

KSSコメント規約
(C 言 語)
イージードキュメントシステム用

第1版 2004/09/05

第2版 2008/04/07



| | |
|---|----|
| 1 はじめに | 3 |
| 4.2 1.1 説明略号の表記または用語説明..... | 3 |
| 2.概要 | 4 |
| 3.基本記述規約 | 5 |
| 4.コメント記述規約 | 8 |
| 4.1 Aタイプのコメント記述説明..... | 9 |
| 4.2 Bタイプのコメント記述説明..... | 12 |
| 5.モジュール仕様書例 | 14 |
| 5.1 Aタイプ記述例（1行コメントタイプ）..... | 14 |
| 5.2 Bタイプ記述例（複数行コメントタイプ）..... | 16 |
| 6.「コメント規約ファイル」作成方法 | 18 |
| 4.3 5.1 ユーザー指定タイプ【CDS98.keyの場合】..... | 18 |
| 6.1 CDS98.keyソース記述例..... | 19 |
| 7 おわりに | 20 |

1.はじめに

本書は、「イージードキュメントシステム」(以下、「EasyDocSys」という。)で完成度の高いドキュメントを作成するためのC言語コメント規約について説明します。

この規約は、株式会社セイケン・ソフトサービス(以下、「KSS」という。)が独自に創作した規約

で、必ず、このように記述しなければならないということではありませんし、強制するものでもありません。只、この規約に則ってコメントを付与すると「イージードキュメントシステム」で非常に完成度の高い各種ドキュメントを作成することができます。

また、特にコメント規約を決めていないでプログラム作成されている方や開発プロジェクトで、これから規約を考えて運用しようとしている方で、「EasyDocSys」を使ってドキュメント作成して運用したいという場合、この規約をそのまま使って頂ければと思います。

本書の配布は自由ですが、一部コピーや変更しての配布、掲載、販売は堅くお断りいたしますので、ご注意願います。雑誌やホームページ等への掲載は、必ず、当社へご連絡下さい。

1.1 説明略号の表記または用語説明

| | |
|------------|--|
| KSS | 株式会社セイケン・ソフトサービスの略称 |
| EasyDocSys | イージードキュメントシステム |
| ファイルヘッダ | ファイルの先頭に記述するファイル全体のコメント |
| モジュールヘッダ | モジュール(関数)の先頭に記述するモジュールに関するコメント |
| CPチャート | 従来から存在しているHCPチャートを「EasyDocSys」用のCPチャートにしたもので、一部の記号をパソコンで使える記号に変更したもの |

2.概要

KSS コメント規約（以下、「本規約」という。）は、プログラマがC言語でプログラム作成していく場合に、ファイルヘッダやモジュールヘッダを規約に則って付与するためのものです。

本規約は、特に、EasyDocSys で完成度の高いドキュメントを作成するために必要な規約で基本記述規約とコメント記述規約の構成になっています。

また、EasyDocSys では、既存のC言語ソースプログラムにファイルヘッダ、モジュールヘッダの雛形を簡単に挿入する機能も備えていますので、それ程手間をかけなくても本規約に対応したコメントを簡単に挿入することができます。

以下に、コメントが反映されるドキュメントについて述べます。

本規約でコーディングした場合

| 反映先ドキュメント | 反映元の内容 |
|-------------|-------------|
| ファイル一覧 | ファイルヘッダの内容 |
| モジュール(関数)一覧 | モジュールヘッダの内容 |
| モジュール仕様書 | モジュールヘッダの内容 |
| C Pチャート | 各処理(行)のコメント |

既に、独自のコメント規約がある場合

項番5の「コメント規約ファイル作成方法」に詳細が記載されていますが、コメント規約ファイルを作成して、それを EasyDocSys へ認識させることにより、独自のコメント規約を、ある程度対応させることができます。(100%互換とはならない場合もあります。)

コメント規約が全くない場合(コメントを付けていない)

そのままでは、作成するドキュメントにコメントを反映させることができません。

しかし、ドキュメント雛形を自動作成しますので、その後、手作業による修正(エディット)を行って下さい。それでも、新規に手作業で作成するより数十倍の工数が削減できるでしょう。

ファイルヘッダとモジュールヘッダをソースファイルへ挿入したい場合

コメント規約がなく、ファイルヘッダやモジュールヘッダを記述していないプログラムに対し EasyDocSys では、あらかじめ2種類(Aタイプ・Bタイプ)の雛形を用意していますので、簡単にプログラムに挿入する機能があります。

また、全ソースファイルに一括して挿入する機能もありますので、比較的簡単に本規約に移行することができます。

3.基本記述規約

コメント規約のBタイプを例に説明します。

```
/*
--file
[ファイル名]
sample_check.c
[概要]
サンプル用チェックプログラム
[履歴/作成者]
01 2004-07-03 新規作成 R.Sakamoto
[備考]
--end
*****/

/* 標準インクルード */
#include <string.h>
/* ユーザーインクルード */
#include "host.h"
/* 外部関数・変数定義 */
extern sub1();
extern int G_flag;
/* DEFINE 定義 */
#define NORMAL 0
#define ERROR 1
*****/

--function
[関数名]
int sample_check(int flag,char cp)
[概要]
サンプル用入力チェック
[引数]
int flag 入力フラグ
char *cp 入力文字列ポインタ
[戻り値]
正常時 NORMAL
エラー時 ERROR
[ヘッダファイル]
host.h
[解説]
パラメータで指定したフラグをチェックして、その結果を返す。
[関連項目]
[使用例]
[履歴/作成者]
01 2004-07-03 新規作成 R.Sakamoto
02 2004-07-11 不具合修正 S.Ohara
[備考]
--end
*****/
```

< 次ページへ続く >

< 前ページ続き >

```
int sample_check(int flag,char cp)
{
    int    i,status;
    char   *w_cp;
    char   w_text[128];

    /* エリア初期化 */
    for(i=0;i<128;i++) {
        w_text[i]=0;
    }
    /* エリア設定 */
    if ( flag == 1 ) {
        strcpy(w_text,cp);
    } else if ( flag == 2 ) {
        strcpy(w_text, " **NONE** ");
    } else {
        strcpy(w_text, " **ERROR** ");
    }
    /* 戻り値設定 */
    switch( flag ) {
    case 1:  status=NORMAL;
            break;
    default: status=ERROR;
            break;
    }
    /* エリアチェック */
    w_cp=cp;
    while( *w_cp ) {
        if ( *w_cp == # ) {           /*02*/
            status=ERROR;
            break;
        }
        w_cp++;
    }
    return(status);
}
```

基本的な記述規則を以下に示します。

- (1) 1 ファイルに 1 つのファイルヘッダと 1 つ以上のモジュールヘッダとモジュール（関数）を記述します。
ヘッダファイルの場合は、ファイルヘッダのみ記述します。（モジュールがない場合）詳細は項番 4 のコメント記述規約を参照して下さい。
- (2) ファイルヘッダの後に標準インクルード、ユーザーインクルード、外部関数・変数定義、DEFINE 定義等を記述します。
- (3) インデントは、スペース 2 個または 4 個で記述し、1 行が 80 バイト以内で収まるようにする。（コメントも含めて）印刷時に、見やすいように工夫して下さい。
タブの基本は 8 スペースで、階層が深くなるとすぐに 80 バイトを超えてしまい、プログラムが見づらくなるので、できれば避けて下さい。但し、既にコーディング済みの物は、あえて修正する必要はありません。
- (4) if 文,switch 文,while 文の条件は、前後にスペースを 1 ヶ入れます。
- (5) 代入文の“=”の前後は、スペースを入れないようにします。
- (6) 修正した場合は、修正した場所がわかるようにコメントにバージョン番号や日付など付けると良い。例：/*02*/ , /*2004.7.3*/ 等
- (7) if 文,for 文,while 文には、“{ }” を付けて記述して下さい。
（「EasyDocSys」での解析を容易にするためにも、極力付けて下さい。）

< 制限事項 >

- (1) if 文などの階層は、最大 32 階層として下さい。
- (2) コメント A タイプでは、コメントを含めて 1 行を 128 バイト以内で記述して下さい。
- (3) ファイル名、モジュール名は 31 文字以内で記述して下さい。
- (4) ファイル名、モジュール名の付け方には、特に制限はありません。
- (5) 変数名についても特に制限はありません。

上記の規約は、あくまでも「EasyDocSys」での解析を助けるための規約であり、C コンパイラの制限ではありません。C 言語は、本来、フリーフォーマットですから C コンパイラの規約に則っていれば、自由に記述できることが良い点です。しかし、「EasyDocSys」にコンパイラ並の解析機能を持たせるのは困難なため、このような規約でカバーしています。

4.コメント記述規約

EasyDocSys で推奨するコメント記述には、**Aタイプ**/**Bタイプ**の2種類あります。

| | |
|-------------|------------------------|
| Aタイプ | コメントが1行で記述できるタイプ |
| Bタイプ | コメントが複数行で記述できるタイプ (推奨) |

コメント記述は、以下の項目について記述します。

(1)ファイルヘッダ (ファイル概要説明)

ソースファイル先頭に、そのファイル全体の概要について記述します。
このコメントは、ファイル一覧に反映されます。

(2)モジュールヘッダ (モジュール(関数)概要説明)

モジュール(関数)ごとの概要や入出力パラメータ、リターン値等について記述します。
このコメントは、関数一覧とモジュール仕様書に反映されます。

(3)行単位処理説明

1行単位の処理についてコメント記述をおこないます。

例：

```
/* 入力パラメータチェック */  
    if ( in_data == 0 ) in_data=1;  
    if ( in_data == 1 ) in_data=0;          /* データ反転 */
```


“ /* */ ” や “ //...” で記述したコメントのことをいいます。

このコメントは、CPチャートに反映されます。

以下に、タイプ別に例を示します。

4.1 Aタイプのコメント記述説明

コメント記述のAタイプは、コメントを1行で記述するタイプです。
簡単なコメントを付ける場合に適しています。

(1)ファイルヘッダ (ファイル概要説明)

```
/*
 * ソースファイル : source-file.c
 * ファイル概要   :
 * 作成者         : l.name
 * 更新日(作成日) : 2000.00.00(2000.00.00)
 * 備考           :
 * HISTORY        :
 * ID  -- DATE -- ----- NOTE -----
 * 00  2000.00.00 Created
 *****/
キーワード
```

C言語のソースファイル全体に関する情報を記述します。原則的にソースファイルの先頭に記述して下さい。「ファイル概要:」等が、ファイル一覧へ反映されます。

Aタイプでは、【キーワード】が非常に重要で、行の先頭からコロンの1つ前までを、【キーワード】と認識して、【キーワード】が一致すればその行はコメント記述とみなしドキュメント反映します。従って、【キーワード】を誤った場合、正しくコメント反映されませんので注意願います。また、モジュールヘッダも同様なので、ファイルヘッダと重複する【キーワード】は使えません。

【ファイルヘッダ規約】

- ・上記のフォームで、"/*"で始まり、"*/"で終わります。
- ・キーワードはスペース・タブも含めて、必ず、統一して下さい。
- ・キーワードに一致しないコメントは、無視されます。
- ・キーワードとコメントの間には、必ず、半角のコロン(:)を入れて下さい。
- ・各項目は、半角で80文字以内にして下さい。(キーワードを除く)
「備考」や「HISTORY」など全てのキーワードがなくても、問題ありません。
不要なものは削除して下さい。

(2)モジュールヘッダ (モジュール【関数】概要説明)

```

/*****
* 関数名          :
* 機能概要       :
* 解説           :
*                :
* 引数(入力)     : p1:
* 引数(出力)     : p2:
* 引数(入出力)   : p3:
* 戻り値         :
* 作成者         :
* 更新日(作成日) :
* 備考           :
/*****
  キーワード

```

各モジュール【関数】に関する情報を記述します。原則的にモジュールの先頭に記述して下さい。
「機能概要：」以下、記述されている項目を、モジュール一覧やモジュール仕様書へ反映されます。

ファイルヘッダと重複する【キーワード】は使えません。

【モジュールヘッダ規約】

- ・上記のフォームで、"/*"で始まり、"*/"で終わります。
 - ・キーワードはスペース・タブも含めて、必ず、統一して下さい。
 - ・キーワードに一致しないコメントは、無視されます。
 - ・キーワードとコメントの間には、必ず、半角のコロン(:)を入れて下さい。
 - ・各項目は、半角で80文字以内にして下さい。(キーワードを除く)
- 但し、解説は256文字以内で複数行に渡って記述できます。
「備考」など全てのキーワードがなくても、問題ありません。
不要なものは削除して下さい。

(3)行単位処理説明

各処理に関する情報を記述します。

C Pチャートに処理説明を反映させたい時、以下の規約に従って記述するとコメントが反映されます。

処理単位の説明

処理単位でコメントを付けたい場合に使用します。

コメント記述 (`/* <...> */`) のなかに「<」、>」で囲まれた記述を処理単位のコメントと見なします。「<」、>」の範囲外は対象外となりますので、適当に見やすいように文字を入れてもかまいません。

例：

```
/* <処理単位のコメント> */
/* ---<処理単位のコメント>--- */
// <処理単位のコメント>
// ---<処理単位のコメント>---
```

処理の詳細説明

各処理ごとに詳細説明を入れたい場合に使用します。

コメント記述 (`/* */`) で記述されたものを詳細説明と見なします。

また、コメント記述のみの場合も詳細説明の対象となりますので、複数行にわたるコメントの場合は、以下のように記述して下さい。

<命令と詳細説明の場合>

例：

```
printf(...); /* 詳細説明 */
printf(...); // 詳細説明
```

<命令と詳細説明が複数行になる場合>

例：

```
printf(...);
/* 詳細説明が長い場合の */
/* 記述例です。          */

printf(...);
// 詳細説明が長い場合の
// 記述例です。
```

4.2 Bタイプのコメント記述説明

コメント記述のBタイプは、複数行のコメントを記述するタイプです。

詳細なコメントを付ける場合に適しています。

EasyDocSys では、こちらを推奨しています。

(1) ファイルヘッダ (ファイル概要説明)

```

/*****
--file
[ファイル名]
XXXX.C
[概要]
(ここにファイルの概要を記述して下さい)
[履歴 / 作成者]
00 2001-mm-dd 作成          X.Name
[備考]

--end
*****/

```

ソースファイルの全体に関する情報を記述します。原則的にファイルの先頭に記述して下さい。

(必ず先頭でなくても問題はありません。)

Bタイプでは、ファイルヘッダの場合【キーワード】がモジュールヘッダを同じでも、コメント内で区別できるように “ -file ” と “ -end ” を必ず、記述します。

従って、“ -file ” と “ -end ” が無い場合、キーワードが一致してもコメント反映されませんので、ご注意願います。

【ファイルヘッダ規約】

- ・上記のフォームで、“/”で始まり、“*/”で終わります。
- ・コメント内で、“--file”から、“--end”までを対象範囲としますので絶対に削除しないで下さい。
- ・キーワードは、行の先頭から[]で囲まれた以下の項目です。

[ファイル名]

[概要]

[履歴 / 作成者]

[備考]

- ・キーワードに一致しないコメントは、無視されます。
- ・各項目は、次のキーワードまで、複数行に渡って記述できます。
全てのキーワードがなくても、問題ありません。不要なものは削除して下さい。

(2) モジュールヘッダ (モジュール(関数)概要説明)

```

/*****
--function
[関数名]
func_name
[概要]
(ここにファイルの概要を記述して下さい)
[引数]
(例：int i          カウンタ)
(例：char *p       ポインタ)
[戻り値]
正常時          0
エラー時       エラーコード
[ヘッダファイル]

[解説]
(ここに解説を記述して下さい)
[関連項目]
[使用例]
[履歴 / 作成者]
2001-mm-dd 作成          X.Name
[備考]

--end
*****/

```

各モジュール【関数】に関する情報を記述します。原則的にモジュールの先頭に記述して下さい。
 [関数名]以下、記述されている項目を、モジュール一覧やモジュール仕様書へ反映されます。

Bタイプでは、モジュールヘッダの場合【キーワード】がモジュールヘッダを同じでも、コメント内で区別できるように“-function”と“-end”を必ず、記述します。
 従って、“-function”と“-end”がない場合、キーワードが一致してもコメント反映されませんので、ご注意願います。

【モジュールヘッダ規約】

- ・上記のフォームで、“/*”で始まり、“*/”で終わります。
- ・コメント内で、“--function”から、“--end”までを対象範囲としますので、絶対に削除しないで下さい。
- ・キーワードは、行の先頭から[]で囲まれた以下の項目です。

[関数名]、[概要]、[引数]、[戻り値]、[ヘッダファイル]、[解説]、[関連項目]
 [使用例]、[履歴 / 作成者]、[備考]

各項目は、次のキーワードまで、複数行に渡って記述できます。

全てのキーワードがなくても、問題ありません。不要なものは削除して下さい。

(3) 行単位処理説明

Aタイプと同じです。

5.モジュール仕様書例

Aタイプ / Bタイプのコーディング例と出力されるモジュール仕様書について説明します。

5.1 Aタイプ記述例 (1行コメントタイプ)

(1) Cソースファイル記述例

```
/*
 * ソースファイル : test.c
 * ファイル概要   : 概要記述
 * 作成者         : 山田 花子
 * 更新日(作成日) : 2003.07.19(2003.07.19)
 * 備考           : 備考を記述
 * HISTORY       :
 * ID  -- DATE --  ---- NOTE -----
 * 00  2000.00.00 Created
 */
#include <stdio.h>
#define NORMAL 0
#define ERROR -1

/*
 * 関数名       : Test 関数
 * 機能概要     : テスト
 * 解説         : x x xを行う。
 *              : 2行目コメント
 * 引数(入力)   : cnt       : 回数
 * 引数(入力)   : buff      : データバッファアドレス
 * 引数(出力)   : add       : 結果格納アドレス
 * 戻り値       : 0         : 正常終了
 *              : 0以外     : 異常終了
 * 作成者       : 山田 花子
 * 更新日(作成日) : 2003.07.19(2003.07.19)
 * 備考         : テスト用備考
 */
int test( int cnt, char* buff, int* add)
{
    return( 0 );
}
```

(2) 出力モジュール仕様書

| | | | |
|--|--------------|-------------------------|--|
| test | < test_a.c > | | |
| <p>テスト</p> <pre>int test(int cnt,char* buff,int* add);</pre> | | | |
| 関数 | 必要なヘッダー | 互換性 | |
| test | なし | ANSI | |
| 戻り値 | | | |
| int 0 | : 正常終了 | | |
| 0 以外 | : 異常終了 | | |
| 引数 | | | |
| int cnt | 回数 ; | | |
| char* buff | データバッファアドレス; | | |
| int* add | 結果格納アドレス ; | | |
| 解説 | | | |
| <p>×××を行う。 2行目コメント</p> | | | |
| プログラム例 | | | |
| (例) | int | i = test(cnt,buff,add); | |
| 使用変数一覧 | | | |
| 関連モジュール | | | |
| 使用関数一覧 | | | |
| (test)テスト | | | |
| <pre>return(0);</pre> | | | |

5.2 Bタイプ記述例（複数行コメントタイプ）

(1) Cソースファイル記述

```
/*
--file
[ファイル名]
test_b.c
[概要]
Bタイプの概要記述
[履歴 / 作成者]
00 2003-07-19 作成          山田 花子
[備考]
備考機銃Bタイプの例です。
--end
*/

#include <stdio.h>

/*
--function
[関数名]
Test 関数
[概要]
Test 関数機能概要
複数行記述できます。
[引数]
int    cnt    回数
char*  buff   データバッファアドレス
int*   add    結果格納アドレス
[戻り値]
0      正常終了
0以外 異常終了
[ヘッダファイル]

[解説]
関数の詳細説明。
x x xを行う。
複数行記述できます。
[関連項目]
[使用例]
    test();
[履歴 / 作成者]
2003-07-19 作成          山田花子
[備考]

--end
*/

int    test( int cnt, char* buff, int* add)
{
    return( 0 );
}
```

(2) 出力モジュール仕様書

| | |
|--|--------------|
| test | < test_b.c > |
| 【形式】 int test(int cnt,char* buff,int* add); | |
| 【概要】 Test 関数機能概要 複数行記述できます。 | |
| 【引数】 int cnt 回数 char* buff データバッファアドレス int* add 結果格納アドレス | |
| 【戻り値】 0 正常終了 0以外 異常終了 | |
| 【解説】 関数の詳細説明。 x x xを行う。 複数行記述できます。 | |
| 【使用例】 test(); | |
| 【関連項目】 | |
| 【履歴 / 作成者】 2003-07-19 作成 山田花子 | |
| 【備考】 (test)Test 関数機能概要 複数行記述できます。 return(0); | |

6. 「コメント規約ファイル」作成方法

コメント規約ファイルは、コメント行のキーワードを登録してあるテキストファイルです。

EasyDocSys で標準装備されているコメント規約（Aタイプ/Bタイプ）では、コメント規約ファイルは必要ありません。[設定]-[コメントタイプ]から選択して下さい。

コメント規約ファイルは、ユーザーが独自のコメント規約でコーディングしている場合、その規約を EasyDocSys に認識させる為に使用します。

ユーザー指定タイプ：ユーザーが独自に決めたコメント規約タイプ。

例：CDS98.key

CDS98 は、Universal Computer Laboratory co., Ltd.の製品です。

ここでは、CSD98 を例にしていますが CDS98.key に修正を加えキーワードを変えることにより、他のコメント規約に対応した例を示します。

4.2 5.1 ユーザー指定タイプ【CDS98.key の場合】

```

;ファイル概要説明
S_SOURCE      : * MODULE
S_SOCFUNC     : * FUNCTION
S_PROGRAMER   : * PROGRAMMED
S_DATE        : * DATE(ORG)
S_REMARKS     : * REMARKS
;関数概要説明
F_FUNCNAME    : * NAME
F_FUNCTION    : * FUNCTION
F_PROCESS     : * PROCESS
F_INPUT       : * INPUT
F_OUTPUT      : * OUTPUT
F_INOUT       : * UPDATE
F_RETURN      : * RETURN
F_PROGRAMER   : * PROGRAMMED
F_DATE        : * DATE(ORG)
F_REMARKS     : * REMARKS
;処理説明
S_STARTKEY    : -<
S_ENDKEY      : >-
-----
                      キーワード

```

コロン(:)の直後にコーディングされているキーワード記述を定義します。

カスタマイズ(変更)する場合、コロンの直後(スペース含む)から記述して下さい。

最後にスペースを入れると、それもキーワードと認識しますので、余分なスペースやタブは入れないで下さい。

コロンの前 (S_xxx,F_xxx) は、EasyDocSys 内部で使用しているキーワードなので、絶対に修正しないで下さい。誤って修正した場合は、正しく動作しませんので元に戻して下さい。

CDS98.key は、セットアップ後のインストールフォルダにありますので、ソースのフォルダや任意のフォルダへコピーして、必要であればエディタにて修正してご利用下さい。

6.1 CDS98.key ソース記述例

(1) ファイルヘッダ

```

/*****
* MODULE      abc.c
* FUNCTION    テスト
* PROGRAMMED  山田  太郎
* DATE(ORG)   2003.11.18
* REMARKS     <Ver.1.02>
*****/
  キーワード

```

(2) モジュールヘッダ

```

/*****
* NAME        abc
* FUNCTION    テスト関数
* PROCESS     テスト結果を判定して結果を返す
* INPUT       data1
* OUTPUT      result
* UPDATE     2003.11.20
* RETURN     判定結果 ( 0 : 正常、 1 : 異常 )
* PROGRAMMED  山田  太郎
* DATE(ORG)   2003.11.18
* REMARKS     <不具合修正>
*****/
  キーワード

```

キーワードは、先頭のスペースを含めてコメント規約ファイルで指定したキーワードと一致する行を、コメント反映情報として認識します。

キーワードが1文字でも異なる場合は、コメントを認識しませんので注意願います。

コメントが正常に反映されない場合は、コメント規約ファイルのキーワードとソースファイルのキーワードが一致しているか確かめて下さい。

大文字と小文字は、別文字と認識しますので、それも確認して下さい。

コメント規約ファイルの作成方法について不明な点がある場合は、当社ホームページよりメールにてご連絡下さい。

7.おわりに

KSS コメント規約は、当社がこれまで行ってきた現場でのノウハウとなるべくシンプルで分かり易いプログラム記述を目指すとともに、既に存在するコメント規約にたいしてもできるだけサポートできるように考えました。

統一したコメント規約を作成することは、様々な考え方や個人的な趣味、既存の規約との互換性などがあり、非常に難しいものがあると思います。

C 言語の日本コーディング基準や世界コーディング基準というものがあれば、それを基準にできるのですが、特に決まったものがないようにおもわれますので、当社の独自の基準をベースに考えてみました。

この規約が皆様のコメント規約の参考になれば幸いです。

開発チーム一同

販売元：〒144-0032 東京都大田区北糞谷 1 - 6 - 7
株式会社セイケン・ソフトサービス
<http://www.seiken-soft.com/>