインしていても、同様です。

ジョブマネージャではポップアップウィンドウが表示され、そこに現在システム上で実行されているすべてのジョブが表示されます。



Dispatched at	Owner	Application	Events	Run time	Expires	Status	Actions
7/15/09 2:44:01 PM	splunk-system-user	sample scheduled search for dashboards (existing job case)	190	08:00:00	Jul 15, 2009 2:46:31 PM	Done	Save Delete
7/15/09 2:43:01 PM	splunk-system-user	sample scheduled search for dashboards (existing job case)	190	08:00:00	Jul 15, 2009 2:45:31 PM	Done	Save Delete
7/15/09 2:40:01 PM	splunk-system-user	stopping workload	0	08:00:07	Jul 15, 2009 2:40:59 PM	Done	Save Delete
7/15/09 2:40:01 PM	splunk-system-user	Top five sourcetypes	0	08:00:00	Jul 15, 2009 2:39:58 PM	Done	Save Delete
7/15/09 2:39:01 PM	splunk-system-user	Top five sourcetypes	0	08:00:00	Jul 15, 2009 2:40:27 PM	Done	Save Delete

システム上のジョブを保存、一時停止、削除、再開、削除および完了するには、コントロールを使用します。有効にしたい項目の左側のチェックボックスを選択し、ページの下部にある関連ボタンをクリックします。

保存されていない検索ジョブは、完了後に、設定時間内に時間切れとなります。つまり、検索結果(それらの結果を含む)がファイルシステムから削除され、そのジョブを明示的に保存しない限り、回収することができません。

手動で実行する検索ジョブのデフォルトの有効期間は 15 分です(スケジュールされた検索による検索ジョブは一般的にはるかに短い有効期間となる)。期限切れ列は、各リストが削除されるまでの時間を表示しています。期限切れの後、検索ジョブをレビューしたい場合、または他と共有したい場合は、保存してください。

OS でのジョブを管理する

OS でのジョブを管理する

Splunk がジョブを実行しているとき、Splunk 検索と呼ばれる、OS 内のプロセスとして表示します。管理を使ってこのジョブを扱うことができますが、また、OS のコマンドラインでジョブのプロセスを管理することもできます。

ジョブのプロセスとその引数を見るには、次のようにタイプします。

```
> top
> c
```

これは、実行しているすべてのプロセスおよびすべての引数を表示します。

`ps -ef | grep splunk-search` と入力すると、このリスト内のすべての Splunk 検索を分離します。次のように表示されます。

```
[pie@fflanda ~]$ ps -ef | grep splunk-search
pie 21368 19460 96 13:51 ? 00:01:18 splunk-search --search=search sourcetype="access_combined"
pie 21371 21368 0 13:51 ? 00:00:00 splunk-search --search=search sourcetype="access_combined"
```

```
pie 22804 20379 0 13:52 pts/9 00:00:00 grep splunk-search
```

各検索ジョブには2つのプロセスがあります。第2のプロセスは'helper'プロセスで、`<codesplunkd</code>`プロセスが使用し、必要に応じて追加の作業を行います。メインジョブはシステムリソースを使用するジョブです。ヘルパープロセスは、そのメインプロセスを停止すると、自ら停止します。

プロセス情報には次のものがあります。

- 検索文字列(search=)
- そのジョブのジョブ id (id=)
- ジョブ検索結果(生成する出力)がディスクに残り利用できる ttl、つまり時間 (ttl=)
- ジョブを実行しているユーザー (user=)
- ユーザーが属している役割 (roles=)

ジョブを実行している際、そのデータが`$SPLUNK_HOME/var/run/splunk/dispatch/<job_id>/`に書き込まれます。スケジュールされたジョブ(スケジュールされた保存された検索)には、ディレクトリ名の一部として、保存済み検索名が含まれます。

プロセスの ttl の値は、ジョブを停止した後においても、そのデータがこのスポットに留まる時間を決定します。ジョブを OS から停止した場合、その検索結果を削除する必要があるかを見るために、停止前にそのジョブ id を見る必要があるかもしれません。

ファイルシステムを使用してジョブを管理する

Splunk では、ジョブの検索結果ディレクトリで、項目を作成および削除することによりジョブを管理することができます。

- ジョブをキャンセルするには、そのジョブの検索結果ディレクトリに移動し'cancel'という名前のファイルを作成します。
- そのジョブの検索結果を保存するには(および、その ttl 設定を無視する)、'save'とう名前のファイルを作成します。
- ジョブを一時停止するには、'pause'という名前のファイルを作成し、それを一時停止から復帰し、その'pause'ファイルを削除します。

Splunk のコマンドラインインタフェース(CLI)を使用する

CLI について

CLI について

Splunk CLI を使用して、Splunk サーバ上で、検索を監視、設定および実行することができます。Splunk 役割設定は、実行可能なアクションを決定します。ほとんどのアクションの実行には、Splunk 管理者である必要があります。

CLI へのアクセス方法

Splunk CLI にアクセスするには、次のいずれかを行う必要があります。

- Splunk サーバーへのシェルアクセス、または
- リモート Splunk サーバー上で正しいポートへアクセスするための権限

管理者またはルート権限を持つ場合、Splunk インストールのトップレベルディレクトリをシェルパスに追加することにより、CLI 使用を単純化することができます。\$SPLUNK_HOME 変数は、トップレベルディレクトリを参照します。

SPLUNK_HOME 環境変数を設定し、\$SPLUNK_HOME/bin をシェルのパスに追加します。

この例は、デフォルト位置に Splunk をインストールした、Linux/BSD/Solaris ユーザーで機能します。

```
# export SPLUNK_HOME=/opt/splunk
# export PATH=$SPLUNK_HOME/bin:$PATH
```

この例は、デフォルト位置に Splunk をインストールした、Mac ユーザーで機能します。

```
# export SPLUNK_HOME=/Applications/Splunk
# export PATH=$SPLUNK_HOME/bin:$PATH
```

CLI コマンド

管理者権限がある場合、CLI を、検索だけでなく、Splunk サーバーの設定および監視にも使用できます。Splunk の設定および監視に使用する CLI コマンドは、検索コマンドではありません。検索コマンドは、search および dispatch CLI コマンドの引数です。

すべての CLI コマンド資料は、CLI ヘルプレファレンスにあります。詳細は、「CLI でヘルプ情報を得る」を参照してください。

Mac ユーザーのための注意

Mac OS X では、システムファイルまたはディレクトリにアクセスするコマンドを実行するには、スーパーユーザーレベルアクセスが必要になります。CLI コマンドを、**sudo** または "su -" を使用して、ルートで新規シェルを操作します。sudo の使用を推奨します。(デフォルトでは、ユーザー "root" は有効になっていませんが、管理者ユーザーは sudo を使用することができます。)

CLI でヘルプ情報を得る

CLI でヘルプ情報を得る

help コマンドを使って完全な CLI ヘルプレファレンスを見ることができます。デフォルト CLI ヘルプページにアクセスします。そのためには、Splunk の実行中に、コマンドラインに次を入力します。

```
./splunk help
```

特定の CLI コマンドまたはトピックに関するヘルプにアクセスするには、次を入力します。

```
./splunk help command name | topic name
```

例えば、Splunk CLI コマンドに関するヘルプページにアクセスする場合は、

```
./splunk help commands
```

または、Splunk 検索コマンドに関するヘルプページにアクセスする場合は、

```
./splunk help search-commands
```

注意： search"と"commands"の間のダッシュ(-)に注意してください。なぜなら、Splunk CLI がスペースをブレイクを解釈するからです。2 語以上からなるトピック名については、複数語の間にダッシュを使用してください。

認証またはターゲットホスト情報が必要なコマンド

CLI コマンドで、auth および uri パラメータを使用します。

auth

auth は、実行認証が必要なコマンドで使用します。auth は、現在ログインしている権限以外の、実行権限が必要なコマンドの実行が必要な場合に便利です。

注意： auth は、CLI コマンド引数で指定する最後のパラメータである必要があります。

構文:

```
./splunk command object [-parameter value]... -auth username:password
```

un

uri はコマンドを他の Splunk サーバーに送るときに使用します。

構文:

```
./splunk command object [-parameter value]... -uri specified-server (= [http|https]://name_of_server:management_port)
```

例：

```
./splunk search "host=fflanda error 404 *.gif" -auth admin -uri https://splunkserver:8089
```

設定ファイルレファレンス

admon.conf

admon.conf

次は admon.conf の仕様とファイル例です。

admon.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# このファイルは、Windows アクティブディレクトリを設定する際に使用する可能性のある属性/値ペアを記載しています。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
[<スタンプ名>]
```

ドメインコントローラには複数の設定が可能です。したがって、これは特定の設定セットに関連する固有な名前です。

```
targetDC = <文字列>
```

完全に記述したドメイン名。これは空白でもかまいません。その場合、ローカルのコンピュータ DC を取得し、そのルート DN にバインドします。

```
startingNode = <文字列>
```

モニタリングを開始する AD のディレクトリツリーまでのパスを指定します。空白の場合、そのディレクトリツリーのルートから開始します。

```
monitorSubtree = <整数 0|1>
```

DC パスの場合、シングルレベルの代わりにサブツリーを監視します。

```
disabled = <in 0|1>
```

この特定の設定を有効または無効にします。

admon.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# このファイルは、Windows アクティブディレクトリモニターへの変更モニタリングするための設定例を記載しています。詳細は admon.conf.spec を参照してください。
```

```
# 次は、アクティブディレクトリ監視設定の例です。
```

```
#
```

```

#       この設定の1つまたは複数を使用するには、その設定ブロックを$SPLUNK_HOME/etc/apps/windows/local/の
admon.conf にコピーしてください。 設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[default]
monitorSubtree = 1
disabled = 0
[DefaultTargetDC]
targetDc =
startingNode =

```

alert_actions.conf

alert_actions.conf

次は alert_actions.conf の仕様とファイル例です。

alert_actions.conf.spec

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       このファイルは、alert_actions.conf のグローバル保存済み検索アクションの設定のために使用する可能性のある属性
と値を記載しています。 保存済みの検索は savedsearches.conf で設定されます。
#
#       $SPLUNK_HOME/etc/system/default/に alert_actions.conf があります。カスタム設定を設定するには、
alert_actions.conf を$SPLUNK_HOME/etc/system/local に置いてください。例は、alert_actions.conf の例を参照して
ください。設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
.
#####
#       グローバルオプション： これらの設定は、スタンザ名をつける必要はありません。
#       各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
#####
maxresults = <整数>
*       アラートにより送られる検索結果のグローバル最大数を設定します。

```

```

*      デフォルトは 100。
hostname = <文字列>
*      アラートにより送信される、リンクに表示されるホストネームを設定します。
*      これは、アラートを送信するマシンが FQDN を持たない場合に便利です。
*      デフォルトは現在のホストネーム (Splunk で設定) またはローカルホスト (何も設定されていない場合) です。
ttl = <整数>[p]
*      検索検索結果の最小 ttl を指定するオプション引数 (この無効がトリガされる場合)。
#####
#      EMAIL: これらの設定は、[email] スタンザ名を付けます。
#####
[email]
*      このスタンザ名のもとで E メール通知オプションを設定します。
*      このスタンザ名に、次の属性/値のペアを続けてください。
*      各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
from = <文字列>
*      アラートの発信元メールアドレス。
*      デフォルトは splunk@$LOCALHOST。
subject = <文字列>
*      代替の E メールサブジェクトを指定します。
*      デフォルトは SplunkAlert-<savedsearchname>。
format = <文字列>
*      Eメールのテキストのフォーマットを指定します。
*      利用可能な値: plain、html、raw、および csv。
*      この値はまた、添付ファイルにも適用されます。
inline = <true | false | auto>
*      検索結果をアラート Eメールの本文に含めるかどうかを指定します。
*      デフォルトは false。
mailserver = <文字列>
*      Eメール送信時に使用する SMTP メールサーバー。
*      デフォルトは $LOCALHOST。
#####
#      RSS: これらの設定は、[rss] スタンザ名を付けます。
#####
[rss]
*      このスタンザ名のもとで rss 通知オプションを設定します。
*      このスタンザ名に、次の属性/値のペアを続けてください。
*      各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
items_count = <数>

```

```

*       保存される RSS フィードの数。
*       maxresults を超えて設定できません([email]スタンザ)。
*       デフォルトは 30。
#####
#       summary_index: これらの設定は、[summary_index]スタンザ名を付けます。
#####
command = <文字列>
*       サマリーインデックスアクションをトリガする保存済み検索からの情報で実現するコマンドテンプレート。
#####
#       populate_lookup: これらの設定は、[populate_lookup]スタンザ名を付けます。
#####
command = <文字列>
*       投入ルックアップアクションをトリガする保存済み検索からの情報で実現するコマンドテンプレート。

```

alert_actions.conf.example

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       これは alert_actions.conf の例です。保存済みの検索のアラートアクションを設定するのにこのファイルを使用してく
#       ださい。
#
#       この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の
#       alert_actions.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
#       http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
#       照してください。
[email]
from = <メールアドレス>
#       メールアドレスからカスタムを設定します。
subject = <カスタムサブジェクト>
#       デフォルトでは、サブジェクトは SplunkAlert-<splunk-name>ですが、カスタムサブジェクトを設定することができま
#       す。
format = <html, plain, csv>
#       E メールテキストのフォーマットを指定します。
#       利用可能な値: html, plain, csv.
[rss]
items_count=30
#       RSS フィードのしきい値を設定します。

```

```
[summary_index]
```

```
command = cacher index="myindex" marker="saved_search=\"$name$", nonce=\"$#random$\"
```

```
# myindex での結果を補損、与えられたマーカを各イベントに追加、詳細は reportcache を参照。
```

app.conf

app.conf

次は app.conf の仕様とファイル例です。

app.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# このファイルは、カスタムアプリケーションのユーザー入力フィールドを作成するための属性/値ペアを記載しています。
```

```
# app.conf を通じてユーザー入力の利用可能フィールドを設定してください。
```

```
# デフォルトの app.conf はありません。カスタム設定を設定するには、app.conf を $SPLUNK_HOME/etc/system/local  
に置いてください 例は、app.conf.example を参照してください。
```

```
# 設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参  
照してください。
```

```
#
```

```
# このアプリケーションのためのインストール設定を設定します。
```

```
#
```

```
# Launcher 設定
```

```
#
```

```
# 最初の Launcher App で App を見るには、アイコンの画像を置き、App の appserver/static dir にスクリーンショッ  
トします。
```

```
# 例えば、appIcon_<appname>.png, screenshot_<appname>.png
```

```
# これらイメージの仕様は次のとおりです：
```

```
# アプリケーションアイコンは 36x36 ピクセルで PNG フォーマットの必要があります。
```

```
# スクリーンショットは少なくとも 490 x 275 である必要がありますが、それよりも大きくても可能です。なぜなら、自動  
でトリミングされるからです。または PNG フォーマットの必要があり、ブラウザがクロームしてはなりません。
```

```
# 画像は次のフォーマットが利用可能です： png, jpg, jpeg, gif
```

```
# 作者などの他のプロパティを設定するには、app.conf に [launcher] と呼ばれるスタanzas を作成し、次の情報を入力しま  
す。
```

```
# 作者
```

```
# 説明
```

```

#     バージョン
[install]
state = disabled | enabled
*     App を無効にするか有効にするかを設定します。
*     App が無効の場合は、その設定は無視されます。
*     デフォルトでは、アプリケーションは有効です。
#
#     この App の UI に特定の設定を設定します。
#
[ui]
is_visible = true | false
*     この App が UIApp として見える / ナビゲートするかどうかを指示します。
*     アプリケーションは少なくとも、UI から利用可能な 1 つのビューが必要です。
label = <文字列>
*     App に分かりやすいラベルを定義します。
#
#     この App に関するカスタム設定ファイル設定を設定します。
#
[config:$STRING]
*     スタンザに名前を付けます。
*     設定の始め :
*     $STRING を任意の識別子に設定します。
targetconf = <$CONFIG_FILE>
*     変更する設定ファイルを対象にします。
*     1 つのみ可能です。
*     アプリケーションに含まれているいずれかの設定ファイル。
*     例えば、indexes.conf のインデックス。
targetstanza = <$STANZA_NAME>
*     アプリケーションからのスタンザ名。
targetkey = <$ATTRIBUTE>
*     設定する属性。
targetkeydefault = <$VALUE>
*     属性のデフォルト設定。
*     デフォルト内にするには空白も可能。
conflabel = <$LABEL>
*     Splunk Web で表示する設定の短い説明。

```

app.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# 次は app.conf 設定の例です。カスタムアプリケーションに応じてプロパティを設定してください。
#
# デフォルトの app.conf はありません。
#
# この設定の1つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の props.conf にコ
# ピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[config:coldindexpath]
targetconf=indexes
targetstanza=sampledatab
targetkey=coldPath
targetkeydefault=$SPLUNK_DB/sampledatab/coldddb
conflabel=Cold DB Path for Sample Data Index
[config:thawedindexpath]
targetconf=indexes
targetstanza=sampledatab
targetkey=thawedPath
targetkeydefault=$SPLUNK_DB/sampledatab/thawedddb
conflabel=Thawed DB Path for Sample Data Index
[config:homeindexpath]
targetconf=indexes
targetstanza=sampledatab
targetkey=homePath
targetkeydefault=$SPLUNK_DB/sampledatab/db
conflabel=Home DB Path for Sample Data Index
[launcher]
author=< app の作者 >
description=< app のテキストによる説明 >
version=< app のバージョン >
```

audit.conf

audit.conf

次は audit.conf の仕様とファイル例です。

audit.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、audit.conf で監査およびイベント署名の設定に使用することができる属性と値を記載しています。
#
# デフォルトの audit.conf はありません。カスタム設定を設定するには、audit.conf を
$SPLUNK_HOME/etc/system/local に置いてください。例は、audit.conf.example を参照してください。設定を有効にするに
は Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#####
# EVENT HASHING: SHA256 イベントハッシングをオンにします。
#####
[eventHashing]
* このスタンザはイベントハッシングをオンにします。すべてのイベントが SHA256 でハッシュされます。
* インデックスマシンは、ブロックですべての署名を暗号化します。
* このスタンザ名に、次の属性/値のペアを続けてください。
filters=mywhitelist,myblacklist...
* (オプション) イベントがハッシュされるフィルタ。
* イベントに適用する filtername 値を指定します。
* 注意: 優先順位は左から右。
# イベントハッシングのフィルタ仕様
[filterSpec:<event_whitelist | event_blacklist>:<filtername>]
* このスタンザは、イベントの whitelisting または blacklisting をオンにします。
* filternames を "filters" エントリ(上記)で使用します。
* 例えば、[filterSpec:event_whitelist:foofilter]。
all=<true | false>
* 'all' タグは、'all' イベントを停止するようブラックリストに伝えます。
* デフォルトは 'false'。
ブラックリスト/ホワイトリストされたソース、ホストまたはソースタイプのオプションリスト(左から右の順番)。
* 例:
```

source=s1,s2,s3...

host=h1,h2,h3...

sourcetype=st1,st2,st3...

#####

KEYS: 暗号化の公開鍵および秘密鍵を指定します。

#####

[auditTrail]

* このスタanzasは、監査追跡イベントの暗号署名をオンにし(inputs.conf で設定)、イベントをハッシュします(イベントハッシュングが上記で有効化されている場合)。

privateKey=@OsDirSep@some@OsDirSep@path@OsDirSep@to@OsDirSep@your@OsDirSep@private@OsDirSep@key@publicKey=@OsDirSep@some@OsDirSep@path@OsDirSep@to@OsDirSep@your@OsDirSep@public@OsDirSep@key@OsDirSep@

* 署名を暗号化し、復号化するには秘密鍵が必要です。

* 自身のキーへのパスを設定します。

* \$SPLUNK_HOME/bin の openssl を使用して自身のキーを生成します。

queuing=<true | false>

* 監査イベントを indexQueue に送信しない(オフ)ようにします。代わりにオーデイトイベントを追跡します。

* false に設定された場合、inputs.conf スタanzasを追加して、audit log を追跡する必要があります。

* デフォルトは true です。

audit.conf.example

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

#

これは audit.conf の例です。このファイルを、監査およびイベントハッシュングの設定に使用してください。

#

デフォルトの audit.conf はありません。

#

この設定の1つまたは複数を使用するには、その設定ブロックを \$SPLUNK_HOME/etc/system/local の audit.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。

#

設定ファイル(優先順位を含む)についての詳細は、

<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork> にあるドキュメントを参照してください。

[auditTrail]

privateKey=@OsDirSep@some@OsDirSep@path@OsDirSep@to@OsDirSep@your@OsDirSep@private@OsDirSep@key@publicKey=@OsDirSep@some@OsDirSep@path@OsDirSep@to@OsDirSep@your@OsDirSep@public@OsDirSep@key@OsDirSep@

このスタanzasが存在する場合、監査追跡イベントは暗号化して署名されます。

署名を暗号化し、復号化するには秘密鍵が必要です。

```

# $SPLUNK_HOME/bin の openssl を使用して自身のキーを生成します。

# 例#1 - 全イベントをハッシュ:
[eventHashing]
# これは、_audit index にいくもの以外(これらは独自の方法で扱われます)のすべてのイベントについて SHA256 ハッシュ
# を実行します。
# 注意: ハッシングを有効にするのに必要なことは、スタンザ 'eventHashing' が存在することだけです。

# 例 # 2 - 単純ブラックリスティング
[filterSpec:event_blacklist:myblacklist]
host=somehost.splunk.com, 45.2.4.6, 45.3.5.4
[eventHashing]
filters=myblacklist
# Splunk は、リストアップされたホストからのイベントをハッシュしません。それらは 'ブラックリスト' に記載されます。
# それ以外の全イベントはハッシュされます。

# 例#3 - マルチプルブラックリスティング
[filterSpec:event_blacklist:myblacklist]
host=somehost.splunk.com, 46.45.32.1
source=@OsDirSep@some@OsDirSep@source
sourcetype=syslog, apache.error
[eventHashing]
filters=myblacklist
# ソース、ソースタイプおよびホストを含むイベントはすべてハッシュしないでください。それらはブラックリストに記載さ
# れています。それ以外の全イベントはハッシュされます。

# 例#4 - ホワイトリスティング
[filterspec:event_whitelist:mywhitelist]
sourcetype=syslog
#source=aa, bb (these can be added as well)
#host=xx, yy
[filterspec:event_blacklist:nothingelse]
# 'all' タグは特別な論理演算子(デフォルトは false)であり、すべてのイベントにマッチします。
all=True
[eventSigning]
filters=mywhitelist, nothingelse
# ソースタイプが syslog のイベントのみハッシュします。それ以外の全イベントはハッシュされません。
# フィルタのリストを作成することができ、すべてのイベントについて左から右に実行されます。
# 1 つのイベントがホワイトリストを通過すると、フィルタの残りは実行されません。したがって、"すべて 2 のブラックリス
# トフィルタの前にホワイトリストフィルタを置くと、"ホワイトリストにマッチするイベントだけをハッシュする"となります。

```

authentication.conf

authentication.conf

次は、authentication.conf の仕様および例です。

authentication.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、認証を authentication.conf で設定するための属性および値を記載しています。
#
#      authentication.conf が $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、
authentication.conf を $SPLUNK_HOME/etc/system/local に置いてください。例は、authentication.conf.example を
参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[authentication]
*      このスタンザ名に、次の属性/値のペアを続けてください。
authType = <文字列>
*      使用する認証システムを指定します。
*      現在利用可能： Splunk、LDAP、スクリプト済。
*      デフォルトは Splunk。
authSettings = <文字列>
*      選択された認証システムの特定の設定をロックアップするキー。
*      <文字列>は、スタンザヘッダ [<authSettingsKey>] の名前です。
*      これは、LDAP およびスクリプト済認証により使用されます。
#####
#      LDAP 設定
#####
[<authSettings-key>]
*      このスタンザ名に、次の属性/値のペアを続けてください。
host = <文字列>
*      LDAP サーバーのホストネーム
*      Splunk サーバーがホスト名を解釈できることを確認してください。
port = <整数>
*      LDAP サーバーとの接続に Splunk が使用するポートを指定します。
*      デフォルトでは、LDAP サーバーは TCP ポート 389 でリッスンします。
```

pageSize = <整数>

- * 一度に返すレコード数を決定します。
- * 0 を入力すると無効にし、LDAPv2 に戻ります。
- * デフォルトは 800。

SSLEnabled = <整数>

- * 0 で無効になります。
- * 1 で有効になります。
- * SSL LDAP 設定については、\$SPLUNK_HOME/etc/openldap/openldap.conf のファイルを参照してください。

bindDN = <文字列>

- * LDAP レコードを取得するマネージャのバインド文字列。
- * このユーザーは、Splunk を追加したい全 LDAP ユーザーにアクセスできる必要があります。

bindDNpassword = <文字列>

- * bindDN ユーザーのパスワード。

groupBaseDN = <文字列>

- * LDAP でのユーザグループの位置。
- * ';' で区切られたリストを提供して multiple trees を検索することもできます。

groupBaseFilter = <文字列>

- * この属性はグループ名を定義します。
- * デフォルト値は objectclass=*。これはほとんどの設定で機能します。
- * Splunk はまた、POSIX-style GID をグループベースフィルタとして受け入れることができます。

groupMappingAttribute = <文字列>

- * 1 つのグループ内のユーザーのリストがユーザーの dn にマッチしない場合の、LDAP グループマッピングの名前。
 - * これは、uid 属性のリストであり、dn 属性でないことがあります。
- * 通常は、このフィールドを空白にします。

groupMemberAttribute = <文字列>

- * これは普通、member または memberOf で、メンバーシップがグループエントリまたはユーザエントリにリストアップされているかどうかによります。
 - * 標準 POSIX 値は member です。

groupNameAttribute = <文字列>

- * ユーザーおよびグループが同じツリーに定義されている場合にのみ設定します。
- * 通常 cn です。

realNameAttribute = <文字列>

- * Splunk の realname フィールドにマップする LDAP ユーザーフィールドの名前。
- * 例 : cn

userBaseDN = <文字列>

- * LDAP でのユーザーレコードの位置。
- * 複数のツリーを検索するには、 ';' で区切られたリストを入力します。
- * この値を設定しないと、認証は機能しません。

userBaseFilter = <文字列>

- * ユーザーをフィルタリングしたいオブジェクトクラス。
- * デフォルト値は objectclass=*。これはほとんどの設定で機能します。
- * または、ユーザーに関する特定のフィルタを設定します。
- * 例：

```
userBaseFilter = (|(department=IT)(department=HR))
```

これは、IT 部または HR 部にあるユーザーにマッチします。

userNameAttribute = <文字列>

- * 注意： username 属性は空白を含んではなりません。 username は大文字と小文字を区別します。
- * アクティブディレクトリでは、これは sAMAccountName となります。
- * 値 uid はほとんどの設定で機能します。

failsafeLogin = <文字列>

- * このログインにより、LDAP サーバーが接続できない場合に、Splunk にログインできるようになります。
- * 重要： このユーザーは、Splunk インストールで管理者権限を持ちます。

failsafePassword = <文字列>

- * フェイルセーフユーザーのためのデフォルトパスワード。

```
#####
```

```
# 役割の割り当て
```

```
#####
```

```
[roleMap]
```

- * このスタanzasに、次の属性/値のペアを続けてください。

<RoleName> = <文字列>

- * LDAP 役割を Splunk 役割に割り当てます (authorize.conf の定義を使用)。
- * このリストはセミコロンで区切られています (スペースなし)。

```
#####
```

```
# スクリプト認証
```

```
#####
```

```
[<authSettings-key>]
```

- * このスタanzas名に、次の属性/値のペアを続けてください。

scriptPath = <文字列>

- * スクリプトまでのフルパス。
- * 例 \$SPLUNK_HOME/etc/system/bin/\$MY_SCRIPT.

scriptSearchFilters = 0|1

- * 1 に設定すると、スクリプトを呼び出し、検索フィルタを追加します。
- * 0 で無効。
- # キャッシュタイミング：
- # Splunk がアプリケーションを呼び出す頻度を調整するには、これらの設定を使用します。
- # 各呼び出しは、それぞれ秒で指定したタイムアウトがあります。キャッシングは、これを指定しないと実行されません。

```
[cacheTiming]
getUserInfoTTL = <整数>
*     getUserInfo のタイムアウトを秒で指定。
getUserTypeTTL = <整数>
*     getUsertype のタイムアウトを秒で指定。
getUsersTTL = <整数>
*     getUsers のタイムアウトを秒で指定。
userLoginTTL = <整数>
*     userLogin 呼び出しのタイムアウト。
getSearchFilterTTL = <整数>
*     検索フィルタのタイムアウト。
```

authentication.conf.example

```
#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#     これは authentication.conf の例です。LDAP の設定または、LDAP と Splunk のネイティブ認証システムとの切替のため
#     にこのこのファイルを使用してください。
#     この設定の 1 つまたは複数使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の
#     authentication.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
#     http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
#     照してください。
#     Splunk 内蔵の認証を使用する :
[auth]
authType = Splunk
#     LDAP を使用
[authentication]
authType = LDAP
authSettings = ldaphost
[ldaphost]
host = ldaphost.domain.com
pageSize = 0
port = 389
SSLEnabled = 0
failsafeLogin = failsafe
failsafePassword = fail
bindDN = cn=Directory Manager
```

```

bindDNpassword = password
groupBaseDN = ou=Groups,dc=splunk,dc=com;
groupBaseFilter = (objectclass=*)
groupMappingAttribute = dn
groupMemberAttribute = uniqueMember
groupNameAttribute = cn
realNameAttribute = givenName
userBaseDN = ou=People,dc=splunk,dc=com;
userBaseFilter = (objectclass=*)
userNameAttribute = uid
#       認証ユーザーに対して authorize.conf で作成した役割を割り当てるためにスタanzasを設定することもできます。
[roleMap]
Admin = SplunkAdmins
#       Scripted Auth examples
#       次の例は、RADIUS 認証の場合です。
[authentication]
authType = Scripted
authSettings = script
[script]
scriptPath = $SPLUNK_HOME/bin/python
$SPLUNK_HOME/share/splunk/authScriptSamples/radiusScripted.scriptSearchFilters = 1
#       次の例は PAM 認証の場合です。
[authentication]
authType = Scripted
authSettings = script
[script]
scriptPath = $SPLUNK_HOME/bin/python $SPLUNK_HOME/share/splunk/authScriptSamples/pamScripted.py
scriptSearchFilters = 1
[cacheTiming]
userLoginTTL = 1
searchFilterTTL = 1
getUserInfoTTL = 1
getUserTypeTTL = 1
getUsersTTL = 1

```

authorize.conf

authorize.conf

次は authorize.conf の仕様とファイル例です。

authorize.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、authorize.conf で役割を作成するための属性/値ペアを記載しています。
# 独自の authorize.conf を作成することにより、役割および細かいアクセスコントロールを設定することができます。
# authorize.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、
authorize.conf を $SPLUNK_HOME/etc/system/local に置いてください。例は、authorize.conf.example を参照してくだ
さい。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[capability::<capability>]
* Splunk で機能を定義します。
* これはまた、システムにソフトウェアを登録することにより動的に追加することもできます (restmap.conf.spec を参照)。
* Splunk は、ほとんどの機能をこのように追加することができ、これによって、レファレンス用のファイルの最後に列挙す
ることができます。
* 機能のデフォルトリストは下記を参照。
[role_<roleName>]
<capability_name> = <enabled|disabled>
* この役割に追加する機能。
* 複数の機能を入力することができます。
importRoles = <文字列>
* インポートする他の役割機能をセミコロンで区切ったリスト。
srchFilter = <文字列>
* この役割用の検索フィルタをセミコロンで区切ったリスト。
srchTimeWin = <文字列>
* 検索の最大時間の幅。
srchDiskQuota = <整数>
* この役割に属するユーザーの検索ジョブに与えられたディスクスペースの最大量 (MB)。
srchJobsQuota = <整数>
* この役割のメンバーが持つことのできる同時に実行している検索の最大数。
# 次は Splunk の機能のリストです。注意：このリストは、新規機能が追加され、古い機能が廃止された際に変更されること
```

があります。authorize.conf の設定中に問題が発生したときは、チケットを http://www.splunk.com/page/submit_issue に提出してください。

[role_Admin]

edit_user = CLI/UI でのユーザー情報の変更。

edit_search_server = \$SPLUNK_HOME/etc で XML 設定ファイルを書き込む能力を与えます。

delete_user = UI/CLI のユーザーを削除します。

change_authentication = 認証設定の保存を可能にします。

bounce_authentication = UI/CLI の認証を再ロードします。

delete_by_keyword = アクセス削除検索演算子。

license_tab = アクセスライセンスタブ。

edit_alert_action = アラートアクションの変更。

edit_roles = ロールに対するユーザーマッピングの変更。

edit_deployment_server = デプロイメントサーバー設定の変更。

edit_deployment_client = デプロイメントクライアント設定の変更。

indexes_edit = インデックス設定の変更。

edit_input_defaults = デフォルト入力設定の変更。

edit_monitor = 監視入力設定の変更。

edit_scripted = スクリプト入力設定の変更。

edit_splunktcp = TCP で分散データ設定を設定。

edit_splunktcp_ssl = TCP SSL 設定を設定。

edit_tcp = TCP 入力設定を変更。

edit_udp = UDP 入力設定を変更。

edit_server = server.conf のサーバー設定を変更。

edit_web_settings = web.conf の設定を変更。

edit_forwarders = フォワーディング側の設定を変更。

use_file_operator = ファイルシステムの検索のためのファイル演算子を使用。

request_auth_token = 他のユーザーのための認証トークンを取得。

rest_apps_management = REST エンドポイントを使ったアプリケーションの管理。

rest_properties_get = REST サービス/プロパティの読み込み。

rest_properties_set = REST サービス/プロパティの書き込み。

admin_all_objects = システム内のすべてのオブジェクトの管理を許可(ユーザーオブジェクト、検索ジョブなど)。

importRoles = Power;User

srchFilter =

[role_Power]

schedule_search = 検索のスケジューリング。

allow_livetail = UI でライブ追跡を表示。

edit_tags = イベントのタグを設定。

importRoles = ユーザー

```

srchFilter      =
[role_User]
get_metadata    =      メタデータ検索プロセッサのメタデータにアクセス。
get_typeahead   =      入力先読みの許可。
search =        検索を実行。
#      スクリプト実行機能
list_inputs     =      入力をリスト表示。
importRoles     =
srchFilter      =

```

authorize.conf.example

```

#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      これは authorize.conf の例です。役割および機能の設定にこのこのファイルを使用します。
#
#      この設定の1つまたは複数使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の authorize.conf
にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[role_Ninja]
edit_save_search      =      有効
schedule_search       =      有効
edit_eventtype =      有効
edit_role_search      =      有効
edit_local_search     =      有効
savesearch_tab =      有効
edit_tags             =      有効
importRoles           =      User;Everybody
srchFilter             =      host=foo
#      これは、Ninja 役割を作成します。これは、デフォルト役割 User および Everybody から機能を引き継ぎます。
#      Ninja は、Power とほぼ同じ機能を持ちます。ただし、アラートを作成できないことが異なります(保存済みの検索のみ)。
#      また、Ninja は host=foo での検索に制限されます。

```

commands.conf

commands.conf

次は `commands.conf` の仕様とファイル例です。

commands.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、作成されたカスタム検索スクリプトの検索コマンドを作成するための属性/値ペアを記載します。カスタム
検索スクリプトを $SPLUNK_HOME/etc/searchscripts/ または $SPLUNK_HOME/apps/MY_APP/bin/ に追加してください。後者につ
いては、カスタムの commands.conf を $SPLUNK_HOME/apps/MY_APP に置いてください。前者については、カスタムの
commands.conf を $SPLUNK_HOME/etc/system/local に置いてください。
# commands.conf は $SPLUNK_HOME/etc/system/default/ にあります。例は、commands.conf.example を参照して
ください。設定を有効にするには Splunk の再起動が必要です。
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[ $STANZA_NAME ]
* 各スタンザは、検索コマンドを表します。コマンドはスタンザ名です。
* スタンザ名は、検索言語のコマンドを呼び出します。
* 次の属性/値をコマンドに設定します。設定しない場合、Splunk はデフォルトを使用します。
type = <文字列>
* スクリプトのタイプ: python, perl
* デフォルトは python。
filename = <文字列>
* コマンドのスクリプトファイルの名前。
* <stanza-name>.pl は perl 用。
* <stanza-name>.py は python 用。
streaming = <true/false>
* これはコマンドストリーミング可能です。
* デフォルトは false。
maxinputs = <整数>
* 各呼び出しで、コマンドに渡すことができるイベントの最大数。
* 0 は制限なし。
* デフォルトは 50000。
passauth = <true/false>
* true に設定すると、認証トークンを入力の際に渡します。
* デフォルトは false。
```

run_in_preview = <true/false>

- * 結果を最終出力ではなく閲覧用に生成する場合に、このコマンドを実行します。
- * デフォルトは true です。

enableheader = <true/false>

- * スクリプトがヘッダ情報を生成するかどうかを指示します。
- * 現在のところ、ヘッダ情報で可能なのは、認証トークンのみです。
- * true に設定すると、ヘッダセクション + '\n' の入力と、csv 入力を仮定します。
- * 注意： splunk.Intersplunk を使う場合には true に設定する必要があります。
- * デフォルトは true。

retainsevents = <true/false>

- * コマンドがイベントを保持するかどうかを指定します。
- * 例：sort/dedup/cluster。
- * または transform するか。

例：stats。

- * デフォルトは false。

generating = <true/false>

- * コマンドが新規イベントを生成するかどうかを指定します。
- * 例：イベントがコマンドに渡されない場合、イベントを生成するか？
- * デフォルトは false。

generates_timeorder = <true/false>

- * generating = true の場合、コマンドが時間の降順(最新が最初)でイベントを生成します。
- * デフォルトは false。

overrides_timeorder = <true/false>

- * generating = false の場合、コマンドが時間に関してイベントの順番を変更します。
- * デフォルトは true。

requires_preop

- * プリストリーミング操作が必要。
- * デフォルトは false。

streaming_preop = <文字列>

- * 要求されたプリストリーミング検索文字列を表示する文字列。

supports_multivalues = <true/false>

- * コマンドが複数をサポートするかどうかを指定します。true の場合、複数は、フラット文字列ではなく、python の文字列リストとして扱われます(Intersplunk を使用して stdin/stdout を解釈する場合)。

supports_getinfo = <true/false>

- * コマンドが、最初に呼び出された引数(= __GETINFO__ または __EXECUTE__)により設定のためのダイナミックプロローピングをサポートするかどうかを指定します。

commands.conf.example

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      外部検索コマンドの設定
#
#####
#      すべての外部コマンドのデフォルト。例外は、個々のスタンザに記載。
#      スクリプトのタイプ: 'python', 'perl'
TYPE = python
#
#      デフォルトのFILENAMEは、pythonでは<stanza-name>.py、perlでは<stanza-name>.pl、その他は<stanza-name>。
#      コマンドはストリーミング可能かどうかを指定します。
STREAMING = false
#      コマンドに渡す事ができる最大データ(0=制限なし)。
MAXINPUTS = 50000
#      デフォルトの最後
#####
[crawl]
FILENAME = crawl.py
[createrss]
FILENAME = creatorsss.py
[diff]
FILENAME = diff.py
[gentimes]
FILENAME = gentimes.py
[head]
FILENAME = head.py
[iplocation]
FILENAME = iplocation.py
[loglady]
FILENAME = loglady.py
[marklar]
FILENAME = marklar.py
[reportcache]
FILENAME = reportcache.py
[runshellscript]
FILENAME = runshellscript.py
[sendemail]
```

```
FILENAME = sendemail.py
[translate]
FILENAME = translate.py
[transpose]
FILENAME = transpose.py
[uniq]
FILENAME = uniq.py
[windbag]
filename = windbag.py
supports_multivalues = true
[xmlkv]
FILENAME = xmlkv.py
[xmlunescape]
FILENAME = xmlunescape.py
```

crawl.conf

crawl.conf

次は `crawl.conf` の仕様とファイル例です。

crawl.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、クローラ 1 の設定用の属性/値ペアを記載しています。
#
#      crawl.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、acrawl.conf を
#      $SPLUNK_HOME/etc/system/local に置いてください。ヘルプは、crawl.conf.example を参照のこと。設定を有効にするには
#      Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
#      http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
#      照してください。
#
#      crawl を用いて attribute-values を設定します。
#
#      属性がある場合、_list で終わります。形式は：
#
#      attr = val, val, val, etc.
```

```

#
#       カンマの後のスペースは必要です。これにより、BAD_FILE_PATTERNS の使用と同様、", "を使用することができます。
[default]
[files]
*       ファイルのクローラ別属性をこのスタンザヘッダのもとに設定します。
*       このスタンザに、次の属性を続けてください。
root = <ディレクトリのセミコロンで区切られたリスト>
*       このクローラが検索するディレクトリのリストを設定します。
*       デフォルトは /;/Library/Logs です。
bad_directories_list = <悪いディレクトリのカンマで区切られたリスト>
*       crawl を望まないディレクトリを記載します。
*       デフォルトは：
bin, sbin, boot, mnt, proc, tmp, temp, dev, initrd, help, driver, drivers, share,
bad_extensions_list = <スキップするファイル拡張子のカンマで区切られたリスト>
*       ファイル拡張子を記載すると、クローラはこれらの拡張子で終わるファイルをスキップします。
*       デフォルトは：
0t, a, adb, ads, ali, am, asa, asm, asp, au, bak, bas, bat, bmp, c, cache, cc,
bad_file_matches_list = <正規表現のカンマで区切られたリスト>
*       crawl は指定された正規表現を適用し、そのパターンと一致するファイルをスキップします。
*       各パターンの終わりに "$" (ファイル名の最後) が暗黙に付加されています。
*       デフォルトは：
*~, *#, *,v, *readme*, *install, (/|^).*, *passwd*, *example*, *makefile, core.*
packed_extensions_list = <拡張子のカンマで区切られたリスト>
*       実行する圧縮ファイルの拡張子を指定します。
*       デフォルトは：
bz, bz2, tbz, tbz2, Z, gz, tgz, tar, zip
collapse_threshold = <整数>
*       ソースがディレクトリとみなすべきファイルの最大数を指定します。
*       デフォルトは 1000。
days_sizek_pairs_list = <整数をハイフンでつないたペアで、カンマで区切られたもの>
*       クロールすべきファイルを制限するための、期限(日数)とサイズ(kb)のペアをカンマで区切ったリストを指定します。
*       例： days_sizek_pairs_list = 7-0, 30-1000 は、Splunk が、7 日以内に修正され、0kb 以上のサイズののファイ
ルをクロールし、また、30 日以内の修正で 1000kb のサイズのファイルををクロールします。
*       デフォルトは 30-0。
big_dir_filecount = <整数>
*       上記<整数>を上回るファイルがあるディレクトリをスキップします。
*       デフォルトは 10000。
index = <INDEX>

```

* クロールされたファイルを追加する先のインデックスを指定します。

* デフォルトは main。

max_badfiles_per_dir = <整数>

* クロールするファイルがあるディレクトリまでの距離を指定します。

* 指定された max_badfiles_per_dir 内に有効なファイルが見つからないとき、ディレクトリをクロールから除外します。

* デフォルトは 100。

[network]

* ネットワーククローラ別属性をこのスタンザヘッダのもとに設定します。

* このスタンザに、次の属性を続けてください。

host = <host または ip>

* ネットワークをクロールする際の開始点として使うデフォルトホスト。

* デフォルトは 'localhost'。

subnet = <整数>

* サブネットマスクで使用するデフォルトのビット数。 IP 123.123.123.123 のホストとすると、サブネット値が 32 の場合、そのホストのみをスキャンし、24 の場合は、123.123.123.* をスキャンします。

* デフォルトは 32。

crawl.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# 次は crawl.conf 設定の例です。クローラのプロパティを設定します。
```

```
#
```

```
# この設定の1つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の crawl.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
[files]
```

```
bad_directories_list= bin, sbin, boot, mnt, proc, tmp, temp, home, mail, .thumbnails, cache, old
```

```
bad_extensions_list= mp3, mpg, jpeg, jpg, m4, mcp, mid
```

```
bad_file_matches_list= *example*, *makefile, core.*
```

```
packed_extensions_list= gz, tgz, tar, zip
```

```
collapse_threshold= 10
```

```
days_sizek_pairs_list= 3-0,7-1000, 30-10000
```

```
big_dir_filecount= 100
```

```
index=main
```

```
max_badfiles_per_dir=100
```

```
[network]
host = myserver
subnet = 24
```

deploymentclient.conf

deploymentclient.conf

次は、deploymentclient.conf の仕様とファイル例です。

deploymentclient.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、デプロイメントサーバーからコンテンツ(アプリケーションおよび設定)を受信するためのデプロイメント
クライアントの設定で使用する属性と値を記載しています。
#
# デプロイメントクライアントの動作をカスタマイズするには、deploymentclient.conf を、その Splunk インスタンス
上の$SPLUNK_HOME/etc/system/local に起きます。serverclass.conf でデプロイメントクライアントに展開するアプリケーシ
ョン/設定コンテンツを設定します。詳細は、serverclass.conf.spec および serverclass.conf.example を参照してください。
#
# この設定ファイルの変更を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#*****
# Splunk デプロイメントクライアントの設定
#
# 注意： 少なくとも、[deployment-client]スタanzasは、有効にするデプロイメントクライアントの
deploymentclient.conf に必要です。
#*****
[deployment-client]
disabled = <false または true>
* デフォルトは false。
* デプロイメントクライアントを有効/無効にします。
clientName=deploymentClient
* また、'tag'ともいう。デフォルトは'deploymentClient'。
* デプロイメントサーバーがフィルタリングに利用することができる名前であり、hostnames に優先します。
workingDir=$SPLUNK_HOME/var/run/deploy-client
```

* アプリケーションおよび設定コンテンツをダウンロードするために deploymentClient が使用する一時フォルダ。

```
repositoryLocation = $SPLUNK_HOME/etc/apps
```

* デプロイメントサーバーからダウンロードされた後にコンテンツがインストールされる場所。

* アプリケーションおよび設定コンテンツは、デフォルト位置(\$SPLUNK_HOME/etc/apps)にインストールする必要ありません。さもないと、デプロイメントクライアント上の Splunk インスタンスに認識されません。

* 注意： 展開されるアプリケーションおよび設定コンテンツは、デプロイメントサーバーの別の場所に置くことができます。それにより、コンテンツがデプロイメントクライアント上の正しい位置(\$SPLUNK_HOME/etc/apps)にインストールされたことを確実にするために、repositoryLocation および serverRepositoryLocationPolicy を設定することができます。

* デプロイメントクライアントは、下記に定義する 'serverRepositoryLocationPolicy' を使い、repositoryLocation のどの値を使用するかを決定します。

```
serverRepositoryLocationPolicy = acceptSplunkHome, acceptAlways, rejectAlways のいずれか>
```

* デフォルトは acceptSplunkHome。

* acceptSplunkHome - deploymentServer が供給した repositoryLocation を受け入れます。ただし、\$SPLUNK_HOME によりルーティングされた場合のみ。

* acceptAlways - deploymentServer が供給した repositoryLocation を常に受け入れます。

* rejectAlways - サーバーが供給した値を拒否し、ローカル deploymentClient.conf で指定した repositoryLocation のみを使用します。

```
endpoint=$deploymentServerUri$/services/streams/deployment?name=$serviceName:$appName$
```

* コンテンツのダウンロード元の HTTP エンドポイント。

* 注意： デプロイメントサーバーは、コンテンツの各セットのダウンロード元の異なるエンドポイントを指定する場合があります(個別のアプリケーションなど)。

* デプロイメントクライアントは下記で定義する 'serverEndpointPolicy' を使用して、使用する値を決定します。

* \$deploymentServerUri\$は、下記の 'target-broker' スタンザで定義される 'targetUri' を決定します。

* \$serviceName\$と \$appName\$は、それぞれが何を提供するかを意味します。

```
serverEndpointPolicy = < acceptAlways, rejectAlways のいずれか>
```

* デフォルトは acceptAlways。

* acceptAlways - サーバーが提供するエンドポイントを常に受け入れます。

* rejectAlways - サーバーが提供するエンドポイントを拒絶します。常に 'endpoint' 定義を使用してください。

```
phoneHomeIntervalInSecs = <N>
```

* デフォルトは 30。

* これは、このデプロイメントクライアントが新規コンテンツをチェックする頻度を決定します。

上級者向け！

階層 DS インストールの場合にのみこのプロパティを使用し、deploymentClient と deploymentServer の両方として動作する Splunk インスタンスを持たせます。

```
reloadDSOnAppInstall = <false または true >
```

* デフォルトは false。

* このフラグを true に設定すると、この Splunk 上の deploymentServer は、アプリケーションがこの deploymentClient によってインストールされるたびに再ロードされます。

次のスタanzasはデプロイメントサーバー接続情報を指定します。

```
[target-broker:deploymentServer]
targetUri= <deploymentServer>:<mgmtPort>
```

* デプロイメントサーバーの URI。

deploymentclient.conf.example

例 1

デプロイメントクライアントはアプリケーションを受信し、受信ごとにそれを同じ repositoryLocation(ローカルで、\$SPLUNK_HOME に相対)に置きます。これは一般的に \$SPLUNK_HOME/etc/apps です。

[deployment-client] には何もありません。なぜなら、デプロイメントクライアントは、デプロイメントサーバー側で設定された値を上書きしないからです。

```
[deployment-client]
[target-broker:deploymentServer]
targetUri= deploymentserver.splunk.mycompany.com:8089
```

例 2

デプロイメントサーバーは、アプリケーションを、サーバー側の標準でない位置に展開し続けます(おそらく組織の目的上)。

デプロイメントクライアントはアプリケーションを受信し、それを標準位置に置きます。

注意: デプロイメントクライアント側の \$SPLUNK_HOME/etc/apps 以外の場所に展開されたアプリケーションは認識されず実行されません。

この設定は、デプロイメントサーバーが指定した位置を拒否し、それを標準クライアント側位置で置き換えます。

```
[deployment-client]
serverRepositoryLocationPolicy = rejectAlways
repositoryLocation = $SPLUNK_HOME/etc/apps
[target-broker:deploymentServer]
targetUri= deploymentserver.splunk.mycompany.com:8089
```

例 3

デプロイメントクライアントは、デプロイメントサーバーが指定したサーバーと異なる HTTP サーバーからアプリケーションを取得する必要があります。

```
[deployment-client]
serverEndpointPolicy = rejectAlways
endpoint = http://apache.mycompany.server:8080/$serviceName/$appName$.tar
[target-broker:deploymentServer]
targetUri= deploymentserver.splunk.mycompany.com:8089
```

例 4

デプロイメントクライアントは、ファイルシステム上の位置からアプリケーションを取得する必要があり、デプロイメントサーバーが指定した位置から取得しません。

```
[deployment-client]
serverEndpointPolicy = rejectAlways
```

```
endpoint = file://<some_mount_point>/$serviceClassName$/AppName$.tar
[target-broker:deploymentServer]
targetUri= deploymentserver.splunk.mycompany.com:8089
```

distsearch.conf

distsearch.conf

次は distsearch.conf の仕様とファイルの例です。

distsearch.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、分散検索の設定に使用可能な属性と値を記載しています。
#
# デフォルトの distsearch.conf はありません。
#
# カスタム設定を設定するには、distsearch.conf を $SPLUNK_HOME/etc/system/local に置いてください。
# 例は、distsearch.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[distributedSearch]
* 分散検索設定オプションをこのスタンザ名で設定します。
* このスタンザ名に、次の属性/値のペアを続けてください。
* 属性を何も設定しない場合、Splunk はデフォルト値を使用します(リストにある場合)。
disabled = true | false
* 分散検索のオン/オフを切り替えます。
* デフォルトは false(分散検索スタンザはデフォルトで有効です)。
heartbeatFrequency = <秒>
* ハートビートを秒で指定。
* 0 はすべてのハートビートを無効にします。
* ハートビートが無効の場合、他の Splunk サーバーもこのインスタンスを自動発見できません。
* デフォルトは 0。
heartbeatMcastAddr = <IP アドレス>
* マルチキャストアドレスを設定。
* デフォルトは 255.0.0.37。
heartbeatPort = <ポート>
```

- * ハートビートポートを設定します。
- * デフォルトは 60。

```
serverTimeout = <秒>
```

- * サーバーとの接続待ち時間。
- * 接続された場合、検索はこの値の 10 倍でタイムアウトします。
- * 例えば、10 秒に設定する場合、許される最大検索は 100 秒になります。
- * この設定は 'removeTimedOutPeers' と平行して動作します。
- * デフォルトは 10。

```
statusTimeout = <秒>
```

- * サーバーがステータスを返す待ち時間。
- * ピアされたサーバーが遅い場合、または、サーバー名が Splunk Web から消えた場合、この数値を上げます。

```
removedTimedOutServers = true | false
```

- * true の場合、'serverTimeout' 内で確立できないサーバー接続を削除します。
- * false の場合、そのサーバーに対するすべての呼び出しが接続を試みます。
- * 注意： これにより、ユーザーインターフェースが遅くなることがあります。

```
checkTimedOutServersFrequency = <秒>
```

- * このタグは、'removeTimedOutServers' が true に設定されている場合にのみ意味があります。
- * 'removeTimedOutServers' が false の場合、この属性は無視されます。
- * この頻度(秒)でサーバーを再チェックします。
- * これが 0 に設定されていると、再チェックされません。
- * デフォルトは 60。

```
autoAddServers = true | false
```

- * このタグが 'true' に設定されている場合、このノードは自動的に発見されたすべてのサーバーを追加します。
- * デフォルトは false。

```
skipOurselves = true | false
```

- * これが 'true' に設定されている場合、このサーバーはサーバーとしてその検索その他の呼び出しにも参加しません。
- * これは、他のサーバーからの結果をマージする以外にはなにも行わないノードを構築するのに使用されます。
- * デフォルトは false。

```
tTl = <整数>
```

- * 有効期間。
- * この数値を増やすと、UDP マルチキャストパケットが現在のサブネットを超え、指定されたホップ数まで広がります。
- * 注意： これは、すべてのルーターが UDP マルチキャストパケットを通過させるように設定されている場合にのみ有効となります。
- * デフォルトは 1(このサブネット)

```
servers = <サーバーのカンマで区切られたリスト>
```

- * サーバの初期リスト。
- * 'autoAddServers' モードで完全に動作している場合(すべてのサーバーを検出)、サーバーをここに入力する必要はありません。

```

blacklistNames = <サーバー名のカンマで区切られたリスト>
*     ピアリングしたくないサーバー名のリスト。
*     サーバー名は、スタートアップ時に作成された'server name'です。
blacklistURLs = <サーバ名または URI のカンマで区切られたリスト>
*     ブラックリストに載せるサーバーを指定します。
*     サーバー名(上記)またはサーバーURI(x.x.x.x:port)上でブラックリストに載せることができます。
shareBundles = true | false
*     このサーバーがそのアプリケーションをピアと共有するかどうかを指示します。
*     このフラグは検索分散が必要です。
*     デフォルトは true。
#*****
#     複製設定オプション
#     これらのオプションは[replicationSettings]エントリーで設定することができます。
#*****
connectionTimeout = <数>
*     ピアへの最初の接続がタイムアウトになるまでの待ち時間の最大値(秒)
*sendRcvTimeout = <数>
*     完全複製をピアに送信するための待ち時間の最大値(秒)
#*****
#     複製ホワイトリストオプション
#     これらのオプションは[replicationWhitelist]エントリーで設定することができます。
#*****
<whitelist_regex> = <enabled|disabled>
*     複製の候補ファイルと一致する場合($SPLUNK_HOME/etc)、そのファイルが複製されるパターン。
*     これらは複数持つことができ、enabled/disabled によりそれらを有効化および無効化できます。
*     注意： ワイルドカードと複製：
*     ワイルドカードを使用して複製ファイルのパスを指定できます。 . . . をパス、* をファイルに使用します。
*     一致するまでディレクトリを再帰的に読み込みます。つまり、/foo/.../bar は、foo/bar, foo/1/bar, foo/1/2/bar
などにマッチします。ただし、bar がファイルの場合です。
*     サブディレクトリも再帰的に読み込むには、別の . . . を使用します。例えば、/foo/.../bar/...は、その特定のパス
セグメントにあるすべてにマッチします。これはディレクトリパスの内側では使用できません。これはパスの最後のセグメントで使用
する必要があります。例えば、/foo/*.log は/foo/bar.log にマッチしますが、/foo/bar.txt や/foo/bar/test.log にはマ
ッチしません。
*     より特定のなマッチには、*と...を混ぜて使います。
*     foo/.../bar/*は、指定したパス内のパーディレクトリ内のファイルにマッチします。

```

distsearch.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# これは distsearch.conf の例です。このファイルを分散検索の設定に使用してください。利用可能なすべての属性/値ベ
アは、distsearch.conf.spec を参照してください。
#
# デフォルトの distsearch.conf はありません。
#
# この設定の1つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の distsearch.conf
にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[distributedSearch]
heartbeatFrequency = 10
servers = 192.168.1.1:8059,192.168.1.2:8059
blacklistNames = the-others,them
blacklistURLs = 192.168.1.3:8059,192.168.1.4:8059
# このエントリは検索を 192.168.1.1:8059,192.168.1.2:8059 に分散します。
# サーバーはハートビートを 10 秒毎に送信します。
# 4 つのブラックリストに載せられたインスタンスがあり、blacklistNames および blacklistURLs に渡ってリストに記載
されます。
# 属性はここでは設定されません。distsearch.conf.spec にリストアップされたデフォルトを使用します。
# このスタanzas は、リモートピアへの接続と送信タイムアウトのタイミング設定をコントロールします。
[replicationSettings]
connectionTimeout = 10
sendRcvTimeout = 60
# このスタanzas は、そのファイルを他のピアに複製するかをコントロールします。それぞれ正規表現です。
[replicationWhitelist]
.*app\.conf$ = enabled
.*authorize\.conf$ = enabled
.*commands\.conf$ = enabled
.*eventtypes\.conf$ = enabled
.*literals\.conf$ = enabled
.*multikv\.conf$ = enabled
.*props\.conf$ = enabled
.*tags\.conf$ = enabled
```

```

.*transforms\conf$ = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@system@OsDirSep@local@OsDirSep@.*\conf$ = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@system@OsDirSep@default@OsDirSep@.*\conf = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@apps@OsDirSep@.*@OsDirSep@local@OsDirSep@.*conf = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@apps@OsDirSep@.*@OsDirSep@default@OsDirSep@.*\conf = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@system@OsDirSep@bin@OsDirSep@.* = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@system@OsDirSep@lookups@OsDirSep@.* = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@apps@OsDirSep@.*@OsDirSep@lookups@OsDirSep@.* = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@apps@OsDirSep@.*@OsDirSep@bin@OsDirSep@.* = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@system@OsDirSep@metadata@OsDirSep@.*\meta = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@apps@OsDirSep@.*@OsDirSep@metadata@OsDirSep@.*\meta = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@searchscripts@OsDirSep@.* = enabled
$SPLUNK_HOME@OsDirSep@etc@OsDirSep@users@OsDirSep@.* = enabled

```

eventdiscoverer.conf

eventdiscoverer.conf

次は eventdiscoverer.conf の仕様とファイル例です。

eventdiscoverer.conf.spec

```

# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
# このファイルは、イベント発見を検索コマンド"typelearner"で設定するために使用可能な属性と値を記載しています。
#
# eventdiscoverer.conf は$SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、
eventdiscoverer.conf を$SPLUNK_HOME/etc/system/local に置いてください。例は、eventdiscoverer.conf.example
を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWorkにあるドキュメントを参
照してください。
ignored_keywords = <用語のカンマで区切られたリスト>
* このリストの用語は、イベントタイプの定義には決して考慮されません。
* eventtypes が考慮したくない用語を含んでいることがわかった場合("mylaptopname"など)、その用語はこのリストに
記載されます。
* Default = "sun, mon, tue,..." ($SPLUNK_HOME/etc/system/default/eventdiscover.conf を参照)。
ignored_fields = <フィールドのカンマで区切られたリスト>
* ignored_keywords と同様、Splunk で定義されたフィールドを除外します。
* デフォルトには、イベントタイプの定義には便利ではない時間関連フィールドを含んでいます。

```

important_keywords = <用語のカンマで区切られたリスト>

* eventtype 検索を生成で複数の語句がある場合、important_keyword 語を含む語句が優先されます。例えば、"fatal error"は"last message repeated"よりも好まれます。なぜなら、"fatal"は重要な用語だからです。

* Default = "abort, abstract, accept, ..." (\$SPLUNK_HOME/etc/system/default/ eventdiscover を参照)。

eventdiscover.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

これは eventdiscover.conf の例です。これらの設定は、typelearner 検索コマンドが使用する共通 eventtypes の発見のコントロールに使用されます。

```
#
```

この設定の1つまたは複数を使用するには、その設定ブロックを \$SPLUNK_HOME/etc/system/local の eventdiscover.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。

```
#
```

設定ファイル(優先順位を含む)についての詳細は、

<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork> にあるドキュメントを参照してください。

このリストにある語は、イベントタイプの定義には決して考慮されません。

```
ignored_keywords = foo, bar, application, kate, charlie
```

このリストにあるフィールドは、イベントタイプの定義には決して考慮されません。

```
ignored_fields = pid, others, directory
```

eventtypes.conf

eventtypes.conf

次は eventtypes.conf の仕様とファイルの例です。

eventtypes.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

このファイルは、eventtypes.conf ファイルのすべての属性と値ペアを記載しています。

このファイルを使用してイベントタイプとそのプロパティを設定してください。また検索を typelearner コマンドにパイプして、イベントタイプを作成することもできます。このようにして作成されたイベントタイプは、

\$SPLUNK_HOME/etc/systems/local/eventtypes.conf に書き込まれます。

```
#
```

eventtypes.conf は \$SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、

eventtypes.conf を \$SPLUNK_HOME/etc/system/local に置いてください。例は、eventtypes.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。

```
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
```

```
[$EVENTTYPE]
```

- * イベントタイプのヘッダ。
- * \$EVENTTYPE はイベントタイプの名前。
- * イベントタイプはいくつでも持つことができます。それぞれが 1 つのスタンザまたは次の属性/値ペアで表されます。
- * 注意： イベントタイプの名前がパーセント文字で囲まれたフィールド名を持つ場合、\$FIELD の値はそのイベントのイベン
ト名に置き換えられます。例えば、[cisco-%code%]というヘッダを持つイベントタイプでは、"code=432"は"cisco-432"でラベ
ルされます。

```
disabled = <1 または 0>
```

- * トグルイベントタイプのオンオフ。
- * 0 に設定すると無効。

```
search = <文字列>
```

- * このイベントタイプの検索語
- * 例： error OR warn.

```
tags = <文字列>
```

- * イベントタイプのタグ付けに使用するスペースで区切られた用語

eventtypes.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
# このファイルは eventtypes.conf の例を記載しています。このファイルを使用して、カスタム eventtypes を設定して
ください。
```

```
#
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の eventtypes.conf
にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
```

```
#
# 次の例では、検索"error OR fatal"に基づいて、"error"という eventtype を作成します。
```

```
[error]
```

```
search = error OR fatal
```

```
# 次の例は、eventtype テンプレートを作成します。なぜなら、パーセント文字で囲まれたフィールド名を含むからです(こ
の場合、"%code%")。
```

```
# "%code%"の値は、そのイベントのイベントタイプ名に置き換えられます。
# 例えば、次の例のイベントタイプが"code=432"をもつイベントでインスタンス化される場合、それは"cisco-432"になります。
[cisco-%code%]
search = cisco
```

fields.conf

fields.conf

次は fields.conf の仕様とファイル例です。

fields.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、ダイナミックフィールド抽出を作成するための属性と値のペアを記載しています。
# このファイルは、フィールドの追加情報を設定するための属性と値のペアを記載しています。
# インデックス時間にフィールドを作成する場合は、このファイルを使用してください(推奨しない)。
# また、設定したフィールドの値がトークンの一部になっているかどうかを示すために、このファイルを使用してください。
# 例えば、フィールド値が"123"であるが、イベントでは"foo123"となる場合。
# fields.conf を次のように設定します：
# * Splunk に、複数値フィールドの扱い方を伝える。
# * インデックス化されたフィールドを抽出されたフィールドを区別する。
# * 検索プロセッサにフィールド値の扱い方を伝えることにより、検索性能が改善する。
#
# fields.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、fields.conf を
# $SPLUNK_HOME/etc/system/local に置いてください。例は、fields.conf.example を参照してください。
# 設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
[<field name>]
* 設定しているフィールドの名前。
* このスタンザ名に、次の属性/値のペアを続けてください。
TOKENIZER = <正規表現>
* このフィールドに同時に複数の値を取り込む方法を示す正規表現。
* この設定を使用して複数値フィールドを設定ます
(http://www.splunk.com/doc/current/admin/MultivalueFields)。
```

- * 空白の場合、フィールドは 1 つの値のみを取得します。
- * それ以外の場合、最初のグループは各一致から取得され、値セットを形成します。
- * この設定は、`search/where` (検索コマンド)、同期検索 API のサマリーおよび XML 出力、および `トップ`、`時間軸` および `stats` コマンドにより使用されます。

* デフォルトは空白です。

INDEXED = true | false

- * フィールドをインデックス化するかどうかを指示します。
- * フィールドがインデックス化される場合、`true` に設定します。
- * `false` に設定すると、フィールドが検索時に抽出されます (フィールドのほとんど)。
- * デフォルトは `false`。

INDEXED_VALUE = true | false

- * 値がイベントのローテキストにある場合、`{{indexed_value}}` を `true` に設定します。
- * 値がイベントのローテキストにない場合、`false` に設定します。
- * これを `true` に設定すると、`key=value` の検索を `value AND key=value` の検索に拡張します (値がインデックス化されるため)。
- * デフォルトは `true`。
- * 注意: `indexed = false` の場合にみ、`indexed_value` を設定する必要があります。

fields.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# このファイルは fields.conf の例を記載します。このファイルを使用してダイナミックフィールド抽出を設定します。
```

```
#
```

```
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の fields.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
# 設定ファイル (優先順位を含む) についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
#
```

```
# 次の例は IMAP バンドルで使用することができます (参照 :
```

```
http://www.splunkbase.com/addons/All/Technologies/Mail/addon:IMAP+Addon)
```

```
# これらのトークナイザは、To, From および Cc の値をリストにします。ここで、各リスト要素は、データのロー文字列にあるメールアドレスです。
```

```
[To]
```

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

```
[From]
```

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

[Cc]

TOKENIZER = (\w[\w.\-]*@[\w.\-]*\w)

indexes.conf

indexes.conf

次は indexes.conf の仕様とファイル例です。

indexes.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、indexes.conf ファイルのすべてのオプションを記載します。このファイルを使用して Splunk のインデックスとそのプロパティを設定してください。
#
# indexes.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、indexes.conf を $SPLUNK_HOME/etc/system/local に置いてください。例は、indexes.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
#
# 警告：これらの設定により Splunk インストールに大きな影響を与えることができます。
# このファイルの設定方法が確かでない場合は、テクニカルサポート (http://www.splunk.com/page/submit_issue) に相談してください。
#
# テクニカルサポートに相談することなく、QueryLanguageDefinition の属性を変更しないでください。
#*****
# グローバルオプション
# これらのオプションはすべてのインデックスに影響します。
#*****
sync = <整数>
* インデックスプロセッサは、イベントの<整数>毎にイベントに同期します。
* マイナスは設定できません。
* 0 に設定すると無効。
* デフォルトは 0。
defaultDatabase = <データベース名>
* 検索でインデックスが指定されていない場合、Splunk はデフォルトデータベースを検索します。
```

* また、データベースはデフォルトでホームページに表示されます。

* デフォルトは main。

queryLanguageDefinition = <ファイルへのパス>

* 検索言語定義ファイルへのパス。

* この設定は編集しないでください。

* デフォルトは \$SPLUNK_HOME/etc/searchLanguage.xml。

blockSignatureDatabase = <データベース名>

* これは、イベントのブロック署名を保存するデータベースです。

* デフォルトは _blocksignature。

memPoolMB = <数または "auto">

* "auto" または無効な値を指定すると、Splunk はシステムに基づいてこのパラメータを自動調整します。

* インデックスマシンメモリプールに与えられるメモリ量は、指定の時間でインデックスマシンにある残りのイベントの数を制限します。

* 0 ~ 1048576 (1TB) で指定します。

* 高く設定しすぎると、splunkd のメモリ使用料が莫大になります。

* 低く設定しすぎると、splunkd インデキシング性能が低下します。

* 十分な知識がある、または Splunk サポートのアドバイスを得不限り、この値は設定しないでください。

* この設定を誤ると、永久的な傷害または業務の損害をもたらす恐れがあります。

indexThreads = <数または "auto">

* "auto" または無効な値を指定すると、Splunk はシステムに基づいてこのパラメータを自動調整します。

* インデキシングに使用するスレッド数。

* この数は、1 ~ 16 以下で指定します。

* 0 未満または 16 を超える数を設定すると、自動調整されます。

* この数は、ボックス内のプロセッサ数より大きい数を設定してはいけません。

* splunkd がまた構文解析および合成を行う場合、その数全プロセッサ数マイナス 2 未満にする必要があります。

* 十分な知識がある、または Splunk サポートのアドバイスを得不限り、この値は設定しないでください。

* この設定を誤ると、永久的な傷害または業務の損害をもたらす恐れがあります。

インデックス毎のオプション

これらのオプションは [\$INDEX] エントリーで設定することができます。

disabled = true | false

* インデックスエントリーのオン/オフを切り替えます。

* true に設定するとインデックスを無効にします。

* デフォルトは false。

homePath = <サーバ上のパス>

* ホットおよびウォームデータベース、およびインデックスのフィールドを含むパス。

* Splunkd は、ウォームデータベースについては常にファイルハンドルをオープンに維持します。@

* 警告：パスは書き込み可能でなければなりません。

coldPath = <サーバ上のパス>

* インデックスのコールドデータベースを含むパス。

* コールドデータベースは、検索時に必要に応じて開かれます。

* 警告：パスは書き込み可能でなければなりません。

thawedPath = <サーバ上のパス>

* インデックスの thawed(復活)データベースを含むパス。

次のオプションは、インデックス毎または、デフォルトで全インデックスのファイルの先頭に設定することができます。

ファイル先頭のデフォルト設定は、インデックス毎が設定されると上書きされます。

maxWarmDBCount = <整数>

* ウォーム DB_N_N_N ディレクトリの最大数。

* 全ウォーム DB はインデックスの <homePath> にあります。

* ウォーム DB はオープン状態に維持されます。

* デフォルトは 300。

maxColdDBCount = <整数>

* 任意の時間におけるオープンコールドデータベースの最大数。

* これはコールドデータベースの総数ではありません。

* 検索の際、Splunkd はすべてのオープンコールド DB の LRU キャッシュを維持します。この数は、そのキャッシュのサイズをコントロールします。

* デフォルトは 10。

maxTotalDataSizeMB = <整数>

* インデックスの最大サイズ(MB)

* インデックスが大きくなると、最も古いデータが圧縮されます。

* デフォルトは 500000。

rotatePeriodInSecs = <整数>

* 新規ホット DB の作成が必要かどうかをチェックする頻度(秒)。

* また、圧縮する必要があるコールド DB があるかどうかをチェックする頻度。

* デフォルトは 60。

frozenTimePeriodInSecs = <整数>

* インデックス化されたデータが圧縮された後の秒数。

* coldToFrozenScript を指定しない場合、このデータは消去されます。

* 重要：DB でのすべてのイベントは、それがロールされる前までは、frozenTimePeriodInSecs より古くなければなりません。

* frozenTimePeriodInSecs は、Splunkd がチェックする次の時間を凍結します。

* デフォルトは 188697600。

warmToColdScript = <スクリプト>

* データをウォームからコールドに移動する際に実行するスクリプトを指定します。

* スクリプトは 2 つの変数を受け入れる必要があります。

- * 第一の変数： コールドに割り当てられるウォームディレクトリ。
- * 第 2 の変数： コールドパスのあて先。
- * ウォームおよびコールド DB を別のパーティションに保存する場合のみ、これを設定する必要があります。
- * この設定の設定についての質問は、Splunk サポートに連絡してください。
- * デフォルトは空白です。

coldToFrozenScript = <スクリプト>

- * アーカイブスクリプトを<スクリプト>を変更することにより指定します。
- * Splunk には 2 つのデフォルトアーカイブスクリプトが標準装備されています(または自身のスクリプトを作成)。
- * compressedExport.sh - tsidx files を gz で圧縮してエクスポート。
- * flatfileExport.sh - フラットテキストファイルでエクスポート。
- * <\$script>パスを\$SPLUNK_HOME/bin に対して定義する。
- * Windows のユーザーはこの表記法を使用する：

coldToFrozenScript = <スクリプト> "\$DIR"

- * <スクリプト>は compressedExport.bat または flatfileExport.bat のどちらでもよい。

compressRawdata = true | false

- * true に設定すると、Splunk はローデータを圧縮 gz ファイルで書き出します。
- * false に設定すると、Splunk はデータを非圧縮ローファイルに書き出します。
- * デフォルトは true。

maxConcurrentOptimizes = <整数>

- * ホット DB に対して実行することができる並列最適化プロセスの数。
- * この数は、次の場合増やす必要があります：

1. ホット DB に常に小さな tsidx ファイルがたくさんある場合。
2. ローリングの後、ウォームまたはコールド DB に tsidx ファイルがたくさんある場合。

maxDataSize = <整数または "auto">

- * ウォームへのロールがトリガされる前に、ホット DB が成長する最大サイズ(MB)。
- * "auto" または "auto_high_volume" を指定すると、Splunk はシステムに基づいてこのパラメータを自動調整します(推奨)。
- * 大量のインデックス(メインインデックスなど)については "auto_high_volume" を使用する必要があります。他の場合には、"auto" を使います。
- * 設定できる最大値は 1048576MB(1TB) ですが、実用的には 100 - 50000 の間です。
- * この範囲を超える数値は、Splunk サポートの承認が必要です。
- * maxDataSize に不正な数値または文字列を指定すると、maxDataSize は自動調整されます。
- * 注意： ウォームバケツの正確なサイズは、maxDataSize とは異なる場合があります。これは、割り当て基準について、ポストプロセッシングとタイミングの問題があるからです。

maxMemMB = <整数>

- * インデキシングに割り当てるメモリの量。
- * このメモリ量は PER INDEX THREAD で割り当てられます。
- * または、indexThreads が 0 に設定される場合、インデックス当たり 1 回です。

- * 重要： この数は注意して計算する必要があります。
- * Splunkd は、利用できる数よりもこれを高く設定するとクラッシュします。
- * デフォルトは 50。

blockSignSize = <整数>

- * ブロック署名で、1 つのブロックを形成するイベントの数をコントロールします。
- * 0 に設定する場合、署名はこのインデックスで無効になります。
- * デフォルトは 0。

maxHotSpanSecs = <マイナスでない数>

- * ホット/ウォームバケツの目標最大タイムスパンの上限を秒で指定します。
- * デフォルトは 90 日。
- * 注意： この設定が小さすぎると、ファイルシステムでホット/ウォームバケツが爆発する可能性があります。システムはこのパラメータの下限をを暗黙に 3600 に設定します。ただし、これは高度なパラメータであり、注意して設定する必要があり、データの特性をよく理解して行う必要があります。

maxHotIdleSecs = <マイナスでない数>

- * ホットバケツの有効期限の上限を秒で指定します。
- * この時間が過ぎると、ホットバケツはウォームバケツに“ロール”されます。
- * 0 に設定すると、アイドルチェックにします (INFINITE アイドル時間に相当)。
- * デフォルトは 0。

maxHotBuckets = <マイナスでない数>

- * インデックス毎に存在可能な最大ホットバケツ。
- * LRU ポリシーは、この数値を超えると、ホットバケツを期限切れにするときに使用します。
- * デフォルトは 1。

quarantinePastSecs = <マイナスでない数>

- * "now" よりも古い quarantinePastSecs のタイムスタンプを持つイベントは検疫バケツにドロップされます。
- * デフォルトは 157680000 (5 年)。
- * これは、メインホットバケツが小さなイベントで汚染されることを防止するためのしくみです。

quarantineFutureSecs = <マイナスでない数>

- * "now" よりも新しい quarantineFutureSecs のタイムスタンプを持つイベントは検疫バケツにドロップされます。
- * デフォルトは 2592000 (1 ヶ月)。
- * これは、メインホットバケツが小さなイベントで汚染されることを防止するためのしくみです。

indexes.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは indexes.conf の例を記載します このファイルを使って、インデキシングプロパティを設定します。
#
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の indexes.conf に
コピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```

#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#
#       次の例は、"hatch"という新規デフォルトインデックスを設定します。
defaultDatabase = hatch

[hatch]
homePath = $SPLUNK_DB@OsDirSep@hatchdb@OsDirSep@db
coldPath = $SPLUNK_DB@OsDirSep@hatchdb@OsDirSep@colddb
thawedPath = $SPLUNK_DB@OsDirSep@hatchdb@OsDirSep@thaweddb
indexThreads = 1
#       所与のインデックスに使用する物理メモリの最大量(メガバイト)。
maxMemMB = 200
maxDataSize = 10000
#       次の例は、Splunk インデックスが使用するデフォルトの容量およびメモリを変更します。
maxTotalDataSizeMB = 650000
maxMemMB = 75
#       次の例は、デフォルトで維持されている時間データを変更します。
#       また、エクスポートスクリプトを設定します。注意：このスクリプトは、実行する前に、エクスポート位置を設定するた
めに編集する必要があります。
maxWarmDBCount = 200
maxColdDBCount = 5
frozenTimePeriodInSecs = 432000
rotatePeriodInSecs = 30
coldToFrozenScript = /opt/bin/compressedExport.sh

```

inputs.conf

inputs.conf

次は inputs.conf の仕様とファイル例です。

inputs.conf.spec

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#       このファイルは、inputs.conf で、入力、分散入力およびファイルシステムモニタリングの設定に使用可能な属性と値を
記載します。
#
#       inputs.conf は $SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、inputs.conf を

```

```

$SPLUNK_HOME/etc/system/local に置いてください。例は、inputs.conf.example を参照してください。
#     新しい設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#
#*****
#     一般設定
#     次の属性/値ペアは、すべての入力タイプで有効です(ファイルシステム変更モニターを除く)。
#     まず、スタンザヘッダを入力して入力タイプを指定する必要があります。
#     次に、次の属性/値ペアを使います。
#*****
host = <文字列>
*     デフォルトホストを固定値に設定します。
*     "host=" は自動的に<文字列>の先頭に追加されます。
*     デフォルトは、元データがあるホストの完全修飾ドメイン名の IP アドレス。
index = <文字列>
*     イベントをこの入力で保存するためのインデックスを設定します。
*     " index =" は自動的に<文字列>の先頭に追加されます。
*     デフォルトは "index=main"(または、デフォルトインデックスで設定したもの)。
source = <文字列>
*     この入力からのイベントのソースを設定。
*     " source =" は自動的に<文字列>の先頭に追加されます。
*     デフォルトはファイルパス。
sourcetype = <文字列>
*     この入力のイベントのソースタイプを設定。
*     " sourcetype =" は自動的に<文字列>の先頭に追加されます。
*     Splunk は自動的にソースタイプをデータのさまざまなアスペクトに基づいて取り上げます。queue = parsingQueue |
indexQueue はありません。
*     入力プロセッサが、それら読み込むイベントを保存する場所を指定します。
*     props.conf を適用し、他の構文解析ルールをデータに適用するためには、"parsingQueue"に設定します。
*     データを直接インデックスに送信するには、 "indexQueue"に設定します。
*     デフォルトは parsingQueue。
#*****
#     有効な入力タイプが、リストアップされた入力別属性とともに続きます：
#*****
#*****

```

```
#      監視：
#*****

[monitor://<パス>]
*      これは Splunk が、<パス>にあるすべてのファイルを監視するように命令します。
*      <パス>には、ディレクトリ(全体)または、単一ファイルが指定可能です。
*      入力タイプを指定し、次にパスを指定する必要があります。したがって、ルートから開始する場合、3つのスラッシュをパスに入れます。

host_regex = <正規表現>
*      これを指定すると、<正規表現>はホストを各入力のファイル名から抽出します。
*      特に、正規表現の最初のグループはホストとして使用されます。
*      正規表現が一致しない場合、デフォルト"host ="属性が使用されます。

host_segment = <整数>
*      これを指定すると、パスの '/' で区切られたセグメントがホストとして設定されます。
*      この値が整数でない場合、または 1 より小さい場合、デフォルト"host ="属性が使用されます。

_whitelist = <正規表現>
*      これが設定されると、このパスからのファイルが、それらが指定された正規表現に一致した場合にのみ監視されます。

_blacklist = <正規表現>
*      これが設定されると、このパスからのファイルは、それらが指定された正規表現に一致した場合に、監視されません。

注意： ワイルドカードと監視：
*      ワイルドカードを使用して、監視される入力の入力パスを指定することができます。...をパスに使用すると、...は一致するまでディレクトリを再帰的に読み込みます。つまり、/foo/.../bar が一致します。
*      サブディレクトリも再帰的に読み込むには、別の...を使用します。例えば、/foo/.../bar/....。
*      *は、その特定のパスセグメントですべてと一致します。ディレクトリパスの中で使用することはできません。
*      より具体的なマッチには、*と...を併用します。
*      foo/.../bar/*は、指定したパス内のパーディレクトリ内のファイルに一致します。

crcSalt = <文字列>
*      これを使用して、Splunk が、CRC に一致するファイルを取り込み後に削除するように強制します。
*      CRC に追加するには文字列を設定します。
*      "crcSalt = <SOURCE>"に設定すると、ソースのフルパスが CRC に追加されます。

followTail = 0 | 1
*      これが 1 に設定されると、ファイルの最後から監視を始めます(tail -f など)。
*      これは、Splunk がファイルを最初に監視する場合にのみ適用されます。
*      その後は、Splunk の内部ファイル位置レコードがそのファイルを追跡します。

dedicatedFD = 0 | 1
*      ファイル記述子を入力専用にします。
*      監視パスがファイルにポイントする場合にのみ許容されます。(ディレクトリと逆)。
*      limits.conf で FD の利用可能数を設定します。
*      すべての FD を使用しないようにしてください。他のデータが無視される原因になることがあります。
```

* 警告：この設定は Splunk インストールならびサーバーに大きく影響します。

alwaysOpenFile = 0 | 1

* ファイルを開き、インデックス化されたかどうかをチェックします。

* modtime を更新しないファイルにのみ効果的です。

* Windows のファイル、そのほとんどを IIS ログのモニタリングにのみ使用すること。

* 注意：このフラグは、最後の手段としてのみ使用すること。負荷が高まり、インデキシングを遅くします。

バッチ:

注意：バッチは、大量の履歴データファイルにのみ使用してください。ディレクトリを監視し続ける場合、または小さなアーカイブをインデックスする場合は、監視を使用します(上記)。

[batch://<path>]

* 一回限り、取り込み後に削除される入力。

* 継続的な取り込み後に削除されない入力は、**監視**を使用します。

追加属性:

move_policy = sinkhole.

* Important = には、move_policy = sinkhole を設定する必要があります。

* これはロードしたファイルを取り込み後に削除します。

* 取り込み後に削除したくないファイルは、この入力タイプを使用しないでください。

host_regex (上記の監視を参照)

host_segment (上記の監視参照)

重要：次は、バッチにより使用されません。

source = <文字列>

<KEY> = <文字列>

TCP:

[tcp://<remote server>:<port>]

* Splunk が特定のポートをリッスンするように設定します。

* 接続が<リモートサーバー>の場合、このスタanzasはその入力を設定する場合のみ使用されます。

* <リモートサーバー>が空白の場合、このスタanzasは指定されたポート上のすべての接続に一致します。

追加属性:

connection_host = ip | dns

* "ip" または "dns" に設定。

* "ip" は、TCP 入力プロセッサが、ホストを、リモートサーバーの IP アドレスで書き換えるように設定します。

* "dns" は、ホストを、リモートサーバーの DNS エントリに設定します。

* デフォルトは 0。

```

# データ分散：
#*****

[ Splunktcp://<リモートサーバー>:<ポート> ]
* これは TCP と同じです。ただし、リモートサーバーが Splunk サーバーであると仮定することを除く。
* SplunkTCP では、ホストまたは connection_host は、リモート Splunk サーバーがホストを設定しない場合、またはそのホストは host::localhost に設定されない場合に使用します。
* 詳細は、説明書 (http://www.splunk.com/doc/latest/admin/ForwardingReceiving) を参照してください。
enableS2SHeartbeat = true | false
* これは、ネットワーク、ファイアウォールなどが原因のデッドフォワードの検出を可能にします。
* Splunk は、ハートビートの存在に関する接続を監視し、そのハートビートが s2sHeartbeatTimeout で見えないとき、受信ホストがその接続を閉じます。
* これは、グローバル SplunkTCP スタンザで指定されたデフォルト値を上書きします。
* これはデフォルトでは true です。
s2sHeartbeatTimeout = <秒>
* タイムアウト値は秒です。
* Splunk は、フォワード接続を監視し、そのハートビートが s2sHeartbeatTimeout 秒で見えないとき、その接続を閉じます。
* これは、グローバル SplunkTCP スタンザで指定されたデフォルト値を上書きします。
* デフォルト値は 600 秒とします。

[ splunktcp ]
route = has_key | absent_key:<key>:<queueName>;...
* ライトフォワードの設定。
* Splunk はこれらのパラメータを自動設定します。これらを設定する必要はありません。
* プロパティルーティングは、';' で区切られたルールからなります。
* Splunk は、ルーティングルールに対して、処理済 TCP ポートを通じて、受信データペイロードをチェックします。
* マッチルールが見つかったら、Splunk はペイロードを指定された<queueName>に送信します。
* マッチングルールが見つからないと、Splunk はペイロードを、このスタンザの queue= で指定されたデフォルトキューに送信します。queue= key がスタンザまたはグローバルに設定されない場合、そのイベントは parsingQueue に送信されます。
compressed = true | false
* 圧縮されたデータを送るかどうか (y/n) を指定します。
* デフォルトは false。
* これが true に設定されていると、フォワードポートは圧縮をオンにします。
enableS2SHeartbeat = true | false
* これは、すべての SplunkTCP ポートに関するグローバルキープアライブ設定を指定します。
* これはデフォルトでは true です。
s2sHeartbeatTimeout = <秒>
* これは、すべての SplunkTCP ポートに関するグローバルキープアライブ設定を指定します。
* デフォルト値は 600 秒です。

```

```

#      データ分散のための SSL 設定：
[splunktcp-ssl:PORT]
*      暗号化された処理済データを Splunk から送信する場合、このスタンプザを使います。
*      どのフォワーダが処理済の暗号化されたデータを送信するかについて、ポートに PORT を設定します。
*      フォワーダ設定は、フォワーダ側の outputs.conf に設定されます。
enableS2SHeartbeat = true | false
*      [splunktcp:PORT]の文書を参照してください。
s2sHeartbeatTimeout = <秒>
*      [splunktcp:PORT]の文書を参照してください。
compressed = true | false
*      圧縮されたデータを送るかどう (y/n) を指定します。
*      デフォルトは false。
*      これが true に設定されていると、フォワーダポートは圧縮をオンにします。
[tcp-ssl:PORT]
*      暗号化されたローデータを他社システムから送信する場合に、このスタンプザを使います。
*      どのフォワーダがローの暗号化されたデータを送信するかについて、ポートに PORT を設定します。
[SSL]
*      次の仕様を、このスタンプザ名の下にある SSL に設定します。
serverCert = <パス>
*      サーバ証明書のフルパス
password = <文字列>
*      サーバ証明書パスワード (存在する場合)
rootCA = <文字列>
*      証明書権限リスト (ルートファイル)
requireClientCert = true | false
*      クライアントが認証する必要があるかどうかを切り替える。
supportSSLV3Only = <true|false>
*      true の場合、inputproc に対し、SSLv3 クライアントからの接続のみを受け入れることを伝える。
*      デフォルトは false。
cipherSuite = <cipher suite 文字列>
*      これが設定されると、入力プロセッサで指定された暗号文字列を使用します。
*      設定されない場合、OpenSSL が提供するデフォルト暗号文字列を使います。これは、サーバーが、弱い暗号化プロトコル
を使って接続を受け入れないことを確実にするために使用します。
#*****
#      UDP：
#*****
[udp://<port>]
*      TCP と同様。ただし、それが UDP ポートでリッスンする部分は異なります。

```

```

#      追加属性：
_rcvbuf = <整数>
*      UDP ポートの受信バッファを指定する(バイト)。
*      その値が 0 の場合、無視されます。
*      デフォルトは 1,000,000(または 1MB)。
*      注意： OS によりデフォルトは異なります。
no_priority_stripping = true
*      この属性は true に設定された場合、Splunk は、受信イベントから<priority>syslog フィールドをストリップしません。
*      注意： <priority>をストリップしたい場合、このキーは含めないでください。
no_appending_timestamp = true
*      この属性が true に設定された場合、Splunk はタイムスタンプとホストを受信イベントに追加しません。
*      注意： タイムスタンプとホストを受信イベントに追加したい場合、このキーは含めないでください。
#*****
#      FIFO：
#*****
[fifo://<path>]
*      これは、Splunk に、指定されたパスで FIFO を読み込むように指示します。
#*****
#      スクリプト入力：
#*****
[script://<cmd>]
*      設定されたインターバル(下記)で<cmd>を実行し、その出力をインデックス化します。
*      このコマンドは、$SPLUNK_HOME/etc/system/bin/または../etc/apps/$YOUR_APP/bin/になければなりません。
*      Windows WMI およびレジストリ入力はスクリプト入力です。 inputs.conf.example の最後をチェックしてください。
interval = <整数>|<クーロンスケジュール>
*      指定されたコマンドの実行頻度(秒)または有効なクーロンスケジュール。
*      注意： クーロンスケジュールが指定されたとき、スクリプトはスタートアップで実行されません。
*      デフォルトは 60 秒です。
passAuth = <ユーザー名>
*      スクリプトを実行するユーザー。
*      ユーザー名を指定した場合、Splunk は、そのユーザーのために認証トークンを生成し、それを stdin 経由でそのスクリプトに渡します。
#*****
#      ファイルシステム変更監視
#*****
注意： fs 変更監視と監視(上記)を使用して、同時にディレクトリを監視できません。
[fschange:<パス>]
*      このディレクトリおよびサブディレクトリに対する追加/更新/削除を監視します。

```

* 注意: <パス>はダイレクトパスです。他の入力のように、//を先頭につける必要はありません。

* 変更のたびにイベントを送信します。

追加属性:

注意: fschange は、他の入力タイプ(上記)と同じ属性は使いません。次の属性のみを使います。

index = <インデックス名>

* 生成されたすべてのイベントを保存するインデックス。

* デフォルトは_audit。ただし、signedaudit(下記)を設定しない場合、または signedaudit = false を設定しない場合を除く。この場合は、イベントはデフォルトインデックスへ行く。

signedaudit = true | false

* 暗号化して署名された、追加/更新/削除イベントを送信する。

* true に設定すると、イベントは*常に*_audit'インデックスに送信され、*常に*ソースタイプ'audittrail'を持ちます。

* false に設定すると、イベントはデフォルトインデックスに置かれ、そのソースタイプは指定したもの(または、デフォルトで'fs_notification')となります。

* インデックスを設定したい場合、signedaudit を false に設定する必要があります。

* 注意: また、audit.conf で監査を有効にする必要があります。

* デフォルトは false。

filters = <フィルタ>,<フィルタ>,...<フィルタ N>

* 各フィルタは、監視のポーリングの際に見つかった各ファイルまたはディレクトリで左から右に適用されます。

* フィルタの定義に関するヘルプは、下記の“ファイルシステム監視フィルタ”を参照のこと。

recurse = true | false

* true の場合、[fschange]で指定されたディレクトリ内で再帰的に読み込みます。

* デフォルトは true。

followLinks = true | false

* true の場合、シンボリックリンクをフォローします。

* これを true に設定しないことを推奨します。設定すると、ファイルシステムのループが発生する可能性があります。

* デフォルトは false。

pollPeriod = <整数>

* <整数>秒毎の変更のたびに、このディレクトリをチェックします。

* デフォルトは 3600。

hashMaxSize = <整数>

* <整数>バイト以下のファイルのすべてについて SHA256 ハッシュを計算します。

* このハッシュは、ファイル/ディレクトリに対する変更を検知するためのもう一つの方法として使われます。

* デフォルトは-1(無効)。

fullEvent = true | false

* 追加または更新の変更が検出された場合にフルイベントを送信するには、true に設定します。

* 'sendEventMaxSize'属性による追加の制限。

* デフォルトは false。

sendEventMaxSize = <整数>

- * イベントのサイズが<整数>バイト以下の場合のみ、フルイベントを送信します。
- * これは、インデックス化されたファイルデータのサイズを制限します。
- * デフォルトは-1(制限なし)。

sourceType = <文字列>

- * この入力からのイベントのソースタイプを設定。
- * " sourceType =" は自動的に<文字列>の先頭に追加されます。
- * デフォルトは audittrail (signedaudit=true の場合)または fschange(signedaudit=false の場合)。

host = <文字列>

- * この入力のイベントのホストを設定。
- * デフォルトは、あらゆるホストにイベントを送信。

index = <文字列>

- * この入力からのイベントのインデックスを設定。
- * デフォルトはメインインデックス。

filesPerDelay = <整数>

- * <整数>ファイルのプロセス後、'delayInMills'により指定された遅延を注入。
- * これは、ファイルシステム監視を絞って、CPU を多く消費しないようにするために使います。

delayInMills = <整数>

- * 'filesPerDelay'で指定されたとおり、<整数>ファイル毎のプロセス後に、使用する遅延をミリ秒で指定します。
- * これは、ファイルシステム監視を調整して、CPU を多く消費しないようにするために使います。

ファイルシステム監視グフィルタ :

[filter:<フィルタタイプ>:<フィルタ名>]

- * タイプ<フィルタタイプ>のフィルタを定義し、それに名前<フィルタ名>を付けます。

<フィルタタイプ>

- * フィルタタイプはブラックリストまたはホワイトリストです。
- * ホワイトリストフィルタは、正規表現リストに一致するすべてのファイル名を処理します。
- * ブラックリストフィルタは、正規表現リストに一致するすべてのファイル名をスキップします。

<フィルタ名>

- * フィルタ名は、ファイルシステム監視を定義する際に、カンマ区切りリストを使用します。

regex<整数> = <正規表現>

- * ブラックリストとホワイトリストフィルタは正規表現を含むことができます。
- * 各正規表現の名前は、'regex<整数>'である必要があります。ここで、<整数>は 1 から始まり、1 ずつ増加します。
- * Splunk は各正規表現を数値順に適用します。

regex1=<正規表現>

regex2=<正規表現>

...

```

#*****
#      Windows 入力：
#*****
*      Windows プラットフォーム専用の入力プロセッサ。
*      セキュリティ、アプリケーション、システムはデフォルトで有効になります。入力タイプを無効にするには、コメント文に
するか、$SPLUNK_HOME\etc\apps\windows\local\inputs.conf で disabled = 1 に設定します。
*      他の Windows イベントログを読み込むように Splunk を設定することもできます。ただし、まず Windows Event Viewer
にインポートしてから、次に inputs.conf($SPLUNK_HOME\etc\apps\windows\local\inputs.conf)のローカルコピーに追加す
る必要があります。
下記に示すもの[WinEventLog:<イベントログ名>]と同じフォーマットと disabled = 0 の行を使います。
[WinEventLog:<ログ名>]
*      監視する Windows イベントログを定義します。
disabled = <整数> 1|0
*      この入力を有効または無効にします。
start_from = <文字列> oldest|newest
*      oldest - Windows イベントログを古いものから新しいものに時系列で読み込みます。
*      newest -Windows イベントログを逆、つまり新しいものから古い順で読み込みます。イベントのバックログが消費される
と、新しいイベントの取り込みを始めます。
current_only = <整数> 1|0
*      1 に設定すると、追跡をエミュレートし、新規に受信したイベントのみを監視します。0 に設定すると、まずシステムに存在
するすべてのイベントを読み込み、次にリアルタイムで送られるイベントを監視します。
checkpointInterval = <整数> 秒
*      0 より大きい整数。Windows イベントログが、保存されるチェックポイントをチェックする頻度(インターバル)を設定しま
す。
evt_resolve_ad_obj = <整数> 1|0
特定の Windows イベントログチャンネルの GUID/SID オブジェクトのように、アクティブディレクトリオブジェクトの分解を有効/無
効にします。デフォルトでは、このオプションは、セキュリティイベントログでオンになります。 オプションで、ドメインコントロ
ーラ名またはドメインの DNS 名を指定し、Splunk が AD オブジェクトの解決に使用するためにバインドできます。
evt_dc_name = <文字列>
オプション。このパラメータは空白にすることができます。
バインドするドメインコントローラの名前。この名前は、ドメインコントローラの名前またはドメインコントローラの完全に記述され
た DNS 名でも可。いずれのの名前のタイプも、オプションで、2 つのバックスラッシュ文字で保護することができます。次の例はすべ
て、正しくフォーマットされたドメインコントローラ名を示します。
*      "FTW-DC-01"
*      "\\FTW-DC-01"
*      "FTW-DC-01.splunk.com"
*      "\\FTW-DC-01.splunk.com"
evt_dns_name = <文字列> オプション,

```

このパラメータは空白にすることができます。

ドメインをバインドする完全修飾 DNS 名。

inputs.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# これは inputs.conf の例です。このファイルを使ってデータ入力を設定します。
#
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の inputs.conf にコ
# ピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
# 次の設定は、Splunk に、/var/log ディレクトリにあるすべてのファイルを読み込むように指示します。
[monitor:///var/log]
# 次の設定は、Splunk に、/var/log/httpd にあるすべてのファイルを読み込み、それらを sourcetype::access_common
として分類するように指示します。
[monitor:///var/log/httpd]
sourcetype = access_common
# 次の設定は、Splunk に、/mnt/logs のすべてのファイルを読み込むように指示します。パスが /mnt/logs/<host>/...
のとき、ホスト名を(ファイルで)<ホスト>に設定します。
[monitor:///mnt/logs]
host_segment = 3
# 次の設定は、Splunk に、すべてのリモートサーバー(Splunk インスタンスだけでない)からのローデータについて、TCP
ポート 9997 でリッスンするように指示します。データのホストは、リモートサーバーの IP アドレスに設定されます。
[tcp://:9997]
# 次の設定は、Splunk に、すべてのリモートサーバーからのローデータについて、TCP ポート 9995 でリッスンするよう
に指示します。
# データのホストは、リモートサーバーのホスト名として設定されます。全データはまた、ソースタイプ "log4j" とソース
"tcp:9995" を割り当てられます。
[tcp://:9995]
connection_host = dns
sourcetype = log4j
source = tcp:9995
# 次の設定は、Splunk に、10.1.1.10.からのローデータについて、TCP ポート 9995 でリッスンするように指示します。
# 全データはホスト "webhead-1"、ソースタイプ "access_common"、およびソース
"/10.1.1.10/var/log/apache/access.log" に割り当てられます。
```

```
[tcp://10.1.1.10:9995]
host = webhead-1
sourcetype = access_common
source = //10.1.1.10/var/log/apache/access.log
# 次の設定は、ライトフォワードから送信されたデータペイロードについてグローバルデフォルトを設定します。
# ルーティングパラメータは、処理済データの各ペイロードのために評価されたルールの順序化されたセットです。
[splunktcp]
route=has_key:_utf8:indexQueue;has_key:_linebreaker:indexQueue;absent_key:_utf8:parsingQueue;absent_
# 次の設定は、Splunk に、すべてのリモートサーバからの分散検索データについて、TCP ポート 9996 でリッスンするように指示します。データは、追加の処理なしで、ローカルマシン上のインデックス化されに直接送信されます。データのホストは、リモートデータがホスト設定を持たない場合、または、"localhost"に設定されている場合にのみ、リモートサーバのホスト名に設定されます。
[splunktcp://:9996]
queue = indexQueue
connection_host = dns
# 次の設定は、Splunk に、10.1.1.100 からの分散検索データについて TCP ポート 9998 でリッスンするように指示します。
[splunktcp://10.1.1.100:9996]
# 次の設定は、Splunk に、syslog.corp.company.net からのデータについて、TCP ポート 514 でリッスンするように指示します。データはソースタイプ"syslog"に割り当てられ、ホストはリモートサーバのホスト名に設定されます。
[tcp://syslog.corp.company.net:514]
sourcetype = syslog
connection_host = dns
# SSL の設定 :
[SSL]
serverCert=$SPLUNK_HOME/etc/auth/server.pem
password=password
rootCA=$SPLUNK_HOME/etc/auth/cacert.pem
requireClientCert=false
[splunktcp-ssl:9996]
# Windows レジストリモニタを有効にします (Windows のみ)。
# この例は、Windows レジストリモニタをスクリプト入力として有効にする方法を示します。
# Windows レジストリは大量のイベントを生成することができるため、Windows レジストリモニタも他の 2 つの設定ファイルに影響を受けます。2 つの設定ファイルとは sysmon.conf と regmon.conf です。
# * sysmon.conf は、グローバル設定を含んでおり、イベントタイプ(追加、削除、名前書き換えなど)をモニタするのか、regmon-filters.conf ファイルからのどの正規表現フィルタを使用するのか、および Windows レジストリイベントがモニタするかどうかを設定するものです。
# * regmon-filters.conf は、Splunk にモニタさせたいハイブキーパスを選別しフィルタするために作成する特定の正規表現を含んでいます。
```

Splunk は、その詳細について、<http://www.splunk.com/base> にある Windows レジストリモニタに関する文書を参照することを推奨します。

\$SPLUNK_HOME/etc/system/local の inputs.conf で下記の変更を行う必要があります。

設定を有効にするには Splunk の再起動が必要です。

```
[script://$SPLUNK_HOME\bin\scripts\splunk-regmon.py]
```

```
interval = 60
```

```
sourcetype = WinRegistry
```

```
source = WinRegistry
```

```
disabled = 0
```

WMI 入力を有効にします (Windows のみ)。

この例は WMI 入力をスクリプト済入力として有効にする方法を示します。

WMI 入力はまた、wmi.conf の設定に影響されます。

Splunk は、詳細について、<http://www.splunk.com/base> にある WMI 入力に関する文書を参照することを推奨します。

\$SPLUNK_HOME/etc/apps/windows/local/ の inputs.conf でこの変更を行う必要があります。

設定を有効にするには Splunk の再起動が必要です。

```
[script://$SPLUNK_HOME\bin\scripts\splunk-wmi.py]
```

```
disabled = 0
```

ファイルシステム変更監視を使用する

```
[fschange:/etc/]
```

```
fullEvent=true
```

```
pollPeriod=60
```

```
recurse=true
```

```
sendEventMaxSize=100000
```

```
index=main
```

Windows イベントログセキュリティの監視:

```
[WinEventLog:Security]
```

```
disabled = 0
```

```
start_from = oldest
```

```
current_only = 0
```

```
evt_dc_name =
```

```
evt_dns_name =
```

```
evt_resolve_ad_obj = 1
```

```
checkpointInterval = 5
```

Windows イベントログフォワーダの監視:

```
[WinEventLog:ForwardedEvents]
```

```
disabled = 0
```

```
start_from = newest
```

```
current_only = 1
```

```
checkpointInterval = 5
```

limits.conf

limits.conf

次は limits.conf の仕様とファイル例です。

limits.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、検索コマンドの制限を設定するための属性/値ペアを記載しています。
#
# limits.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、limits.conf を
# $SPLUNK_HOME/etc/system/local に置いてください。例は、limits.conf.example を参照してください。設定を有効にするに
# は Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
# http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
# 照してください。
#
# 警告： 詳細がわからない場合は、limits.conf の設定を変更しないでください。
# 不適切に制限を設定すると、Splunkd のクラッシュ、またはメモリの使いすぎを招く恐れがあります。
* 各スタンプは、検索コマンドの異なるパラメータをコントロールします。
[searchresults]
* このスタンプは、さまざまな Splunk 検索パイプライン演算子の検索結果をコントロールします。
maxresultrows = <整数>
* multikv などの結果のサイズを増やしたり、イベントを作成したりする検索演算子により生成されるイベントの最大数を設
# 定します。検索演算子の中には、下記の特定のスタンプで明示的にコントロールされるものがあります。
* デフォルトは 50000。
tocsv_maxretry = <整数>
* アトミック書き込み動作で、トライする最大回数。
* 1 = 再試行なし。
* デフォルトは 5。
tocsv_retryperiod_ms = <整数>
* 再試行期間。
* デフォルトは 500。
[subsearch]
* このスタンプはサブ検索結果をコントロールします。
```

maxout = <整数>

- * サブ検索から返す結果の最大数。
- * デフォルトは 100。

maxtime = <整数>

- * 完了させる前に、サブ検索を実行する最大秒数。
- * デフォルトは 10。
- * 既に実行しているサブ検索を最大待ち時間。
- * デフォルトは 30。

ttl = <整数>

- * 与えられたサブ検索の結果をキャッシュする時間。
- * デフォルトは 300。

[anomalousvalue]

maxresultrows = <整数>

- * 一度にメモリに存在することができるイベントの最大数を設定します。
- * デフォルトは searchresults::maxresultsrows (例: 50000)。

maxvalues = <整数>

- * フィールドの各値の最大数。
- * デフォルトは 100000。

maxvaluesize = <整数>

- * 単一値の最大サイズ(バイト)(これより大きい場合は、このサイズに切り詰められる)
- * デフォルトは 1000。

[associate]

maxfields = <整数>

- * 分析するフィールドの最大数。
- * デフォルトは 10000。

maxvalues = <整数>

- * 追跡するフィールドの値の最大数。
- * デフォルトは 10000。

maxvaluesize = <整数>

- * 1つの値で対象とする最大長。
- * デフォルトは 1000。

[ctable]

- * このスタンプは、contingency、ctable、および counttype コマンドをコントロールします。

maxvalues = <整数>

- * 生成する列/行の最大数(つまり、行フィールドの最大個別値)
- * デフォルトは 1000。

[correlate]

maxfields = <整数>

- * 関連させるフィールドの最大数。
- * デフォルトは 1000。

[discretize]

- * このスタンザは bin/bucket/discretize の属性を設定します。

maxbins = <整数>

- * 離散化するバケツの最大数。
- * maxbins が指定されていない、または 0 の場合、デフォルトを searchresults::maxresultrows(例: 50000) に設定します。

[inputcsv]

mkdir_max_retries = <整数>

- * 一時ディレクトリ作成時に再試行する最大回数(SPLUNK_のサブディレクトリとしてのランダム名)。
- * デフォルトは 100。

[kmeans]

maxdatapoints = <整数>

- * k-means クラスタリングを行う最大データポイント。
- * デフォルトは 1000000000。

[kv]

maxcols = <整数>

- * 0 でない場合、kv が新規フィールドの作成を停止する点。
- * デフォルトは 512。

[metrics]

maxseries = <整数>

- * metrics.log で per_x_thruput レポートに含めるシリーズの数。
- * デフォルトは 10。

[rare]

maxresultrows = <整数>

- * 作成する結果の最大行数。
- * 指定されていない場合、デフォルトを searchresults::maxresultrows(例: 50000) に設定します。

maxvalues = <整数>

- * 追跡する個々のフィールドベクトル値の最大数。
- * デフォルトは 100000。

maxvaluesize = <整数>

- * 1 つの値で対象とする最大長。
- * デフォルトは 1000。

[report]

maxresultrows = <整数>

- * 作成する結果の最大行数。
- * デフォルトは 300。

[restapi]

maxresultrows = <整数>

- * REST API から /events または /results ゲッターによりリターンされる最大結果行数。
- * デフォルトは 50000。

[search]

ttl = <整数>

- * 検索終了後、保存する期間。
- * デフォルトは 86400。

status_buckets = 300

- * 保持する時間軸バケツのおよその最大数。
- * デフォルトは 300。

max_count = <整数>

- * ベースとバウンドを取る呼び出しの最後のアクセス可能イベント。
- * デフォルトは 10000。

min_prefix_len = <整数>

- * *がインデックスを問い合わせる前の接頭辞の最小長さ。
- * デフォルトは 1。

max_results_raw_size = <整数>

- * メモリに読み込む "_raw" の最大ボリューム。
- * _raw フィールドの全ボリューム(イベントのテキスト)がこの値を超えると、それ以上の結果はなし。
- * デフォルトは 100000000(100MB.)。

cache_ttl = <整数>

- * 検索キャッシュエントリを継続する時間の長さ(秒)。
- * デフォルトは 300。

reduce_freq = <整数>

- * 中間結果を減らす試みをするチャンクの数指定します(0=なし)。
- * デフォルトは 10。

[slc]

maxclusters = <整数>

- * 作成するクラスタの最大数。
- * デフォルトは 10000。

[stats]

maxresultrows = <整数>

- * 作成する結果の最大行数。
- * 指定されていない場合、デフォルトを searchresults::maxresultrows(例: 50000)に設定します。

maxvalues = <整数>

- * 追跡するフィールドの値の最大数。
- * デフォルトは 10000。

```

maxvaluesize = <整数>
*     1つの値で対象とする最大長。
*     デフォルトは 1000。

[thruput]
maxKBps = <整数>
*     0 以外を指定する場合、thruput プロセッサに対する速度を specified[top] に制限します。

maxresultrows = <整数>
*     作成する結果の最大行数。
*     指定されていない場合、デフォルトを searchresults::maxresultrows (例: 50000) に設定します。

maxvalues = <整数>
*     追跡する別々のフィールドベクトル値の最大数。
*     デフォルトは 100000。

maxvaluesize = <整数>
*     1つの値で対象とする最大長。
*     デフォルトは 1000。

[inputproc]
max_fd = <整数>
*     Splunk が Select Processor で使用することができるファイル記述子の最大数。
*     与えられる最大値は、プロセスあたりに許されるファイル記述子の現在の数の半分。
*     選択された値は最大許容値より高い場合、最大値が代わりに使用されます。
*     デフォルトは 32。

time_before_close = <整数>
*     Splunk がファイルを EOF で閉じる前に必要な modtime delta。
*     過去<整数>秒に更新されていないファイルを閉じないようにシステムに伝えます。
*     デフォルトは 5。

fishbucketSyncTime = <整数>
*     fishbucket DB キューがディスクをフラッシュする頻度。
*     デフォルトは 10 秒。

```

limits.conf.example

```

#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#     警告: 詳細がわからない場合は、limits.conf の設定を変更しないでください。
#     不適切に制限を設定すると、Splunkd のクラッシュ、またはメモリの使いすぎを招く恐れがあります。

[searchresults]
maxresultrows = 50000
#     アトミック書き込み動作で、トライする最大回数(1=再試行なし)。
tocsv_maxretry = 5
#     再試行期間は 1/2 秒(500 ミリ秒)

```

```

tocsv_retryperiod_ms = 500

[subsearch]
#       サブ検索から返す結果の最大数。
maxout = 100
#       完了させる前に、サブ検索を実行する最大秒数。
maxtime = 10
#       既に実行しているサブ検索の最大待ち時間。
timeout = 30
#       与えられたサブ検索の結果をキャッシュする時間。
ttl = 300

[anomalousvalue]
maxresultrows = 50000
#       フィールドの各値の最大数。
maxvalues = 100000
#       単一値の最大サイズ(バイト)(これより大きい場合は、このサイズに切り詰められる)
maxvaluesize = 1000

[associate]
maxfields = 10000
maxvalues = 10000
maxvaluesize = 1000

#       contingency、ctable、および counttype コマンド用。
[ctable]
maxvalues = 1000

[correlate]
maxfields = 1000

#       bin/bucket/discretize 用
[discretize]
maxbins = 50000
#       maxbins が指定されていない、または 0 の場合、デフォルトを searchresults::maxresultrows に設定します。
[inputcsv]
#       一時ディレクトリ作成の再試行の最大数(SPLUNK_HOME/var/run/splunk のランダム名)。
mkdir_max_retries = 100

[kmeans]
maxdatapoints = 100000000

[kv]
#       0 でない場合、kv が新規列の作成を停止する点。
maxcols = 512

[rare]

```

```

maxresultrows = 50000
#      追跡する個々の値ベクトル値の最大数。
maxvalues = 100000
maxvaluesize = 1000
[report]
maxresultrows = 300
[restapi]
#      REST API から /events または /results ゲッタによりリターンされる最大結果行数。
maxresultrows = 50000
[search]
#      検索終了後、保存する期間。
ttl = 86400
#      保持する時間軸バケツのおよその最大数。
status_buckets = 300
#      ベースとバウンドを取る呼び出しの最後のアクセス可能イベント。
max_count = 10000
#      *がインデックスを問い合わせる前の接頭辞の最小長さ。
min_prefix_len = 1
#      メモリーに読み込む "_raw" の最大ボリューム。
max_results_raw_size = 100000000
#      検索キャッシュエントリーを継続する時間の長さ(秒)。
cache_ttl = 300
[slc]
#      作成するクラスタの最大数。
maxclusters = 10000
[stats]
maxresultrows = 50000
maxvalues = 10000
maxvaluesize = 1000
[top]
maxresultrows = 50000
#      追跡する個々の値ベクトル値の最大数。
maxvalues = 100000
maxvaluesize = 1000
[inputproc]
max_fd = 32
time_before_close = 5

```

literals.conf

literals.conf

次は literals.conf の仕様とファイル例です。

literals.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、literals.conf で外部文字列の設定をするための属性/値ペアを記載しています。
#
# literals.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、literals.conf
# を $SPLUNK_HOME/etc/system/local に置いてください。例は、literals.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
# http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
#
# 上書き可能なすべてのリテラルの全リストは $SPLUNK_HOME/etc/system/default/literals.conf をチェックしてください。
#####
#
# 警告：
#
# literals.conf を不適切に変数することで、Splunkd の性能が劣化することがあります。
#
# -属性値のみを編集すること ( '=' の右側にある )
# 属性名 ( '=' の左側 ) の編集はしないこと。
#
# -文字列が "%s" を含む場合、%s を追加したり削除したり、それらの順番を変えないこと。
#
# -文字列が html タグを含む場合、すべてのタグと引用属性が適切に閉じており、&などのすべてのエントリーがエスケープされていることを特に確認すること。
#
```

literals.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
```

```
# このファイルは literals.conf の例を記載しています。これは、Splunk の外部文字列の設定に使用されます。
#
# 上書きできるリテラルの全リストについては、$SPLUNK_HOME/etc/system/default/literals.conf にある詳細リス
トを参照のこと。
#
[ui]
PRO_SERVER_LOGIN_HEADER = Splunk へのログイン (guest/guest)
INSUFFICIENT_DISK_SPACE_ERROR = サーバーの空きディスク容量が少なすぎます。インデキシングは一時的に、
SERVER_RESTART_MESSAGE = この Splunk サーバーの設定は変更されました。サーバーは、XXが必要です。
UNABLE_TO_CONNECT_MESSAGE = %s で Splunkd に接続できません。
```

macros.conf

macros.conf

次は macros.conf の仕様とファイル例です。

macros.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは検索言語マクロの属性/値ペアを記載しています。
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[$STANZA_NAME]
* 各スタンザは、検索で参照可能な検索マクロを表します。
* スタンザ名は、マクロが引数をとらない場合のマクロの名前です。その他の場合、スタンザ名は "<numargs>" が追加され
たマクロ名です。ここで、<numargs>は、このマクロをとる引数の数です。
* マクロは過負荷となる場合があります。したがって、[foobar] および [foobar(1)] および [foobar(2)] などを持つこ
とができます。
* マクロは、マクロ名および `foobar(arg1,arg2)` または `foobar` などのチェックマーク内の引数リストを囲むことによ
り、検索言語で使用することができます。
* "foo`bar`baz" などの引用された値の内部では、マクロ拡張は行われません。
args = <文字列>
* 引数名のカンマで区切られた文字列
* 引数名は、英数字とアンダースコア '_' およびダッシュ '-' のみを含むことができます。
* スタンザ名が、このマクロは引数を取らないことを示す場合、このキーは無視されます。
* このリストが繰り返しエレメントを含む場合はエラーとなります。
definition = <文字列>
```

- * マクロが拡張され、引数が記載された文字列。(例外: `iseval = true` の場合は下記参照)
- * 記載すべき引数はドル文字で囲まれている必要があります。例: "the last part of this string will be replaced by the value of argument foo \$foo\$".
- * \$が引数(`args` リストで指定)を囲んでいない場合、それは無視され、注釈の\$と解釈されます。
- * \$...\$のパターンは、文字列でグローバルに置換されます。引用の内側でもそうなります。

`validation = <文字列>`

- * 'eval' 表現である検証文字列。この表現は、静的に論理演算子または文字列を評価する必要があります。
- * この検証は、このマクロの呼び出しに使われる引数値が受容できるかどうかを検証するためのものです。
- * 検証表現が論理演算子表現の場合、それが `true` を返すと検証は成功です。それが `false` または `null` を返す場合は、検証は失敗し、'errmsg' により定義されたエラーメッセージが返されます。
- * 検証表現が論理演算子表現でない場合、文字列または `null` が返されます。この場合、それが `null` を返す場合、検証は成功したと見なします。その他の場合、返される文字列はエラー文字列として生成されます。

`errmsg = <文字列>`

- * 検証が論理演算子表現で、`true` と評価されない場合に表示されるエラーメッセージ。

`iseval = <true/false>`

- * `true` の場合、'definition' は、このマクロの拡張を表す文字列を返す評価表現である必要があります。
- * デフォルトは `false`。

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

#

Example macros.conf

```
#
# 引数をとらないマクロフットバーは`foobar`により呼び出されます。
[foobar]
# マクロの定義は、別のマクロを呼び出すことができます。ネスティングは無限とすることができ、サイクルが検出され、エラーになります。
definition = `foobar(foo=defaultfoo)`
# 引数を1つとるマクロフットバーで、`foobar(someval)`により呼び出されます。
[foobar(1)]
args = foo
# 注意: この定義は語頭と語末に引用符を含みます。つまり`foobar(someval)`は"foo = someval"に拡張します。
definition = "foo = $foo$"
# 引数を2つとるマクロ。
# 注意: マクロ引数は、このマクロが同等に呼び出されるように名前を付けることができます。例: `foobar(1,2)`
foobar(foo=1,bar=2)` または `foobar(bar=2,foo=1)`。
[foobar(2)]
args = foo, bar
definition = "foo = $foo$, bar = $bar$"
```

```

#       検証を行わない引数を1つとるマクロ。
[foovalid(1)]
args = foo
definition = "foovalid = $foo$"

#       検証評価機能は偶数の引数を取ります(>=2)。ここで、第1の引数は論理演算子表現、第2は文字列、第3は論理演算子、
第4は4thなど。
validation = validate(foo>15,"foo must be greater than 15",foo<=100,"foo must be <= 100")

#       簡単な論理演算子検証を表すマクロ。ここで foo > bar な true でない場合、errmsg が表示されます。
[foovalid(2)]
args = foo, bar
definition = "foo = $foo$ and bar = $bar$"
validation = foo > bar
errmsg = foo must be greater than bar

#       評価ベース定義の例。例えば、この場合、`fooeval(10,20)`は 10 + 20 で置き換えられます。
[fooeval(2)]
args = foo, bar
definition = if (bar > 0, "$foo$ + $bar$", "$foo$ - $bar$")
iseval = true

```

multikv.conf

multikv.conf

次は multikv.conf の仕様とファイル例です。

multikv.conf.spec

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       このファイルは、multikv ルールを作成するための属性と値のペアを記載しています。
#       multikv は、イベントをテーブル型イベントからイベントを抽出するプロセスです。例えば、top、ps、ls、netstat などの出力です。
#
#       デフォルトの multikv.conf はありません。カスタム設定を設定するには、multikv.conf を
$SPLUNK_HOME/etc/system/local に置いてください。例は、multikv.conf.example を参照のこと。
#       設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。

```


注意: Splunk の自動 multikv の振る舞いに満足できない場合にのみ multikv.conf を設定してください。 multikv
検索コマンドで満足できる結果が得られるのであれば、このファイルを作成する理由はありません。

テーブル型イベントには、一つのテーブル、つまり 4 つの部分またはセクションからなるテーブルを含みます:

#-----
セクション名 | 説明
#-----
pre | オプション: 情報/説明(例: トップのシステムサマリー出力)
header | オプション: 定義されていない場合、フィールドは Column_N と名付けられます。
body | 必須: これは、子イベントが構成するテーブルの本体です。
post | オプション: 情報/説明
#-----

注意: 各セクションとプロセスのためのセクションは、セクション定義(下記)と処理(下記)のセットの両方がなければなりません。

[multikv_config_name]

* multikv 検索コマンドと共に使用するスタンザに名前を付けます:

例: '.... | multikv conf=\$STANZA_NAME rmorig=f |'

* このスタンザ名に、次の属性/値のペアを続けてください。

#####

セクション定義

#####

各セクションが開始し終了する箇所を定義します。

#

section_\$NAME.start = <正規表現>

* この正規表現に一致する行は、このセクションの開始を示します(この行を含む)。

または

section_\$NAME.start_offset = <整数>

* event-start からの行オフセット、またはこのセクションが開始する前のセクションの最後(この行を含む)。

* セクションの開始を正規表現で定義できない場合にこれを使います。

section_\$NAME.member = <正規表現>

* 行メンバーシップテスト。

* メンバーiff 行は正規表現に一致します。

section_\$NAME.end = <正規表現>

* この正規表現に一致する行は、このセクションの最後を示します(この行を含む)。

または

section_name.linecount = <整数>

```

*       このセクションの行数を指定します。
*       セクションの最後を正規表現で定義できない場合にこれを使います。
#####
#       セクション処理
#####
#       各セクションの処理を設定します。
#
section_${NAME}.ignore = <文字列 matcher>
*       この文字列 matcher に一致するメンバー行は無視されるため、それ以上処理されません。
*       <string-matcher> = _all_ | _none_ | _regex_ <regex-list>

section_${NAME}.replace = <quoted-str> = <quoted-str>, <quoted-str> = <quoted-str>....
*       フォームのリスト toReplace = replaceWith.。
*       toReplace = replaceWith は複数指定可能。
*       例: "%" = "_", "#" = "_"

section_${NAME}.tokens    = <chopper> | <tokenizer> | <aligner> | <token-list>
*       $VAL の定義は以下を参照。
<chopper> = _chop_, <int-list>
*       各文字列を<int-list>で指定されたトークンのリストに変換します。
*       <int-list> は(オフセット, 長さ) の組のリストです。
<tokenizer> = _tokenize_ <max_tokens (int)> <delims>
*       <delims> = 区切り文字のカンマで区切られたリスト。
*       区切り文字を使って文字列をトークン化します。
*       これは、最大の max_tokens トークンで生成します。
*       max_tokens を次に設定します:
*       -1 は完全トークン化。
*       0 は前のセクションからの引き継ぎ(通常はヘッダ)。
*       または、0 でない数字は、特定のトークン数。
*       トークン化が max_tokens により制限されている場合、残りの文字列は最後のトークンに追加されます。
*       注意: 連続区切り文字扱われは空フィールドとして扱われます。
<aligner> = _align_, <header_string>, <side>, <max_width>
*       指定されたヘッダフィールドに配列されたテキストを抽出してトークンを生成します。
*       header_string: 列が配列される、完全または一部のヘッダフィールド値。
*       side: L または R(左または右アライン)
*       max_width: 抽出されたフィールドの最大幅。
*       max_width を-1 に設定すると自動幅となります(これはフィールドを、次の区切り文字が見つかるま
で拡張します: " ", "\t")。

```

```

<token_list> = _token_list_ <カンマで区切られたリスト>
    *      1つのセクションで固定トークンのリストを定義する。
    *      これは、ヘッダがないテーブルで便利です。例えば、ヘッダがない'ls -lah'の出力など。
multikv.conf.example
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは複数キー/値抽出設定の例を記載します。
#
#      この設定の1つまたは複数を使用するには、その設定ブロックを$SPLUNK_HOME/etc/system/localのmultikv.confに
コピーしてください。設定を有効にするにはSplunkの再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWorkにあるドキュメントを参
照してください。
#      この例は出力をトップから分解します：
#      出力例：
#      Processes: 56 total, 2 running, 54 sleeping... 221 threads 10:14:07
#.....
#
#      PID COMMAND      %CPU TIME      #TH      #PRTS      #MREGS      RPRVT      RSHRD      RSIZE      VSIZE
#      29960 mdimport    0.0% 0:00.29      3         60         50         1.10M     2.55M     3.54M     38.7M
#      29905 pickup     0.0% 0:00.01      1         16         17         164K      832K      764K      26.7M
#....
[top_mkv]
#      pre tableは"Process..."から開始し、"PID"を含む行で終了します。
pre.start = "Process"
pre.end = "PID"
pre.ignore = _all_
#      テーブルヘッダ位置と処理を指定します。
header.start = "PID"
header.linecount = 1
header.replace = "%" = "_", "#" = "_"
header.tokens = _tokenize_, -1, " "
#      テーブルボディは次の"Process"行(つまり、別のトップの開始)で終わり、前のセクション(ヘッダー)からトークンの数を
トークン化して継承します。
body.end = "Process"
body.tokens = _tokenize_, 0, " "
##      この例は、'ls -lah' コマンドの出力を扱います：

```

```

#
#      total 2150528
#      drwxr-xr-x 88 john john 2K Jan 30 07:56 .
#      drwxr-xr-x 15 john john 510B Jan 30 07:49 ..
#      -rw----- 1 john john 2K Jan 28 11:25 .hidden_file
#      drwxr-xr-x 20 john john 680B Jan 30 07:49 my_dir
#      -r--r--r-- 1 john john 3K Jan 11 09:00 my_file.txt

[ls-lah]
pre.start = "total"
pre.linecount = 1
#      ヘッダが内場合、列名をリストアップします。
header.tokens = _token_list_, mode, links, user, group, size, date, name
body.end = "\s*$"
body.member = ".cpp"
#      日付を一つの解読不能項目に連結します。
body.replace = "(\\w{3})\\s+(\\d{1,2})\\s+(\\d{2}:\\d{2})" = "\\1_\\2_\\3"
#      ignore dirs
body.ignore = _regex_ "^drwx.*",
body.tokens = _tokenize_, 0, " "

```

outputs.conf

outputs.conf

次は outputs.conf の仕様とファイル例です。

outputs.conf.spec

```

#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、outputs.conf の設定のための属性/値ペアを記載しています。Splunk のデータフォワーディングア
#      ションを自身の outputs.conf を作成することにより設定してください。
#
#      デフォルトの outputs.conf はありません。カスタム設定を設定するには、outputs.conf を
#      $SPLUNK_HOME/etc/system/local に置いてください。例は、outputs.conf.example を参照してください。
#      設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
#      http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
#      照してください。

```

```

#
#     注意： outputs.conf を分散 Splunk デプロイメントのフォワーディング側に置いてください。
#     分散設定についての詳細は、TODO のドキュメントを参照してください。下記の文章は最新のドキュメント：http://www.splunk.com/doc/latest/admin/ForwardingReceiving にはありません。
#####
#----TCP 出力 ----
#####
#     これらの設定は、特定のターゲットグループで上書きされない場合に使用されます。
#     ターゲットグループメタデータがないすべてのイベントはこのグループに送信されます。
#     複数のグループが指定された場合、そのイベントはクローニングされすべてリストアップされます。
[tcput]
defaultGroup= Group1, Group2, ...
attribute1 = val1
attribute2 = val2
...
#     TODO：下記の機能を低下させます。syslog を使用してください。
#     注意：これは一般的な使用方法ではありません：
#     この設定項目は<key>についてイベントを見ます。イベントがこのキーを含む場合、その値は、宛先サーバに送信されるロ
ーデータの先頭に追加されます。これは、'sendCookedData = false'の場合にみ有効になることに注意してください。キー/値ベ
アおよびその引き出し方は、props.conf と transforms.conf で設定します。
#     使用ケース：syslog ファイルを監視し、それを syslog サーバーに送信して得た syslog イベントに<priority>を追加
します。
prependKeyToRaw = key
#     このパラメータは、トップレベル [tcput] スタンザからのみ利用可能です。ターゲットグループで上書きできません。
indexAndForward = true | false
*     他のアクションに加えて、このデータをローカルですべてインデックス化し、それをフォワーディングします。
*     これはインデックスおよび転送設定として知られています。
*     デフォルトは false。
#     ----ターゲットグループ設定----
#     ターゲットグループはいくつでも持つことができます。
#     複数のグループが指定された場合、フォワードはすべてのイベントを各ターゲットグループにクローニングします。
[tcput:$TARGET_GROUP]
server=$IP:$PORT, $IP2:$PORT2...
attribute1 = val1
attribute2 = val2
...
#     ----シングルサーバー設定----
#     ここで、個々のインデックスマシン特有の設定をサーバー別に定義することができます。ただし、1つのターゲットグルー

```

プまたはデフォルトグループで、データを送信するために1つのサーバを含める必要があります。

```
[tcpout-server://$IP:$PORT]
```

```
attribute1 = val1
```

```
attribute2 = val2
```

```
...
```

```
# #----オプション設定----
```

```
# これらの属性はオプションです。
```

```
sendCookedData = true | false
```

```
* true の場合、イベントは処理されます (Splunk により処理され、ローの状態ではない)。
```

```
* false の場合、イベントはローの状態で、送信前には何の処理もされません。
```

```
* サードパーティシステムに送信する場合は、false に設定してください。
```

```
* デフォルトは true。
```

```
heartbeatFrequency = <整数>
```

```
* ハートビートパケットを受信ホストに送信する頻度を秒で指定。
```

```
* ハートビートは、'sendCookedData' は true の場合にのみ送信されます。
```

```
* デフォルトは 30 です。
```

```
blockOnCloning = true | false
```

```
* true の場合、TcpOutputProcessor は、少なくとも1つのクローニングされたグループがイベントを取得するまでブロックします。これは、すべてのクローニンググループがダウンした場合、イベントをドロップしません。
```

```
* false の場合、TcpOutputProcessor は、すべてのクローニンググループがダウンし、各クローニンググループのキューがいっぱいになると、イベントをドロップします。少なくとも1つのクローニンググループがアップしキューがいっぱいになると、イベントはドロップされません。
```

```
* デフォルトは true。
```

```
compressed = true | false
```

```
* 圧縮されたデータを送るかどうかを指定します。
```

```
* デフォルトは false。
```

```
* これが true に設定されていると、受信ホストポートは圧縮をオンにします。
```

```
#----キュー設定----
```

```
maxQueueSize = <整数>
```

```
* フォワーディングサーバでキューに入ったイベントの最大数(キューサイズ)。
```

```
* デフォルトは 1000。
```

```
dropEventsOnQueueFull = <整数>
```

```
* プラスの数値に設定されると、キューにスペースができるまで、すべての新規イベントを捨てる前に N * 5 秒待ちます。
```

```
* これを-1 または 0 に設定すると、キューがいっぱいになるとブロックします。これにより、プロセスサチエーション遮断します。
```

```
* ターゲットグループキューにブロックされたものがあると、データはそれ以上他のターゲットグループに到達しなくなりません。
```

```
* ロードバランシンググループの利用は、この状態を軽減するための最良の方法です。なぜなら、複数の受信ホストがダウンしない限り、キューブロッキングが生じないからです。
```

* デフォルトは-1(イベントをドロップしない)。

* ファイルを監視する場合は、この値をプラスの整数(true)に設定しないでください！

#----バックオフ設定---

このセクションでの設定は、インデックスマシンが利用できなくなったとき、フォワーダがどのように再試行するかを決定します。

backoffAtStartup = <整数>

* 最初の再試行までに必要な待ち時間を秒数で設定します。

* デフォルトは 5。

initialBackoff = <整数>

* 最初の再試行の後、毎回再試行するまでの待ち時間を秒数で設定します。

* デフォルトは 2。

maxNumberOfRetriesAtHighestBackoff = <整数>

* 最大バックオフ期間に到達した後、完全に停止する前に、システムが再試行する回数を指定します。

* -1 は永久に再試行します。

* Splunk は、デフォルトから変更しないことを推奨します。変更すると、フォワーダは、ダウンした URI への転送をある時点で完全に停止します。

* デフォルトは-1(永久)。

maxBackoff = <整数>

* 最大バックオフ頻度に到達するまでの秒数を指定します。

* デフォルトは 20。

#----フォワーダの自動ロードバランシング

autoLB = true | false

* true に設定すると、フォワーダは自動ロードバランシングモードに切り替わります。このモードでは、フォワーダは、autoLBFrequency 毎にランダムに新規インデックスマシンを選択します。インデックスマシンへの接続がある時点で失われた場合、新しい有効なインデックスマシンを選択し、データをそれに転送します。

* このフラグがない場合、フォワーダはロードバランシングをラウンドロビン戦略で使用します。

* デフォルトは false。

autoLBFrequency = <秒>

* これは自動ロードバランシングモードで使用されます。autoLBFrequency 毎に、新規インデックスマシンが、サーバーパラメータで提供されたインデックスマシンのリストからランダムに選択されます。

* デフォルトは 3600 秒(1 時間)。

#----SSL 設定---

フォワーダで SSL を設定するには、次の属性/値ペアを設定します。

認証のために SSL を使用する場合は、認証する必要のある各受信ホストについてスタンザを追加します。

sslPassword = <パスワード>

* CAcert に関連するパスワード。

* デフォルト Splunk CAcert は "password" というパスワードを使います。

* デフォルト値はなし。

```

sslCertPath = <パス>
*     これを指定すると、この接続は SSL を使います。
*     これはクライアント証明書へのパスです。
*     デフォルト値はなし。
sslRootCAPath = <パス>
*     ルート証明書権限ファイルへのパス(オプション)。
*     TODO: default = ?
sslVerifyServerCert = true | false
*     true に設定する場合、接続しているサーバーが有効なものであるかどうかを確認します(認定)。
*     次に、サーバーの共通名および代替名は一致するかどうかチェックされます。
*     デフォルトは false。
sslCommonNameToCheck = <文字列>
*     この名前に対してサーバーの証明書の共通名をチェックします。
*     一致しない場合、Splunk はこのサーバーに対して認定されていないと考えられます。
*     「sslVerifyServerCert」が true の場合、この設定を指定する必要があります。
altCommonNameToCheck = <文字列>
*     この名前に対してサーバーの証明書の代替名をチェックします。
*     一致しない場合、Splunk はこのサーバーに対して認定されていないと考えられます。
*     「sslVerifyServerCert」が true の場合、この設定を指定する必要があります。
#####
#---- syslog 出力---
#####
#     次の設定は syslog は syslog を使用して出力を送信するために使います。
[syslog]
defaultGroup = Group1, Group2, ...
[syslog:$TARGET_GROUP]
attribute1 = val1
attribute2 = val2
...
#----必要な設定----
#     syslog 出力に必要な設定：
server = ip/servername:<ポート>
*     syslog サーバーが実行されている IP またはサーバ名
*     syslog サーバーがリスニングしているポート。
*     デフォルト値はなし。ポートを指定する必要があります。syslog はデフォルトで 514 を使用。
#----オプション設定----
#     syslog 出力に必要なオプション設定：
type = tcp | udp

```

```

*      使用プロトコル。タイプが指定されない場合、デフォルトは udp。
priority = <ddd>
*      ddd は、syslog ヘッダで<ddd>として表示される値。
*      ユーザーは ddd を (<facility> * 8) + <severity>で計算します。
*      facility が 4(セキュリティ/認証メッセージ)、severity が 2(Critical:危険状態)、優先値は 34 = (4 * 8) + 2
となります。
*      TODO: default = ?
syslogSourceType = <string>
*      string は syslog のソースタイプを示します。
*      この属性がない場合、「sourcetype::syslog」は syslog メッセージのソースタイプとなります。
timestampformat = <%b %e %H:%M:%S>
*      これが指定されると、そのフォーマットは、タイムスタンプをヘッダに追加する際に使用されます。
*      TODO: default = ?
#---- syslog サーバーへのデータのルーティング----
データを syslog サーバーにルーティングする場合：
1)まず、どのイベントをどのサーバーにルーティングするかを決定します。
2)次に、フォワーディングサーバーにある props.conf、transforms.conf、および outputs.conf を編集します。
$SPLUNK_HOME/etc/system/local/props.conf を編集し、次の TRANSFORMS-routing の属性を設定します。
[<spec>]
TRANSFORMS-routing=$UNIQUE_STANZA_NAME
<spec> には、以下が指定できます。
<sourcetype>、イベントのソースタイプ
host::<host>、<host>はイベントに対するホスト
source::<source>、<source>はイベントに対するソース
transforms.conf でエントリーを作成する際は、$UNIQUE_STANZA_NAME を使用します。
$SPLUNK_HOME/etc/system/local/transforms.conf を編集し、props.conf スタンザを一致させるルールを設定します。
[$UNIQUE_STANZA_NAME]
REGEX=$YOUR_REGEX
DEST_KEY=_SYSLOG_ROUTING
FORMAT=$UNIQUE_GROUP_NAME
$UNIQUE_STANZA_NAME は、props.conf で作成した名前と一致する必要があります。
$YOUR_REGEX で正規表現のルールを入力し、どのイベントを条件によりルーティングするかを決定します。
DEST_KEY は、イベントを SYSLOG を通じて送信するために、_SYSLOG_ROUTING に設定する必要があります。
FORMAT を $UNIQUE_GROUP_NAME に設定します。これは、outputs.conf で作成したグループ名に一致する必要があります。
#####
#---- HTTP 出力---
#####
#      次の設定は出力を HTTP を通じて送信するために使用します。

```

```
[httpoutput]
defaultGroup = Group1, Group2, ...
[httpoutput:$TARGET_GROUP]
attribute1 = val1
attribute2 = val2
...
#----必要な設定----
# HTTP 出力に必要な設定：
username = <ユーザー名>
* ユーザー名は Splunk インデックスマシンに対する認証で使用されます。
password = <パスワード>
* パスワードは Splunk インデックスマシンに対する認証で使用されます。
server = ip/servername:port
* Splunk 受信ホストの ip/servername
* port は Splunk 受信ホストがリスニングするポートです。
#----オプション設定----
# HTTP 出力に必要なオプション設定：
ssl = true | false
* HTTP 出力の SSL を設定します。
* デフォルトは true。
#---- http でデータを Splunk インスタンスにルーティング----
データを Splunk サーバーにルーティングする場合：
1)まず、どのイベントをどのサーバーにルーティングするかを決定します。
2)次に、フォワーディングサーバーにある props.conf、transforms.conf、および outputs.conf を編集します。
$SPLUNK_HOME/etc/system/local/props.conf を編集し、次の TRANSFORMS-routing の属性を設定します。
[<spec>]
TRANSFORMS-routing=$UNIQUE_STANZA_NAME
<spec> に、以下が指定できます。
<sourcetype>、イベントのソースタイプ
host::<host>、<host>はイベントに対するホスト
source::<source>、<source>はイベントに対するソース
transforms.conf でエントリーを作成する際は、$UNIQUE_STANZA_NAME を使用します。
$SPLUNK_HOME/etc/system/local/transforms.conf を編集し、props.conf スタンザを一致させるルールを設定します。
[$UNIQUE_STANZA_NAME]
REGEX=$YOUR_REGEX
DEST_KEY=_HTTP_ROUTING
FORMAT=$UNIQUE_GROUP_NAME
$UNIQUE_STANZA_NAME は、props.conf で作成した名前と一致する必要があります。
```

\$YOUR_REGEX で正規表現のルールを入力し、どのイベントを条件によりルーティングするかを決定します。

DEST_KEY は、イベントを HTTP を通じて送信するために_HTTP_ROUTING に設定する必要があります。

FORMAT を\$UNIQUE_GROUP_NAME に設定します。これは、outputs で作成した syslog グループ名に一致する必要があります。

#####

#---- インデックスおよび転送----

#####

IndexAndForward プロセッサは、データをインデキシングするためのデフォルトの動作を決定します。

フォワーダ(tcpout、httpoutput)に設定すると、'index'を'false'にします。

フォワーダが設定されていない時、'index'は'true'に設定されます。

#

tcpout スタンザが'indexAndForward'で設定される場合、'index'の値は'indexAndForward'の値に設定されます。

#

'index'の設定は、前に決定された値を上書きする[indexAndForward] スタンザで上書きすることができます。

[indexAndForward]

index = true | false

* true に設定すると、データはインデックス化されます。

* false に設定すると、データはインデックス化されません。

outputs.conf.example

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

#

このファイルは outputs.conf の例を記載します。このファイルを使用して、分散設定でフォワーディングを設定します。

#

この設定の1つまたは複数を使用するには、その設定ブロックを\$SPLUNK_HOME/etc/system/localの outputs.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。

#

設定ファイル(優先順位を含む)についての詳細は、

<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork> にあるドキュメントを参照してください。

一つの受信ホストからなる IP:PORT のターゲットグループを指定します。

これは最も簡単な設定です。データをポート 9997 の 10.1.1.197 でホストに送信します。

[tcpout:group1]

server=10.1.1.197:9997

1つの受信ホストからなるホスト名のターゲットグループを指定します。

[tcpout:group2]

server=myhost.Splunk.com:9997

2つの受信ホストからなるターゲットグループを指定します。この場合、データは、これら2つの受信ホストでバランス(ラウンドロビン)されます。受信ホストはいくつでも指定することができます。必要に応じて、ホスト名と IP を同じにすることができます。

```

す。
#       注意： この設定は SplunkLightForwarder で使用しないでください。
[tcpout:group3]
server=myhost.Splunk.com:9997,10.1.1.197:6666
#       グローバル設定値を各ターゲットグループベースで上書きすることができます。
#       グローバル設定を上書きしないすべてのターゲットグループはグローバル設定を継承します。
#       すべてのイベントを受信ホストに foo.Splunk.com:9997 で送信し、ハートビートを 45 秒毎に、最大キューサイズ
100,500 イベントで送信します。
[tcpout:group4]
server=foo.Splunk.com:9997
heartbeatFrequency=45
maxQueueSize=100500
#       ハービート頻度を各グループ 15 に設定し、イベントをグループ indexer1 および indexer2 にクローニングします。また、
このすべてのデータをローカルでインデックスします。
[tcpout]
heartbeatFrequency=15
indexAndForward=true
[tcpout:indexer1]
server=Y.Y.Y.Y:9997
[tcpout:indexer2]
server=X.X.X.X:6666
#       Y.Y.Y.Y と X.X.X.X の間のラウンドロビンデータバランス
[tcpout:indexerGroup]
server=Y.Y.Y.Y:9997, X.X.X.X:6666
#       2 つのデータのバランスされたグループ間のイベントのクローニング
[tcpout:indexer1]
server=A.A.A.A:1111, B.B.B.B:2222
[tcpout:indexer2]
server=C.C.C.C:3333, D.D.D.D:4444
#       Syslog 出力設定
#       この例は、Splunk デーモンが生成したイベントのみをリモート syslog ホストに送信します。
[syslog:syslog-out1]
disabled = false
server = X.X.X.X:9099
type = tcp
priority = 34
timestampformat = %b %e %H:%M:%S
#       HTTP 出力設定

```

```

[httpoutput:httpout1]
server=indexer1:8089
ssl = true
username=admin
password=changeme
#       4.0 で新規：自動ロードバランシング
#
#       この例は、1.2.3.4:4433 と 1.2.4.5:4433.上で実行する 2 つのインデックスマシン間の出力をバランスします。
#       これを行うには、インデックスマシンの 2 つの IP アドレスにポイントしている splunkLB の DNS エントリーを作成します。
#
#       $ORIGIN example.com.
#       splunkLB A 1.2.3.4
#       splunkLB A 1.2.3.5
[tcpout]
defaultGroup = lb
[tcpout:lb]
server = splunkLB.example.com:4433
autoLB = true
#       代わりに、sans DNS を autoLB することができます：
[tcpout]
defaultGroup = lb
[tcpout:lb]
server = 1.2.3.4:4433, 1.2.3.5:4433
autoLB = true
#       圧縮
#
#       この例は圧縮されたイベントをリモートインデックスマシンに送信します。
#       注意： 圧縮は、TCP または SSL 出力でのみ有効にできます。
#       受信ホスト入力ポートもまた圧縮が有効である必要があります。
[tcpout]
server = splunkServer.example.com:4433
compressed = true
#       SSL
#
#       この例はイベントを、Splunk の自己署名認定書を使って SSL を通じてインデックスマシンに送信します。
[tcpout]
server = splunkServer.example.com:4433
sslPassword = password

```

```

sslCertPath = $SPLUNK_HOME/etc/auth/server.pem
sslRootCAPath = $SPLUNK_HOME/etc/auth/cacert.pem
#
# 次の例は、イベントの syslog サーバーへのルーティング方法を示します。
# これは tcpout ルーティングに似ていますが、DEST_KEY を _SYSLOG_ROUTING に設定する部分のみが異なります。
#
1. $SPLUNK_HOME/etc/system/local/props.conf を編集し、TRANSFORMS-routing= の属性を設定します：
[default]
TRANSFORMS-routing=errorRouting

[syslog]
TRANSFORMS-routing=syslogRouting

2. $SPLUNK_HOME/etc/system/local/transforms.conf を編集し、errorRouting と syslogRouting のルールを設定します：
[errorRouting]
REGEX=error
DEST_KEY=_SYSLOG_ROUTING
FORMAT=errorGroup

[syslogRouting]
REGEX=.
DEST_KEY=_SYSLOG_ROUTING
FORMAT=syslogGroup

3. $SPLUNK_HOME/etc/system/local/outputs.conf を編集し、どの syslog 出力がどのサーバーまたはグループに行くかを設定します。
[syslog]
defaultGroup=everythingElseGroup

[syslog:syslogGroup]
server = 10.1.1.197:9997

[syslog:errorGroup]
server=10.1.1.200:9999

[syslog:everythingElseGroup]
server=10.1.1.250:6666
#
# 次の例は、HTTP を使用してイベントを Splunk インスタンスにルーティングする方法を示します。
# これは tcpout ルーティングに似ていますが、DEST_KEY を _SYSLOG_ROUTING に設定する部分のみが異なります。
#
1. $SPLUNK_HOME/etc/system/local/props.conf を編集し、TRANSFORMS-routing= の属性を設定します：
[default]
TRANSFORMS-routing=errorRouting

```

```
[syslog]
```

```
TRANSFORMS-routing=httpRouting
```

2. `$(SPLUNK_HOME)/etc/system/local/transforms.conf` を編集し、`errorRouting` と `httpRouting` のルールを設定します。

```
[errorRouting]
```

```
REGEX=error
```

```
DEST_KEY=_HTTP_ROUTING
```

```
FORMAT=errorGroup
```

```
[httpRouting]
```

```
REGEX=.
```

```
DEST_KEY=_HTTP_ROUTING
```

```
FORMAT=httpGroup
```

3. `$(SPLUNK_HOME)/etc/system/local/outputs.conf` を編集し、`http` 出力が出力されるサーバーまたはグループを設定します。[`httpoutput`]

```
defaultGroup=everythingElseGroup
```

```
[httpoutput:httpGroup]
```

```
server=10.1.1.197:8089
```

```
ssl = true
```

```
username=admin
```

```
password=changeme
```

```
[httpoutput:errorGroup]
```

```
server=10.1.1.200:8089
```

```
ssl = true
```

```
username=admin
```

```
password=changeme
```

```
[httpoutput:everythingElseGroup]
```

```
server=10.1.1.250:8089
```

```
ssl = true
```

```
username=admin
```

```
password=changeme
```

procmon-filters.conf

procmon-filters.conf

次は、`procmon-filters.conf` の仕様とファイル例です。

procmon-filters.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
#     このファイルは、Windows レジストリモニタリングを設定する際に使用する可能性のある属性/値ペアを記載しています。
procmon-filters.conf は、sysmon.conf と連携して使用され、Splunk に監視させたいプロセスを調整しフィルタリングするた
めに作成した特定の正規表現を含みます。設定を有効にするには Splunk の再起動が必要です。
```

```
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
```

```
[<スタンザ名>]
```

```
*     定義されるフィルタ名。
```

```
proc = <文字列>
```

```
*     Splunk に監視させたいプロセスイメージを指定する正規表現。
```

```
type = <文字列>
```

```
*     Splunk に監視させたいプロセスイベントのタイプを指定する正規表現。
```

```
これは、regmon-filters の event_types attribute で定義されたサブセットである必要があります。
```

```
hive = <文字列>
```

```
*     このコンテンツでは使用されないが、常に値".*"を持っている必要があります。
```

procmon-filters.conf.example

```
#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
#     このファイルはレジストリモニタフィルタの例を記載します。自分のフィルタを作成するには、
procmon-filters.conf.spec にある情報を使用してください。
```

```
#
```

```
#     この設定の1つまたは複数を使用するには、その設定ブロックを$SPLUNK_HOME/etc/system/local の
procmon-filters.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
#     設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
```

```
[default]
```

```
hive = .*
```

```
[not-splunk-optimize]
```

```
proc = (?<!splunk-optimize.exe)$
```

```
type = create|exit|image
```

props.conf

props.conf

次は props.conf の仕様とファイル例です。

props.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、props.conf で Splunk の処理プロパティを設定する際の属性/値ペアを記載しています。
#
# props.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、props.conf を
# $SPLUNK_HOME/etc/system/local に置いてください。ヘルプは、props.conf.example を参照してください。
# props.conf に施した設定の変更は、次の検索文字列を Splunk Web で入力することにより有効化できます：
#
# | extract reload=T
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
```

[<spec>]

- * このスタンザは、所与の<spec>についてプロパティを有効にします。
- * props.conf ファイルは、任意の数の異なる<spec>について複数のスタンザを含むことができます。
- * このスタンザ名に、次の属性/値のペアを続けてください。
- * 所与の<spec>について属性を設定しない場合、デフォルトが使用されます。

<spec> には、以下を指定できます。

1. <sourcetype>、イベントのソースタイプ
2. host::<<host>、<host>はイベントに対するホスト
3. source::<<source>、<source>はイベントに対するソース。
4. rule::<<rulename>、<rulename>はソースタイプ分類ルールに固有の名前です。
5. delayedrule::<<rulename>、<rulename>は遅延ソースタイプ分類ルールに固有の名前です。

これらは、表示されたソースに基づいて新規ソースタイプを生成する前の、最後の手段とみなされます。

優先順位：

マッチングスタンザの複数カテゴリーで指定された設定については、host:: spec 設定は sourcetype:: spec 設定を上書きしま
す。さらに、source:: spec 設定は、host:: および sourcetype:: 設定の両方を上書きします。

注意： <spec>を設定する際には、次の正規表現タイプ構文を使用してください。

... =一致するまでディレクトリを再帰的に読み込みます。

* = /以外に 0 回以上一致します。

| = or

() = | の範囲を制限するために使用されます。

例: [source::...(?<!tar.)(gz|tgz)]

言語に一致させます。

これらの一致表現は、サブストリングだけでなく、キー値全体に一致する必要があります。

正規表現をご存知の方は、これらは...、* および .を言い換えた PCRE の完全実装です。

したがって、.はピリオドに一致し、*はディレクトリ以外のセパレーター、...は任意の数の任意の文字に一致します。

詳細は、<http://www.splunk.com/base/Documentation/latest/Admin/FilesAndDirectories#Inputs.conf> のワイルドカードセクションを参照してください。

props.conf の属性/値ペアおよびそのデフォルト値:

国際文字

CHARSET = <文字列>

- * 設定すると、Splunk は、指定した<spec>からの入力が指定されたエンコーディングであると見なします。
- * 有効なエンコーディングのリストは、ほとんどの*nix システム上でコマンド"iconv -l"を使って取得できます。
- * 無効なエンコーディングが指定された場合、警告が最初の設定の際に記録され、その<spec>からの以後の入力は廃棄されます。
- * 指定されたエンコーディングにおいて、ソースエンコーディングが有効で、<spec>からの文字のいくつかが無効でない場合、この文字は 16 進数としてエスケープされます(例: "\xF3")。
- * "AUTO" に設定すると、Splunk は自動的に文字エンコーディングを決定し、そのエンコーディングからのテキストを UTF-8 に変換します。
- * Splunk が自動的に検出する文字セットの全リストはオンラインドキュメントを参照してください。
- * デフォルトは.ASCII。

改行

行の長さを定義するには次の属性を使用します。

TRUNCATE = <マイナスでない整数>

- * デフォルトの最大行長を変更します。
- * 切り捨てたくない場合は、0 を設定します(ただし、非常に長い行はゴミデータの兆候)。
- * デフォルトは 10000。

LINE_BREAKER = <正規表現>

- * ラインマーキングが生じる前に、ローテキストストリームがどのように最初のイベントに分解されるかを決定する正規表現を指定します。(SHOULD_LINEMERGE を参照)
- * デフォルトは([\r\n]+)。これは、データが\r または \n で区切られて、各行で分解されます。
- * 正規表現はマッチンググループを含んでいる必要があります。
- * 正規表現はマッチすると、最初のマッチグループの最初は、前のイベントの最後と見なされ、最初のマッチグループの最後は次のイベントの最初と見なされます。

```

*     最初のマッチグループのコンテンツはイベントテキストとして無視されます。
*     注意： ラインマーキングを使用して個々の行をイベントに再結合するよりも、LINE_BREAKER を使って複数行のイベント
を区切る方が、大幅な速度改善がみられます。
LINE_BREAKER_LOOKBEHIND = <整数>
*     正規表現ベースの改行コードのデフォルト後方参照を変更します。
*     前のローチャンクのデータが残っている場合、これで、正規表現の適用を開始するローチャンクの最後(次のチャンクが連結
されて)までの長さを示します。
*     デフォルトは 100。
#     複数行のイベントを追加の属性と値で定義するには、次の属性を使用します。
SHOULD_LINEMERGE = true | false
*     true に設定すると、Splunk は、データの数行を、次の設定属性に基づいて 1 つのイベントに結合します。
*     デフォルトは true。
#     SHOULD_LINEMERGE = True の場合、次の属性を使用して複数行イベントを定義します。
BREAK_ONLY_BEFORE_DATE = true | false
*     true に設定すると、Splunk は、ある日付を持つ新しい行に遭遇した場合にのみ新規イベントを作成します。
*     デフォルトは false。
BREAK_ONLY_BEFORE = <正規表現>
*     設定すると、Splunk は、正規表現にマッチする新しい行に遭遇した場合にのみ、新規イベントを作成します。
*     デフォルトは空白です。
MUST_BREAK_AFTER = <正規表現>
*     設定した場合に、正規表現が現在の行と一致すると、Splunk は、次の入力行のために新規イベントを作成します。
*     Splunk は、別のルールが一致する場合、現在の行の前で改行します。
*     デフォルトは空白です。
MUST_NOT_BREAK_AFTER = <正規表現>
*     設定した場合、現在の行が正規表現と一致すると、Splunk は、MUST_BREAK_AFTER 表現が一致するまで、次の行を改行し
ません。
*     デフォルトは空白です。
MUST_NOT_BREAK_BEFORE = <正規表現>
*     設定した場合、現在の行が正規表現と一致すると、Splunk は、現在の行の前の最後のイベントを改行しません。
*     デフォルトは空白です。
MAX_EVENTS = <整数>
*     イベントに追加する入力の最大行数を指定します。
*     Splunk は、指定された行数が読み込まれた後に改行します。
*     デフォルトは 256。
*****
#     タイムスタンプ抽出設定
*****
DATETIME_CONFIG = <$SPLUNK_HOME に対する相対ファイル名>

```

- * タイムスタンプエクストラクターを設定するファイルを指定します。
- * この設定はまた、"NONE"に設定し、タイムスタンプエクストラクターが実行されるのを防止することもでき、また、"CURRENT"に指定して、現在のシステム時間を各イベントに割り当てることもできます。
- * デフォルトは/etc/datetime.xml(例：\$SPLUNK_HOME/etc/datetime.xml)。

MAX_TIMESTAMP_LOOKAHEAD = <整数>

- * Splunk でタイムスタンプを探すイベントの長さ(文字数)を指定します。
- * デフォルトは 150。

TIME_PREFIX = <正規表現>

- * タイムスタンプ抽出の必要条件を指定します。
- * タイムスタンピングアルゴリズムのみが、最初の正規表現一致の後、タイムスタンプを探します。
- * デフォルトは空白です。

TIME_FORMAT = <strftime 型フォーマット>

- * 日付を抽出するための strftime フォーマット文字列を指定します。
- * strftime の詳細は、Splunk 管理マニュアルの`man strftime`または"タイムスタンプ認識の設定"を参照してください。
- * この日付抽出の方法は、in-event タイムゾーンをサポートしません。
- * TIME_FORMAT は、TIME_PREFIX の後に読み込みを開始します。
- * 正しい結果を得るには、<strftime 型フォーマット>は、その年の日とその日の時間を記述する必要があります。
- * デフォルトは空白です。

TZ = <タイムゾーン識別子>

- * 特定のイベントに関するタイムゾーンを決定するためのアルゴリズムは次のとおり：
- * イベントがローテキストにタイムゾーンを持つ場合(例：UTC, -08:00)、それを使います。
- * TZ が有効タイムゾーン文字列に設定される場合、それを使います。
- * その他の場合は、Splunkd を実行しているシステムのタイムゾーンを使います。
- * デフォルトは空白です。

MAX_DAYS_AGO = <整数>

- * 抽出された日付を有効にするために、現在の日付から、過去の日数の最大数を指定します。
- * 10 に設定すると、例えば、Splunk は 10 日より古い日付を無視します。
- * デフォルトは 2000。
- * 重要： データが 2000 日より古い場合、この設定を変更すること。

MAX_DAYS_HENCE = <整数>

- * 抽出された日付を有効にするために、現在の日付から、未来の日数の最大数を指定します。
- * 3 を設定すると、例えば、3 日を超える未来の日付は無視されます。
- * 正偽は、厳しく設定するとより発生しにくくなります。
- * デフォルト値は、1 日未来からの日付を含みます。
- * サーバーの日付設定が正しくない場合や 1 日進んだタイムゾーンにある場合、この値を少なくとも 3 増加させてください。
- * デフォルトは 2。

MAX_DIFF_SECS_AGO = <整数>

- * イベントのタイムスタンプが、前のタイムスタンプの前より<整数>秒より大きいと、ソースからのタイムスタンプの大部分

と同じ正確な時間のフォーマットを持つ場合にのみ受け入れます。

* 重要：タイムスタンプが正しくない場合には、この値を増やすことを検討すること。

* デフォルトは 3600 (1 時間)。

MAX_DIFF_SECS_HENCE = <整数>

* イベントのタイムスタンプが、前のタイムスタンプの後より<整数>秒より大きいと、ソースからのタイムスタンプの大部分と同じ正確な時間のフォーマットを持つ場合にのみ受け入れます。

* 重要：タイムスタンプが正しくない場合、またはログを週に 1 度以下に書き込む場合、この値を増やすことを検討すること。

* デフォルトは 604800 (1 週間)。

transform 設定

インデックス化フィールドを作成するには TRANSFORMS クラスを使用します。抽出フィールドを作成するには REPORT クラスを使用します。

抽出フィールドが最良の方法として推奨します。

注意：インデックス化フィールドは性能上の変動があり、特定の状況においてのみ使用することを推奨します。

インデックス化フィールドは、例えば、foo!="bar" や NOT foo="bar" で、そのフィールド foo がほとんど常に値 bar をとるような表現を検索する場合に使用することが考えられます。

インデックス化フィールドを使用するもう 1 つの一般的な理由は、フィールドの値が、そのフィールドの内部よりも外部により多く存在するかどうかです。

例えば、通常 foo="1" を検索し、1 が foo="1" のない多くのイベントに 1 が生じる場合、index foo を使用することが考えられます。

詳細は、<http://www.splunk.com/doc/latest/admin/ExtractFields> にあるドキュメントを参照のこと。

例は、props.conf.spec および transforms.conf.spec を参照してください。

クラスの優先順位ルール

* 各クラスについて、Splunk は、最も高い優先順位設定ブロックから設定を取得します(このファイルの最初にある優先順位ルールを参照)。

* 特定のクラスがあるソースおよびソースタイプを指定した場合、ソースのクラスが優先します。

* 同様に、特定のクラスが<spec>について../local/で指定された場合、そのクラスを../default/で上書きします。

TRANSFORMS-<値> = <unique_stanza_name>

* <unique_stanza_name>は、transforms.conf のスタンザ名です。

* <値>は、スタンザに与えてその name-space を特定する値です。

* Transforms は指定された順番で適用されます。

* 順番を変えたい場合、リストを再整理してコントロールします。

REPORT-<値> = <unique_stanza_name>

* <unique_stanza_name>は、transforms.conf のスタンザ名です。

* <値>は、スタンザに与えてその name-space を特定する値です。

* Transforms は指定された順番で適用されます。

* 順番を変えたい場合、リストを再整理してコントロールします。

EXTRACT<クラス> = <正規表現> (in <src_field>)?

- * 正規表現ベースフィールド抽出をソースフィールドの値で実行します。
- * 正規表現は、グループをキャプチャできるように命名する必要があります。
- * 正規表現が命名されたキャプチャリンググループと一致すると、それらの値がそのイベントに追加されます。
- * 注意： この抽出は、検索時にのみ実行されます。
- * このとき、
- * 正規表現： perl 互換正規表現であり、命名キャプチャリンググループを含む。
- * src_field： 正規表現に一致するフィールド名(デフォルトは_raw)。

KV_MODE = none | auto | multi

- * データのキー/値の抽出モードを指定します。
- * KV_MODE 次のうちいずれかに設定します。
- * none： キー/値抽の出が必要ない場合。
- * auto： 等号で区切られたキー/値ペアを抽出。
- * multi： multikv を呼び出し、表イベントを複数イベントに拡張。
- * デフォルトは auto。

CHECK_FOR_HEADER = true | false

- * あるファイルについてヘッダーベースのフィールド抽出を有効にするには true に設定します。
- * ファイルが列のリストを持ち、各イベントがフィールド値(フィールド名なし)を含む場合、Splunk は、フィールド名を抽出するために適切なヘッダ行を取り上げます。
- * デフォルトは false。

SEDCMD<クラス> = <sed スクリプト>

- * インデックス時間のみに、_raw フィールドに適用する sed スクリプトを指定。
- * sed スクリプトは、sed コマンドのスペースで区切られたリストです。
- * 現在、次の sed コマンドサブセットがサポートされています。

置換(s)および文字置き換え(y)

- * 構文：
- * 置換 - s/regex/replacement/flags
- * ここで、正規表現は perl 正規表現である(オプションでキャプチャリンググループを含む)
- * 置換は正規表現一致を置き換える文字列であり、後方参照には $\backslash N$ を使用します。
- * フラッグには次を指定できます。g はすべての一致を置換、数字は指定された一致を置換。
- * 置き換え- y/string1/string2/
- * string1[i]を string2[i]で置き換えます。

LOOKUP<クラス> = \$TRANSFORM (<match_field> (AS <match_field_in_event>)?)+ (OUTPUT|OUTPUTNEW (<output_field> (AS <output_field_in_event>)?)+)?

- * 特定のルックアップテーブルおよびそのルックアップテーブルをイベントに適用する方法を指定します。
- * <match_field>は、一致するルックアップテーブルのフィールドを指定します。
- * デフォルトでは、一致するイベントの同じ名前を持つフィールドを探します(<match_field_in_event>が指定されない場合)。

- * 複数一致フィールドを指定することができる。1つ以上が必要です。
- * <output_field>は、各マッチングイベントにコピーするためのルックアップエントリーのフィールドを指定します。各マッチングイベントはフィールド<output_field_in_event>にあります。
- * これが指定されない場合、<output_field>が使われます。
- * 出力フィールドのリストは必要ありません。
- * 指定されない場合、一致フィールド(およびタイムスタンプフィールド(指定された場合))以外のルックアップテーブル内のすべてのフィールドが、各マッチングイベントに出力されます。
- * 出力フィールドリストが、キーワード"OUTPUT"の代わりに"OUTPUTNEW"で開始する場合、そのルックアップは、出力フィールドがすでにそのイベントに存在しない場合のみ適用されます。その他の場合、出力フィールドは常に上書きされます。

match_fields のすべてを持ち、ルックアップテーブルにマッチングエントリーを持たないイベントは、出力フィールドのすべてをクリアします。

FIELDALIAS<クラス> = (<orig_field> AS <new_field>)+

- * 新規フィールドとしてエイリアスするフィールドのリスト。
- * 両方のフィールドが存在、つまり、元のフィールドは削除されません。
- * フィールドエイリアシングは、kv 抽出の後、ルックアップの前に実行されます。
- * したがって、フィールドエイリアスに基づいてルックアップを指定することが可能です。
- * さらに、検索時に抽出されたフィールドはエイリアスすることが可能です。

バイナリファイル設定

NO_BINARY_CHECK = true | false

- * [source::...] スタンザについてのみ設定できます。
- * true に設定すると、Splunk はバイナリファイルを処理します。
- * デフォルトでは、バイナリファイルは無視されます。
- * デフォルトは false。

セグメンテーション設定

SEGMENTATION = <文字列>

- * インデックスタイムに使用するために、segmenters.conf からのセグメンタを指定します。
- * このファイルの始めに概要説明した<spec>のためのセグメンテーションを設定します。

SEGMENTATION-<セグメント選択> = <文字列>

- * Splunk Web が、与えられた<セグメント選択>選択で特定したセグメントを(segmenters.conf から)使用するよう指定します。
- * デフォルトの<セグメント選択>選択: all, inner, outer, none.

ファイルチェックサム設定

CHECK_METHOD = entire_md5 | modtime

* デフォルトでは、ファイルの最初および最後の 256 バイトのチェックサムが既存の保存されたチェックサムと一致すると、Splunk は、すでにインデックス化されたものとしてリストアップするため、それは無視されます。

* ファイル全体のチェックサムを使用するには、これを "entire_md5" に設定します。

* または、ファイルの修正時間のみチェックするには、これを "modtime" に設定します。

* デフォルトは entire_md5。

小ファイル設定

PREFIX_SOURCETYPE = true | false

* 注意：この属性は、"[too_small]" ソースタイプにのみ関係します。

* 100 行未満で分類されないファイルに与えるソースタイプを決定します。

* false はソースタイプを "too_small" に設定します。

* true は、ソースタイプを "<sourcename>-too_small" に設定します。ここで、"<sourcename>" は、ファイル名のクレーンアップバージョンです。

* true の場合の利点は、小ファイルがすべて同じソースタイプに分類されないことです。また、ワイルドカード検索が有効なことです。

* 例えば、Splunk 検索 "sourcetype=access*" は、"access-too_small" ファイルと同様に "access" ファイルも取得します。

* デフォルトは true。

ソースタイプ設定

sourcetype = <文字列>

* [<ソース>::...] スタンザについてのみ設定できます。

* その<ソース>にはすべて指定されたソースタイプが割り当てられます。

* デフォルトは空白です。

次の属性/値ペアは、 [<ソースタイプ>] で始まるスタンザのみに設定することができます。

rename = <文字列>

* <ソースタイプ>を<文字列>として名前を変更します。

* 名前を変更すると、そのソースタイプを sourcetype=<文字列>で検索することができます。

* 名前の変更なしでオリジナルソースタイプを検索するには、field _sourcetype を使用します。

invalid_cause = <文字列>

* [<ソースタイプ>] スタンザについてのみ設定できます。

* Splunk は invalid_cause 設定されたデータはインデックス化しません。

* ファイルを "archive" してアーカイブプロセッサに送るには、<文字列>を設定します (unarchive_cmd で設定)。

* Splunklogger をデバックモードで実行している場合、エラーを splunkd.log に送るには、他の文字列に設定します。

* デフォルトは空白です。

```

is_valid = true | false
*     invalid_cause で自動設定されます。
*     これは設定しないでください。
*     デフォルトは true。
unarchive_cmd = <文字列>
*     invalid_cause が "archive" に設定されている場合にのみ呼び出されます。
*     <文字列>は、アーカイブされたソースに抽出するために実行するシェルコマンドを指定します。
*     stdin で入力を受け、stdout で出力を生成するシェルコマンドである必要があります
*     バッチ処理ファイルで行わないでください。preprocessing_script を使用してください。
*     デフォルトは空白です。
LEARN_MODEL = true | false
*     ソースタイプが分かる場合は、fileclassifier はモデルファイルを learned ディレクトリに追加します。
*     多様なソースタイプについてこの動作を無効にするには(ソースタイプの作成に適切な原型がないソースコードなど)、
LEARN_MODEL = false に設定します。
*     デフォルトは空白です。
maxDist = <整数>
*     ソースタイプと現在のファイルの異なる度合いを決定します。
*     値が大きくなるほど、許容度が大きくなります。
*     例えば、この値が非常に小さい場合(例: 10)、指定されたソースタイプのファイルは非常に異なります。
*     大きな値を設定した場合は、与えられたソースタイプのファイルはあまり異なりません。
*     デフォルトは 300。
#     rule::and delayedrule::configuration
MORE_THAN<optional_unique_value>_<数値> = <正規表現> (空白)
LESS_THAN<optional_unique_value>_<数値> = <正規表現> (空白)
例:
[rule::bar_some]
sourcetype = source_with_lots_ofBars
#     if more than 80% of lines have "----", but fewer than 70% have "####"
#     declare this a "source_with_lots_ofBars"
MORE_THAN_80 = ---LESS_
THAN_70 = ####
1つのルールは複数の MORE_THAN と LESS_THAN パターンを持つことができますが、すべてがそのルールに一致する必要があります。
*****
#     内部設定
*****
#     ユーザーが設定する項目ではありません。設定しないでください。
_actions = <文字列>
*     オブジェクトのユーザーインタフェースコントロールで使用する内部フィールドです。

```

- * デフォルトは "new,edit,delete"。
- pulldown_type = <論理式>
- * ソースタイプのユーザーインターフェイスコントロールで使用する内部フィールドです。
- * デフォルトは空白。

props.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# 次は props.conf 設定の例です。データのプロパティ設定します。
#
# この設定の1つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の props.conf にコ
# ピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#####
# ラインマーキング設定
#####
# 次の例は、apache_error ソースタイプに関するソースデータを複数行イベントにラインマージします。
[apache_error]
SHOULD_LINEMERGE = True
#####
# チューニングの設定
#####
# 次の例は、host::small_events のイベント別にインデックス化される文字数を制限します。
[host::small_events]
TRUNCATE = 256
# 次の例は、/mylogs/*.log で終わるパスの DATETIME_CONFIG(インデキシングのスピードを上げる)をオフにします。
[source:.../mylogs/*.log]
DATETIME_CONFIG = NONE
#####
# タイムスタンプ抽出設定
#####
# 次の例は、ホストが nyc* と一致する場合、東部時間帯に設定します。
[host::nyc*]
TZ = US/Eastern
# 次の例は、作成されたカスタム datetime.xml を使用し、カスタムディレクトリに置きます。これは、dharma で始めるホ
```

ストから来るイベントのすべてをこのカスタムを使用するように設定します。

```
[host::dharma*]
DATETIME_CONFIG = <etc/apps/custom_time/datetime.xml>
#####
#       transform 設定
#####
#       次の例は、host::foo が transforms でスタンザに固定されている場合、その検索フィールドを作成します。
[host::foo]
TRANSFORMS-foo=foobar
#       次の例は、ソースタイプ access_combined が transforms.conf でスタンザに固定されている場合、そのための抽出されたフィールドを作成します。
[eventtype::my_custom_eventtype]
REPORT-baz = foobaz
#       次のスタンザは IP アドレスを_raw から抽出します。
[my_sourcetype]
EXTRACT-extract_ip = (?<ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})
#       次の例は、ルックアップテーブルの設定方法を示します。
[my_lookuptype]
LOOKUP-foo = mylookuptable userid AS myuserid OUTPUT username AS myusername
#       次はフィールドエイリアスを指定する方法を示します。
FIELDALIAS-foo = user AS myuser id AS myid
#####
#       ソースタイプ設定
#####
#       次の例は、ファイル web_access.log のソースタイプを設定します。
[source::.../web_access.log]
sourcetype = splunk_web_access
#       次の例は、syslog イベントを解凍します。
[syslog]
invalid_cause = archive
unarchive_cmd = gzip -cd
#       次の例は、カスタムソースタイプを認識し、より小さいデフォルト maxDist で、異なる例の範囲を制限します。
[custom_sourcetype]
LEARN_MODEL = true
maxDist = 30
#       rule::and delayedrule::configuration
#       次の例は、カスタムソースタイプについて正規表現でソースタイプルールを作成します。
[rule::bar_some]
```

```

sourcetype = source_with_lots_of_bars
MORE_THAN_80 = ---
[delayedrule::baz_some]
sourcetype = my_sourcetype
LESS_THAN_70 = ####
#####
#       ファイル設定
#####
#       バイナリファイル設定
#       次の例は host::sourcecode からバイナリファイルを受け取ります。
[host::sourcecode]
NO_BINARY_CHECK = true
#       ファイルチェックサム設定
#       次の例は、似ているファイルをスキップすることなく、web_access ディレクトリにあるすべてのファイルをチェックしま
す。
[source:.../web_access/*]
CHECK_METHOD = entire_md5

```

pubsub.conf

pubsub.conf

次は pubsub.conf の仕様とファイル例です。

pubsub.conf.spec

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       このファイルは PubSub システムのクライアントを設定するための属性と値を記載します。
#
#       カスタム設定を設定するには、pubsub.conf を $SPLUNK_HOME/etc/system/local に置いてください。
#       例は、pubsub.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#####
#       deploymentServer が実行中の物理位置を設定します。
#       この設定は、PubSub システムのクライアントが使用します。
#####

```

```
[pubsub-server:deploymentServer]
disabled = <false または true>
*      デフォルトは 'false'。
targetUri = <IP:Port or hostname:Port or "direct">
*      ブローカーがリモートの場合に、リモートサーバーの URL を指定する、または単に、ブローカーが in-process の場合、キ
ーワード "direct" を指定します。
*      通常、ブローカーとデプロイメントサーバーを同じ Splunk に置くと良い結果をもたらします。このような設定では、デプ
ロイメントクライアントはすべて targetUri を deploymentServer:port に設定します。
#*****
#      次のセクションは Splunk デベロッパーにのみ関係があるものです。
#*****
#      この "direct" 設定は常に利用することができ、上書きできません。
[pubsub-server:direct]
disabled = false
targetUri = direct
[pubsub-server:<logicalName>]
*      Splunk であればすべてブローカーになることができます。複数のブローカーがある場合、それが参照するクライアントが
使用する logicalName を割り当てます。
disabled = <false または true>
*      デフォルトは 'false'。
targetUri = <IP:Port or hostname:Port or "direct">
*      ブローカーが使用している Splunk の URI。
*      キーワード "direct" は、クライアントがブローカーと同じ Splunk インスタンス上で実行中であることを意味します。
pubsub.conf.example
[pubsub-server:deploymentServer]
disabled=false
targetUri=somehost:8089
[pubsub-server:internalbroker]
disabled=false
targetUri=direct
```

regmon-filters.conf

regmon-filters.conf

次は regmon-filters.conf の仕様とファイル例です。

regmon-filters.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```

#
#     このファイルは、Windows レジストリモニタリングを設定する際に使用する可能性のある属性/値ペアを記載しています。
regmon-filters.conf ファイルは sysmon.conf と連携して使用され、Splunk に監視させたいハイブキーパスを選別しフィルタ
するために作成する特定の正規表現を含んでいます。設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[<スタンザ名>
*     定義されるフィルタ名。
proc = <文字列>
*     Splunk に監視させたいプロセスイメージを指定する正規表現。
hive = <文字列>
*     Splunk に監視させたいレジストリキーパスを指定する正規表現。
type = <文字列>
*     Splunk に監視させたいレジストリイベントのタイプを指定する正規表現。
これは、regmon-filters の event_types attribute で定義されたサブセットである必要があります。
baseline = <整数 0|1>
*     このフィルタが定義するキーのベースライン値を確立するかどうかを指定します。
baseline_interval = <整数>
*     スナップショットを再取得する前に Splunk を停止しておく必要がある時間のしきい値を秒で指定します。
disabled = <整数 0|1>
*     指定のフィルタの無効および有効を切り替えます。

regmon-filters.conf.example
#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#     このファイルはレジストリモニタフィルタの例を記載します。
#
#     この設定の1つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の
regmon-filters.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#     次は、レジストリモニタフィルタの例です。自分のフィルタを作成するには、regmon-filters.conf.spec で説明する仕
様に従って値を修正してください。
[default]

```

```
disabled = 1
baseline = 0
baseline_interval = 86400
[User keys]
proc = \\Device\\.
hive = \\REGISTRY\\USER\\.
type = set|create|delete|rename
```

restmap.conf

restmap.conf

次は restmap.conf の仕様とファイル例です。

restmap.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、新規 rest エンドポイントを作成するための属性と値のペアを記載しています。
#
#      restmap.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、restmap.conf
#      を $SPLUNK_HOME/etc/system/local に置いてください。ヘルプは、restmap.conf.example を参照してください。設定を有効
#      にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
#      http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
#      照してください。
#      注意： REST エンドポイントを利用可能にするには、すべてこのファイルにより登録する必要があります。
#####
#      グローバルスタンザ
[global]
*      このスタンザは、すべての REST エンドポイントのグローバル設定を設定します。
*      このスタンザ名に、次の属性/値のペアを続けてください。
allowGetAuth=<true | false>
*      user/password が GET パラメータとしてエンドポイント services/auth/login にパスするようにします。
*      必要に応じてこれを true に設定すると、user/password が Splunk のログおよびその間のプロキシサーバーにクリアテ
キストとして記録される恐れがあります。
*      デフォルトは false。
pythonHandlerPath=<パス>
*      'main'python スクリプトハンドラへのパス。
```

```

*       スクリプトハンドラが使用して、実際の 'main' スクリプトの位置を決定します。
*       通常、これを変更する必要はありません。
*       デフォルトは $SPLUNK_HOME/bin/rest_handler.py。
#####
#       エンドポイント毎スタンプ
#       ハンドラと他のハンドラー別設定を指定します。
#       ハンドラは、各 REST エンドポイント下にある任意のネームスペースを実装します。
[script:<uniqueName>]
*       注意: uniqueName は各ハンドラで異なっている必要があります。
*       このエンドポイントを実行するときに指定されたハンドラを呼び出します。
*       次の属性/値ペアはスクリプトハンドラをサポートします。
scripttype=python
*       このエンドポイントを使う際に実行するスクリプトのタイプをシステムに伝えます。
*       デフォルトは python。
*       Python は現在 scripttype の唯一のオプションです。
handler=<SCRIPT>.<CLASSNAME>
*       実行するファイルの名前およびクラス名。
*       ファイルは、アプリケーションの ../rest/サブディレクトリにある必要があります。
*       例えば、$SPLUNK_HOME/etc/apps/<APPNAME>/default/rest/TestHandler.py は、MyHandler というクラスを持ちます (python が 'splunk.rest.BaseRestHandler' と呼ばれるベースクラスから来る必要がある場合)。このタグ/値ペアは、
"handler=TestHandler.MyHandler" です。
match=<path>
*       ハンドラを呼び出す URI を指定します。
*       例えば、match=/foo の場合、https://$SERVER:$PORT/services/foo がこのハンドラを呼び出します。
*       注意: パスは /. で始まる必要があります。
requireAuthentication=<true | false>
*       このオプションのタグは、このエンドポイントに認証が必要かどうかを決定します。
*       デフォルトは true です。
capability=<capabilityName>
capability.<post|delete|get|put>=<capabilityName>
*       HTTP の方法に依存します。認定セッションユーザーの機能をチェックします。
*       'capability.post|delete|get|put' を使用する場合、関連方法は、認定ユーザーの役割に対してチェックされます。
*       単純に 'capability' を使う場合は、すべての呼び出しは、この機能に対してチェックされます (HTTP 方法に依存しない)。
xsl=< XSL transform ファイルまでのパス>
*       オプション。
*       ハンドラから返されたデータに対してオプションの XSL transform を実行します。
*       データが XML の場合にのみ、これを使用します。
script=<実行可能スクリプトへのパス>

```

```

* オプション。
* 'splunk.rest.BaseRestHandler' からではないスクリプトを実行します。
* ここで、そのパスをそのスクリプトに入力します。
* これは、ほとんど使用されません。
* 詳細がわからない場合は、使用しないでください。
#####
# 'admin'
# 拡張可能管理インタフェースのための内蔵ハンドラ
# リストアップされた EAI ハンドラを与えられた URL で公開します。
#
[admin:<uniqueName>]
match=<URL の一部>
* URL は、アクセスされると、下記のハンドラを表示します。
members=<csv list>
* この URL で公開するハンドラのリスト。
* すべてのハンドラ一覧については、https://localhost:8089/services/admin を参照してください。
#####
# 'admin_external'
# 拡張可能管理インタフェースの Python ハンドラを登録します。
# ハンドラはその "uniqueName" で公開されます。
#
[admin_external:<uniqueName>]
handlertype=<スクリプトタイプ>
* 現在、値 'python' のみが有効です。
handlerfile=<固有のファイル名>
* 実行するスクリプト。bin/myAwesomeAppHandler.py については、myAwesomeAppHandler のみを指定します。
restmap.conf.example
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは REST エンドポイント設定の例を記載します。
#
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の restmap.conf に
コピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
# 次はデフォルトの REST 設定です。エンドポイントを自作するには、restmap.conf.spec で説明した spec に従って値を

```

修正してください。

```
# ////////////////////////////////////////////////////////////////////
#   グローバル設定
#   ////////////////////////////////////////////////////////////////////
[global]
#   GET パラメータで認証を許可するかどうかを指示します。
allowGetAuth=false
#   デフォルトハンドラ (PYTHONPATH セットがあると仮定)。
pythonHandlerPath=$SPLUNK_HOME/bin/rest_handler.py
#   ////////////////////////////////////////////////////////////////////
#   内部 C++ ハンドラ
#   注意： これらは内部 Splunk 作成エンドポイントです。他社ディベロッパーのみがスクリプトを使用することができます。
#   検索はハンドラとして使用することができます。(設定に関するヘルプは、restmap.conf.spec を参照)。
#   ////////////////////////////////////////////////////////////////////
[streams:livetail]
match=/streams
capability=allow_livetail
[SBA:sba]
match=/properties
capability=get_property_map
[asyncsearch:asyncsearch]
match=/search
capability=search
```

savedsearches.conf

savedsearches.conf

次は savedsearches.conf の仕様とファイル例です。

savedsearches.conf.spec

```
#   Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#   このファイルは、savedsearches.conf の保存済み検索エントリーの属性/値ペアを記載します。
#   自分の savedsearches.conf を作成することにより、保存済み検索を設定することができます。
#
#   デフォルトの savedsearches.conf は $SPLUNK_HOME/etc/system/default にあります。カスタム設定を設定するに
#   は、savedsearches.conf を $SPLUNK_HOME/etc/system/local に置いてください。
#   例は、savedsearches.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
```

```

#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#*****
#       savedsearches.conf の属性/値ペアには、以下が指定できます。
#*****
[<スタンプ名>]
*       保存済み検索に名前を付けます。
*       各保存済み検索について固有のなスタンプ名を作成します。
*       このスタンプ名に、次の属性/値のペアを続けてください。
*       属性を指定しない場合、Splunk はデフォルトを使用します。
disabled = 0 | 1
*       1 に設定すると検索が無効になります。
*       1 に設定すると、Splunk Web で検索が見えなくなります。
*       デフォルトは 0。
search = <文字列>
*       保存済み検索の実際の検索語。
*       例えば、search = index::sampledata http NOT 500。
*       検索は、マクロ検索で置き換えることができます。
*       マクロ検索を作成するには、http://www.splunk.com/doc/latest/admin/MacroSearch のドキュメントを参照して
ください。
#*****
#       スケジューリングオプション
#*****
enableSched = 0 | 1
*       検索をスケジュールにより実行するには、これを 1 に設定します。
*       デフォルトは 0。
schedule = <cron 型文字列>
*       このフィールドは、4.0 で廃止されます。cron_schedule を代わりに使用してください。
*       cron 型スケジュール。
*       例えば、*/12 * * * *
*       注意： Splunk の現在の cron の実装は標準 POSIX cron とは異なります。
*       */n を "divide by n" として使用します(標準 POSIXcron では "every n")。
cron_schedule = <cron 文字列>
*       この検索の実行に使用する cron スケジュール。
*       例： */5 * * * *
*       上記の cron 文字列により、検索は 5 分毎に実行されます。

```

```

#*****
#       通知オプション
#*****
counttype = number of events | number of hosts | numbers of sources | always
*       アラートのカウントタイプを設定します。
*       relation と quantity を併用します(下記)。
*       注意: "always" を指定した場合は、relation と quantity は使用しないでください(下記)。
relation = greater than | less than | equal to | drops by | rises by
*       counttype との比較方法。
quantity = <整数>
*       relation と counttype の値を指定します。
*       例えば、"number of events [is] greater than 10" は、イベントのカウントが 10 を超えたときにアラートを送信
    します。
*       例えば、"number of events drops by 10%" は、イベントのカウントが 10%減少したときにアラートを送信します。
alert_condition = <文字列>
*       アラートをトリガするかどうかを決めるために保存済み検索の検索結果を評価する検索。
*       アラートは、指定された検索が空でない検索結果リストを生成したときトリガされます。
#*****
#       一般アクション設定
#*****
action.<action_name> = 0 | 1
*       アクションを有効にするか無効にするかを指定します。
action.<action_name>.<パラメータ> = <値>
*       alert_actions.conf で定義されたアクションのパラメータを上書きします。
#*****
#       E メールアクション設定
#*****
action.email.to = <E メールリスト>
*       受取側メールアドレスをカンマ区切りしたリスト。
action.email.from = <E メールアドレス>
*       送信側アドレスとして使用されるメールアドレス。
action.email.subject = <文字列>
*       受取側に送られる E メールリストの題名。
action.email.mailserver = <文字列>
*       Eメールの送信使用する MTA サーバーのアドレス。
#*****
#       サマリーインデックス設定
#*****

```

```

action.summary_index = 0 | 1
* サマリーインデックスを有効にするかどうかを切り替えます。
* 1 は有効、0 は無効。
* デフォルトは 0。
action.summary_index._name = <index>
* スケジュールされた検索の結果が保存されるサマリーインデックス。
* スケジュールされた検索の結果を保存するサマリーインデックスを指定します。
* デフォルトは summary。
action.summary_index.<KEY> = <文字列>
* オプション<KEY> = <文字列>は、サマリーインデックスで保存する際に、各イベントに追加します。
#*****
# ルックアップテーブル投入設定
#*****
action.populate_lookup = 0 | 1
* ルックアップ投入アクションの有効を切り替えます。
action.populate_lookup.dest = <文字列>
* 検索結果をコピーする先のルックアップ csv ファイルへのパス。
* 注意： このパスは、次のいずれかのディレクトリで.csv ファイルをポイントする必要があります。
* $SPLUNK_HOME/etc/system/lookups/
* $SPLUNK_HOME/etc/apps/<app-name>/lookups
* 注意： 上記ファイルの目的ディレクトリはすでに存在する必要があります。
run_on_startup = true | false
* この検索を、Splunk の開始時に実行するかどうかを切り替えます。
* スタートアップ時に実行しない場合は、次のスケジュールされた時間に実行されます。
* ルックアップテーブルに投入するスケジュール検索については、これを true に設定することを推奨します。
#*****
# ディスパッチ検索オプション
#*****
* HTTP 接続(Splunkd との)に関する読み込み/書き込み/接続タイムアウト(秒)。
* スケジュールされた検索およびそのアクション/アラートの実行に使用します。
dispatch.ttl = <整数>[p]
* スケジュールされた検索の検索結果の有効期間(秒)。
* 整数の次に文字 'p' を続けると、ttl は複数のスケジュールされたと解釈します。
* デフォルトは 10p。
dispatch.buckets = <整数>
* 時間軸バケツの最大数。
* デフォルトは 0。
dispatch.max_count = <整数>

```

```

*      検索を終了する前の結果の最大数。
*      デフォルトは 10000。
dispatch.max_time = <整数>
*      検索を終了する前の最大時間(秒)。
*      デフォルトは 0。
dispatch.lookups = true | false
*      この検索でフックアップを有効にするかどうかを切り替えます。
*      デフォルトは true。
dispatch.earliest_time = <time-str>
*      検索の最も早い時間。
dispatch.latest_time = <time-str>
*      検索の最近の時間。
dispatch.time_format = <time format str>
*      最も早い時間と最近の時間の指定に使用する時間フォーマット。
dispatch.spawn_process = <論理式>
*      この保存済み検索が実行される時に、新規検索プロセスを実施するかどうか(デフォルトは true)を指定します。
#*****
#      UI 別設定
#*****
displayview
*      結果をロードするデフォルト UI ビュー名を定義(ラベルではない)します。
*      アクセスのしやすさは、ユーザーが十分な許可を有しているかに左右されます。
vsid
*      'displayview' にリストアップされる UI ビューに関連するビューステート id を定義します。
*      viewstates.conf のスタンザに一致する必要があります。

```

savedsearches.conf.example

```

#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは保存済みの検索およびアラートの例を記載します。
#
#      この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の
savedsearches.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#      次の検索は検索の例です。分の検索を自作するには、savedsearches.conf.spec で説明した spec に従って値を修正し

```

てください。

[Daily indexing volume by server]

```
search = index=_internal todaysBytesIndexed LicenseManager-Audit NOT source=*web_service.log NOT
_Indexing_Volume_in_MBs = todaysBytesIndexed/1024/1024 | timechart
avg(Daily_Indexing_Volume_in_dispatch.earliest_time = -7d
```

[Errors in the last 24 hours]

```
search = error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
dispatch.earliest_time = -1d
```

[Errors in the last hour]

```
search = error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
dispatch.earliest_time = -1h
```

[KB indexed per hour last 24 hours]

```
search = index=_internal metrics group=per_index_thruput NOT debug NOT sourcetype=splunk_web_access
sum(kb) | rename sum(kb) as totalKB
dispatch.earliest_time = -1d
```

[Messages by minute last 3 hours]

```
search = index=_internal eps "group=per_source_thruput" NOT filetracker | eval events=eps*kb/kbps
um(events) by series
dispatch.earliest_time = -3h
```

[Splunk errors last 24 hours]

```
search = index=_internal " error " NOT debug source=*/splunkd.log*
dispatch.earliest_time = -24h
```

searchbnf.conf

searchbnf.conf

次は searchbnf.conf の仕様とファイル例です。

searchbnf.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#
# このファイルは、searchbnf.conf で search-assistant を設定するためのスタンザと属性/値ペアを説明します。
#
# searchbnf.conf は $SPLUNK_HOME/etc/system/default/ にあります。これは修正してはいけません。アプリケーションが自身のカスタム python 検索コマンドを持つ場合、コマンドを search-assistant に記述するためにアプリケーションは自身の searchbnf.conf を含めることができます。
#
```

設定ファイル(優先順位を含む)についての詳細は、
<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork>にあるドキュメントを参照してください。

```
[<search-commandname>-command]
```

- * このスタanzasは、指定の<search-command>についてプロパティを有効にします。
- * searchbnf.conf ファイルは、コマンドの数に関わらず、複数のスタanzasを含むことができます。
- * このスタanzas名に、次の属性/値のペアを続けてください。
- * 指定の<spec>について属性を設定しない場合、デフォルトが使用されます。デフォルト値は空白です。
- * スタanzas名の例: "geocode"コマンドの場合、"geocode-command"とできます。
- * 検索コマンドスタanzasは、他のスタanzasで定義された定義を参照することができます。また、"-command"をそれらに追加する必要はありません。

例:

```
[geocode-command]
```

```
syntax = geocode <geocode-option>*
```

```
...
```

```
[geocode-option]
```

```
syntax = (maxcount=<int>) | (maxhops=<int>)
```

```
...
```

```
*****
```

```
# searchbnf.conf に関する属性/値ペア
```

```
*****
```

```
SYNTAX = <文字列>
```

- * 検索コマンドの構文を記述します。詳細は、searchbnf.conf のヘッドを参照してください。
- * 必須です。

```
SIMPLESYNTAX = <文字列>
```

- * オプションの構文の簡単バージョンで、完全性は犠牲になるものの、理解が容易になります。通常、ほとんど使用しないオプションを削除したり、表現方法を変えたりします。

- * 例えば、ある検索コマンドは "m|min|mins|minute|minutes"などの値を受け入れる場合、その構文記述を不必要に分解します。この場合、simplesyntax は単純に 1 つのみを取り上げることができます(例: "minute")。

```
ALIAS = <コマンドリスト>
```

- * 検索コマンドの代替名。これはまた、構文を明確にします。したがって、ユーザーは 'savedsearch' が 'macro' や 'savedsplunk' による呼び出しについて知る必要はありません。

```
DESCRIPTION = <文字列>
```

- * 検索コマンドの詳細なテキスト説明。説明は、その行が"\n"で終わる場合、次の行に続けることができます。
- * 必須です。

```
SHORTDESC = <文字列>
```

- * 検索コマンドの簡略説明。完全な説明は、検索アシスタントの画面の面積を多く使いすぎる場合があります。
- * 必須です。

EXAMPLE = <文字列>

COMMENT = <文字列>

* 'example' は、検索コマンドを使用する、役に立つ例を掲げます。また、'comment' はその例を説明します。

* 'example' and 'comment' は、複数のサンプルと相当するコメントを許可するために、マッチングインデックスに追加することができます。

* 例：

example2 = geocode maxcount=4

command2 = run geocode on up to four values

example3 = geocode maxcount=-1

comment3 = run geocode on all values

USAGE = public|private|deprecated

* コマンドに public、private、deprecated のいずれかを指定します。検索アシスタントのみがパブリックコマンドを操作できます。

* 必須です。

TAGS = <タグリスト>

* この検索コマンドを記述するタグのリスト。ユーザーが同義語を入力した場合に使用してコマンドを見つけます(例：

"graph" -> "chart")。

RELATED = <コマンドリスト>

* 1つのコマンドを使って他のコマンドについて学ぶときにユーザーを助ける関連コマンドのリストです。

Splunk で主に内部的に使用されるオプション属性

maintainer, appears-in, note, supports-multivalue, appears-in

searchbnf.conf.example

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

#

次は、searchbnf.conf 設定のスタンザの例です。

#

#####

selfjoin

#####

[selfjoin-command]

syntax = selfjoin (<selfjoin-options>)* <field-list>

shortdesc = 結果を結合。

description = 結果を結合。結合するフィールドを少なくとも1つ指定する必要があります。

usage = public

example1 = selfjoin id

```
comment1 =      'id'フィールドで結果を結合。
related =      join
tags = join combine unite
[selfjoin-options]
syntax = overwrite=<論理式> | max=<int> | keepsingle=<int>
description =  selfjoin は、join フィールドの同じ値を持つ他の結果と各結果を結合します。join の基礎として使用する結果のフィールドを、これら他の結果からのフィールドが上書きする場合は、'overwrite' がコントロールします(デフォルト=true)。
max は、各メイン結果が結合することができる他の結果の最大数を表します。
(デフォルト= 1、0 は制限なし)。「keepsingle」は、join フィールドの固有の値を持つ結果(従って結合する他の結果はない)を保持すべきかどうかをコントロールします。(デフォルト=false)
```

segmenters.conf

segmenters.conf

次は segmenters.conf の仕様とファイル例です。

segmenters.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、segmenters.conf のイベントのセグメンテーションの設定のための属性/値ペアを記載します。
#
#      デフォルトの segmenters.conf は $SPLUNK_HOME/etc/system/default にあります。カスタム設定を設定するには、
segmenters.conf を $SPLUNK_HOME/etc/system/local に置いてください。
#      例は、segmenters.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
[SegmenterName]
*      スタンザに名前を付けます。
*      このスタンザ名に、次の属性/値のペアを続けてください。
*      属性/値ペアを指定しない場合、Splunk はデフォルトを使用します。
MAJOR = <分解文字のスペースで区切られたリスト>
*      メジャーブレーカーを設定します。
*      メジャーブレーカーは、セットブレーキングにより囲まれた、データにある語句です。
*      デフォルトでは、メジャーブレーカーはほとんどの文字および空白スペースに設定されています。
*      通常、メジャーブレーカーは 1 文字です。
*      デフォルトは、[ ] < > ( ) { } | ! ; , ' " * \n \r \s \t & ? + %21 %26 %2526 %3B %7C %20 %2B。
```

* 注意：\s はスペース、\n は新しい行、\r はキャリッジリターン、\t はタブを表します。

MINOR = <文字列のスペースで区切られたリスト>

* マイナーブレイカーを設定します。

* メジャーブレイカーで指定されたセグメントに加えて、各マイナーブレイカーが見つかった場合、Splunk は、そのトークンを、最後のメジャーブレイカーから現在のマイナーブレイカーまで、および、最後のマイナーブレイカーから現在のマイナーブレイカーまでインデックスします。

* デフォルトは / : = @ . - \$ # % \ \ _ 。

FILTER = <正規表現>

* 設定されると、セグメンテーションは、正規表現が一致した場合にのみ行われます。

* さらに、セグメンテーションは、一致する正規表現の最初のグループでのみ行われます。

* デフォルトは空白です。

LOOKAHEAD = <整数>

* Splunk が指定イベントをセグメントする範囲(文字数)を設定します。

* LOOKAHEAD は、FILTER ルールの適用後に適用されます。

* セグメンテーションを無効にするには、0 に設定します。

* デフォルトは-1(すべてのイベントの読み込み)。

MINOR_LEN = <整数>

* マイナートークンの長さを指定します。

* それより長いマイナートークンは例外なく廃棄されます。

* デフォルトは-1。

MAJOR_LEN = <整数>

* メジャートークンの長さを指定します。

* それより長いマイナートークンは例外なく廃棄されます。

* デフォルトは-1。

MINOR_COUNT = <整数>

* イベント毎に作成可能なマイナーセグメントの数を指定します。

* 指定された数のマイナートークンが作成された後、それ以後のものは例外なく廃棄されます。

* デフォルトは-1。

MAJOR_COUNT = <整数>

* イベント毎に作成可能なメジャーセグメントの数を指定します。

* 指定された数のメジャーセグメントが作成された後、それ以後のものは例外なく廃棄されます。

* デフォルトは-1。

segmenters.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# 次はセグメンテーション設定の例です。
```

```
#
```

```

# この設定の1つまたは複数を使用するには、その設定ブロックを$SPLUNK_HOME/etc/system/localのsegmenters.conf
にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWorkにあるドキュメントを参
照してください。
# syslog データで、日付をセグメントとしてインデックス化しないセグメントの例：
[syslog]
FILTER = ^.*?\d\d:\d\d:\d\d\s+\S+\s+(.*)$
# イベントの最初の 256b のみをインデックス化するセグメントの例：
[limited-reach]
LOOKAHEAD = 256
# イベントの最初の行だけをインデックス化するセグメントの例：
[first-line]
FILTER = ^(.*)($|\n)
# セグメンテーションを完全にオフにする場合：
[no-segmentation]
LOOKAHEAD = 0

```

server.conf

server.conf

次は server.conf の仕様とファイル例です。

server.conf.spec

```

# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、server.conf で SSL と HTTP サーバーオプションを設定するのに使用することができる属性と値を記載
しています。
#
# server.conf は$SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、server.conf を
$SPLUNK_HOME/etc/system/local に置いてください。例は、server.conf.example を参照してください。
# 設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWorkにあるドキュメントを参
照してください。
# このファイルは、サーバー設定をコントロールするためのオプションを記載しています。

```

```

#      現在利用可能なオプションは、サーバーの SSL 設定のコントロールのみです。
#####
#      一般サーバ設定
#####
[general]
serverName = <ascii 文字列>
*      分散検索などの機能でこの Splunk インスタンスを特定するために使用される名前。
*      デフォルトは<ホスト名>-<ユーザーが実行している Splunk >。
#####
#      SSL 設定詳細
#####
[sslConfig]
*      このスタンザ名の Splunk のバンクエンドでコミュニケーションするための SSL を設定します。
*      注意： Splunk Web およびそのブラウザの SSL (例：HTTPS) を設定するには、web.conf を使用してください。
*      このスタンザ名に、次の属性/値のペアを続けてください。
*      各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
enableSplunkdSSL = <true | false>
*      Splunkd 管理ポート (8089) での SSL を有効化/無効化。
*      デフォルトは true。
useClientSSLCompression = <true | false>
*      HTTP クライアント圧縮をオンにします。
*      サーバー側圧縮はデフォルトでオンになります。クライアント側でこれを設定すると、サーバーとクライアントとの間で圧縮が有効になります。
*      これを有効にすると、複数の Splunk インスタンスで、遙かに速い分散検索が可能になる場合があります。
*      デフォルトは true。
supportSSLV3Only = <true|false>
*      true の場合、HTTP サーバーに対して、接続を受け入れることのみを伝えます。
*      SSLv3 クライアントから。
*      デフォルトは false。
sslVerifyServerCert = <true|false>
*      分散検索により使用：その検索クラスターで、検索要求を他のサーバーに行う場合。
*      分散デプロイメントクライアントにより使用：デプロイメントサーバーをポーリングする場合。
*      true に設定する場合、接続しているサーバーが有効なもの (認証済) であるかどうかを確認してください。サーバーの共通名と代替名の両方が、それらがこの設定ファイルで指定された場合、一致するかどうかチェックされます。
*      デフォルトは false。
sslCommonNameToCheck = <commonName>
*      'sslVerifyServerCert' が true に設定されているときチェックする共通名。
*      オプション。デフォルトは共通名のチェックなし。

```

```

sslAltNameToCheck = <alternateName>
*      'sslVerifyServerCert'が true に設定されているときチェックする代替名。
*      オプション。デフォルトは代替名チェックなし。
requireClientCert = <true|false>
*      splunkds 内部 HTTPS サーバーに接続している HTTPS クライアントが、我々の証明機関により署名された証明書を有する
ことを要求する。
*      分散検索により使用：Splunk インデキシングインスタンスは、他の Splunk インデキシングインスタンスに接続するた
めに認証される必要があります。
*      分散デプロイメントにより使用：デプロイメントサーバーは、デプロイメントクライアントが新規設定/アプリケーションに
対するポーリングを許可する前に、それが認証されていることを要求します。
*      true の場合、クライアントは、我々の証明機関により作成された証明書がこのクライアントで使用された場合にのみ接続で
きます。
*      デフォルトは false。
cipherSuite = <cipher suite 文字列>
*      これが設定されると、HTTP サーバーで指定された暗号文字列を使用します。
設定されない場合、OpenSSL が提供するデフォルト暗号文字列を使います。これは、サーバーが、弱い暗号化プロトコルを使って接続
を受け入れないことを確実にするために使用します。
sslKeysfile = <ファイル名>
*      サーバー証明書ファイル。
*      証明書は、Splunk が起動すると、Splunkd により自動生成されます。
*      デフォルトの証明書を自分の PEM フォーマットファイルで置き換えることができます。
*      証明書は caPath に保存されます(下記参照)。
*      デフォルトは server.pem。
sslKeysfilePassword = <パスワード>
*      サーバー証明書パスワード。
*      デフォルトは password。
caCertFile = <ファイル名>
*      署名機関の公開鍵。
*      デフォルトは cacert.pem。
caPath = <パス>
*      これらすべての証明書が保存されるパス。
*      デフォルトは $SPLUNK_HOME/etc/auth。
certCreateScript = <スクリプト名>
*      Splunk のスタートアップ時に証明書を生成するための作成スクリプト。
*      デフォルトは genSignedServerCert.sh。
#####
#      SplunkdHTTP サーバー設定
#####

```

[httpServer]

- * このスタンザ名において、Splunk のスタンドアロン HTTP 設定を設定します。
- * このスタンザ名に、次の属性/値のペアを続けてください。
- * 各属性のエントリーを指定しない場合、Splunk はデフォルト値を使用します。

atomFeedStylesheet = <文字列>

- * デフォルト Atom フィードを適用するためのスタイルシート相関 URL を定義します。
- * 'none' に設定すると、xsl-stylesheet 指令の書き出しを停止します。
- * デフォルトは /static/atom.xsl。

max-age = <整数>

- * '/static' ディレクトリの稼働していないスタティックアセットをキャッシュする最大時間(秒)を設定します。
- * この値は 'Cache-Control' HTTP ヘッダに渡されます。
- * デフォルトは 3600。

follow-symlinks = <true|false>

- * スタティックファイルハンドラ('/static'ディレクトリをサブ)が、ファイルをサブする際、ファイルシステムシムリンクをフォローするかどうかを切り替えます。
- * デフォルトは false。

disableDefaultPort = <true|false>

- * true の場合、Splunkd 管理ポート(デフォルトでは 8089)のリスニングをオフにします。
- * デフォルト値は 'false'。

#####

Splunkd HTTP サーバリスナー設定

#####

[httpServerListener:<ip>:<ポート>]

- * Splunkd HTTP サーバが、<ip>で指定されたネットワークインタフェース(NIC)、および<ポート>で指定されたポート番号をリッスンすることを有効にします。<ip>を空白にすると(しかしまだなお ':' を含む)、Splunkd は、カーネルが取り上げた NIC をポート<ポート>を使用してリッスンします。

ssl = <true|false>

- * このリスニング IP ポートが SSL を使用するかどうかを切り替えます。
- * デフォルト値は 'true'。

#####

スタティックファイルハンドラ MIME タイプマップ

[mimetype-extension-map]

- * このスタンザ名で、スタティックファイルハンドラからサブされるファイルの MIME タイプにファイル名拡張子を割り当てます。

<file-extension> = <MIME-type>

- * HTTP スタティックファイルサーバに、'file-extension'で終わり、ヘッダが 'Content-Type: <MIME-type>' のファイルをマークするように指示します。
- * デフォルトは :

```

[mimetype-extension-map]
gif = image/gif
htm = text/html
jpg = image/jpeg
png = image/png
txt = text/plain
xml = text/xml
xsl = text/xml
#####
#      リモートアプリケーション設定(例 : Splunkbase)
#####

[applicationsManagement]
*      このスタンプ名で、Splunk のリモートアプリケーション設定を設定します。
*      このスタンプ名に、次の属性/値のペアを続けてください。
*      各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
url = <URL>
*      アプリケーションリポジトリ。
*      デフォルトは http://www.splunkbase.com/api/apps。
loginUrl = <URL>
*      アプリケーションリポジトリログイン。
*      デフォルトは http://www.splunkbase.com/api/account:login/。
useragent = <splunk-version>-<splunk-build-num>-<platform>
*      アプリケーションリポジトリにコンタクトする際に使用するユーザー-エージェント文字列。
*      <platform>は、オペレーティングシステムや CPU アーキテクチャなどの情報を含む。
#####
#      その他の設定
#####

[scripts]
initialNumberOfScriptProcesses = N
*      N は、システムが立ち上がる際に開始されるプリフォークスクリプトプロセスの数です。これらのスクリプトは、スクリプト REST エンドポイントおよび検索スクリプトが実行される際に再利用されます。その目的は、スクリプトインタプリターが呼び出される際のパフォーマンスオーバーヘッドを取り除くことです。これらのプロセスはプールの中に置かれます。そのプールが、スクリプトが呼び出される際に完全にビジーな場合、新たなプロセスが開始され、その新たな呼び出しを扱います。しかし、その呼び出しが終わると、削除されます。
#####
#      ディスク利用設定(インデックスマシン用、Splunk ログファイルでは使用しない)
#####

[diskUsage]

```

```

minFreeSpace = <num>
*       ディスク利用プロセッサは、許容最小ディスクスペースに達した場合に、Splunk がデータをインデックスに追加すること
を防ぎます。
*       デフォルト設定は 2000 メガバイトです。
pollingFrequency = <num>
*       デフォルトのポーリング頻度は 100000 イベント毎です。
#####
#       キュー設定
#####
[queue=<queueName>]
maxSize = <num>
*       キューが保持できるパイプラインデータの最大数。
*       デフォルトは 1000。
#       HTTP エンドポイントの PubSub サーバーを設定します。
#####
[pubsubsvr-http]
disabled=<true or false>
*       無効にすると、HTTP エンドポイントは登録されません。PubSub サーバーを HTTP に公開するには、この値を 'false' に設
定します。
*       デフォルトは 'true'。
stateIntervalInSecs=300
*       非アクティブのために接続をフラッシュするまでの秒数。接続が閉じられると、その接続のメッセージのみがフラッシュさ
れます。
*       デフォルトは 300 秒/5 分。

```

server.conf.example

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       このファイルは server.conf の例を記載します。SSL および HTTP サーバーオプションを設定するには、このファイルを
使用してください。
#
#       この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の server.conf にコ
ピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#       SSL をオンにする：

```

```
[sslConfig]
enableSplunkdSSL = true
useClientSSLCompression = true
sslKeysfile = server.pem
sslKeysfilePassword = password
caCertFile = cacert.pem
caPath = $$SPLUNK_HOME/etc/auth
certCreateScript = genSignedServerCert.sh
```

serverclass.conf

serverclass.conf

次は serverclass.conf の仕様とファイル例です。

serverclass.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、デプロイメントクライアントが属するサーバークラスを定義するための属性と値を記載しています。これ
らの属性と値は、指定サーバークラスメンバーがデプロイメントサーバーから受信するコンテンツを指定します。
#
#      コンテンツをデプロイメントクライアントに展開する際に使用するための、このデプロイメントサーバーのサーバークラス
を定義するには、serverclass.conf を $$SPLUNK_HOME/etc/system/local/ に置いてください。
#      例は、serverclass.conf.example を参照してください。この設定ファイルの変更を有効にするには Splunk の再起動が
必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#*****
#      デプロイメントサーバーインスタンスが使用するサーバークラスの設定。
#
#      サーバークラスにより、アプリケーションの収集の作成、およびホスト名、クライアントマシンのビルド番号を使用してそ
れらにフィルタを適用することができるようになります。
#      ターゲットマシンがフィルタと一致すると、サーバークラスを構成するアプリケーションと設定コンテンツが展開されます。
#
#      プロパティ継承
#      serverclass.conf 内のスタanzasは、標準から具体的なものへ、次の順番で移行します。
#      [serverClass] -> [serverClass:<name>] -> [serverClass:<scname>:app:<appname>]
```

```

#
# 標準レベル([serverClass]など)で定義されるプロパティの中には、適用される際に、個別のスタンザで上書きされるものがあります。継承可能なプロパティはすべてこのようにマークされます。
#*****
# 全てのサーバークラスのプロパティを定義するグローバルスタンザです。
[global]
# serverClass で上書きされます。
repositoryLocation = $SPLUNK_HOME/etc/apps
* サーバマシン上のアプリケーションのリポジトリ。
targetRepositoryLocation = $SPLUNK_HOME/etc/apps
* このサーバークラスについて定義されたアプリケーションと設定コンテンツがインストールされるデプロイメントクライアント上の位置。
tmpFolder = $SPLUNK_HOME/var/run/tmp
* デプロイメントサーバーが使用するワーキングフォルダ。
# serverClass で上書きされます
continueMatching = <true or false>
* デフォルトは true。
* 最初のマッチの後、マッチングサーバークラスを継続します。
* serverClass はこのプロパティを overload し、マッチングを停止することができます。
* マッチングは、サーバークラスが定義された順番で行われます。
# serverClass で上書きされます。
endpoint = $deploymentServerUri$/services/streams/deployment?name=$serverClassName$: $appName$
* デプロイメントクライアントでダウンロード可能なコンテンツのエンドポイント。デプロイメントクライアントは、URL の変数の値の置換の方法を知っています。
* カスタム URL は、指定された変数を使用する限り、ここで設定することができます。
# serverClass および serverClass:app により上書きされます。
filterType = <whitelist または blacklist>
* デフォルトは whitelist。
# serverClass および serverClass:app により上書きされます。
whitelist.<n> = <ipAddress または hostname または clientName>
blacklist.<n> = < clientName の hostname の ipAddress>
* 'n' は 0 から開始し、1 ずつ増えます。'n' がブレイクするとフィルタの閲覧を停止します。
* デプロイメントクライアントの ipAddress。 10.1.1.* のようにワイルドカードを使用することもできます。
* デプロイメントクライアントの hostname。 *.splunk.com のようにワイルドカードを使用することもできます。
* clientName-deploymentClient で、各デプロイメントクライアントに割り当てることができる論理名または 'タグ' 名。
# 注意: 1 種類のフィルタ(ホワイトリスト/ブラックリスト)を上書きすると、他のものも上書きされます。
# ホワイトリストを上書きすると、ブラックリストは親から継承されません。スタンザでそれを指定する必要があります。
# フィルタタイプがホワイトリストになる場合の例

```

```

#       whitelist.0=*.splunk.com
#       blacklist.0=printer.splunk.com
#       blacklist.1=scanner.splunk.com
#       これにより、'printer'と'scanner'を除いて、splunk.comにあるすべてのホストがこのサーバークラスに一致するよ
うになります。
#       フィルタタイプがブラックリストになる場合の例
#       blacklist.0=*
#       whitelist.0=*.web.splunk.com
#       whitelist.1=*.linux.splunk.com
#       これにより、'web'および'linux'ホストのみがサーバークラスに一致するようになります。他のホストは一致しません。
#       クライアント machineTypes もクライアントとの一致に使用できます。
#       この設定により、デプロイメントクライアントのハードウェアタイプをフィルタとして使用できるようになります。
#       このフィルタは、クライアントがホワイトリスト/ブラックリストフィルタを使用して一致できない場合にのみ使用されます。
#       machineTypes の値は、ハードウェアプラットフォーム自身が指定した特定の文字列です。
#       クライアント上でこの文字列を見つける方法は、プラットフォームにより異なります。しかし、デプロイメントクライアン
トがすでにデプロイメントサーバーに接続されている場合、デプロイメントサーバー上のこの SplunkCLI コマンドを使用して、この
文字列が何であるかを決定することができます：
#       <code>./splunk list deploy-clients</code>
#       これは、<code>machineTypes</code>の指定に使うことができる<code>utsname</code>の値を返します。
#       serverClass および serverClass:app により上書きされます。
machineTypes = <マシンタイプのカンマで区切られたリスト-例：linux-i686, linux-x86_64>
*       指定しない限り使用されません。
*       カンマで区切られたリストでマシンタイプと一致します。
*       通常使用されるマシンタイプ： linux-x86_64, windows-intel, linux-i686, freebsd-i386, darwin-i386,
sunos-sun4u
*       このフィルタは、クライアントがホワイトリスト/ブラックリストフィルタを使用して一致できない場合にのみ使用されます。
*       注意： "machineTypes"の最後に 's' を含めてください。
#       serverClass および serverClass:app により上書きされます。
restartSplunkWeb = <True or False>
*       デフォルトは false。
*       サーバークラスまたはアプリケーションのインストールがクライアントの SplunkWeb の再起動を要求するかどうかを指定
します。
restartSplunkd = <True or False>
*       デフォルトは false。
*       サーバークラスまたはアプリケーションのインストールがクライアントの Splunkd の再起動を要求するかどうかを指定し
ます。
#       serverClass および serverClass:app により上書きされます。
stateOnClient = <enabled, disabled, noop>

```

* インストール後にクライアントのアプリケーションを有効または無効にします。

* 値が `noop` の場合、状態を変えずに、DS での状態を保持します。

[serverClass:<serverClassName>]

* このスタンザはサーバークラスを定義します。serverClass はアプリケーションの集合です。

* serverClassName は、この serverClass に割り当てられる固有の名前です。

* serverClass は、[serverClass]スタンザから継承したすべてのプロパティを上書きできます。

このレベルで通常上書きする必要があるプロパティには次のものがあります。

repositoryLocation

continueMatching

filtering using whitelist/blacklist, startBuild, endBuild, machineType

requiresRestart

stateOnClient

注意：アプリケーションバージョンのサポートはまだありません！！

[serverClass:<server class name>:app:<* or appName>]

* このスタンザは、repositoryLocation にあるアプリケーションをサーバークラスに追加します。

* serverClassName-このコンテンツが追加されるサーバークラス。

* appName = * - repositoryLocation にあるすべてのコンテンツをこの serverClass に追加する。

* appName = some name - アプリケーション/設定コンテンツを明示的にサーバークラスに追加する。

* マッチングに関する重要事項：サーバークラスは、そのサーバークラスに属するコンテンツがシステムにより一致される前に一致する必要があります。

appFile=<ファイル名>

* appName がファイル名またはディレクトリ名と異なる場合、このパラメータを使用してそのファイル名を指定することができます。サポートされるフォーマットは、アプリケーションディレクトリ、.tar または .tgz ファイルです。

serverclass.conf.example

例 1

すべてのクライアントと一致し、すべてのアプリケーションをサーバークラスに含めます。

[global]

whitelist.0=*

ホワイトリストはすべてのクライアントと一致します。

[serverClass:AllApps]

[serverClass:AllApps:app:*]

repositoryLocation ですべてのアプリケーションを包含するサーバークラス-この場合、\$SPLUNK_HOME/etc/apps。

例 2

サーバークラスをホストネームに基づいて割り当てます。

[global]

[serverClass:AppsForOps]

whitelist.0=*.ops.yourcompany.com

```

[serverClass:AppsForOps:app:unix]
[serverClass:AppsForOps:app:SplunkLightForwarder]
[serverClass:AppsForDesktops]
filterType=blacklist
#      Windows デスクトップマシンを除いて、すべてをブラックリストに掲載します。
blacklist.0=*
whitelist.0=*.desktops.yourcompany.com
[serverClass:AppsForDesktops:SplunkDesktop]
#      例 3
#      サーバークラスをマシンタイプに基づいて展開します。
[global]
[serverClass:AppsByMachineType]
#      このサーバークラスがすべてのクライアントと一致することを確認します。ここに標準フィルタ、アプリケーションレベル
により個別のフィルタがあることが重要です。アプリケーションは、それが擁するサーバークラスが正しく一致している場合にのみ一
致します！
whitelist.0=*
[serverClass:AppsByMachineType:app:SplunkDesktop]
#      このアプリケーションを Windows ボックスにのみ展開します。
machineTypes=Windows-Intel
[serverClass:AppsByMachineType:app:unix]
#      このアプリケーションを Unix ボックス-32/64 ビットにのみ展開します。
machineTypes=linux-i686, linux-x86_64

```

serverclass.seed.xml.conf

serverclass.seed.xml.conf

次は、serverclass.seed.xml.conf の仕様とファイル例です。

serverclass.seed.xml.conf.spec

```

<!--
#      この設定は、Splunk インストールをアプリケーションでスタートアップ時にシードするために、deploymentClient が使
用します。
#      このファイルは、deploymentclient.conf が定義する workingDir フォルダに位置する必要があります。
#
#      興味ある事実-通信中の DS -> DC コミュニケーションもこの XML フォーマットを使用します。
-->
<?xml version="1.0"?>
<deployment name="somenam">

```

```

<!--
#   すべてのアプリケーションをダウンロードすることができるエンドポイント。この値は、serviceClass または次の ap 宣
言により上書きすることができます。
#   さらに、deploymentclient.conf はこのプロパティが deploymentClient によりどのように使用されるかをコントロ
ールします。deploymentclient.conf.spec を参照してください。
-->

<endpoint>${deploymentServerUri$}/services/streams/deployment?name=${serviceClassName$}:${appName$}</endpo
int>
<!--
#   すべてのアプリケーションがインストールされる deploymentClient 上の位置。この値は、serviceClass または次の
App 宣言により上書きすることができます。
#   さらに、deploymentclient.conf はこのプロパティが deploymentClient によりどのように使用されるかをコントロールし
ます。deploymentclient.conf.spec を参照してください。
-->

<repositoryLocation>${SPLUNK_HOME}/etc/apps</repositoryLocation>
<serviceClass name="serviceClassName">
<!--
#   このサービスクラスが処理される順番。
-->
<order>N</order>
<!--
#   DeploymentClients も、serverRepositoryLocationPolicy と serverEndpointPolicy を使用してこ
れらの値を上書きすることができます。
-->

<repositoryLocation>${SPLUNK_HOME}/etc/myapps</repositoryLocation>
<endpoint>splunk.com/spacecake/${serviceClassName$}/${appName$.tgz</endpoint>
<!--
#   これらのプロパティの使用方法については、serverclass.conf.spec を参照してください。
-->

<continueMatching>>true</continueMatching>
<restartSplunkWeb>>false</restartSplunkWeb>
<restartSplunkd>>false</restartSplunkd>
<stateOnClient>enabled</stateOnClient>
<app name="appName1">
<!--
#   アプリケーションはエンドポイントプロパティを上書きすることができます。
-->

```

```
<endpoint>splunk.com/spacecake/$appName$</endpoint>
</app>
<app name="appName2"/>
</serviceClass>
</deployment>
```

serverclass.seed.xml.conf.example

例なし

source-classifier.conf

source-classifier.conf

次は、source-classifier.conf の仕様とファイル例です。

source-classifier.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは、source-classifier.conf のファイル分類子の設定を設定するためのすべてのオプションを記載します。
#
#      source-classifier.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、
source-classifier.conf を $SPLUNK_HOME/etc/system/local/ に置いてください。
#      例は、source-classifier.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
ignored_model_keywords = <用語のスペースで区切られたリスト>
*      ソースタイプモデルを生成する際に無視する用語。
*      ソースタイプ "bundles/learned/*-model.xml" ファイルが、データファイルで非常に頻繁に生じる敏感な語 (例 :
"bobslaptop") を含むことを防ぐには、それらの語を ignored_model_keywords に追加します。
ignored_filename_keywords = <用語のスペースで区切られたリスト>
*      ソースを分類する目的で、新しいソース名と既存のソース名を比較する際に無視する用語。
```

source-classifier.conf.example

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは source-classifier.conf の例を記載します。ソースのソースタイプへの分類を設定するには、このフ
イルを使用します。
```

```

#
#     この設定の1つまたは複数を使用するには、その設定ブロックを$SPLUNK_HOME/etc/system/localの
source-classifier.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#     モデルにサーバー名を含まないようにするため、ソースタイプを生成する際に無視する用語、
ignored_model_keywords = sun mon tue wed thurs fri sat sunday monday tuesday wednesday thursday friday
saturday jan feb mar apr may jun jul aug sep oct nov dec january february march april may june july august
september october november december 2003 2004 2005 2006 2007 2008 2009 am pm ut utc gmt cet cest cetdst
met mest metdst mez mesz eet eest eetdst wet west wetdst msk msd ist jst kst hkt ast adt est edt cst cdt
mst mdt pst pdt cast cadt east eadt wast wadt
#     ソース名と既存のソース名を比較する際に無視する用語
ignored_filename_keywords = log logs com common event events little main message messages queue server
splunk

```

sourcetypes.conf

sourcetypes.conf

次は sourcetypes.conf の仕様とファイル例です。

sourcetypes.conf.spec

```

#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#     注意: sourcetypes.conf は、ソースタイプを生成するために、ファイル分類子が使用するドキュメントモデルを保存す
るマシン生成ファイルです。
#     通常、ほとんどの属性はマシンが生成するため sourcetypes.conf は編集してはいけません。
#     ただし、変更可能な属性が2つあります。
#
#     sourcetypes.conf は$SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、
sourcetypes.conf を$SPLUNK_HOME/etc/system/local/に置いてください。
#     例は、sourcetypes.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
_sourcectype = <値>

```

- * モデルのソースタイプを指定します。
- * モデルのソースタイプを変更するにはこれを変更します。
- * 将来、ソースがモデルと一致する場合は、この新しい名前のソースタイプを受信します。

`_source = <値>`

- * モデルのソース(ファイル名)を指定します。

sourcetypes.conf.example

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

#

このファイルは `sourcetypes.conf` の例を記載します。ソースタイプモデルを設定するにはこのファイルを使用します。

#

注意: `sourcetypes.conf` は、ソースタイプを生成するために、ファイル分類子が使用するドキュメントモデルを保存するマシン生成ファイルです。

#

通常、ほとんどの属性はマシンが生成するため `sourcetypes.conf` は編集してはいけません。

ただし、変更可能な属性が 2 つあります。

#

この設定の 1 つまたは複数を使用するには、その設定ブロックを `$SPLUNK_HOME/etc/system/local` の `sourcetypes.conf` にコピーしてください。設定を有効にするには Splunk の再起動が必要です。

#

設定ファイル(優先順位を含む)についての詳細は、

<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork> にあるドキュメントを参照してください。

#

これは、架空ソースタイプ `cadcamlog` のマシン生成ソースタイプモデルの例です。

#

`[/Users/bob/logs/bnf.x5_Thu_Dec_13_15:59:06_2007_171714722]`

`_source = /Users/bob/logs/bnf.x5`

`_sourcetype = cadcamlog`

`L----- = 0.096899`

`L-t<_EQ> = 0.016473`

sysmon.conf

sysmon.conf

次は sysmon.conf の仕様とファイル例です。

sysmon.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは Windows システム上でレジストリモニタリングを設定するための属性/値ペアを記載します。どのイベント
# タイプ(追加、削除、名前変更など)を監視するのか、どの正規表現が regmon-filters.conf ファイルからフィルタリングするか、
# および Windows レジストリイベントが監視されるのかどうか、に関するグローバル設定を含みます。
#
# このファイルは regmon-filters.conf と連携して使用されます。
#
# 設定を有効にするには Splunk の再起動が必要です。
#
# 設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[<スタンザ名>]
* デフォルトは[RegistryMonitor]
* このスタンザに、次の属性/値のペアを続けてください。
filter_file_name = <文字列>
* この監視用のフィルタが保存されるファイルの名前を表す文字列です。
event_types = <文字列>
* 監視するイベントのタイプを指定する正規表現文字列です。削除、設定、作成、名前変更、開く、閉じる、クエリ
が可能です。
inclusive = <1 または 0>
* 1 は、active_filters field で指定されたフィルタルールが含まれていることを指定します(ホワイトリスト)。
0 は、フィルタルールが除外されてることを指定します(ブラックリスト)。
disabled = <1 または 0>
* 1 は有効、0 は無効。
```

sysmon.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、Windows レジストリの変更をモニタリングするための設定例を記載しています。詳細は
sysmon.conf.spec を参照してください。
#
# 次は、レジストリモニタフィルタとプロセスモニタ用フィルタの例です。
#
# フィルタを自作するには、regmon-filters.conf.spec 内の情報を使用して値を修正してください。
```

```

#
#       この設定の1つまたは複数を使用するには、その設定ブロックを$SPLUNK_HOME/etc/system/localの
sysmon-filters.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[RegistryMonitor]
filter_file_name = regmon-filters
event_types = set.*|create.*|delete.*|rename.*
disabled = 0
[ProcessMonitor]
filter_file_name = procmon-filters
event_types = create.*|exit.*|image.*
inclusive = 0
disabled = 1

```

tags.conf

tags.conf

次は tags.conf の仕様とファイル例です。

tags.conf.spec

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       このファイルは、タグの設定のための属性/値ペアを記載しています。インデックス化されたフィールドまたは抽出されたフ
ィールドのタグ数を設定します。
#
#       tags.conf は$SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、tags.conf を
$SPLUNK_HOME/etc/system/local に置いてください。ヘルプは、tags.conf.example を参照してください。
#       設定を有効にするには Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[<fieldname>]
*       スタンザ内のタグが適用するフィールド名(例: host, source, ip)
*       tags.conf ファイルは複数のスタンザを含むことができます。

```

* 各スタンプは 1 つのフィールド名を参照することができます。

```
tag::::<tag1> = <enabled|disabled>
```

```
tag::::<tag2> = <enabled|disabled>
```

```
tag::::<tag2> = <enabled|disabled>
```

```
tag::::<tag3> = <enabled|disabled>
```

* フィールド<fieldname>の特定の<value>に関する各<tag>を有効および無効を設定します。

* <value>は、フィールド<fieldname>に可能な値です。

* 1 つのタグが、1 つのスタンプに許されます。

tags.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# これは tags.conf の例です。正規表現および transforms のルールを作成するには、このファイルを使用します。
```

```
# このファイルは props.conf と併用します。
```

```
#
```

```
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の transforms.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
#
```

```
# この最初の例は、フィールドは "host"、3 つのホスト名が、"hostswitch"、"emailbox"、および "devmachine." である状況を示します。各ホスト名は、それに適用する 2 つのタグを持ち、"building1" タグが 2 つのホスト名値(emailbox および devmachine)に適用されています。
```

```
[host]
```

```
tag::hostswitch::pci = enabled
```

```
tag::hostswitch::cardholder-dest = enabled
```

```
tag::emailbox::email = enabled
```

```
tag::emailbox::building1 = enabled
```

```
tag::devmachine::development = enabled
```

```
tag::devmachine::building1 = enabled
```

```
[src_ip]
```

```
tag::192.168.1.1::firewall = enabled
```

```
[seekPtr]
```

```
tag::lcb58000::EOF = enabled
```

```
tag::ld158000::NOT_EOF = disabled
```

tenants.conf

tenants.conf

次は tenants.conf の仕様とファイル例です。

tenants.conf.spec

```
#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      このファイルは PubSub システムのクライアントを設定するための属性と値を記載します。
#
#      カスタム設定を設定するには、pubsub.conf を $SPLUNK_HOME/etc/system/local に置いてください。
#      例は、pubsub.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#*****
#      同一の Splunk サーバ内のテナント(deploymentServer インスタンス)を設定します。
#
#      デプロイメントサーバーの複数インスタンスは、この設定ファイルを使用して同一の Splunk インスタンス内で設定するこ
とができます。
#      このファイルがない場合、serverclass.conf または default-serverclass.conf が存在すれば、tenant='default'
のデフォルトの deploymentServer がシステムにより設定されます。
#
#      ホワイトリスト/ブラックリストのメカニズムを使用することにより、デプロイメントクライアントをデプロイメントサーバ
ーの適当なインスタンスにリダイレクトすることが可能です。これは、serverclass.conf の場合と類似しています。
#
#      これがどのように機能するのか？
#      deploymentClient は TenantService とハンドシェイクし、通信する deploymentServer を決定します。
TenantService はこの設定を使用し、クライアントを適当なデプロイメントサーバー(phoneHomeTopic により代表される)にリダ
イレクトします。
#
#      マルチテナント設定はどのように保存されるのか？
#      各テナントのサーバークラス設定は <tenantName>-serverclass.conf で利用可能となる必要があります。
#
#      'default'テナントの設定も 'serverclass.conf' に行うことができます。tenantName の接頭辞がないことに注意する
こと。
#*****
```

```
[tenant:<tenantName>]
filterType = <whitelist または blacklist>
*     デフォルトは whitelist。
whitelist.<n> = <ipAddress または hostname または clientName>
blacklist.<n> = <clientName の hostname の ipAddress>
*     'n' は 0 から開始し、1 ずつ増えます。'n' がブレイクするとフィルタの利用を停止します。
*     デプロイメントクライアントの ipAddress。 10.1.1.* のようにワイルドカードを使用することもできます。
*     デプロイメントクライアントの hostname。 *.splunk.com のようにワイルドカードを使用することもできます。
*     clientName-deploymentClient で、各デプロイメントクライアントに割り当てることができる論理名または 'タグ' 名。
#     内部。
phoneHomeTopic=deploymentServer/phoneHome/$tenantName$
*     何らかの固有の接尾辞。デフォルトはテナント名を使用。この値が固有であることを確認します。
*     自分のデプロイメントサーバーをスクリプトし割り当てるとは異なる場合以外はこの値を上書きします。
```

tenants.conf.example

```
#     2 つのテナントを定義-dept1 および dept2。
#     dept1 の DS 設定は、一致する dept1-serverclass.conf で行います。
#     dept2 の DS 設定は、一致する dept2-serverclass.conf で行います。
[tenant:dept1]
whitelist.0=*.dept1.splunk.com
[tenant:dept2]
whilelist.0=*.dept2.splunk.com
```

times.conf

times.conf

次は times.conf の仕様とファイル例です。

times.conf.spec

```
#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#     このファイルはカスタム時間範囲を作成するための属性/値ペアを記載します。
#
#     カスタム設定を設定するには、times.conf を $SPLUNK_HOME/etc/system/local に置いてください。
#     ヘルプは、tags.conf.example を参照してください。設定を有効にするには Splunk の再起動が必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
```

照してください。

```
[<timerange_name>]
```

* 時間範囲に API またはコマンドラインでアクセスする際に使用するトークン。

* times.conf ファイルは複数のスタンザを含むことができます。

```
label = <文字列>
```

* この時間範囲を参照するために UI が使用するテキスト記述。

* 必須です。

```
earliest_time = <relative_time_identifier>
```

* 返すべき最も早いイベントを表す相対時間識別子文字列(含める)。

* オプション。省略された場合、最も早い時間限界が使用されます。

```
latest_time = <relative_time_identifier>
```

* 返すべき最も最近のイベントを表す相対時間識別子文字列(含めない)。

* オプション。省略された場合、最も最近の時間限界が使用されます。注意：将来生じるイベント(サーバータイムゾーンに対して)が返されることもあります。

```
order = <整数>
```

* すべてのカスタム時間範囲をソートするキー(昇順)。

* UI のデフォルト時間範囲セレクターはマージし、すべての時間範囲を 'order' キーに従って並べ替え、次にアルファベット順に並べ替えます。

* オプション。デフォルト値は 0。

times.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# これは times.conf の例です。検索システムを相互作用する際に使用することができるカスタム時間範囲を作成するにはこのファイルを使用します。
```

```
#
```

```
# この設定の1つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の times.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
# 注意：これは例です。カスタマイズした値と入れ替えてください。
```

```
# スタンザ名は、時間範囲を個別に特定する英数字の文字列(スペースなし)です。
```

```
[this_business_week]
```

```
# Define the label used in the UI
```

```
label = This business week
```

```
earliest_time = +1d@w1
```

```

latest_time = +6d@w6
#     この時間範囲の順番を定義します。すべての時間範囲はソートされます。
#     英数字の昇順。
order = 110
#
#     最も早い時間の限界だけを持つ時間範囲
#
[last_3_hours]
label = Last 3 hours
earliest_time = -3h
order = 30

```

transactiontypes.conf

transactiontypes.conf

次は、transactiontypes.conf の仕様とファイル例です。

transactiontypes.conf.spec

```

#     Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#     このファイルは、transactiontypes.conf ファイルのすべての属性と値ペアを記載しています。トランザクション検索
とそのプロパティを設定するには、このファイルを使用します。
#
#     transactiontypes.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、
transactiontypes.conf を $SPLUNK_HOME/etc/system/local に置いてください。設定を有効にするには Splunk の再起動が
必要です。
#
#     設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
[<TRANSACTIONTYPE>]
*     任意の数のトランザクションタイプを作成します。それぞれは、スタンザ名と任意の数の*により表現されます。 スタンザ
名 [<TRANSACTIONTYPE>] を使用して、Splunk Web 内のトランザクションを検索します。
*     次の各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
maxspan = [<integer> s|m|h|d]
*     トランザクションの最大時間幅を設定します。
*     秒、分、時間または日で設定することができます。
*     例： 5s、6m、12h、30d など。

```

* デフォルトは 5m。

maxpause = [<整数> s|m|h|d]

* 1つのトランザクション内のイベント間の最大一時停止時間を設定します。

* 秒、分、時間または日で設定することができます。

* 例： 5s、6m、12h、30d など。

* デフォルトは 2s。

fields = <フィールドのカンマで区切られたリスト>

* これが設定されると、各イベントは、同一のトランザクションの一部としてみなされる同一のフィールドを持つ必要があります。

* デフォルトは ""P。

transactiontypes.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# これは transactiontypes.conf の例です。トランザクションを設定するためのテンプレートとしてこのファイルを使用  
# します。
```

```
#
```

```
# この設定を1つまたは複数使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の transactiontypes.conf  
# にコピーします。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参  
照してください。
```

```
[default]
```

```
maxspan = 5m
```

```
maxpause = 2s
```

```
match = closest
```

```
[purchase]
```

```
maxspan = 10m
```

```
maxpause = 5m
```

```
fields = userid
```

transforms.conf

transforms.conf

次は transforms.conf の仕様とファイル例です。

transforms.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
# このファイルは、transforms.conf で transform およびイベント署名の設定に使用することができる属性と値を記載し
# ています。
#
# transforms.conf は $SPLUNK_HOME/etc/system/default/ にあります。カスタム設定を設定するには、
# transforms.conf を $SPLUNK_HOME/etc/system/local に置いてください。例は、transforms.conf.example を参照してく
# ださい。
# transforms.conf への設定変更は、次の検索文字列を Splunk Web で入力することにより有効にできます。
#
# | extract reload=T
#
# 設定ファイル(優先順位を含む)についての詳細は、
# http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
# 照してください。
[<unique_stanza_name>]
* スタンザに名前を付けます。props.conf を設定する際にこの名前を使用します。
# 例、props.conf スタンザで、TRANSFORMS-<value> = <unique_stanza_name>を入力します。
* このスタンザ名に、次の属性/値のペアを続けてください。
* 各属性のエントリを指定しない場合、Splunk はデフォルト値を使用します。
REGEX = <正規表現>
* データを操作する正規表現を入力します。
* 検索時に変換が生じる場合：
* 正規表現の名前キャプチャリンググループはフィールドに直接抽出されます。つまり、フォーマットを指定する必要はあり
# ません。
* フィールド名とフィールド値の両方が、正規表現を使用して抽出される場合、次の特別キャプチャリンググループを使用し
# て、FORMAT _KEY_<文字列>, _VAL_<文字列>でマッピングの指定をスキップできます。
* 例、次は同じことを意味します。
* フォーマットを使用する：
* REGEX = ([a-z]+)=([a-z]+)
* FORMAT = $1::$2
* フォーマットを使用しない
```

* REGEX = (?<_KEY_1>[a-z]+)=(?<_VAL_1>[a-z]+)

* デフォルトは空白。

* 注意：このオプションは、インデックス/検索時間 KV 抽出の両方で有効です。

LOOKAHEAD = <整数>

* イベントを検索する文字数を指定します。

* デフォルトは 256。

* 注意：このオプションは、インデックス時間 KV 抽出にのみ有効です。

DEST_KEY = <KEY>

* 正規表現の結果を保存する場所を指定します。

* 下記の KEY を使用します。

* 注意：このオプションは、インデックス時間 KV 抽出にのみ有効です。

format = <文字列>

* イベントのフォーマットを指定します。追加したフィールド名または値などを指定。

* \$n (例：\$1、\$2 など)を使用して、それぞれの正規表現が一致したときの出力を指定します。

* 正規表現が n グループを持たない場合、そのマッチングは失敗します。

* 特別な識別子 \$0 は、この正規表現が実行される前に DEST_KEY にあるものと表示します。

* デフォルトは \$1。

* 注意：このオプションは、インデックス/検索時間 KV 抽出の両方で有効です。

WRITE_META = <true | false>

* 自動的正規表現をメタデータに書き出します。

* DEST_KEY = meta の代わりに使用します。

* デフォルトは false。

* 注意：このオプションは、インデックス時間 KV 抽出にのみ有効です。

DEFAULT_VALUE = <文字列>

* これが設定され、正規表現(上記)が失敗すると、この値を DEST_KEY に書き込みます。

* デフォルトは空白。

* 注意：このオプションは、インデックス時間 KV 抽出にのみ有効です。

SOURCE_KEY = <文字列>

* この KEY に正規表現の実行を設定します。

* デフォルトは _raw(ローイベント)。

* インデックス時間転換には、下記の KEY を使用します。

* 検索時間抽出には、このフィールド抽出の実行時に利用できるフィールドを使用します。

* 注意：このオプションは、インデックス/検索時間 KV 抽出の両方で有効です。

REPEAT_MATCH = <true | false>

* SOURCE_KEY で正規表現を複数回実行するかどうかを指定します。

* REPEAT_MATCH は、最後の一致が停止すると開始し、その以上一致しなくなるまで続きます。

* デフォルトは false。

* 注意：このオプションは、インデックス時間 KV 抽出にのみ有効です。

DELIMS = <引用符で囲まれた文字列リスト>

- * データをキー-値ペアに分離し、次にそのペアからキーを分離するための区切り文字を設定します。
- * 注意：区切り文字は必ず " " を使って囲みます (エスケープするには、¥ を使います)。
- * 通常、2 セットの区切り文字を指定する必要があります。
 - * 第 1 の区切り文字はキー/値ペアを抽出します。
 - * 第 2 の区切り文字はそのキーをその値から分離します。
- * 区切り文字を 1 セットだけ入力した場合は、抽出されたトークンは次のようになります。
 - * FIELDS が入力された場合 (下記)、FIELDS からの名前が付けられます。
 - * または、偶数のトークンがフィールド名として使用され、奇数トークンがフィールド値になります。
- * 連続区切り文字は、フィールド名のリストが指定される場合を除いて、消費されます。
- * 注意：このオプションは、検索時間 KV 抽出にのみ有効です。

FIELDS = <引用符で囲まれた文字列リスト>

- * DELIMS を使用して抽出したフィールド値の名前をリストにします。
- * 注意：フィールド名がスペースまたはカンマを含む場合、それらを " " で囲む必要があります (エスケープするには ¥ を使用する)。
- * デフォルトは ""。
- * 注意：このオプションは、検索時間 KV 抽出にのみ有効です。

MV_ADD = <論理式>

- * 既に存在するフィールドを見つけたときのエクストラクターの動作コントロールするオプションです。
- * true に設定すると、エクストラクターはそのフィールドを複数値フィールドにし、その新規に見つけた値を付加します。その他の場合、その新規に見つけた値は廃棄されます。
 - * デフォルトは false。
 - * 注意：このオプションは、検索時間 KV 抽出にのみ有効です。

CLEAN_KEYS = <論理式>

- * 検索タイムで抽出されたキーをクリーニングするかどうかをコントロールするオプションです。キークリーニングは、英数字でない文字のアンダーバー置換として定義されます。先頭のアンダーバーと数字は切り落とされます。
 - * デフォルトは true。

CAN_OPTIMIZE = <論理式>

- * Splunk がこの抽出を最適化できるかどうかをコントロールするオプションです。抽出は、抽出されたフィールド (抽出による) のどれもが検索の正しい評価のためには必要ないと Splunk が判断できる場合に、無効にされます。このオプションを false に設定することはほとんどありません。
 - * デフォルトは true。

ルックアップテーブル

- # 注意：ルックアップテーブルは検索中のみ使用されます。

filename = <文字列>

- * 固定ルックアップファイルの名前。

* ファイルは、<app_name>の場合に、\$SPLUNK_HOME/etc/<app_name>/lookups/にある必要があります。または \$SPLUNK_HOME/etc/system/lookups/にある必要があります。

* ファイルが複数 'lookups' ディレクトリにある場合、階層化は実行されません。

* 標準設定ファイルの優先順位が、あいまいさの解消のために使用されます。

max_matches = <整数>

* 各入カルクアップ値に関するマッチングの最大数。

* デフォルトは、時間情報が付いていない場合は 100、時間情報が付いている (time_field を指定) 場合は 1。

* 時間情報が付いていない場合、最初 (ファイルの順番) の <整数> エントリーが使用されます。

* 時間情報が付いている場合、時間の降順による最初の <整数> が使用されます。

min_matches = <整数>

* 各入カルクアップ値に関するマッチの最小数。

* デフォルトは、時間情報が付いている場合と付いていない場合の両方で 0 です。つまり、マッチしない場合、何も出力されません。

* ただし、min_matches > 0 の場合、min_matches より小さい数のマッチを取得して、その default_match を排出します。

default_match = <文字列>

* min_matches > 0 で、指定の入力で min_matches 未満を得た場合、min_matches のしきい値に達するようにこの default_match 値を 1 回以上書き出します。

external_cmd = <文字列>

* 検索の実行を呼び出すためのコマンドおよび引数。

* この文字列は、シェルコマンドのようにパースされます。

* 第 1 の引数は、\$SPLUNK_HOME/etc/<app_name>/bin * にある python スクリプトとなります。このフィールドが存在するということは、ルックアップが外部コマンドベースであることを意味します。

fields_list = <文字列>

* 外部コマンドベースがサポートするすべてのフィールドの、カンマおよびスペースで区切られたリスト。

external_type = python

* 外部コマンドベースのタイプ。

* 現在、python のみがサポートされています。

time_field = <文字列>

* 時間情報が付いている (つまり時間に拘束される) ルックアップについては、タイムスタンプを表すルックアップテーブルのフィールドを指定します。

* デフォルトは <空文字列> で、これはルックアップが時間情報が付いていないことを意味します。

time_format = <文字列>

* 時間情報が付いているルックアップについては、タイムスタンプフィールドの 'strptime' フォーマットを指定します。

* サブセカンドを含めることができますが、それらは無視されます。

* デフォルトフォーマットは純粋 UTC 時間です。

max_offset_secs = <整数>

* 時間情報が付いているルックアップで一致させるため、イベント時間がルックアップエントリー時間より進むことが許され

る最大時間(秒)

* デフォルトは 2000000000(最大なし)。

min_offset_secs = <整数>

* 時間情報が付いているルックアップで一致させるため、イベント時間がルックアップエントリー時間より進んでいなければ
ならない最小時間(秒)

* デフォルトは 0。

キー:

* 注意: キーは、大文字と小文字を区別します。次のキーを記載とおりに正確に使用してください。

queue : イベントの送信先キューを指定します(parsingQueue、nullQueue、indexQueue を指定可能)。

_raw : イベントのローテキスト。

_done : 何らかの文字列を設定すると、これがストリームの中で最後のイベントとなります。

_meta : あるイベントのメタデータのスペースで区切られたリスト。

_time : イベントのタイムスタンプ。1/1/1970 UTC からの秒数で指定。

MetaData:FinalType : イベントのイベントタイプ。

MetaData:Host : イベントに関連するホスト。

この値は、"host::"の接頭辞を付ける必要があります。

_MetaData:Index : イベントを保存するインデックス。

MetaData:Source : イベントに関連するソース。

この値は、"source::"の接頭辞を付ける必要があります。

MetaData:Sourcetype : イベントのソースタイプ。

この値は、"sourcetype::"の接頭辞を付ける必要があります。

* 注意: '_'が先頭に付いたキーは、通常、Splunk でインデックスされません。

transforms.conf.example

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

#

これは transforms.conf の例です。正規表現および transforms のルールを作成するには、このファイルを使用します。

このファイルは props.conf と平行して使用します。

#

この設定の1つまたは複数を使用するには、その設定ブロックを \$SPLUNK_HOME/etc/system/local の transforms.conf
にコピーしてください。設定を有効にするには Splunk の再起動が必要です。

#

設定ファイル(優先順位を含む)についての詳細は、

<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork> にあるドキュメントを参
照してください。

注意: これは例のため、カスタマイズした値を入れ替えてください。

```

#       インデックス化フィールド：
[netscreen-error]
REGEX = device_id=[^ ]+\s+\[w+\](.*)"(?
FORMAT = err_code:.$1
WRITE_META = true

#       抽出フィールド：
[netscreen-error]
REGEX = device_id=[^ ]+\s+\[w+\](.*)"(?
FORMAT = err_code:.$1

#       上書きホスト：
[hostoverride]
DEST_KEY = MetaData:Host
REGEX = \s(\w*)$
FORMAT = host:.$1

#       抽出フィールド：
[netscreen-error]
REGEX = device_id=[^ ]+\s+\[w+\](.*)"(?
FORMAT = err_code:.$1

#       固定ルックアップテーブル
[mylookuptable]
filename = mytable.csv

#       相対ルックアップ
#       各入力値に対して1つのルックアップ値の出力を保証します。一致しなかった場合、"default_match"の値を使用します。
デフォルトは、"NONE"です。
[mylook]
filename = mytable.csv
max_matches = 1
min_matches = 1
default_match = nothing

#       外部コマンドベースルックアップテーブル
[myexternaltable]
external_cmd = testadapter.py blah
fields_list = foo bar

#       一時的静的検索テーブル
[staticwtime]
filename = mytable.csv
time_field = timestamp
time_format = %d/%m/%y %H:%M:%S

```

```

#      秘密データのマスク：
[session-anonymizer]
REGEX = (?m)^(.*)SessionId=\w+(\w{4}[\&"].*)$
FORMAT = $1SessionId=#####$2
DEST_KEY = _raw

#      代替インデックスへのルーティング：
[AppRedirect]
REGEX = Application
DEST_KEY = _MetaData:Index
FORMAT = Verbose

#      カンマで区切られた値をフィールドに抽出：
[extract_csv]
DELIMS = ","
FIELDS = "field1", "field2", "field3"

#      この例は、_raw から抽出された値を field1、field2、および field3 に割り当てます(抽出の順番で)。4 つ以上の値が
抽出された場合、マッチングフィールド名がない値が無視されます。

#      キー/値ペアを抽出します。
#      この例は、'|'で区切られたキー/値ペアを抽出します。一方、キーは値から'='で区切られます。
[pipe_eq]
DELIMS = "|", "="

#      この例は、'|'で区切られたキー/値ペアを抽出します。一方、キーは値から'='で区切られます。

```

user-seed.conf

user-seed.conf

次は user-seed.conf の仕様とファイル例です。

user-seed.conf.spec

```

#      Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#      user-seed.conf の仕様。Splunk のユーザー名とパスワードの設定を行います。
#      現在、1 人のユーザーのみが user-seed.conf で設定することができます。
#
#      デフォルトのユーザー名とパスワードを上書きするには、user-seed.conf を $SPLUNK_HOME/etc/system/default に
置いてください。設定を有効にするには Splunk の再起動が必要です。
#
#      設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参

```

照してください。

```
[user_info]
```

```
rename = <文字列>
```

* パスワードに関連付けるユーザー名。

* デフォルトは Admin。

```
password = <文字列>
```

* そのユーザーに設定するパスワード。

* デフォルトは changeme。

user-seed.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# これは user-seed.conf の例です。初期ログインを作成するにはこのファイルを使用します。
```

```
#
```

```
# 注意：デフォルトのスタートアップログインとパスワードを変更するには、このファイルは、初めて Splunk を開始する前に、$SPLUNK_HOME/etc/system/default/にある必要があります。
```

```
#
```

```
# この設定を使用するには、設定ブロックを$SPLUNK_HOME/etc/system/local/の user-seed.conf にコピーしてください。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
[user_info]
```

```
USERNAME = admin
```

```
PASSWORD = myowndefaultPass
```

web.conf

web.conf

次は web.conf の仕様とファイル例です。

web.conf.spec

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# このファイルは、Splunk のウェブインタフェースの設定に使用することができる属性と値を記載しています。
```

```
#
```

```
# web.conf は$SPLUNK_HOME/etc/system/default/にあります。カスタム設定を設定するには、web.conf を
```

`$SPLUNK_HOME/etc/system/local` に置いてください。例は、`web.conf.example` を参照してください。

設定を有効にするには Splunk の再起動が必要です。

#

設定ファイル(優先順位を含む)についての詳細は、

<http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork> にあるドキュメントを参照してください。

[settings]

* このスタンザ名で一般的な SplunkWeb 設定オプションを設定します。

* このスタンザ名に、次の属性/値のペアを続けてください。

* 各属性のエントリーを指定しない場合、Splunk はデフォルト値を使用します。

startwebservice = [0 | 1]

* SplunkWeb を開始するかどうかを設定します。

* 0 は SplunkWeb を無効化、1 は有効化します。

* デフォルトは 1。

httpport = <port_number>

* SplunkWeb を開始するにはこれが 필요합니다。

* 省略される、または 0 の場合、サーバーは HTTP リスナーを開始しません。

* デフォルトは 8000。

mgmtHostPort = <IP:port>

* Splunkd の場所。

* `http[s]://` を含めずに、IP アドレスだけを入力してください。

* デフォルトは `127.0.0.1:8089`。

enableSplunkWebSSL = [True | False]

* `http` と `https` を切り替えます。

* `https` と SSL を有効にするには、`true` に設定します。

* デフォルトは `false`。

sslport = <port_number>

* フロントエンドを開始するにはこれがなければなりません。

* 省略する、または 0 の場合、サーバーは SSL リスナーを開始しません。

privKeyPath = @OsDirSep@certs@OsDirSep@privkey.pem

caCertPath = @OsDirSep@certs@OsDirSep@cert.pem

* ウェブ SSL 証明書のパスおよび名前を指定します。

* パスは、`$SPLUNK_HOME/share/splunk` からの相対パスです。

serviceFormPostURL = <http://headlamp.Splunk.com/event/add>

userRegistrationURL = https://www.Splunk.com/index.php/pre_reg?destination=prod_reg

updateCheckerBaseURL = <http://quickdraw.Splunk.com/js/>

* これらは、設定可能な、さまざまな Splunk.com です。

* `updateCheckerBaseURL` を 0 に設定すると、SplunkWeb が Splunk.com に Splunk の新規バージョン検索を停止します。

```

enable_insecure_login = [True | False]
*      /account/insecurelogin エンドポイントが有効かどうかを指示します。
*      代替 GET ベース認証メカニズムを提供します。
supportSSLV3Only = [True | False]
*      true の場合、SSLv3 接続のみを許可します。
*      注意：これを有効にすると、ブラウザによっては問題が生じることがあります。
root_endpoint = <URI を先頭に付けた文字列>
*      アプリケーションサーバーがリッスンするルート URI パスを定義します。
*      デフォルト設定は '/' です。
*      例：Splunk UI を http://splunk:8000/splunkui でプロキシしたい場合は、root_endpoint を設定します。
static_endpoint = <URI を先頭に付けた文字列>
*      スタティックコンテンツへのパス。
*      ここでのパスは自動的に上記で定義した root_endpoint に付加されます。
*      デフォルトは /static。
static_dir = <相対ファイルシステムパス>
*      実際にスタティックコンテンツを保持するディレクトリ。
*      他の場所に置く場合、絶対 url となる場合もあります。
*      デフォルトは share/splunk/search_mrsparkle/exposed。
tools.staticdir.generate_indexes = [1 | 0]
*      ウェブサーバーが、スタティックディレクトリをリスト出力するディレクトリをサーブするかどうかを指示します。
*      デフォルトは 0(false)。
template_dir = <相対ファイルシステムパス>
*      mako テンプレートへのベースパス。
*      デフォルトは share/splunk/search_mrsparkle/templates。
module_dir = <相対ファイルシステムパス>
*      UI モジュールアセットへのベースパス。
*      デフォルトは share/splunk/search_mrsparkle/modules。
enable_gzip = [True | False]
*      ウェブサーバーが応答に gzip 圧縮を適用するかどうかを決定します。
*      デフォルトは True。
use_future_expires = [True | False]
*      /static ファイルの Expires ヘッダが far-future の日付に設定されるかどうかを決定します。
*      デフォルトは True。
flash_major_version = <整数>
flash_minor_version = <整数>
flash_revision_version = <整数>
*      Flash プラグインバージョンの最低要件を指定します。
*      Flash サポート、3 つの部分に分解されます。

```

* 現在、最低 Shockwave Flash 9.0 r124 が必要です。

enable_proxy_write = [True | False]

* /splunkd プロキシエンドポイントが POST オペレーションを許可するかどうかを指示します。

* True の場合、GET と POST オペレーションの両方が Splunkd にプロキシされます。

* False の場合、GET オペレーションのみが Splunkd にプロキシされます。

* これは通常、セキュリティのための無効にする必要があります。

* デフォルトは False。

js_logger_mode = [None | Firebug | Server]

* JavaScript ログモード。

* 利用可能モード： None、Firebug、Server

* None モード： ログしない

* Firebug モード： より古いマイナーバージョンがあるか、それに従う場合には、デフォルトで Firebug を使用します。

* Server モード： 定義されたサーバエンドポイントにログします。

* モード実装の詳細、および自分で作成したい場合は、js/logger.js Splunk.Logger.Mode を参照してください。

* デフォルトは Firebug。

js_logger_mode_server_end_point = <URI 相対パス>

* JavaScript ログメッセージをポストするためのサーバエンドポイントを指定します。

* js_logger_mode = Server の場合に使用します。

* デフォルトは util/log/js。

js_logger_mode_server_poll_buffer = <整数>

* JavaScript ログバッファをチェック、ポストおよびクリアするインターバルをミリ秒で指定します。

* デフォルトは 1000。

js_logger_mode_server_max_buffer = <整数>

* JavaScript ログバッファをポストおよびクリアする最大サイズしきい値を指定します。

* デフォルトは 100。

poller_timeout_interval = <整数>

* クライアント側のポーラーがタイムアウトになった場合の、その後の時間を指定します(ミリ秒)。

* デフォルトは 600000(10分)。

js_no_cache = [True | False]

* js キャッシュコントロールを切り替えます。

* デフォルトは False。

enable_insecure_login = [True | False]

* GET ベースログインエンドポイントがアクティブかどうかを指示します。

* True の場合、/account/insecurelogin?username=USERNAME&password=PASSWD が利用可能です。

* False の場合、メインの/account/login エンドポイントのみ利用可能です。

* デフォルトは False。

enable_autocomplete_login = [True | False]

* メインログインページがブラウザにユーザー名のオートコンプリートを許可するかどうかを指示します。

```

* True の場合、ブラウザは、ユーザー名フィールドに、オートコンプリートドロップダウンを表示します。
* False の場合、ユーザー名フィールドにオートコンプリートドロップダウンを表示しないようブラウザに指示します。
* デフォルトは True。
#
# cherryypy HTTP サーバー設定
#
server.thread_pool = <整数>
* アプリケーションサーバーが保持することができるスレッド数を指定します。
* デフォルトは 10。
server.socket_host = <ip アドレス>
* ホスト値は、IPv4 または IPv6 アドレス、またはその他の適切なホスト名。
* 文字列 'localhost' は、'127.0.0.1' と同義です (または、使用しているホストファイルが IPv6 の場合、 ':::1')。文字列 '0.0.0.0' は特別な IPv4 エントリーであり、"任意のアクティブインタフェース"(INADDR_ANY)、また、 ':::' は、IPv6 では類似の IN6ADDR_ANY です。 空白文字列または None は許可されません。
* デフォルトは 0.0.0.0。
log.access_file = <ファイル名>
* HTTP アクセスログファイル名を指定します。
* デフォルトの Splunk /var/log ディレクトリに保存されます。
* デフォルトは web_access.log。
log.error_file = <ファイル名>
* HTTP エラーログファイル名を指定します。
* デフォルトの Splunk /var/log ディレクトリに保存されます。
* デフォルトは web_service.log。
log.screen = [True | False]
* ランタイム出力がインタラクティブ tty の内側で表示されるかどうかを指示します。
* デフォルトは True。
request.show_tracebacks = [True | False]
* 例外トレースバックが致命的例外状態にあるユーザーに表示されるかどうかを指示します。
* デフォルトは True。
engine.autoreload_on = [True | False]
* アプリケーションサーバが、python ファイルが変更されたことを検知した場合に、自動再起動するかどうかを指示します。
* デフォルトは False。
tools.sessions.on = True
* ユーザーセッションサポートが有効かどうかを指示します。
* 常に True である必要があります。
tools.sessions.timeout = <整数>
* ユーザーセッションが期限切れになるまでの非アクティブの時間(分)を指定します。
* デフォルトは 60。

```

`response.timeout = <整数>`

- * サーバーの応答完了を待つ時間(秒)を指定します。
- * 大きなファイルをアップロードするなど、リクエストによっては、長い時間かかる場合があります。
- * デフォルトは 7200。

`tools.sessions.storage_type = [file]`

`tools.sessions.storage_path = <ファイルパス>`

- * セッション情報保存メカニズムを指定します。
- * RAM ベースのセッションを代わりに使用するには、次の 2 行をコメント文とします。
- * Splunk ツリーの外側でセッションを保存するには、絶対パスを使用します。
- * デフォルトは `storage_type=file`、`storage_path=var/run/splunk`。

`tools.decode.on = [True | False]`

- * Cherrypy コントローラメソッドが受信するすべての文字列をユニコードでデコードするかどうかを指示します。
- * 警告： これを無効にすると、アプリケーションがダウンします。なぜなら、すべての受信文字列をユニコードを仮定するからです。
- * デフォルトは `True`。

`tools.encode.on = [True | False]`

- * すべてのコントローラメソッド応答文字列を `python` で UTF-8 文字列オブジェクトにエンコードします。
- * 警告： これを無効にすると、上位バイト文字エンコーディングに失敗する原因となります。
- * デフォルトは `True`。

`tools.encode.encoding = <コーデック>`

- * すべての送信文字を強制的に UTF-8 にエンコードします。
- * これは、`tools.encode.on` を `True` に設定した場合にのみ機能します。
- * これを `utf-8` に設定することにより、Cherryppy が `Accept-Charset` ヘッダをチェックする際のデフォルト動作は上書きされ、`utf-8` による出力が強制されます。特定のブラウザインストールが他の文字エンコーディング(`Latin-1 iso-8859-1` など)を受信する必要がある場合にのみ、これを変更してください。
- * 警告：この変更は、ユーザーの責任で行ってください。
- * デフォルトは `utf08`。

`pid_path = <ファイルパス>`

- * PID ファイルへのパスを指定します。
- * デフォルトは `var/run/splunk/splunkweb.pid`。

web.conf.example

Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0

これは web.conf の例です。データウェブ設定を設定するにはこのファイルを使用します。

この設定の 1 つまたは複数を使用するには、その設定ブロックを `$SPLUNK_HOME/etc/system/local` の `web.conf` にコピーしてください。設定を有効にするには Splunk の再起動が必要です。

```

#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#       このスタンザヘッディングは変更箇所の前に置く必要があります。
[settings]
#       デフォルトのポート番号を変更する：
httpport = 12800
#       SSL をオンにする：
#       注意： パスは$SPLUNK_HOME に対する相対パスです。
enableSplunkWebSSL = true
sslport = 8080
privKeyPath = /certs/myprivatekey.pem
caCertPath = /certs/mycacert.pem

```

wmi.conf

wmi.conf

次は wmi.conf の仕様とファイル例です。

wmi.conf.spec

```

#       Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
#
#       このファイルは、Splunk からの WMI アクセス設定のための属性/値ペアを記載しています。
#
#       wmi.conf は$SPLUNK_HOME\etc\system\default\にあります。カスタム設定を設定するには、wmi.conf を
$SPLUNK_HOME\etc\system\local\に置いてください。これは、wmi.conf.example を参照してください。設定を有効にするには
Splunk の再起動が必要です。
#
#       設定ファイル(優先順位を含む)についての詳細は、
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参
照してください。
#####
#---- グローバル設定----
#####
[settings]
*       この設定スタンザは、さまざまなランタイムパラメータを指定します。
*       この中のスタンザ全体とすべてのパラメータはオプションです。

```

```

*      スタンザがない場合、Splunk はシステムデフォルトを使用します。
initial_backoff = <整数>
*      WMI プロバイダへの接続について、その最初の接続エラーの後再試行するまで待つ時間を秒で指定します。
*      接続エラーが継続する場合、その待ち時間は、max_backoff に達するまで 2 倍になります。
*      デフォルトは 5。
max_backoff = <整数>
*      再接続を試みる最大時間(秒)。
*      デフォルトは 20。
max_retries_at_max_backoff = <整数>
*      max_backoff に達すると、これを何回も試みます。
*      max_retries の後に再接続に失敗すると、影響に試みません(再起動するまで)。
*      デフォルトは 2。
result_queue_size = <整数>
*      WMI プロバイダからの結果をキューに置き、その後出力に送信します。
*      デフォルトは 1000。
checkpoint_sync_interval = <整数>
*      状態データ(イベントログチェックポイント)をディスクに書き込む最小待ち時間(秒)。
*      デフォルトは 2。
heartbeat_interval = <整数>
*      WMI プロバイダとの接続をテストするハートビートインターバル(ミリ秒)。
*      デフォルトは 500。
#####
#---- 入力別設定      ----
#####
[WMI:$NAME]
*      WMI スタンザは 2 種類あります：
*      Event log: イベントログを引き出します。event_log_file 属性を設定する必要があります。
*      WQL: ローカル WQL 要求を発行します。WQL 属性を設定する必要があります。
server = <カンマで区切られたリスト>
*      データを取得する元サーバーのカンマで区切られたリスト。
*      デフォルトはローカルマシン。
interval = <整数>
*      新規データをポーリングする頻度。
*      オプションではありません。
*      デフォルトはなし。
disabled = <1/0>
*      1 は有効、0 は無効。
*      デフォルトはなし。

```

* イベントログに特定の属性：

event_log_file = <アプリケーション、システムなど>

* WQL の代わりにこれを使用してソースを指定します。

* ポーリングするログファイルのカンマで区切られたリストを指定します。

* デフォルトはなし。

* WQL に特定の属性：

wql = <文字列>

* event_log_file を使用していない場合、これを使用します。

* データを WMI プロバイダから抽出する wql を指定します。

* 例えば、Name = "splunkd" の Win32_PerfFormattedData_PerfProc_Process から * を選択します。

namespace = <文字列>

* WMI プロバイダの場所。

* WMI プロバイダがあるネームスペース。

* 直接 WQL クエリ。

* デフォルトは root\。

wmi.conf.example

```
# Copyright (C) 2005-2009 Splunk Inc. All Rights Reserved. Version 4.0
```

```
#
```

```
# これは wmi.conf の例です。これらの設定は、WMI プロバイダからの入力をコントロールするために使用します。
```

```
# このファイルに関する詳細は、wmi.conf.spec および Splunk.com のドキュメントを参照してください。
```

```
#
```

```
# この設定の 1 つまたは複数を使用するには、その設定ブロックを $SPLUNK_HOME/etc/system/local の wmi.conf にコピーしてください。設定を有効にするには Splunk の再起動が必要です。
```

```
#
```

```
# 設定ファイル(優先順位を含む)についての詳細は、
```

```
http://www.splunk.com/base/Documentation/latest/Admin/HowDoConfigurationFilesWork にあるドキュメントを参照してください。
```

```
# このスタンザはランタイムパラメータを指定します。
```

```
[settings]
```

```
initial_backoff = 5
```

```
max_backoff = 20
```

```
max_retries_at_max_backoff = 2
```

```
result_queue_size = 1000
```

```
checkpoint_sync_interval = 2
```

```
heartbeat_interval = 500
```

```
# これらのスタンザはイベントログをローカルシステムから引き出します。
```

```
[WMI:LocalApplication]
```

```
interval = 10
event_log_file = Application
disabled = 0

[WMI:LocalSystem]
interval = 10
event_log_file = System
disabled = 0

[WMI:LocalSecurity]
interval = 10
event_log_file = Security
disabled = 0

# これらスタanzasは、ローカルシステムから性能データを収集します。
[WMI:LocalPhysicalDisk]
interval = 1
wql = select Name, DiskBytesPerSec, PercentDiskReadTime, PercentDiskWriteTime, PercentDiskTime disabled
= 0

[WMI:LocalMainMemory]
interval = 10
wql = select CommittedBytes, AvailableBytes, PercentCommittedBytesInUse, Caption from
Win32_PerfFormattedData_disabled = 0

[WMI:LocalSplunkdProcess]
interval = 1
wql = select * from Win32_PerfFormattedData_PerfProc_Process where Name = "splunkd"
disabled = 0
```

トラブルシューティング

コンタクトサポート

コンタクトサポート

連絡先情報は、メインサポートの連絡先ページを参照してください。

ここでは、Splunk サポートが問題の診断に用いるツールおよびテクニックについての情報を扱います。多くの事項は、自分で試すことができます。

注意： Splunk サポートにファイルまたは情報を送る前に、それを当社に送ることに懸念がないことを確認してください。当社は、下記のコマンドによる出力にセンシティブな情報が含まれていないように努力していますが、各ユーザーに個別のセキュリティポリシーを順守することについて保証はできません。

diag

diag コマンドは、使用している Splunk サーバーに関する基本的情報を収集します。情報には Splunk の設定詳細を含みません (\$SPLUNK_HOME/etc のコンテンツ、およびホストおよびソース名などのインデックスに関する一般情報など)。イベントデータや個人情報は含まれません。

\$SPLUNK_HOME/bin から次を実行します。

UNIX:

```
./splunk diag
```

Windows:

```
splunk diag
```

使用している環境で diag を実行することに問題がある場合、python スクリプトを直接実行することもできます。

cmd を使用して

```
./splunk cmd python /opt/splunk/lib/python2.5/site-packages/splunk/clilib/info_gather.py
```

これにより、splunk-diag.tar.gz (または.zip)が作成され、それを Splunk サポートに送ることでトラブルシューティングを行うことができます。

diag の出力を次のサポート担当にアップロードしてください。

- 企業向けサポート：http://www.splunk.com/index.php/track_issues
- コミュニティメンバー：http://www.splunk.com/index.php/send_to_splunk

ログレベルおよびデバックモードの開始

Splunk のロギングレベルは、\$SPLUNK_HOME/var/log/splunk/splunkd.log にある様々な項目を変更して、より詳細な情報を提供することができます。最も簡単な方法は、すべてのメッセージを--デバックオプションで有効にすること

です。これは性能に影響するため、日常的には使用しないでください。

- Splunk を実行している場合は停止します。
- 既存の `splunkd.log` ファイルを移動して、それを `splunkd.log.old` などの新しいファイル名にします。
- `splunk start --debug` を使って、Splunk をデバックモードで再起動します。
- 問題が生じた場合は、Splunk を停止します。
- 上記の新規 `splunkd.log` ファイルをどこかに移動し、古いファイルを復帰させます。
- 通常どおり Splunk を再起動し (`--debug` フラグなしで)、デバックロギングを無効にします。

特定のエリアを有効にして、性能への影響を最小限に抑えながら長期にわたるデバック詳細を収集することができます。`$SPLUNK_HOME/etc/log.cfg` ファイルのカテゴリ設定で、多くのカテゴリを `--debug` のように有効にすることなく、特定のログレベルを設定することができます。警告やエラーとマークされたメッセージのすべてが、Splunk が実際に問題を生じていることを示すわけではありません。中には、機能が使用されていないことを示すものもあります。

`splunkd.log` のデバックメッセージは検索により動的に有効することもできます。

デバックを有効にするには、次の検索を実行します。

```
| oldsearch !++cmd++::logchange !++param1++::root !++param2++::DEBUG
```

デフォルトのログレベルに戻るには、次の検索を実行します。

```
| oldsearch !++cmd++::logchange !++param1++::root !++param2++::WARN
```

特定のカテゴリメッセージを設定するには、"root"を適当なカテゴリで置き換えます。これにより、`log.cfg` の設定が変更されることはありません。再起動すると、ログレベルは `log.cfg` で定義されたものに戻ります。

注意：この検索は、「・・・のプライオリティが設定されているため検索実行に失敗しました」というメッセージを返します。これは正常です。

問題モニタリングファイルを調査するには、`FileInputTracker` と `selectProcessor` カテゴリを使います。

これらは、非常に詳細な情報を出力するため通常の `--debug` オプションでは有効になりません。

Splunk Web のデバック

`web.conf` で追加の Splunk Web デバックを有効にします。

```
[settings]
appLoggingLevel = DEBUG
```

Splunk Web プロセスを、`./splunk restart splunkweb` コマンドで再起動します。追加のメッセージが `$SPLUNK_HOME/var/log/splunk/web_service.log` ファイルに出力されます。

コアファイル

コアファイルを収集するには、`ulimit` を使用して、Splunk を起動する前に、最大ファイルサイズ設定を削除します。

```
# ulimit -c unlimited
```

```
# splunk restart
```

この設定は、ユーザーが特定のシェルで開始するプロセスにのみ影響します。したがって、新しいセッションでこれを行うこともできます。Linux では、Splunk を `--nodaemon` オプションで起動します (`splunk start --nodaemon`)。他のシェルでは、`splunk start splunkweb` を使って、ウェブインタフェースを手動で起動します。

システムによっては、コアは `core.1234` のように命名されている場合があります。番号はプロセス id を表しており、実行可能 `splunkd` を同じ場所にあります。

LDAP 設定

LDAP の設定で問題が生じた場合、通常、サポートでは次の情報が必要です。

- `$SPLUNK_HOME/etc/system/local/`からの `authentication.conf` ファイル。
- 役割を割り当てようとしているグループの `ldif`。
- 認証対象のユーザーの `ldif`。

場合によっては、`splunkd.log` または `web_service.log` のデバックが有効です。

サポートに送るデータサンプルの匿名化

サポートに送るデータサンプルの匿名化

Splunk には匿名化機能があります。そのアノニマイザーは、サンプルログファイルやイベントファイルを徹底的に調べ、ユーザー名、IP アドレス、ドメイン名などの識別データを、同じ語数やイベントタイプを有する架空の値で置換します。例えば、`user=carol@adalberto.com` という文字列を `user=plums@wonderful.com` に変えるなどです。これにより、Splunk ユーザーは、秘密情報や個人情報を自分のネットワークからさらけ出すことなく、ログデータを共有することができます。

匿名化されたファイルは、同じディレクトリにソースファイルをして書き込まれ、そのファイル名の先頭には `ANON-` が付加されます。例えば、`/tmp/messages` は、`/tmp/ANON-messages` のように匿名化されます。

Splunk の CLI からファイルを匿名化することができます。Splunk の CLI を使用するには、`$SPLUNK_HOME/bin/` ディレクトリに移動して、`./splunk` コマンドを使用します。

簡単な方法

ファイルを匿名化する最も簡単な方法は、アノニマイザーツールのデフォルトを使うことです。下記のセッションで説明します。ただし、現在使用している作業ディレクトリが `$SPLUNK_HOME/bin` である必要があります。これは、各バージョンのリリースで同じです。

CLI で以下のように入力します。

```
# ./splunk anonymize file -source /path/to/[filename]
# cp -p /var/log/messages /tmp
# cd $SPLUNK_HOME/bin
```

```

# splunk anonymize file -source /tmp/messages
Getting timestamp from: /opt/paul207/splunk/lib/python2.4/site-packages/splunk/timestamp.config
Processing files: ['/tmp/messages']
Getting named entities
    Processing /tmp/messages
Adding named entities to list of public terms: Set(['secErrStr', 'MD_SB_DISKS', 'TTY', 'target',
    Processing /tmp/messages for terms.
    Calculating replacements for 4672 terms.
=====
Wrote dictionary scrubbed terms with replacements to "/tmp/INFO-mapping.txt"
Wrote suggestions for dictionary to "/tmp/INFO-suggestions.txt"
=====
Writing out /tmp/ANON-messages
Done.

```

高度な方法

匿名化する(またはしない)用語および置換に使用する用語指定して、アノニマイザーをカスタマイズできます。この高度なコマンドは以下のように記述します。

```

# ./splunk anonymize file -source <filename> [-public_terms <file>] [-private_terms
<file>] [-name_

```

- filename
 - ◆ デフォルト: None
 - ◆ 匿名化するファイルのパスおよび名前。
- public_terms
 - ◆ デフォルト: \$SPLUNK_HOME/etc/anonymizer/public-terms.txt
 - ◆ 匿名化しない、ローカルで使用している用語のリスト。これは dictionary ファイルの付属ファイルとして使われます。
 - ◆ 次は入力のサンプルです:

```

2003 2004 2005 2006 abort aborted am apr april aug august auth
authorize authorized authorizing bea certificate class com complete

```
- private_terms
 - ◆ デフォルト: \$SPLUNK_HOME/etc/anonymizer/private-terms.txt
 - ◆ 秘密情報であるために匿名化する用語のリスト。
 - ◆ 次は入力のサンプルです:

```

481-51-6234
passw0rd

```
- name_terms

- ◆ デフォルト : `$SPLUNK_HOME/etc/anonymizer/names.txt`
- ◆ Splunk が匿名化された用語の置換に使用する、一般的な英語の個人名のグローバルリスト
- ◆ Splunk は、各イベントのデータパターンを同じに維持するために、常に同じ長さの名前で語を置換します。
- ◆ Splunk は、各名前を一旦 `name_terms` で使用し、ファイルを使用して同じ長さの文字列を置換します。名前がなくなると、ランダム化された文字列を使い始めます。それでもなお置換された各パターンを 1 つの匿名化された文字列に割り当てます。
- ◆ 次は入力のサンプルです。

```
charlie
claire
desmond
jack
```

- `dictionary`
- デフォルト : `$SPLUNK_HOME/etc/anonymizer/dictionary.txt`
 - ◆ `private_terms` のエントリで上書きされない限り、匿名化しない共通語のグローバルリスト
 - ◆ 次は入力のサンプルです。

```
algol
ansi
arco
arpa
arpanet
ascii
```

- `timestamp_config`
 - ◆ デフォルト : `$SPLUNK_HOME/etc/anonymizer/anonymizer-time.ini`
 - ◆ タイムスタンプのパーズの方法を決定する Splunk の内蔵ファイル。

出カファイル

Splunk のアノニマイザー機能は、3 つの新たなファイルを同じディレクトリにソースファイルとして作成します。

- `ANON-filename`
 - ◆ ソースファイルの匿名化バージョン。
- `INFO-mapping.txt`
 - ◆ このファイルは、どの用語がどの文字列に匿名化されたのかのリストを含みます。
 - ◆ 次は入力のサンプルです :

Replacement Mappings

```
-----
kb900485 --> L0200231
1718 --> 1608
transitions --> tstymnbkxno
reboot --> SPLUNK
```

cdrom --> pqyvi

- INFO-suggestions.txt
 - ◆ ローカルデータをより正確に匿名化するために、public_terms.txt or to private-terms.txt または public-terms.txt に、用語の外観と頻度に基づいて判断し追加する、ファイルで見つかった用語のレポート。
 - ◆ 次は入力のサンプルです：

Terms to consider making private (currently not scrubbed):

```
['uid', 'pci', 'lpj', 'hard']
```

Terms to consider making public (currently scrubbed):

```
['jun', 'security', 'user', 'ariel', 'name', 'logon', 'for', 'process', 'domain', 'audit']
```

探しているイベントが見つからない場合

探しているイベントが見つからない場合

入力を Splunk に追加する際、その入力は、使用しているアプリケーションに相対的に追加されます。Splunk に付属する *nix や Windows アプリケーションなど、アプリケーションによっては、入力データを特定のインデックス(*Nix および Windows の場合、'OS'インデックス)に書き込むものがあります。確実に Splunk にあるデータが見つからない場合、目的のインデックスを探しているかどうか確認してください。'OS'インデックスをデフォルトインデックスのリストに追加して、使用している役割で使用する場合があります。役割の詳細は、本書の役割に関するトピックを参照してください。

SuSE Linux: サーバーから正しくフォーマットされた応答を取得できません

SuSE Linux: サーバーから正しくフォーマットされた応答を取得できません

Splunk を SuSE サーバー上で実行しているユーザーは、検索実行中に、「サーバーから正しくフォーマットされた応答を取得できません；現在の検索をキャンセルします」というエラーメッセージを受け取ることがあります。あるいは、単にダッシュボードが正しく表示しない場合があります。この問題を解決するには、`/etc/mime.types` を編集します。次の 2 行を削除(またはコメント文にする)してください。

```
text/x-xsl xsl
```

```
text/x-xslt xslt xsl
```

さらに、以下の行を

```
text/xml xml
```

次のように変更してください:

```
text/xml xml xsl
```

これらの変更を行い、Splunk を再起動し、ブラウザのキャッシュをクリアします。

注意： プロキシを使用している場合、それもフラッシュする必要があります。[[Category:SuSE Linux]]

サポートの指示により使用するコマンドラインツール

サポートの指示により使用するコマンドラインツール

警告：事前に Splunk サポートに相談することなく、これらのコマンドを使用しないでください。

cmd

btool

Cmd 行修正およびバンドルの表示

構文

追加

```
./splunk cmd btool application name add
```

削除

```
./splunk cmd btool application name delete [prefix] [entry]
```

リスト表示

```
./splunk cmd btool application name list [prefix]
```

classify

gzdumper

listtails

locktest

locktool

```
./splunk cmd locktool
```

使用方法：

```
lock : [-l | --lock ] [dirToLock] <timeOutSecs>
```

```
unlock [-u | --unlock ] [dirToUnlock] <timeOutSecs>
```

Splunkd と同様にロックを取得/解除します。外部スクリプトを書いて、DB バケツをインデックスにコピーしたりそれからコピーする場合、それらを修正中は、db colddb と thaweddb ディレクトリをロックし、修正が終わったらロックを解除する必要があります。

parsetest

pcregextest

regextest

searchtest

signtool

署名

```
./splunk cmd signtool [-s / --sign] [<dir to sign>]
```

検証

```
./splunk cmd signtool [-v / --verify] [<dir to verify>]
```

/Applications/splunk/etc/log-cmdline.cfg のロギング設定を使用。

Splunk インデックスバケツの検証および署名を可能にします。署名を有する場合には、コールドで設定しスクリプトをリリースします。Signtool は、アーカイブの署名の検証を可能にします。

tsidxprobe

これは、インデックスファイル(.tsidx)を閲覧し、必要なフォーマット条件を満たしているかどうかを検証します。また、問題を生じる可能性のあるファイルの特定も行います。

\$SPLUNK_HOME/bin ディレクトリに移動します。 "source setSplunkEnv"を実行します。

次に、tsidxprobe を使って、この小さなスクリプトで各インデックスファイルを閲覧します。スクリプトはシェルから実行することができます(これは bash とともに機能します) :

```
1. for i in `find $SPLUNK_DB | grep tsidx`; do tsidxprobe $i >> tsidxprobeout.txt; done
```

(デフォルトのデータ保存先パスを変更している場合は、その新しい場所をこれを行う必要があります。)

tsidxprobeout.txt は、インデックスファイルからの結果を含んでいます。これを gzip し、E メールに添付して Splunk サポートに送ります。