

CasePlayer2 仕様書出力サンプル【仕様書生成モード】

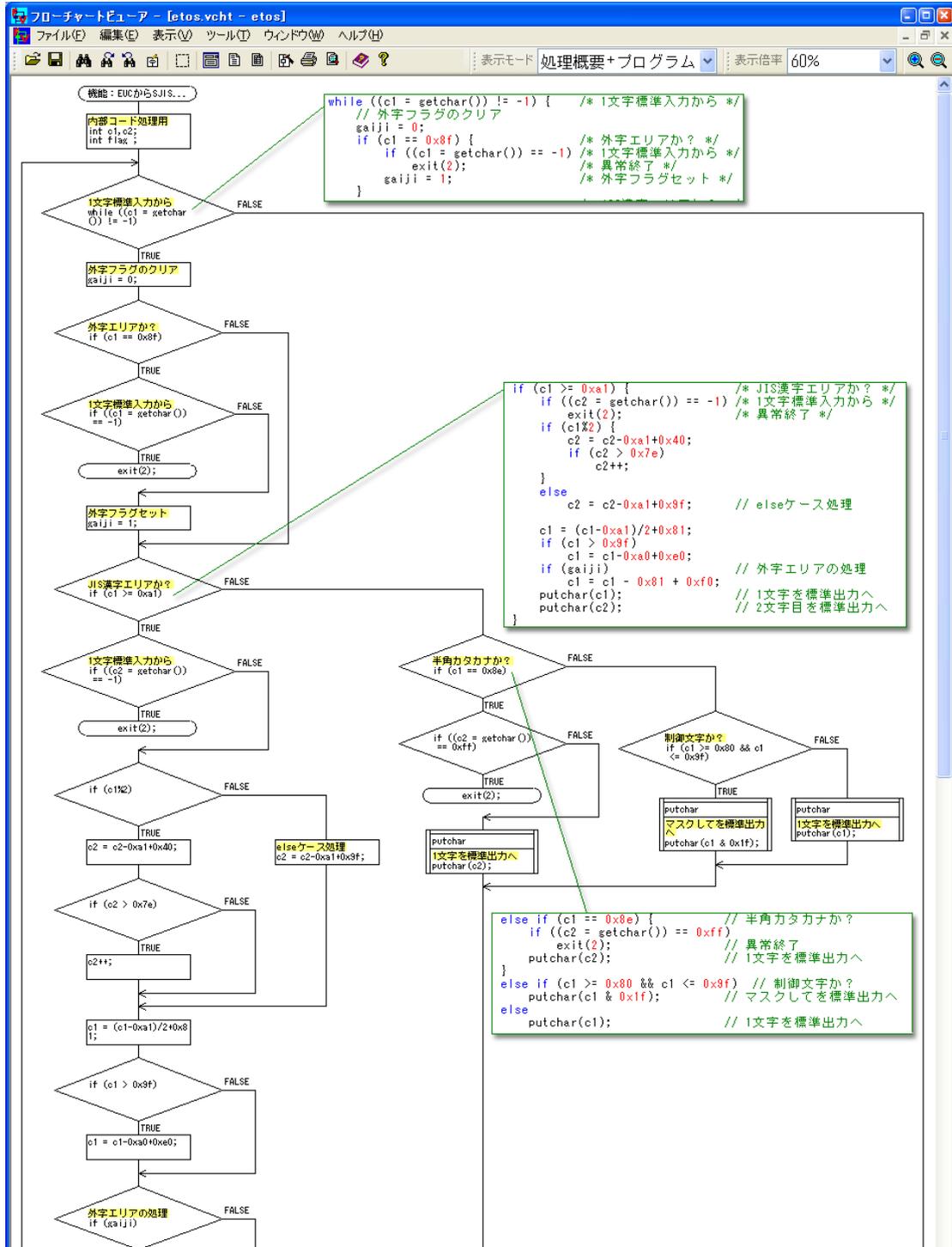
CasePlayer2 の仕様書のうち、本説明書掲載の 6 つは、ソースファイルを登録するだけで簡単に作成できます。ソースが完成している必要はなく、ヘッダファイルが無い場合や開発途中のソースコードの場合で、コンパイルエラー、リンクエラーを発生するようなソースでも対応可能です。ソースの if 文や switch 文、for 文、while 文などのソースの構造を解析し、チャート化することが可能です。

```

3 //各種定義
4 typedef unsigned char uchar8; // 符号無し8ビット
5 typedef signed char char8; // 符号有り8ビット
6 typedef unsigned short uint16; // 符号無し16ビット
7 typedef signed short int16; // 符号有り16ビット
8
9
10 #define NULL 0
11 #define TRUE 1
12 #define FALSE 0
13
14 #define TIMER_ID_RESTART 1 // RESTART状態
15 #define TIMER_ID_AUTO_UPDOWN 2 // 自動UP/DOWN
16 #define TIMER_ID_AUTO_ONOFF 3 // 自動ON/OFF
17 #define TIMER_ID_STARTUP 4 // STARTUP状態
18
19 /* 漢字コードテーブル */
20 struct code_table {
21     char *code_name; // 漢字コード表記 */
22     int code_id; // 漢字コードの識別番号 */
23 };
24
25 /* 外部変数 */
26 int chosenStaff=0; // 選択番号
27 int sortorder=0; // ソート順
28 int sortItemColumn=4; // ソートアイテムカラム
29 int isMakeThreadRunning = FALSE; // スレッド実行中フラグ
30 int isDispMsg = FALSE; // メッセージ表示中
31 int saveCpuPower = FALSE; // GPUセーブモード
32 int saveCpuPowerValue = 10; // GPUセーブ率
33 int saiji; // 外字フラグ
34
35 /*
36 機能：EUCからSJISに変換
37 説明：CasePlayer2仕様書作成サンプルソース
38 Copyright(c) GAIO TECHNOLOGY
39 */
40 int etos(
41     int arg_a, /* 第1引数：パラメータA */
42     int arg_b) /* 第2引数：パラメータB */
43 {
44     int c1,c2; // 内部コード処理用
45     int flag; // フラグ
46
47     while ((c1 = getchar()) != -1) { // 1文字標準入力から */
48         // 外字フラグのクリア
49         saiji = 0;
50         if (c1 == 0x8f) { // 外字エリアか? */
51             if ((c1 = getchar()) == -1) /* 1文字標準入力から */
52                 exit(2); /* 異常終了 */
53             saiji = 1; /* 外字フラグセット */
54         }
55         if (c1 >= 0xa1) { // JIS漢字エリアか? */
56             if ((c2 = getchar()) == -1) /* 1文字標準入力から */
57                 exit(2); /* 異常終了 */
58             if (c1%2) {
59                 c2 = c2-0xa1+0x40;
60                 if (c2 > 0x7e)
61                     c2++;
62             }
63             else
64                 c2 = c2-0xa1+0x9f; // elseケース処理
65
66             c1 = (c1-0xa1)/2+0x81;
67             if (c1 > 0x9f)
68                 c1 = c1-0xa0+0xe0;
69             if (saiji) // 外字エリアの処理
70                 c1 = c1 - 0x81 + 0xf0;
71             putchar(c1); // 1文字を標準出力へ
72             putchar(c2); // 2文字目を標準出力へ
73         }
74         else if (c1 == 0x8e) { // 半角カタカナか?
75             if ((c2 = getchar()) == 0xff)
76                 exit(2); // 異常終了
77             putchar(c2); // 1文字を標準出力へ
78         }
79         else if (c1 >= 0x80 && c1 <= 0x9f) // 制御文字か?
80             putchar(c1 & 0x1f); // マスクしてを標準出力へ
81         else
82             putchar(c1); // 1文字を標準出力へ
83     }
84     return c1;
85 }

```

●フローチャート： 先頭ページ掲載のソースから生成した、関数 etos()のフローチャート(分岐状態の表示、ソースに記述したコメントの取り込まれ方をご確認下さい。)



●モジュール仕様書：先頭ページ掲載のソース内の関数 etos()の、モジュール(関数)仕様書

ソースコードサンプルから生成された、関数 etos()のモジュール仕様書です。関数の上にあるコメントブロックが、仕様書の概要説明書に適用されます。その他、ソース内のコメントが、どのように適用されるか、ご確認下さい。

- ・概要説明欄：関数の直ぐ上にあるコメントブロック(/* */で囲んだ範囲)が適用されます。
- ・引数：引数部分を改行して、各行にコメントを記述すると、仕様書の引数コメントに適用されます。
- ・ローカル変数：変数の右に付けたコメントが適用されます。
- ・戻り値：引数のコメントを仕様書に反映させるには、CasePlayer2 専用の記述を追加する必要があります。このため、戻り値の説明は、上記の「概要説明欄」に含めて記述するか、仕様書生成後に、仕様書に直接書き込む方法をお薦めしています。(仕様書に書き込んだ記述は、仕様書を再生成した場合も保持されます。)

(例)戻り値のコメントを仕様書に埋め込むための特殊な記述。これを関数のコメントヘッダに入れる必要有り。

```
/* ++module<etos>
++arguments
<int><>[戻り値のコメントを記入します。]
++end_module */
```

```
/*
機能：EUCからSJISに変換
説明：CasePlayer2仕様書作成サンプルソース
Copyright(c) GAIO TECHNOLOGY
*/
int etos(
int arg_a, /* 第1引数：パラメータA */
int arg_b) /* 第2引数：パラメータB */
{
int c1,c2; // 内部コード処理用
int flag; // フラグ

while ((c1 = getchar()) != -1) { /* 1文字
// 外字フラグのクリア
```

また、詳細解析機能により、この関数を使用している外部変数(グローバル変数)と、その参照関係が示されます。(↓参照している外部変数一覧、↓代入している外部変数一覧)

specific - [etos.vmod]

ファイル(F) 編集(E) 表示(V) ツール(T) ウィンドウ(W) ヘルプ(H)

名前 etos

| | |
|--------|--------|
| 種別 | 記憶クラス |
| Normal | extern |

定義位置 etos.c 42行目

概要説明

機能：EUCからSJISに変換
説明：CasePlayer2仕様書作成サンプルソース
Copyright(c) GAIO TECHNOLOGY

戻り値

| 型 | コメント |
|-----|------|
| int | |

引数

| 名称 | 型 | コメント |
|-------|-----|-------------|
| arg_a | int | 第1引数：パラメータA |
| arg_b | int | 第2引数：パラメータB |

参照している外部変数一覧

| 変数名 | 定義ファイル名 | 行番号 |
|-------|---------|-----|
| gaiji | etos.c | 32 |

代入している外部変数一覧

| 変数名 | 定義ファイル名 | 行番号 |
|-------|---------|-----|
| gaiji | etos.c | 32 |

ローカル変数一覧

| 変数名 | 型 | 記憶クラス | コメント |
|------|-----|-------|----------|
| c1 | int | auto | |
| c2 | int | auto | 内部コード処理用 |
| flag | int | auto | フラグ |

呼び出し関数一覧

| 関数名 | 定義ファイル名 | 行番号 |
|---------|---------|-----|
| getchar | etos.c | 46 |
| exit | etos.c | 51 |
| putchar | etos.c | 70 |

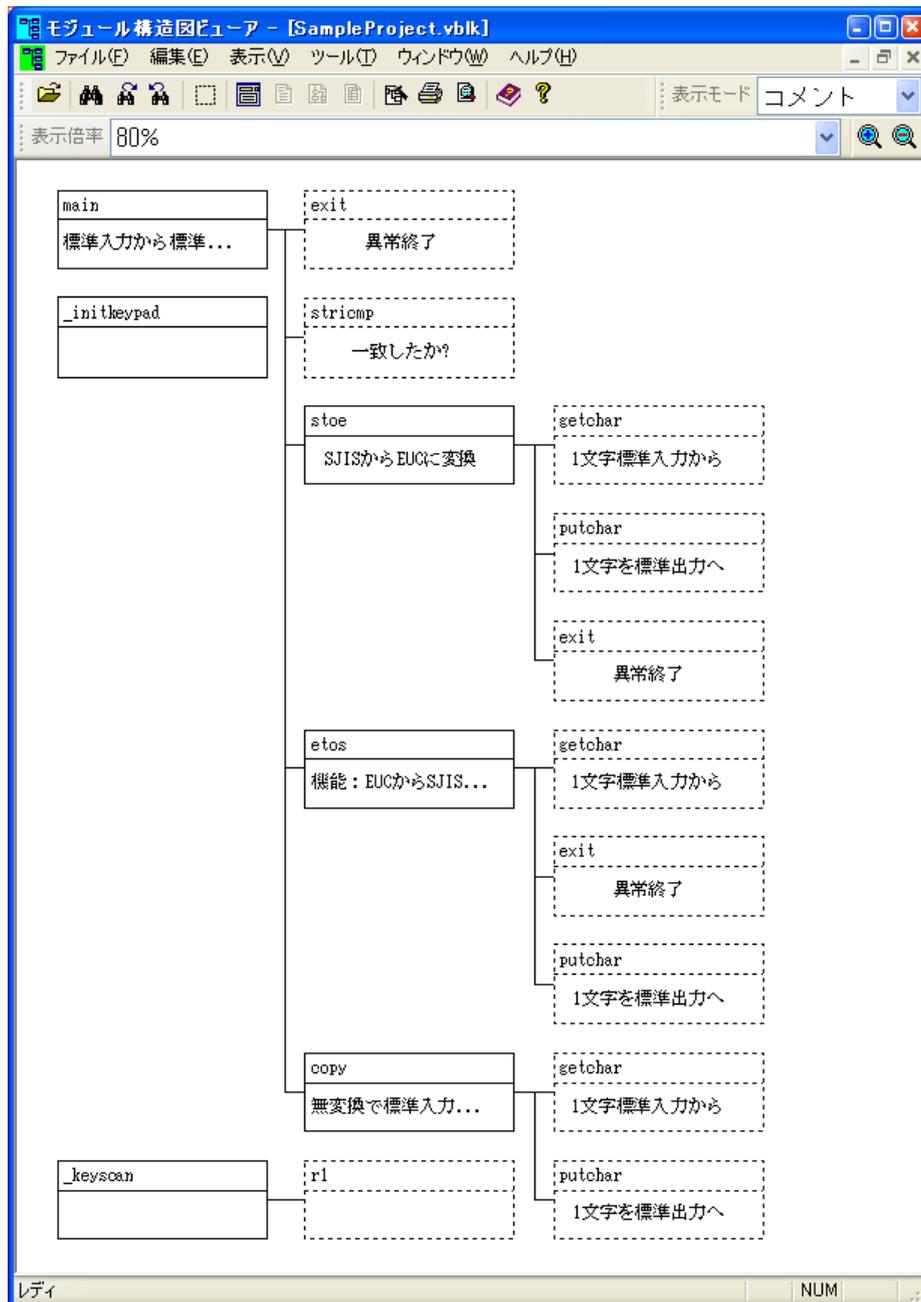
補足説明

レディ NUM 書禁

●モジュール構造図：先頭ページ掲載のソース内の関数 `etos()`を含む、プロジェクト全体のモジュール構造図(関数の呼出階層図)

ソースコードサンプルの関数を含む、プロジェクト全体のモジュール構造図(関数呼出相関図)です。1つのボックスが1つの関数に対応しており、関数が呼び出した関数を、階層的に右側に示しています。

実線のボックスは、プロジェクト内にソースコードが存在する関数、点線は、ライブラリなどのようにプロジェクト内に関数の実体ソースがないものを示しています。



●構造体仕様書： 先頭ページ掲載のソース内の構造体 `code_table` の仕様書

サンプルコード内に定義されている構造体「code_table」の仕様書です。構造体定義の上にあるコメントが、仕様書の概要説明に適用されます。メンバーの横にあるコメントが、要素一覧のコメント欄に適用されます。

```

18 /* 漢字コードテーブル */
19 struct code_table {
20     char *code_name;          /* 漢字コード表記 */
21     int code_id;             /* 漢字コードの識別番号 */
22 };

```

構造体仕様書

| | | | |
|-----------|------------|------------|-----|
| 名称 | code_table | 種別 | 構造体 |
| 定義位置 | etos.c 9行目 | | |
| 概要説明 | 漢字コードテーブル | | |
| 要素一覧 | | | |
| 名称 | 型 | コメント | |
| code_name | char * | 漢字コード表記 | |
| code_id | int | 漢字コードの識別番号 | |
| 補足説明 | | | |

レディ NUM 書禁

●モジュール一覧：先頭ページ掲載のソース内の関数 `etos()`を含む、プロジェクト全体のモジュール(関数)一覧

サンプルコード内に定義されている関数 `etos()` を含む、プロジェクト全体に含まれるモジュール(関数)の一覧です。各項目の行を右クリックすることで、ソースコード上の関数の定義位置へ移動することができます。



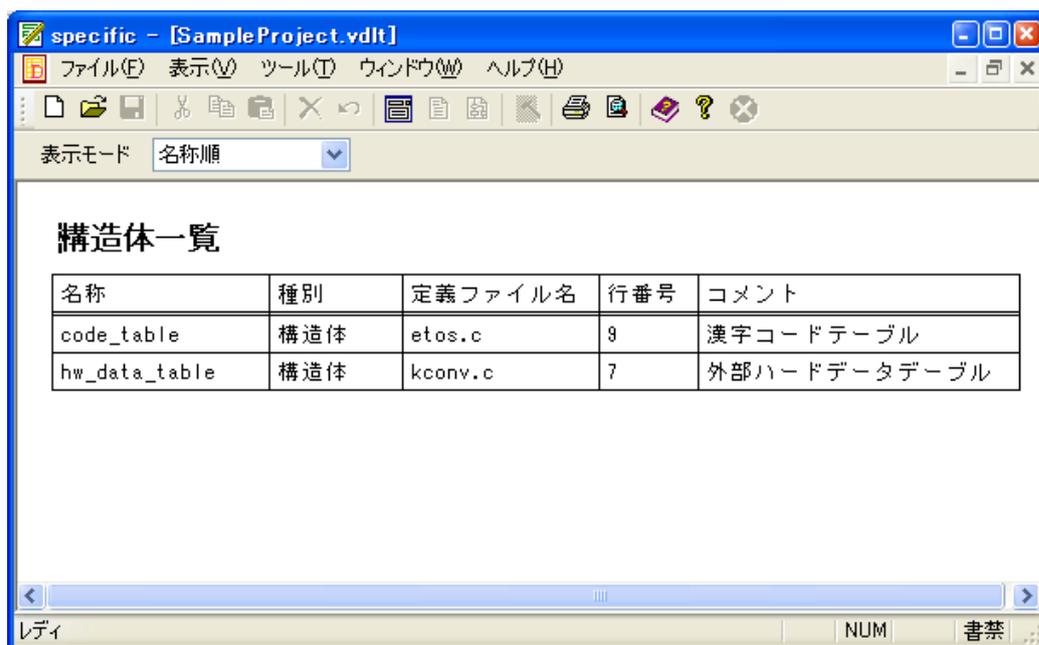
モジュール一覧

| 名称 | 定義ファイル名 | 行番号 | コメント |
|-------------|------------|-----|-----------------------|
| etos | etos.c | 20 | 機能：EUCからSJISに変換 |
| main | kconv.c | 32 | 標準入力から標準出力へ漢字コード変換を行う |
| copy | kconv.c | 71 | 無変換で標準入力から標準出力へコピー |
| _initkeypad | sample.xas | 19 | |
| _keyscan | sample.xas | 54 | |
| stoe | stoe.c | 3 | SJISからEUCに変換 |

レディ NUM 書禁

●構造体一覧： 先頭ページ掲載のソース内の構造体 `code_table` を含む、プロジェクト全体の構造体一覧

サンプルコード内に定義されている構造体 `code_table` を含む、プロジェクト全体に含まれる構造体の一覧です。各項目の行を右クリックすることで、ソースコード上の構造体の定義位置へ移動することができます。



END