

Appendix I *Em* Command Line for Batch

処理の負荷が軽い時に、バッチ、或いはスクリプトファイルを設定して、一晩中や一日のある時間に解析を実行したい場合には、コマンドラインを使ってバッチファイルから *em* を実行することが可能です。解析モニターを使って、開始及び終了時間を指定し、バッチファイルを設定することも可能であることを覚えておいてください。この実行方法については、オンラインヘルプの **How do I create a batch file to run multiple analysis jobs?** をご覧ください。

コマンドラインから実行すると、実行中に解析モニターに表示される、解析の進行状況を示すすべてのステータスは表示されないことを覚えておいてください。

コマンドラインのシンタックスは、以下の通りです。

em -[options] <project name> [external frequency file]

Sonnet User's Guide

ここで

下の表の<options>は1つ以上のRunオプションです。複数のオプションを使用する場合は、マイナス符号の後に、間にスペースを入れずにタイプしてください。お使いのプロジェクトでは、Analysis Setup ダイアログボックスの中で他のRunオプションを設定することができ、解析中に使用できることにご注意ください。

Option	Meaning
-Dlicense	<i>Em</i> のライセンスの問題をデバッグするのに使用。ライセンスに関係のあるすべての環境の情報を表示。
-N	サブセクション数と推定必要メモリ量を表示。 <i>Em</i> はその後、解析を実行せずに終了する。
-fullsolve	Sonnet Lite、LitePlus、Level2、Level3 の使用時に、正確なメモリ量の見積りを算出するために上の -N オプションと共に使用。
-test	<i>Em</i> をテスト用の回路で実行。 <i>Em</i> がライセンスを取得し、うまく実行されるかを調べるのに使用。
-v	解析の実行時に、解析の情報を表示。解析の情報がコマンドプロンプトに、またはパッチが行われた端末に出力される。
-AbsCacheNone	ABS のキャッシング機能を使用不可にする（プロジェクトファイルの設定に優先する）
-AbsCacheStopRestart	ABS のキャッシング機能を停止、再スタートする（プロジェクトファイルの設定に優先する）
-AbsCacheMultiSweep	ABS のマルチスイープを可能にし、キャッシング機能を停止、再スタートする（プロジェクトファイルに優先する）
-AbsNoDiscrete	以前に存在したキャッシュデータで ABS を実行する時に使用。解析エンジンに個々の周波数は解析しないように指示する。以前のキャッシュデータが ABS の解法を収束させるのに十分であれば、出力に書かれる。そうでなければ何の処理も行われない。

Appendix I Em Command Line for Batch

Option	Meaning
-SubFreqHz[value]	ここで [value] はサブセクションする周波数 (Hz)。値の欄の前にはスペースがないことに注意。 このオプションで、サブセクションする周波数をコメント行に指定することができ、プロジェクトファイルの設定に優先する。
-ParamFile <filename>	ここで <filename> はユーザが解析される回路でパラメータに使いたい値を含んだファイルの名前。この値は解析のシミュレーションプロジェクトに含まれる値に優先するが、シミュレーションプロジェクトの内容は変えない。パラメータファイルのシンタックスは <parname>=fnum。ここで <parname> はパラメータ名で、fnum は解析のパラメータの値を定義する不動小数点の数値。

<project name> は解析しようとするプロジェクトの名前です。拡張子がない場合は、拡張子 “.son” がついていると想定されます。このフィールドは必要です。

[external frequency file] は拡張子が “.eff” であるオプションの外部の周波数コントロールファイル名です。この拡張子はコントロールファイルを指定する時に含まれてはなりません。Project editor の中で外部の周波数コントロールファイルを作成できます。詳細は、project editor のオンラインヘルプの Frequency Sweep Combinations をご覧ください。このファイルの周波数は、プロジェクトファイルの周波数を無効にします。

例えば、-v オプションを使って、パッチファイルのプロジェクト steps.son を解析したい時には、コメントラインは次のようになります。

em -v steps.son

以下は複数の解析を実行するパッチファイルの例です。

```
em -v steps.son
em -v filter.son filter.eff
em -v airbridge.son filter.eff
em -v airbridge.son
```

PC上でバッチファイルを実行するには、“bat” 拡張子のついたコマンドラインを含むテキストファイルを作成します。次に DOS プロンプトウィンドウをオープンし、プロンプトにファイル名をタイプして、リターンキーを押します。

UNIX 上でバッチファイルを実行するには、コマンドラインを含んだテキストファイルを作成します。そのファイル名には、拡張子は必要ありません。次にファイルのパーミッションを変えて、これを実行します。例えば、

```
chmod a+x <filename>
```

ここで <filename> は、実行したいバッチファイルの名前です。次に UNIX のプロンプトにファイルの名前をタイプして、リターンキーを押します。

UNIX システムには、知っておくと便利な、コマンドラインへのいくつかの追加があります。コマンドの前に “nice” をつけると、より低い順位で実行されます。コマンドの後ろに “&” をつけると、バックグラウンドで実行され、カーソルは戻ります。コマンドラインの後に “nohup” をつけると、em のジョブを実行させながら動作することができます。“&” または “nohup” を使っている時に、“> outfile” を使って出力を redirect することを考えたくなるかもしれません。これらのオプションについての詳細は、御社のシステムアドミニストレータにお尋ねください。