

Star SHOW ユーザーズガイド

Yusuke Kurihara

2000年12月25日,Vol.2

概要

Star SHOW version1.0 の使用ガイドです。基本的な使い方、セットアップの詳細を説明します。このドキュメント Vol.2 は修正や追加がおこなわれています。

目次

1	アプリケーションのセットアップ	2
1.1	Windows での使用	2
1.2	UNIX での個人での使用	2
1.3	UNIX での管理者によるセットアップ	2
2	クイックスタート	3
3	オプションの詳細	4
4	カスタマイズ	5
5	各種ファイル書式	5
5.1	カスタマイズファイル	5
5.1.1	ファイル書式	5
5.1.2	キーワードタイプ	6
5.2	encode.conf	7
5.3	handler.conf	7
5.4	htmlenc.list	8
6	アプリケーションのアンインストール	8
7	Javadoc 用ドックレットの使用法	8
7.1	使用方法	8

1 アプリケーションのセットアップ

ここでは、いろいろな使用形態におけるアプリケーションのセットアップ法を説明します。各仕様環境、形態に合わせてごらんください。なお、付属 Java ドックレットの使用方法は [7](#) を参照してください。

1.1 Windows での使用

Windows での、最も簡単な実行方法は、`{INST_DIR}\lib\sshow1_0.jar`¹ をダブルクリックすることです。ショートカットをデスクトップやスタートメニューに作成しておき、それを利用することも可能です。実行にコマンドを使用する場合は、

```
java -jar {INST_DIR}\lib\sshow1_0.jar
```

を実行します。

1.2 UNIX での個人での使用

UNIX で個人で使用する場合、あるいは管理者権限がなく、ホームディレクトリ以下にアプリケーションを解凍する場合、実行は `java` コマンドを使用します。

```
% java -jar {INST_DIR}/lib/sshow1_0.jar
```

なお、`java` コマンドにパスが通っていることが前提です。このコマンドを省略するために、スクリプトを用意する方法もあります。 [1.3](#) を参照してください。

1.3 UNIX での管理者によるセットアップ

UNIX 系 OS の管理者が、一般のユーザーも使用できるようにセットアップする場合、`/usr/local/` などで、ダウンロードしたアーカイブを解凍し、実行パスの通ったディレクトリに (例:`/usr/bin/`) アプリケーション起動のためのスクリプトを用意します。`{INST_DIR}` はインストールしたディレクトリ名に置き換えてください。

```
#!/bin/sh
java -jar {INST_DIR}/lib/sshow1_0.jar
```

を `sshow` などの適当な名前でも保存し、これに実行パーミッションを設定します。

¹Star SHOW のインストールされたディレクトリを `{INST_DIR}` であらわします。

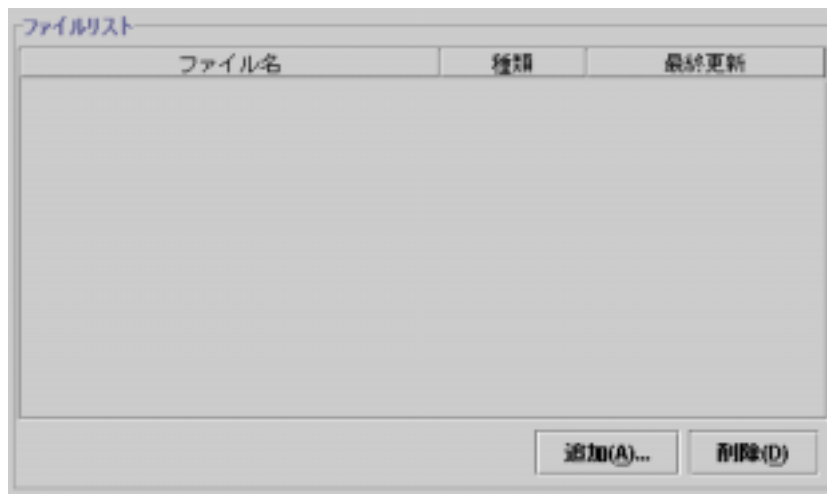


図 1: ファイルリスト

```
% chmod 755 sshow
```

一般ユーザーは、sshow を実行することでアプリケーションを使用できるようになります。

2 クイックスタート

簡単に、HTML を生成するための手順を説明します。オプションなどの詳細は、事項以降を参照してください。

対象ファイルの選択 図 1 のように、生成する対象となるファイル一覧は、ファイルリストのテーブルに表示されます。対象を追加するには、“追加” ボタンを押して選択するか、ファイルをドラッグアンドドロップによりテーブルに追加します。ただし、この方法は、実行確認をおこなった Vine Linux (GNOME 使用) ではドラッグアンドドロップはうまくいきません。対象から外したい場合は、テーブル中の削除したいファイル名を選択して、削除ボタンを押下します。なお、ファイルリストには、ディレクトリを追加することも可能です。この場合、ディレクトリの中のディレクトリも再起的に処理するかどうかは、オプションで設定してください。

オプションの設定 オプションは“SHOW オプション”の項目で設定できます(図 2)。とりあえず必要と考えられるオプションは、再起処理と出力先です。ファイルリストにディレクトリを追加した場合で、そのディレクトリ内のディレクトリも

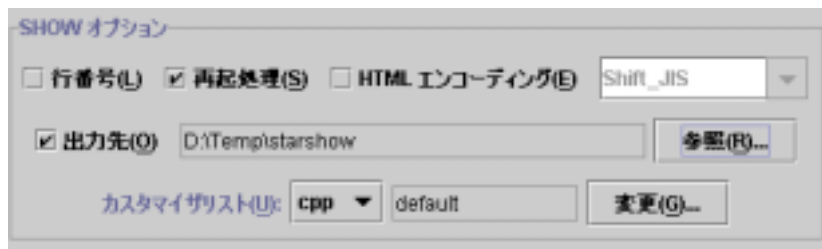


図 2: SHOW オプション

再起的に処理したい場合は、“再起処理”にチェックをいれてください。出力先は、生成する HTML ファイルを置く場所を示します。“出力先”にチェックを入れ、“参照...” ボタンを押下します。出力先として設定したいディレクトリを選択します。

HTML の生成 “HTML の生成” ボタンを押下すれば、処理が始まります。処理対象ファイルが多く時間がかかる場合は進捗ダイアログを表示し、処理の進行状況を表示します。

3 オプションの詳細

以下に SHOW オプションの詳細を列挙します。

行番号 行番号を出力に含める場合にチェックを入れます。チェックを入れると、各行の先頭に行番号が付加されます。

再起処理 HTML を生成する対象ファイルとして、ディレクトリを選択した場合にその中のディレクトリ以下を、再起的に処理する場合にチェックします。チェックしない場合、ディレクトリ内のディレクトリは無視され、ファイルのみが処理されます。

HTML エンコーディング 出力する HTML のエンコーディングを選択したい場合にチェックします。出力したいエンコーディングを選択してください。選択項目に希望のエンコーディングがない場合、新たに入力することが可能ですが、設定が必要です(参照 5.4)。チェックを外している場合、利用している環境のデフォルトエンコーディングが利用されます。

出力先 生成した HTML ファイルの出力先を指定します。指定しない場合、現在のユーザーのカレントディレクトリが使用されます。なお、処理の対象がファイルの場合、出力先ディレクトリに直接置かれ、処理の対象がディレクトリの場合は、出力先ディレクトリに同様のディレクトリ階層が生成されます。出力先の指定を有効にするためには、チェックボックスを ON にしてください。

カスタマイザリスト 処理するファイルタイプに応じてカスタマイズに使用するカスタマイザを設定します。通常は“default”となっていて、あらかじめ用意された設定が使用されます。タイプのコンボボックスからカスタマイザを変更するタイプを選択します。タイプは拡張子で区別されています。“変更...” ボタンをクリックすると、現在選択できるカスタマイザの一覧から選択ができます。もし選択できるカスタマイザがない場合、先に 4 にしたがってカスタマイザを作成してください。

4 カスタマイズ

ここでカスタマイズとは、HTML を生成する際のキーワードなどのハイライト色を好みに合わせて変更することです。このバージョンでは、満足なカスタマイザの作成ユーティリティがまだ実装されておらず、簡易エディタが付属するのみです。カスタマイザを編集するには、“ツール”メニューから”カスタマイザの編集...”を選択します。現在登録されているカスタマイザ一覧が表示されます。新しく作成する場合は、“追加”ボタンを、編集、あるいは削除する場合は、“編集”、“削除”をそれぞれ選択してください。編集が完了すれば“保存”によりその内容を保存します。この際ファイル名を入力しますが、拡張子を .cst としたファイル名で、ホームディレクトリの .show に保存します。こうすることで、カスタマイザ変更の際のリストにそのカスタマイザ名 (拡張子 .cst を省略した名前) が含まれます。なお、これらのユーザーインターフェースは次のバージョン以降で改善の予定です。

5 各種ファイル書式

カスタマイズファイル、その他 {INST_DIR}/lib/conf/ にある設定ファイルの書式です。{INST_DIR}/lib/conf/ にある設定ファイルは通常編集する必要はありません。

5.1 カスタマイズファイル

5.1.1 ファイル書式

カスタマイズファイルは、以下の書式で記述します。

```
# コメント行
keyword-type: color
keywotd-type color
keyword-type=color
```

行頭が # で始まる行はコメントとみなします。keyword-type は、ハイライトの色を

設定するためのキーワードのグループ名です。これは次に説明します。*color* は、スタイルシート (CSS) で指定できる形式で指定します。例えば、*green* といった色名や、*#9966cc* のような形式です。*keyword-type* と *color* の間は、*:*, (空白), *=* のいずれかで区切ります。なお、カスタマイズしたい項目のみの設定で構いません。ほかの値は、デフォルト色 (カスタマイズファイルを使用しない場合の色) が使用されます。なお *keyword-type* に対し色名を指定しなかった場合、ハイライトを中止させることができます。過剰なハイライトが気になる場合、この方法を使用してハイライトの対象を減らすことができます。

5.1.2 キーワードタイプ

上記カスタマイズファイルで指定する *keyword-type* はプログラミング言語によって異なります。それぞれ以下のように分類されています。

Java

modifier アクセス修飾子をはじめとするメンバの修飾子を意味します。

primitive *int* など、プリミティブ型をあらわすキーワードを意味します。

control *if*, *else*, *for* など、制御に関するキーワードを意味します。

class_type キーワード *class* を意味します。

generalizer キーワード *extends*, *implements* を意味します。

class_name クラス名を指します。*class* 宣言においてクラス名がハイライトされます。

package キーワード *package* を意味します。

import キーワード *import* を意味します。

package_name *package*, *import* 宣言に対するパッケージ名を指します。

line_number 行番号を含める際の行番号: を意味します。

comment コメントを意味します。コメントはハイライトされます。

C, C++, ヘッダ, objective-C

line_number 行番号を含める際の行番号: を意味します。

comment コメントを意味します。コメントはハイライトされます。

preprocess プリプロセスに関するキーワードを意味します。

primitive int など、プリミティブ型をあらわすキーワードを意味します。

modifier 関数、クラスメンバなどの、修飾子を意味します。

define typedef や struct といったキーワードを意味します。

extern extern を意味します。

EXTERN EXTERN を意味します。

control if, else, for など、制御に関するキーワードを意味します。

op sizeof を意味します。

obj_c interface や implementation など Objective-C のキーワードを意味します (間違っているかも)。

例 行番号を黒で、修飾子を黄色で表示するようにするカスタマイズファイルの例です。

```
line_number:#000000  
modifier:yellow
```

5.2 encode.conf

encode.conf は、Java エンコーディング名と IANA Charset Registry のエンコーディング名とをマッピングします。Java エンコーディング名は Java 2 SDK ドキュメントのサポートするエンコーディング一覧を参照してください。

```
# コメント行  
Java encoding=HTML encoding
```

行頭が # で始まる行はコメントとみなします。HTML encoding に指定した文字列はそのまま HTML のメタ情報として charset の指定に使用されます。

5.3 handler.conf

handler.conf は、現在のところ編集しても使用できませんので編集しないでください。

5.4 `htmlenc.list`

`htmlenc.list` は、オプションにおいて HTML エンコードを選択する際、コンボボックスに表示するエンコーディングの一覧です。なお現在このリストは、エンコーディングについて `encode.conf` の逆の変換に使用されます。したがって、HTML エンコーディングにあらかじめ用意されていないエンコーディングを指定したい場合、このファイルにそのエンコーディング名を追加しますが、これはコンボボックスに含まれることとなります。また、エンコーディングを追加する場合、ここに記述したものと逆の対応づけを `encode.conf` に追加しなくてはなりません。

6 アプリケーションのアンインストール

アプリケーションを削除するためには、解凍したファイル、作成したシェルスクリプト (作成した場合) を削除してください。各ユーザーのホームディレクトリの `.show` には、カスタマイズファイルが保存されています。必要でないようなら削除してください。

7 Javadoc 用ドックレットの使用法

この Star SHOW には、Javadoc ツールによるドキュメントから、ソースへのリンクを設けられるように SHOW ドックレットを付属しています。このドックレットを使用すると、各クラスドキュメントの関連項目の下に、Source の項が追加され、SHOW により生成された HTML へのリンクが作成されます。生成されるソースの HTML は、各クラスドキュメントと同じディレクトリに、`{source_filename}.html` という名前で生成されます。

なお、このドックレットを使用する場合、通常の SHOW の HTML 生成で設定できるオプション (行番号、カスタマイズなど) は現在のところ指定できるようなっていません。これは次のバージョン以降で対応する予定です。

7.1 使用方法

以下は付属する起動スクリプトあるいはバッチファイルを使用することによるドックレットの使用法です。

Windows 系 SHOW ドックレットを使用するように通常の javadoc ツールに対するオプションをセットしたバッチファイルが `{INST_DIR}` に `showdoc.bat` という名前で含まれます。この引数に、通常 javadoc を使用するときと同様に、オプションをセットすれば使用できます。すなわち、これまで `javadoc [OPTIONS]` のように実

行していたものを `showdoc.bat [OPTIONS]` とタイプすることで使用可能です。なお、バッチファイルは、`{INST_DIR}` において使用することが前提となった内容です。インストール先に応じて、バッチファイルの `-docletpath` の引数 `lib\sshowdlt.jar` の部分を `{INST_DIR}\lib\sshowdlt.jar` に変更してください。また、このバッチファイルを使用し、オプションを与える場合、引数の数が 9 個に制限されますので注意してください。

Unix 系 SHOW ドックレットを使用するように通常の javadoc ツールに対するオプションをセットしたスクリプトが `{INST_DIR}` に `showdoc.sh` という名前で含まれます。この引数に、通常 javadoc を使用するときと同様に、オプションをセットすれば使用できます。すなわち、これまで `javadoc [OPTIONS]` のように実行していたものを `showdoc.sh [OPTIONS]` とタイプすることで使用可能です。なお、スクリプトは、`{INST_DIR}` において使用することが前提となった内容です。インストール先に応じて、スクリプトの `-docletpath` の引数 `lib/sshowdlt.jar` の部分を `{INST_DIR}/lib/sshowdlt.jar` に変更してください。