

hspc

HSP3ソースファイルをコンソールでコンパイル、ビルド、診断を行えるCLIツールです。

充実した機能とオプション

基本機能のHSP3ソースファイルのコンパイル、自動実行ファイル作成、文字コードの自動識別、strmapファイルの出力に対応。診断機能でソースファイルの文法エラーをコンソールに常時表示することが可能です。hsphelp.idxファイルが作成されていれば、そのままコンソールでドキュメントを閲覧できます。

環境変数の対応

環境変数 HSP3_ROOT にHSP3開発環境ディレクトリパスを設定する事で、hspc.exe直下の環境以外を利用可能になりました。柔軟に複数のバージョンや環境を切り替えられるので、様々な環境でhspcが導入し易くなります。

注意事項（困ったら読むところ）

絶対パスと最後の区切り文字

hspcの使用するAPIとhspcmp.dllの都合上、一部引数には絶対パスが必要です。ディレクトリを指定する場合は**パスの最後に\文字が必要**です。HSP3_ROOT環境変数とhsp3rootオプションの引数には、**/文字を使わない**でください。

絶対パスを必要とするものは、下記の通りです。

- HSP3_ROOT並びに --hsp3root オプションの引数
- --compath オプションの引数
- --rtmdir オプションの引数
- --hsp3root オプションの引数

MAX_PATH

hspcの使用するAPIの都合上、取り扱えるファイルパスは260byteまでになります。

パスの文字コード

hspcは内部でShift_JISとして文字列を取り扱っています。なので、Shift_JISで表現できない文字がパスに含まれると、文字化けにより機能しません。

システムロケール

上記の仕様から、システムロケールはShift_JIS系のみ対応しています。Unicode UTF-8 環境では機能しません。

プロセス実行と管理者権限

hspc.exeが子プロセスを実行するとき、自身の権限を子プロセスに継承させます。そのため、管理者権限で起動したhspcは、デバッグ実行などで生まれる子プロセスにも管理者権限を付与することになります。その結果、デバッグ実行したウィンドウにファイルなどをドラッグアンドドロップできない等の症状が生じます。

Wine環境

一部WinAPIが正しく動作しない場合、hspcは例外を返しコマンドを終了します。

-mによるiconinsツール使用でリソースの挿入に失敗する場合、hsed3.exeから自動実行ファイル作成を行ってください。iconinsツールを使用しない場合、-Mを使用して自動実行ファイル作成を行えます。

また、`hspc lint --watch`で子ディレクトリの監視ができない場合があります。

PowerShell

一部CUI環境ではピリオド(.)で引数を区切られてしまい、意図しない結果になる場合があります。ダブルクォーテーションで囲むことで、意図した通りに引数を渡すことができます。(例:"-oout.ax")

HSP3_ROOT 環境変数

HSP3_ROOTの影響を受けたくない場合、HSPC_DISABLE_ENV_HSP3ROOT 環境変数に "true"文字列を設定する事でオミットできます。

VS Code

VS Code のターミナルから呼び出す場合、hspcは現在のターミナルに設定されているコードページに合わせるようにエンコードして標準出力します。そのため、-rオプションもしくは、-Rオプションによる子プロセスの標準出力コードページ次第では、その出力は文字化けになる場合がございます。

HSP3.7以降での自動実行ファイル作成

hspcmp.dllのバージョンが3.7以降の場合、hspc v3.5.x以降で実装されたmaker v3で、自動実行ファイル作成を行います。

maker v3の制約のため、hspc内部のカレントディレクトリは一時的にfileのあるディレクトリに移動します。

fileのあるディレクトリに作成された中間ファイル、packfile、exeファイルは、元のディレクトリへコピーされた後に削除されます。

インストール・アンインストール

hspcはCLIツールなので、解凍した先にパスを通すとコンソールで簡単にhspcコマンドを呼び出せるようになります。

hspc.exeを既存のHSP3開発環境ディレクトリに挿入することで、簡単にインストールできます。アンインストールはhspc.exeの削除で完了します。

配布zipには、最小限のHSP3開発環境を同梱しています。この環境を使用すると、解凍後すぐに使用できます。アンインストールも解凍したディレクトリを削除します。

HSP3_ROOT環境変数、--hsp3rootオプションにHSP3開発環境の絶対パスを設定することでも、その環境を使用することができます。一時的に環境を変更したいときに便利です。

補助ツールを使用した導入について

Readme.mdに記載された方法以外による導入時の問題や不具合につきましては、使用したツール作者様に対応していただきたく存じます。

コマンド

```
hspc <command> <options> <file> <arguments>
```

hspcは、コマンドで使用する機能を切り替えることができます。これにより、コンパイルだけでなく、継続的な診断や、ドキュメント閲覧ができます。

下記には、普段からよく使うコマンドレシピと、各コマンドの詳細な説明を記載しています。

よく使うコマンドレシピ集

デバッグ実行

HSPスクリプトエディタのF5に相当するコマンドです。

```
hspc -aCdwRoout app.hsp args
```

app.hspがコンパイルするファイルです。argsを付けることで、dir_cmdlineでその文字列を受け取ることができます。-Rを-rに変えることで、コマンドはデバッグ実行したスクリプトの終了を待たずに終了します。

自動実行ファイル作成

HSPスクリプトエディタのCtrl-F9に相当するコマンドです。

```
hspc -aPmC app.hsp
```

app.hspに対して、自動実行ファイルを行います。-mを-Mに変えることで、iconinsツールを使用しないよう制限する事ができます。

strmapファイル作成

```
hspc --strmap strmap app.hsp
```

カレントディレクトリにstrmapファイルを作成します。

この機能は、HSP3.6以降で使用できます。

診断（常時監視）

```
hspc lint --watch .\
```

カレントディレクトリのソースファイルを一通りコンパイルが通るか診断します。ファイルを保存するなど、変更を加えると自動で再診断します。診断結果は、標準出力へ書き出されます。

watchを終了するには、コンソールにCtrl-Cキーを送信してください。

compile

```
hspc <options> <file> <arguments>
rem or
hspc compile <options> <file> <arguments>
```

HSPソースファイルを中間ファイルへコンパイルします。オプションを組み合わせることで、普段使いの操作ができます。

中間ファイル、実行ファイル、strmapファイルの保存先に同名のファイルが存在する場合、上書き保存します。ファイルの上書き、もしくは削除に失敗したとき、コマンドは失敗して、その時点で終了します。

一文字オプションの後に正数を記入することで、そのオプションのレベルを指定する事ができます。（例：-D1m3）

レベルを省略した場合、レベル0が設定されます。

一文字オプション 文字	説明
-a	ソースファイルのコードページを推測して、UTF-8の可能性があるなら-iオプションを有効にします。
-C	カレントディレクトリをファイルのディレクトリに移動します。
-D	コマンド終了時に中間ファイルを削除します。 レベル2で使用済みpackfileを削除します。
-d	デバッグ情報を付与してソースファイルをコンパイルします。
-E	コンパイルエラーでもコマンドの終了コードは0を返すようにします。
-I	コンパイル中の詳細を標準出力します。
-i	ソースファイルのコードページをUTF-8として扱います。
-M	自動実行ファイル作成を行います。 レベル3で--makerV3が有効になります。

一文字オプション文字 説明

-m	自動実行ファイル作成とiconinsツールによるリソース挿入を行います。 レベル3で--makerV3が有効になります。
-o???	保存される中間ファイル名を???に設定します。
-P	packfileをカレントディレクトリに生成します。
-p	プリプロセス処理だけ行ってコマンドを終了します。
-R	ソースファイルを実行して、終了するまで待機します。
-r	ソースファイルを実行します。
-S???	strmapファイル作成して、???名で保存します。
-u	ソースファイルをShift_JISからUTF-8へエンコードしてコンパイルします。
-w	中間ファイルにデバッグウィンドウ表示フラグを立てます。
-v	hspcのバージョン情報を標準出力します。

複数文字オプション文字 説明

--compath ???	コモンディレクトリを???に設定します。
--curdir ???	カレントディレクトリを???に設定します。
--hsp3root ???	HSP3開発環境ディレクトリを???に設定します。
--iconins	自動実行ファイル作成時にiconinsツールによるリソース挿入を行います。
--licence	hspcのライセンス情報を標準出力します。
--makerV2	従来の方で自動実行ファイル作成を行います。
--makerV3	新しい方で自動実行ファイル作成を行います。
--nonextramacro	HSP3拡張マクロを無効化してコンパイルします。
--objname ???	保存する中間ファイル名を???に設定します。
--printsymbols	シンボル一覧を標準出力します。
--refname ???	コンパイルログのソースファイル名を???に置き換えます。
--rtmdir ???	ランタイムディレクトリを???に設定します。
--wait	-rで実行したスクリプトが終了するまで待機します。
--version	hspcのバージョン情報を標準出力します。

自動実行ファイル作成のレベルについて

v3.4.x以前では、HSP3.6未満のバージョンでなければ-mオプションは使用できませんでした。

v3.5.xから、HSP3.7betaに対応するために、新しい自動実行ファイル作成方法「maker v3」を導入しました。

maker v3は、hspcが読み込んだhspcmp.dllのバージョンが3.7以上であれば、自動判定により有効化されます。

maker v3を明確に使用するには、--makerV3オプションまたは、-mオプション、-Mオプションにレベル3を指定して使用します。

```
hspc -aPmC --makerV3 app.hsp
rem or
hspc -aPm3C app.hsp
```

--makerv2、--makerv3オプションを使用すると、自動判定より優先して設定されます。つまり、HSP3.7以降でも、--makerv2を使用することで従来の自動実行ファイル作成方法でEXEファイルを作成しようとします。

子プロセスについて

hspcが-Rオプションや-rオプションで子プロセスを作成するとき、子プロセスは親のhspcコンソールへのアクセス権を持ちます。

これにより、hspcを立ち上げたコンソールを使用して子プロセスと対話する事ができます。

hspcは、--Waitオプションや-Rオプションによる子プロセス待機中に自身のプロセスが内外の原因問わず終了した場合、子プロセスも終了させます。

lint

```
hspc lint <options> <files or directorys>
```

ソースファイルをコンパイルして構文エラーが無いか診断するコマンドです。

ファイルとディレクトリは複数指定できます。ファイルを指定した場合、--watchで直接監視することはできません。同じ層のディレクトリを指定してください。

--allow、--ignoreオプションは、MS-DOSワイルドカードを使用してファイルパスを許可、制限することができます。両方のワイルドカードに適合したファイルパスを監視、診断します。

オプション文字	説明
--addhspcopt ???	診断に使用するhspcオプションを追加します。
--allow ???	診断するファイルの拡張子を追加します。
--background	--watch、--onelineprintオプションを有効化、始動、終了テキストを出力します。
--hsp3root ???	使用するHSP3開発環境ディレクトリを???に設定します。
--ignore ???	無視するファイルパスを追加します。
--ignorednoninitvar	未初期化変数の情報を非表示にします。
--inserthspcopt ???	--addhspcoptオプションのエイリアスです。

オプション文字	説明
--iniv	--ignorednoninitvarのエイリアスです。
--lineprint	--onelineprintオプションのエイリアスです。
--nosubdir	子ディレクトリを監視しません。
--onelineprint	診断結果を一行で出力します。
--relativepath	ファイルパスを相対パスで出力します。
--querylimit ???	一度に処理するファイル数を???（整数）で指定します。
--savedb ???	内部データベースを???ファイル名として保存します。
--sethspcopt ???	診断に使用するhspcオプションを設定します。
--watch	ファイルの変更を監視します。

doc

```
hspc doc <options> <keyword>
```

hsphelp.idxからキーワードを検索して、検索結果を閲覧できます。

--keywordオプションは必ず引数を取ります。引数に閲覧したいキーワードを指定してください。（例：
`hspc doc --keyword "mes"`）

オプション文字	説明
--keyword	キーワードを検索して一致したドキュメントを出力します。
--hsp3root ???	使用するHSP3開発環境ディレクトリを???に設定します。

help

```
hspc help <keyword>
```

hspcのヘルプドキュメントを閲覧します。

`hspc help <command>`で、コマンドの概要を閲覧できます。（例：`hspc help compile`）

オプションの説明は、`hspc help compile <option>`で閲覧することができます。（例：`hspc help compile -a`）

`hspc help ls`でコマンドの一覧、`hspc help <command> ls`でコマンドのオプション一覧を閲覧できます。

ライセンス

hspc

- hspc.exe
- hspc146.exe
- hspc211.exe
- hspc334.exe
- hspc343.exe
- readme.md

Copyright 2017-2023 Honobono

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

OpenHSP

- hspcmp.dll
- hsprt
- hsp3.exe
- hsp3utf.exe
- hsp3cl.exe
- runtime/hsp3utf.hrt
- runtime/hsp3cl.hrt
- common/hspdef.as
- common/hsp3cl.as
- common/hsp3utf.as

Hot Soup Processor (HSP) Copyright (C) 1997-2021, Onion Software/onitama, all rights reserved. Made with base technology of OpenHSP.

OpenHSP Copyright (C) 1997-2021, Onion Software/onitama. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Onion Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ソースコード形式かバイナリ形式か、変更するかしないかを問わず、以下の条件を満たす場合に限り、再頒布および使用が許可されます。

- ・ソースコードを再頒布する場合、上記の著作権表示、本条件一覧、および下記免責条項を含めること。
- ・バイナリ形式で再頒布する場合、頒布物に付属のドキュメント等の資料に、上記の著作権表示、本条件一覧、および下記免責条項を含めること。
- ・書面による特別の許可なしに、本ソフトウェアから派生した製品の宣伝または販売促進に、Onion Softwareの名前またはコントリビューターの名前を使用してはならない。

本ソフトウェアは、著作権者およびコントリビューターによって「現状のまま」提供されており、明示黙示を問わず、商業的な使用可能性、および特定の目的に対する適合性に関する暗黙の保証も含め、またそれに限定されない、いかなる保証もありません。著作権者もコントリビューターも、事由のいかんを問わず、損害発生の原因いかんを問わず、かつ責任の根拠が契約であるか厳格責任であるか（過失その他の）不法行為であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、本ソフトウェアの使用によって発生した（代替品または代用サービスの調達、使用の喪失、データの喪失、利益の喪失、業務の中断も含め、またそれに限定されない）直接損害、間接損害、偶発的な損害、特別損害、懲罰的損害、または結果損害について、一切責任を負わないものとします。

ESLint

- ・ hspcのlintコマンドで出力する形式を決定するのに参考しました。

Copyright OpenJS Foundation and other contributors, <www.openjsf.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SQLeLe And SQLite

- sqlite3.dll
- common/sqlele.hsp

SQLeLe (モジュール本体、サンプル) は著作権フリーです。

SQLite はパブリックドメインとして公開されていて、自由に使うことができます。

SQLeLe の開発元 http://spn.php.xdomain.jp/hsp_sqlele.htm

SQLite の開発元 <http://www.sqlite.org/>