

COBOL2002 Professional Tool Kit COBOL ソース解析ガイド

手引・操作書

3021-3-616-40

COBOL2002

前書き

■ 対象製品

P-2636-7344 COBOL2002 Developer Professional 04-40 (適用 OS : Windows 7, Windows 8.1, Windows 10, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019)

P-2936-7344 COBOL2002 Developer Professional(64) 04-40 (適用 OS : Windows 7(x64), Windows 8.1(x64), Windows 10(x64), Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019)

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI, HiRDB, OpenTP1, XDM, XMAP は、株式会社 日立製作所の商標または登録商標です。

Bing は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Excel は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle および Java は、オラクルおよびその関連会社の登録商標です。

UNIX は、The Open Group の商標です。

Visual Basic は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ マイクロソフト製品のスクリーンショットの使用について

マイクロソフトの許可を得て使用しています。

■ 発行

2021 年 10 月 3021-3-616-40



■ 著作権

All Rights Reserved. Copyright (C) 2017, 2021, Hitachi, Ltd.

変更内容

変更内容 (3021-3-616-40) COBOL2002 Developer Professional 04-40, COBOL2002 Developer Professional(64) 04-40

追加・変更内容	変更箇所
制御フロー解析のソース解析情報を利用して、節、段落、文および条件指定のソースコードを実行順にツリー形式で表示/出力できる機能（実行順序表示機能）をサポートした。 これに伴い、COBOL ソース解析で生成するドキュメントに、次の HTML ファイルおよび CSV ファイルを追加した。 <ul style="list-style-type: none">・ [実行順序表示] 画面・ 実行順序リスト	1.2, 1.4, 1.4.7, 2.4.1, 2.7, 3.7, 6.1, 7.4.5, 7.6.2, 8.1, 8.2.7, 10.1, 10.16, 付録 A
実行順序表示機能を有効とするため、開発マネージャおよび cblcsaSetOptions コマンドの [解析生成オプション設定] 画面で次のタブの内容を変更した。 <ul style="list-style-type: none">・ [その他の解析オプション] タブに [プログラムの制御フロー情報（実行順序表示機能で使用する情報）を抽出する] を追加した。・ [生成ドキュメント] タブで一覧情報 CSV に実行順序リストを追加した。	2.1, 5.2.7, 5.2.8
実行順序表示機能のサポートに伴い、[プログラム詳細] 画面の「セクション一覧」に [実行順序表示] 画面に遷移する [実行順序] の説明を追加した。	7.1, 7.4.1
[ソース表示] 画面でソースの最終行が表示されている場合も、左ペインで選択した見出しのソースコードが最上行に表示されるようにした。	7.4.3
[>>D] のあとが空または空白だけの行を空白行として扱うようにした。これに伴い、ソース表示の空白行の色の説明を変更した。	7.6.2
次に示すメッセージの説明を変更した。 KEDL1260A-E	9.3.3
次に示すメッセージを追加した。 KEDL1288A-W, KEDL1289A-W	9.3.3

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、次に示す COBOL2002 Professional 製品の機能である、COBOL ソース解析の使い方について説明したものです。

- COBOL2002 Developer Professional
- COBOL2002 Developer Professional(64)

このマニュアルでは、上記の製品を総称して、COBOL2002 Developer Professional と表記します。

■ 対象読者

COBOL ソース解析を利用して、既存の COBOL プログラムを解析して再利用するプログラマーで、次に示す項目を理解、習得していることを前提とします。

- Windows の基本的な操作方法
- COBOL2002 の知識

■ このマニュアルで使用する記号

このマニュアルで使用する記号を次のように定義しています。

記号	意味
[]	メニュータイトル、メニュー項目、ボタン、キー、およびアイコンの名称を示します。 例：[ファイル] メニュー [OK] ボタン [Enter] キー など
[] – []	前のメニューを選択し、続けて後ろの項目を選択することを示します。 例：[ファイル] – [開く] [ファイル] メニューから [開く] を選択することを示します。
[] + []	+ の前のキーを押したまま、後ろのキーを押すことを示します。 例：[Ctrl] + [C] キー [Ctrl] キーを押したまま [C] キーを押すことを示します。
△	半角の空白を入れること、またはスペースキーを指定された数だけ押すことを示します。
斜体	インストール先フォルダなど可変の個所を示します。
[]	省略できることを示します。 例：INCLUDE [サブシステム名] サブシステム名は省略できることを示します。

記号	意味
{ }	この記号で囲まれている複数の項目のうちから 1 つを選択することを示します。項目が縦に複数行にわたって記述されている場合は、そのうちの 1 行分を選択します。
	横に並べられた複数の項目に対して項目間の区切りを示し、「または」を意味します。
...	記述が省略されていることを意味します。 この記号の直前に示された項目を繰り返して複数個指定できます。
<u>下線</u>	括弧で囲まれた複数の項目のうち 1 項目に対して使用され、括弧内のすべてを省略したときにシステムが仮定する標準値を意味します。

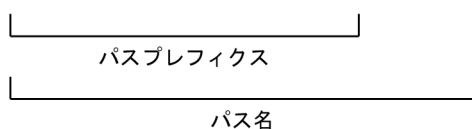
■ 用語の定義

このマニュアルで使用する用語の定義を次に示します。

パス名とパスプレフィクス

パス名とパスプレフィクスは、次のとおりです。

[ドライブ名¥フォルダ名¥・・・¥] ファイル名



絶対パス名

ドライブ名で始まるパス名。

ドライブ名¥フォルダ名¥・・・¥ファイル名

相対パス名

カレントフォルダからの相対のパス名。

- ・カレントフォルダの 1 階層上位のフォルダを経由する場合

..¥フォルダ名¥フォルダ名¥・・・¥ファイル名

- ・カレントフォルダ下のフォルダを経由する場合

フォルダ名¥フォルダ名¥・・・¥ファイル名

絶対パスプレフィクス

ドライブ名で始まるパスプレフィクス。

相対パスプレフィクス

カレントフォルダからの相対のパスプレフィクス。

目次

前書き	2
変更内容	4
はじめに	5

1	COBOL ソース解析の紹介	12
1.1	COBOL ソース解析とは	13
1.2	生成するソース解析情報の内容	15
1.3	解析できる COBOL ソースファイル	17
1.4	便利な使い方	18
1.4.1	変更した COBOL ソースファイルだけを解析する	18
1.4.2	コメント行の文字列を解析結果に表示する	18
1.4.3	設定した解析環境を保存/読み出しをして使う	18
1.4.4	メインフレームの COBOL ソースファイルを解析する	19
1.4.5	バッチモードで COBOL ソース解析を実行する	19
1.4.6	プログラム呼び出しの引数の整合性をチェックする	19
1.4.7	効率良くかつ効果的にソースコードリーディングを実施する	19
2	COBOL ソース解析の基本的な使い方	21
2.1	COBOL ソースファイルをすべて解析する	22
2.2	変更した COBOL ソースファイルだけを解析する	31
2.3	解析した COBOL ソースファイルの結果を参照する	35
2.4	HTML ファイル形式のソース解析情報を参照する	38
2.4.1	解析したプログラムの情報とソースを参照する	40
2.4.2	各種の分布情報を参照する	41
2.4.3	使われていないソースコードの候補を確認する	42
2.5	解析前に前回のソース解析情報を残す	44
2.5.1	ソース解析情報のフォルダを退避する	44
2.5.2	ソース解析情報の出力先を変更する	45
2.6	エラーがある COBOL プログラムを修正する	47
2.7	CSV ファイル形式のソース解析情報を加工する	49
2.8	COBOL ソース解析で操作した内容を参照する	52
3	業務目的別で使う	53
3.1	COBOL 資産を棚卸する	54
3.2	プログラムの処理を調査する	56

3.3	プログラムの入出力情報を調査する	59
3.4	コメント行の文字列を抽出する	62
3.5	メインフレームの固定長形式がある COBOL ソースファイルを解析する	66
3.6	プログラム呼び出しの実引数と仮引数の整合性をチェックする	68
3.7	処理ロジックが複雑なプログラムを調査する	71
4	バッチモードで実行する	76
4.1	バッチモードで COBOL ソース解析を実行する	77
5	画面のリファレンス	79
5.1	開発マネージャでの COBOL ソース解析の操作メニュー項目と設定画面	80
5.2	[解析生成オプション設定] 画面	83
5.2.1	[コンパイラオプション] タブ	83
5.2.2	[主プログラム指定] タブ	85
5.2.3	[プログラムコメント抽出] タブ	86
5.2.4	[節コメント抽出] タブ	88
5.2.5	[呼出先プログラム解決] タブ	90
5.2.6	[引数整合性チェック] タブ	92
5.2.7	[その他の解析オプション] タブ	94
5.2.8	[生成ドキュメント] タブ	96
6	コマンドリファレンス	99
6.1	COBOL ソース解析で使用するコマンド	100
	cblcsa コマンド – バッチモードで COBOL ソース解析を実行する	100
	cblcsaSetOptions コマンド – ソース解析オプションファイルを編集する	113
7	解析結果のリファレンス (HTML ファイル形式)	114
7.1	ソース解析情報の画面遷移	115
7.2	プロジェクトサマリ画面	117
7.3	システム固有画面	121
7.3.1	[COBOL ソースファイル一覧] 画面	121
7.3.2	[登録集原文ファイル一覧] 画面	122
7.3.3	[プログラム一覧] 画面	123
7.3.4	[登録集原文一覧] 画面	126
7.3.5	[ファイル一覧] 画面	130
7.3.6	[表(DB)一覧] 画面	132
7.3.7	[プログラム階層図] 画面	133
7.3.8	[デッドコード情報一覧] 画面	136
7.4	プログラム固有の情報	138
7.4.1	[プログラム詳細] 画面	138

7.4.2	[プログラムデータ] 画面	146
7.4.3	[ソース表示] 画面	147
7.4.4	[セクション階層図] 画面	149
7.4.5	[実行順序表示] 画面	151
7.5	[分布情報一覧] 画面	154
7.6	ソース解析情報の表示規則	160
7.6.1	COBOL ソースファイルの内容と差異があるソース解析情報の項目	160
7.6.2	ソース表示の色と行数カウント	161
7.6.3	ソース表示で表示されるソース行の最大長	166

8 解析結果のリファレンス (CSV ファイル形式) 167

8.1	CSV ファイルの種類	168
8.2	一覧情報 CSV	169
8.2.1	COBOL ソースファイル一覧	169
8.2.2	プログラム一覧	172
8.2.3	セクション一覧	174
8.2.4	ファイル一覧	177
8.2.5	データ項目一覧	180
8.2.6	不整合引数一覧	190
8.2.7	実行順序リスト	196
8.3	関係情報 CSV	200
8.3.1	プログラム呼出関係	200
8.3.2	登録集原文参照関係	202
8.3.3	ファイル参照関係	204
8.3.4	表(DB)参照関係	205

9 エラー時の対処 208

9.1	事例から探す	209
9.1.1	COBOL ソース解析を実行できない	209
9.1.2	出力結果が意図したとおりではない	210
9.1.3	ログが蓄積されない	212
9.1.4	一時ファイルが削除されない	213
9.1.5	開発マネージャから解析結果を参照できない	213
9.1.6	解析結果の出力先の変更が開発マネージャに反映されない	213
9.1.7	ソース解析プロジェクトの解析生成の実行で想定以上に時間が掛かる	213
9.2	ログを参照する	215
9.2.1	解析ログを参照する	215
9.2.2	操作ログを参照する	217
9.3	メッセージから探す	225

9.3.1	メッセージの形式	225
9.3.2	解析エラー発生時のエラー位置の表示	226
9.3.3	メッセージの一覧	226
9.3.4	システムエラー番号	240
9.4	そのほかの方法から探す	242
10	COBOL ソース解析の仕様範囲のリファレンス	243
10.1	解析情報の上限値	244
10.2	制限がある構文	246
10.3	COBOL2002 の言語仕様の解析結果	249
10.4	COBOL ソース解析で使えるコンパイラオプション	251
10.5	COBOL ソース解析で使えるコンパイラ環境変数	258
10.6	デッドコード候補の検知方法	259
10.7	埋め込み SQL 文を含む COBOL ソースファイルの解析	267
10.8	実行文として数える文	276
10.9	解析結果に出力される COBOL ソースファイル中の名称	279
10.10	COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI)	281
10.10.1	COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) の説明	281
10.10.2	COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) の不正なパラメータ値の扱い	285
10.11	引数整合性チェック	287
10.11.1	引数整合性チェックとは	287
10.11.2	引数整合性チェックの検査内容	288
10.11.3	引数整合性チェック使用時の注意事項	299
10.12	主プログラムの指定方法	301
10.12.1	主プログラム自動検出	302
10.12.2	主プログラム名による主プログラム指定	304
10.13	プログラムコメントの抽出	306
10.13.1	プログラムコメント抽出とは	306
10.13.2	プログラムコメントの抽出位置の指定	307
10.13.3	プログラムコメント抽出時の注意事項	313
10.14	節コメントの抽出	314
10.14.1	節コメント抽出とは	314
10.14.2	節コメントの抽出位置の指定	315
10.14.3	節コメント抽出時の注意事項	320
10.15	そと PERFORM 文からの呼出先情報の抽出規則	321
10.16	制御フロー解析による実行順序表示機能	323
10.16.1	制御フロー解析とは	323
10.16.2	実行順序表示機能とは	324
10.16.3	実行順序ツリーでトレースできる実行順序情報	325

10.16.4 実行順序表示機能使用時の注意事項 330

11 使用上の注意事項 333

11.1 解析／生成に関する注意事項 334

付録 336

付録 A ソース解析情報のフォルダ構成 337

付録 B 解析を補助する機能 342

付録 B.1 一意名呼び出し対応付けファイル 342

付録 B.2 サービスルーチンファイル 344

付録 B.3 -SQLAutoCopy コンパイラオプション 345

付録 C 各バージョンの変更内容 347

付録 D このマニュアルの参考情報 351

付録 D.1 関連マニュアル 351

付録 D.2 このマニュアルでの表記 351

付録 D.3 KB（キロバイト）などの単位表記について 352

索引 353

1

COBOL ソース解析の紹介

COBOL ソース解析の概要について説明します。

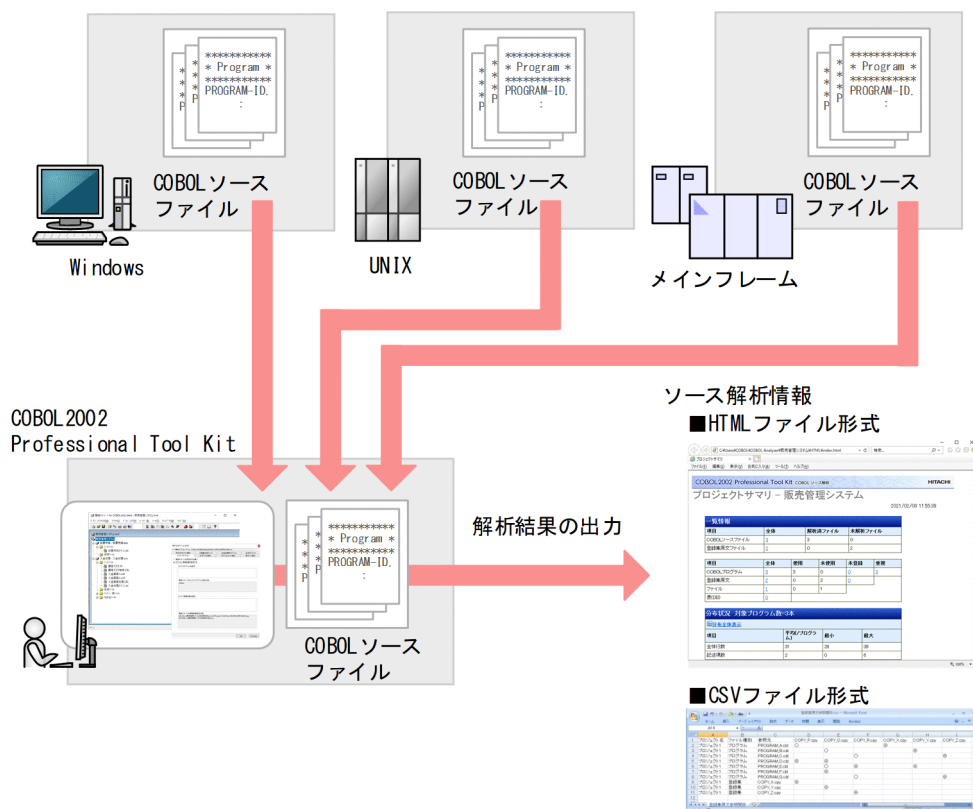
1.1 COBOL ソース解析とは

COBOL ソース解析とは、COBOL プログラムを解析し、プログラムの構造や関係などの解析情報を出力して、プログラムを「見える化」する機能です。

これまで長期間にわたって稼働してきたシステムの場合、システムを改変しないまま稼働してきたため、担当者が不在のケースがあります。また、保守書を紛失し、プログラムの構造を調査する工数が掛かってしまうこともあります。

COBOL ソース解析は、COBOL 開発環境で使用している COBOL ソースファイルをそのまま入力情報として利用して、構造がわからない COBOL プログラムの解析結果を出力できます。

図 1-1 COBOL ソース解析の概要



ポイント

COBOL ソース解析を使用する前に、組み込みデータベースのセットアップが必要です。セットアップの手順については、マニュアル「COBOL2002 Professional 製品 導入ガイド」を参照してください。

COBOL ソース解析では、COBOL プログラムを解析した結果をソース解析情報として出力します。ソース解析情報を参照すれば、プログラム間の入出力、テーブル領域の共有などの COBOL 特有の処理がわかるので、COBOL プログラムの保守に役立ちます。

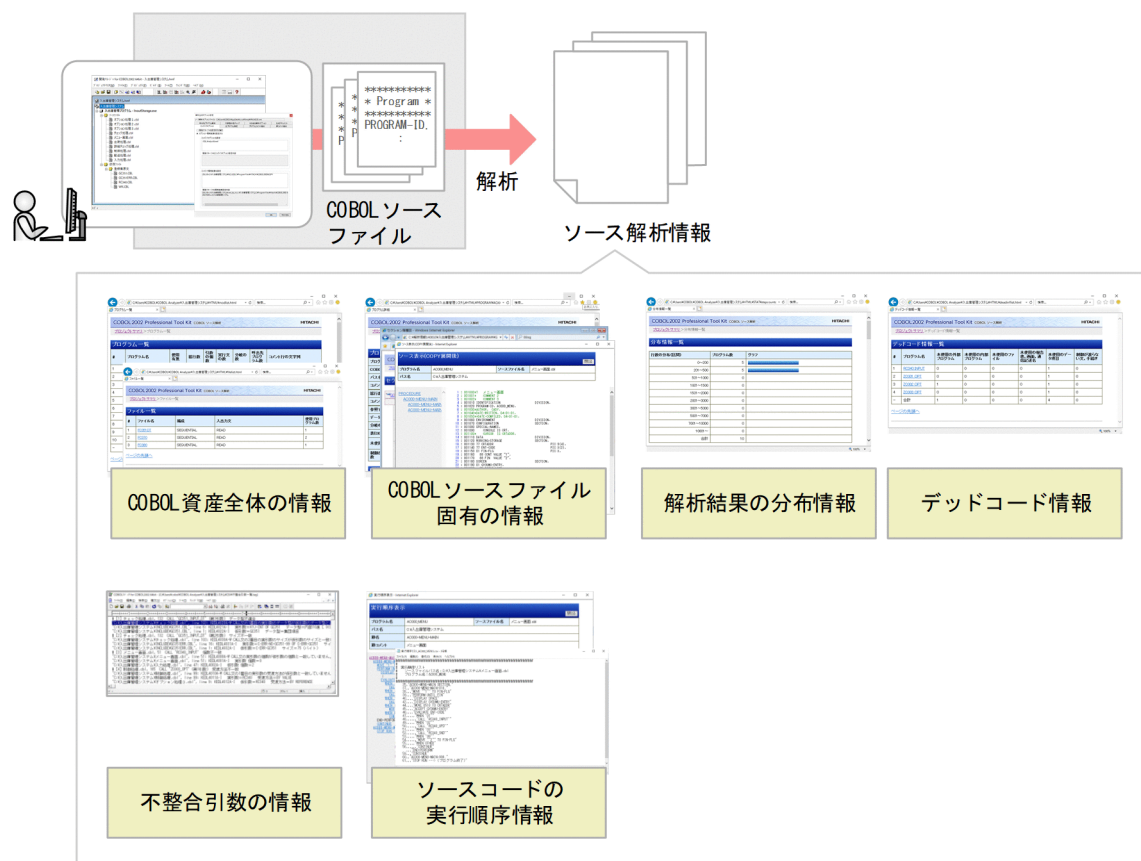
ソース解析情報には、HTML ファイル形式と CSV ファイル形式があります。HTML ファイル形式は、リンクをクリックしデータの関係を確認できます。CSV ファイル形式のソース解析情報は、Excel で参照または保守資料を作成する目的で使用できます。

1.2 生成するソース解析情報の内容

COBOL ソース解析が生成するソース解析情報には、COBOL 資産全体の情報、COBOL ソースファイル固有の情報、解析結果の分布情報、デッドコード情報、および不整合引数の情報が出力されます。

COBOL ソース解析が生成するソース解析情報の概要を次の図に示します。

図 1-2 COBOL ソース解析が生成するソース解析情報の概要



- COBOL 資産全体の情報

解析対象の COBOL ソースファイル、使用されていない COBOL ソースファイル、COBOL ソースファイルが使用しているファイルなどを解析できます。解析対象の COBOL ソースファイルの一覧を参照できるので、解析対象の COBOL 資産全体を理解するために役立ちます。

- COBOL ソースファイル固有の情報

COBOL ソースファイルのプログラム名 (PROGRAM-ID)、COBOL ソースファイルが使用している登録集原文、COBOL ソースファイルの手続き部 (節または段落)などを解析できます。COBOL ソースファイルの内容が項目ごとに参照できるので、COBOL ソースファイルを理解するために役立ちます。

- 解析結果の分布情報

行数の分布、実行文や分岐の数の分布、節数や段落数の分布などを解析できます。分布情報を項目ごとに参照できるので、システムの規模やシステムの状況を把握するときに役立ちます。

- デッドコード情報

実行される可能性がない文、プログラムなど、プログラムの実行結果に影響を与えないソースコードの候補（デッドコード候補）を検出できます。実行されない処理がわかるため、プログラムの分岐条件の修正や、プログラムテストが不要なファイルを発見するときに役立ちます。

- 不整合引数の情報

CALL 文によるプログラム呼び出しで引数の個数、受渡方法、データ属性やサイズが一致していない実引数と仮引数の情報を検出できます。システムの更改や移植の契機で修正が必要な引数不整合の規模を把握したり、テストや障害調査で引数の整合性を検証したりするときに役立ちます。

- ソースコードの実行順序情報

ソースコードのすべての実行順序をツリー形式で表示またはリスト出力できます。実行経路の組み立てや、分岐先および呼出先を探す手間が不要になるので、ソースコードリーディングを効率的かつ効果的に実施するのに役立ちます。

1.3 解析できる COBOL ソースファイル

COBOL ソース解析では、文字コードがシフト JIS コードの COBOL ソースファイルおよび登録集原文ファイルを解析対象としています。

日立 COBOL2002 用の COBOL ソースファイルおよび登録集原文ファイルの場合は、構文的なコンパイルエラーがないものを対象としています。COBOL 言語用の SQL プリプロセサを実行する UAP ソースファイルの場合は、次の条件を満たすものを対象としています。

- SQL プリプロセサでエラーがない
- プリプロセスしたあとの COBOL ソースファイルで構文的なコンパイルエラーがない

COBOL ソースファイル内のソース、および登録集原文ファイル内の登録集原文は、日立 COBOL85 の言語仕様で記述されたものが解析対象です。ほかの言語仕様で記述されたソース、および登録集原文に関する情報は、COBOL ソース解析が生成するソース解析情報には出力されません。

そのほか、COBOL ソース解析の制限については、「[10. COBOL ソース解析の仕様範囲のリファレンス](#)」を参照してください。

1.4 便利な使い方

COBOL ソース解析の便利な使い方です。

- 変更した COBOL ソースファイルだけを解析する
- コメント行の文字列を解析結果に表示する
- 設定した解析環境を保存/読み出しをして使う
- メインフレームの COBOL ソースファイルを解析する
- バッチモードで COBOL ソース解析を実行する
- プログラム呼び出しの引数の整合性をチェックする
- 効率良くかつ効果的にソースコードリーディングを実施する

1.4.1 変更した COBOL ソースファイルだけを解析する

解析済みの COBOL ソースを再度解析するときには、COBOL ソース解析では COBOL ソースファイルを更新したかどうかを検出し、更新したファイルだけを解析します。複数の COBOL ソースファイルの一部を変更した場合、解析する対象の COBOL ソースファイルを手動で選択する手間が省けます。また、更新された COBOL ソースファイルに関係する個所だけを解析・生成するため、解析時間とソース解析情報の生成時間を短縮できます。

1.4.2 コメント行の文字列を解析結果に表示する

COBOL ソース解析では、プログラム段落や手続き部の節に付けたコメントを解析結果に表示できます。表示が不要な場合は解析結果に表示しないこともできます。

また、コメント行に命名規則がある場合、命名規則に従ったコメント行を抽出できます。例えば、コメント行の前に「T」を付ける命名規則がある場合、「T」を検出して、「T」が付いているコメント行だけを抽出できます。

使用例については、「[3.4 コメント行の文字列を抽出する](#)」を参照してください。

1.4.3 設定した解析環境を保存/読み出しをして使う

COBOL ソース解析を使用するときには、COBOL 資産のソースファイル一式を登録します。少数のファイルであれば追加に時間はそれほど掛かりませんが、10,000 ファイル以上を一括して追加する場合は時間が掛かります。さらに、追加した状態を保存しておいて、再度同じ COBOL ソースファイル一式を解析することもあります。

このような場合、COBOL ソース解析で設定した解析環境（環境設定情報）を保存して利用できます。保存した環境設定情報を必要なときに使用、またはほかの Windows マシンで使用できます。

解析環境を保存または読み出すときは、組み込みデータベースのコマンドを使用します。解析環境の保存と読み出しの手順については、マニュアル「COBOL2002 Professional 製品 導入ガイド」の解析環境の操作に関する説明を参照してください。

1.4.4 メインフレームの COBOL ソースファイルを解析する

メインフレーム向けに開発した COBOL ソースファイルでは、機能キャラクタによって記述がずれることがあります。

COBOL ソース解析では、コンパイラオプションを設定して、メインフレーム向けの COBOL ソースファイルを解析できます。COBOL ソースファイルを修正する必要がありません。

使用例については、「[3.5 メインフレームの固定長形式がある COBOL ソースファイルを解析する](#)」を参照してください。

1.4.5 バッチモードで COBOL ソース解析を実行する

COBOL ソース解析をバッチモードで実行すれば、大量のプログラムを自動的にソース解析できます。また、プログラムのビルドに同期させて COBOL ソース解析を実行することで、COBOL プログラムとソース解析情報を常に一致させることもできます。

バッチモードで COBOL ソース解析を実行する方法については、「[4. バッチモードで実行する](#)」を参照してください。

1.4.6 プログラム呼び出しの引数の整合性をチェックする

引数整合性チェックの機能を使用すれば、プロジェクト中のすべての CALL 文の実引数と呼出先プログラムの仮引数の整合性をチェックできます。

引数整合性チェックを実施する方法については、「[3.6 プログラム呼び出しの実引数と仮引数の整合性をチェックする](#)」を参照してください。

1.4.7 効率良くかつ効果的にソースコードリーディングを実施する

実行順序表示機能を使用すれば、ソースコードの実行順序を確認しながら、ソースコードリーディングを実施できます。頭の中で実行経路を組み立てる必要はなく、分岐先や呼出先を探す手間で集中力が中断されることもないため、効率良くかつ効果的にソースコードリーディングができます。

実行順序表示機能を使ったソースコードリーディングを実施する方法については、「[3.7 処理ロジックが複雑なプログラムを調査する](#)」を参照してください。

2

COBOL ソース解析の基本的な使い方

COBOL ソース解析の操作手順，操作画面の見方および解析結果の見方について説明します。

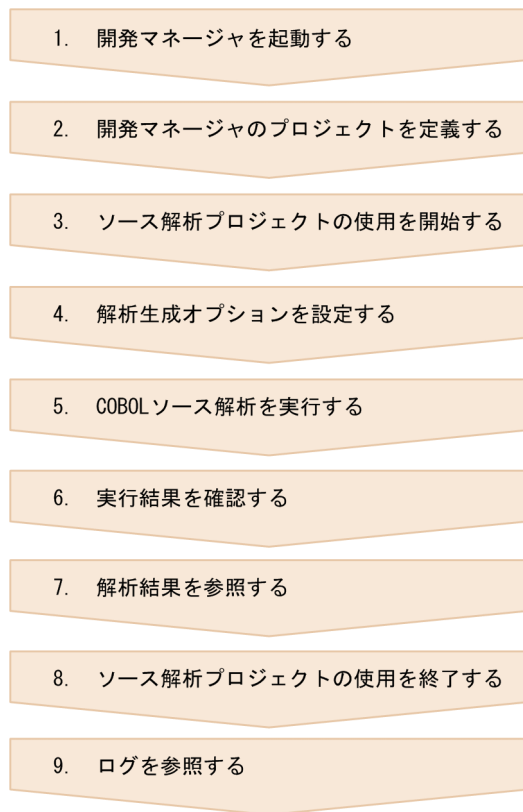
2.1 COBOL ソースファイルをすべて解析する

COBOL ソース解析を使って、COBOL ソースファイルを新規で解析する手順です。

作業の前に確認すること

- DB ユティリティで組み込みデータベースのセットアップが完了している。
- ODBC データソースの設定を完了している。

図 2-1 COBOL ソース解析の操作の手順（新規で解析する場合）

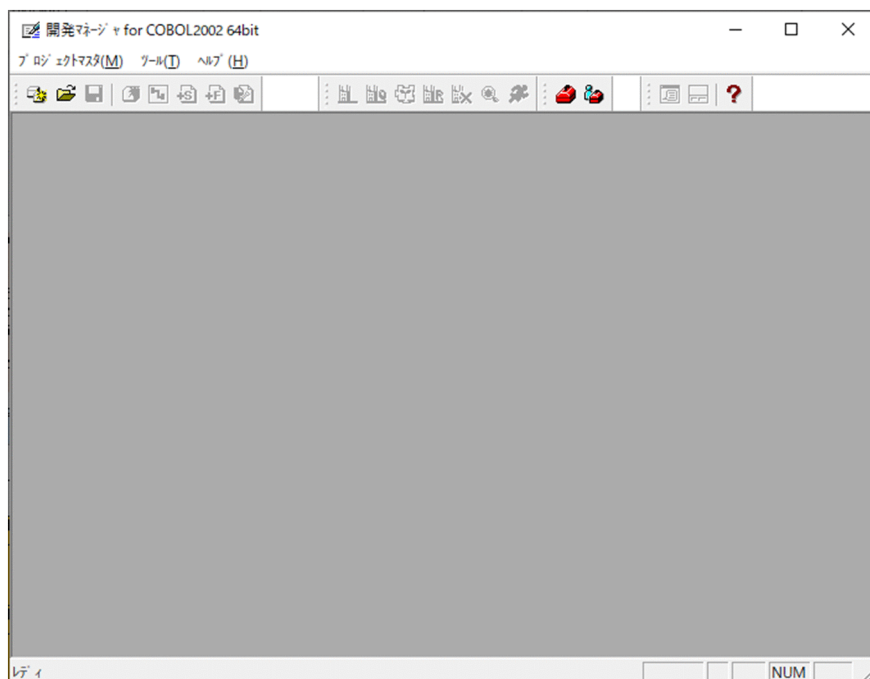


各手順について説明します。手順の番号は図中の番号と対応しています。

作業の手順

1. 開発マネージャを起動します。

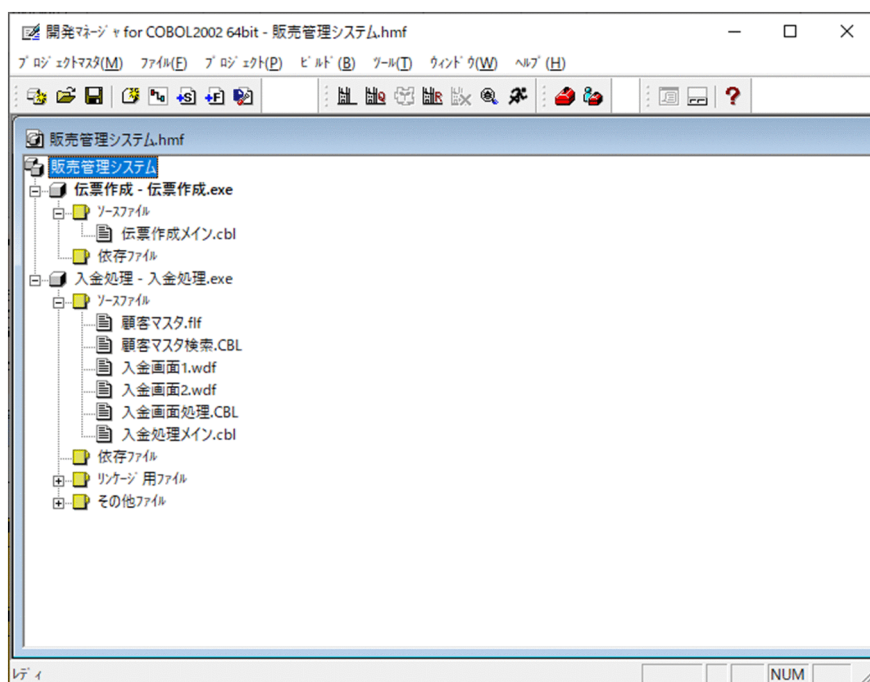
Windows のプログラム一覧から [COBOL2002] – [開発マネージャ for COBOL2002] を選択します。開発マネージャのウィンドウが表示されます。



2. 開発マネージャのプロジェクトを定義します。

プロジェクトマスタの作成からコンパイルまでを実施します。設定したコンパイラオプションは、COBOL ソース解析の解析生成オプションに引き継がれます。

開発マネージャの使用方法については、マニュアル「COBOL2002 操作ガイド」を参照してください。



ポイント

COBOL ソース解析の対象は、プロジェクトマスタで管理するすべてのプロジェクトの COBOL ソースファイルです。

3. ソース解析プロジェクトの使用を開始します。

[ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。



任意のソース解析プロジェクト名を設定し、[OK] ボタンをクリックします。

4. 解析生成オプションを設定します。

[ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。



• 主プログラムを指定する

[解析生成オプション設定] 画面の [主プログラム指定] タブを選択します。

ここでは、解析する COBOL ソースファイルの主プログラムを指定します。主プログラムを指定する方法には次の 2 つがあり、2 つの方法は併用できます。

- ・主プログラムの自動検出を使用する
- ・主プログラムのプログラム名を指定する

主プログラムの自動検出を使用する場合は、[主プログラムを COBOL ソースファイル先頭の最外側プログラムの中から検出する] のチェックボックスをオンにして、USING 検索条件と RETURNING 検出条件を選択します。

主プログラムのプログラム名を指定する場合は、[主プログラム名の指定] テキストボックスに指定します。複数の主プログラム名を指定する場合は、コンマまたは改行で区切って指定します。

注意事項

主プログラムは過不足なく指定してください。解析時に自動検出で検出された主プログラム、および [主プログラム名の指定] テキストボックスに指定した主プログラムをルートとして、プログラムの呼び出し関係が解析され、プログラムの使用／未使用の情報が抽出されます。主プログラムの指定に過不足があると、プログラムの使用／未使用が正しく判定できません。

- ・生成ドキュメント、出力先を指定する

[解析生成オプション設定] 画面の [生成ドキュメント] タブを選びます。

ここでは、出力するソース解析情報と出力先を指定します。

- ・出力するソース解析情報（HTML ファイル形式、CSV ファイル形式）を指定する

ソース解析情報（HTML ファイル形式、CSV ファイル形式）に出力する情報を選択できます。チェックボックスを選択して指定してください。

CSV ファイルは、デフォルトではチェックボックスはすべてオフになっています。

- ・ソース解析情報の出力先を指定する

[参照] ボタンをクリックして、出力先フォルダを設定します。ここに設定した場所に、ソース解析情報（HTML ファイル形式、CSV ファイル形式）が出力されます。

注意事項

出力先フォルダを指定しない場合、「Windows の環境変数 USERPROFILE で定義されているフォルダ¥COBOL Analyzer」の下に、解析対象プロジェクト名のフォルダが作成され、そこにソース解析情報が出力されます。出力先を変更したい場合は、出力先フォルダを指定してください。

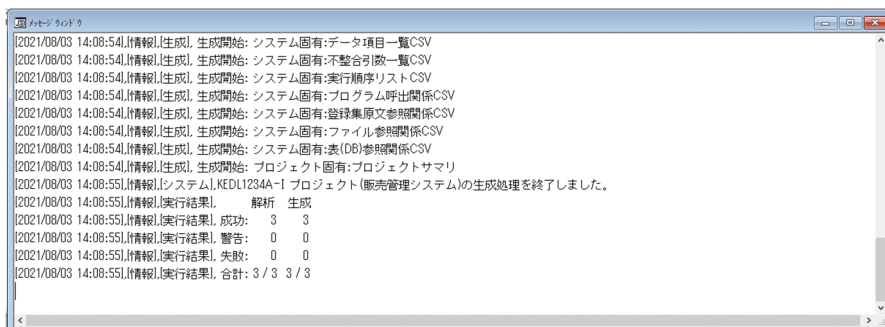
[解析生成オプション設定] 画面では、上記のほかに、ソース解析時のコンパイラオプション、節コメント行およびプログラムコメント行を抽出するときの条件を設定できます。詳細については、「[5.2 \[解析生成オプション設定\] 画面](#)」を参照してください。

5. COBOL ソース解析を実行します。

[ツール] - [COBOL ソース解析] - [解析の実行] を選択します。

6. 実行結果を確認します。

COBOL ソース解析の実行結果は、開発マネージャのメッセージウィンドウに表示されます。



解析に失敗した場合、ソース解析情報が正しく生成されないため、メッセージウィンドウにエラーメッセージが表示されていないかを確認してください。エラーがある場合、エラー原因を調査し、対策後、COBOL ソース解析を再実行してください。

7. 解析結果を参照します。

HTML 形式の解析結果を参照する場合は、開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択すると、プロジェクトサマリ画面 (index.html) が表示されます。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ - 販売管理システム

2021/02/08 14:56:16

一覧情報			
項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	3	3	0
登録集原文ファイル	2	0	2

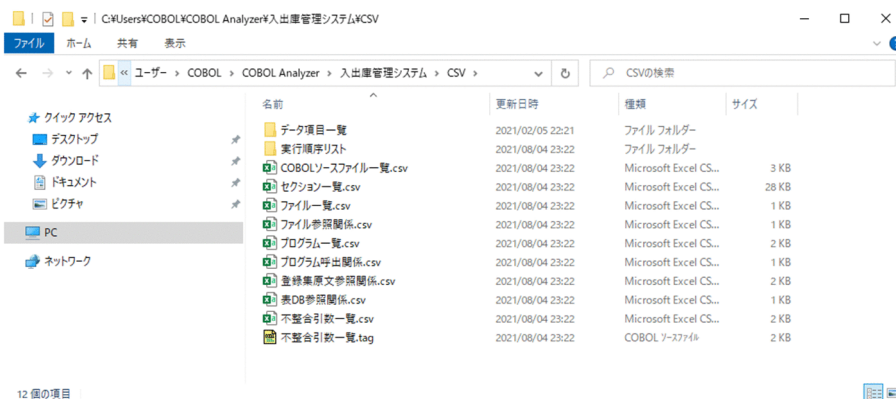
項目	全体	使用	未使用	未登録	重複
COBOLプログラム	3	3	0	0	3
登録集原文	2	0	2	0	0
ファイル	1	0	1	0	0
表(DB)	0	0	0	0	0

分布状況 対象プログラム数=3本

分布全体表示

項目	平均(プログラム)	最小	最大
全体行数	31	28	38
記述項数	2	0	6
実行文の数	2	1	5
分岐の数	0	0	0
節数	1	1	1
段落数	1	1	1
呼出元プログラム数	0	0	0

CSV 形式の解析結果を参照する場合は、開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(CSV/TAG)の表示] を選択すると、CSV ファイルおよびタグファイルを格納したフォルダが表示されます。



ポイント

解析結果は、エクスプローラーなどで出力先フォルダを表示することで、ソース解析プロジェクトの使用を終了したあとでも参照できます。

8. ソース解析プロジェクトの使用を終了します。

[ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用終了] を選択します。

9. ログを参照します。

開発マネージャのメッセージウィンドウで解析状態および生成状態に、エラーまたは未生成が表示されている場合、ソース解析情報が正しく生成されていないことが考えられます。解析ログまたは操作ログを確認して、解析に失敗した原因を調査してください。

- 解析ログ

解析または生成でエラーになった場合に、コンパイルエラーを確認するときに参照します。次に示すファイルに格納されます。

Windows の環境変数 *TEMP* で定義されているフォルダ¥COBOLAnalyzeLog.tag

- 操作ログ

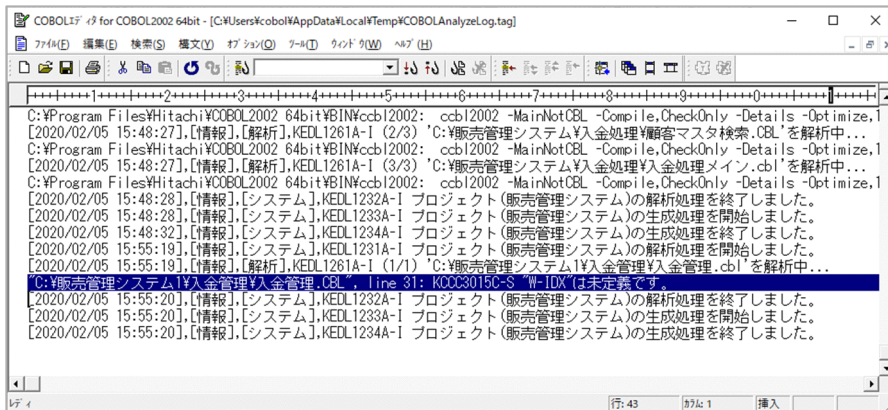
[解析生成オプション設定] 画面に設定した内容および操作の誤りを確認するときに参照します。次に示すファイルに格納されます。

Windows の環境変数 *TEMP* で定義されているフォルダ¥COBOLControlLog.txt

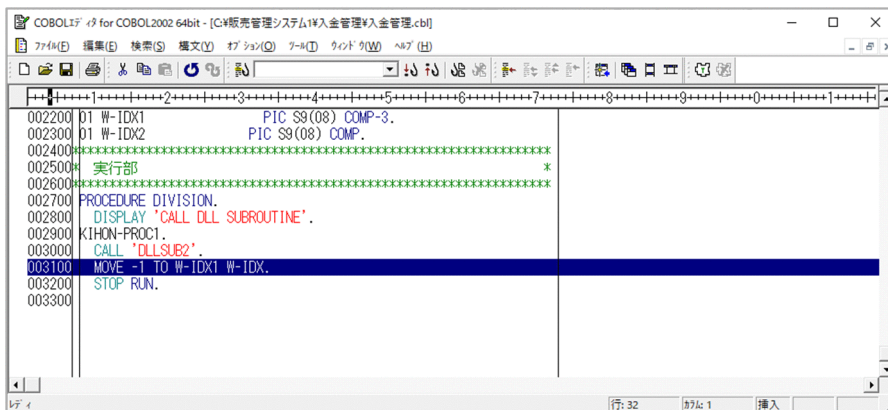
参照方法は、ログの種類によって異なります。

- 解析ログを参照する

解析ログ (COBOLAnalyzeLog.tag) を選択すると、COBOL エディタが起動します。



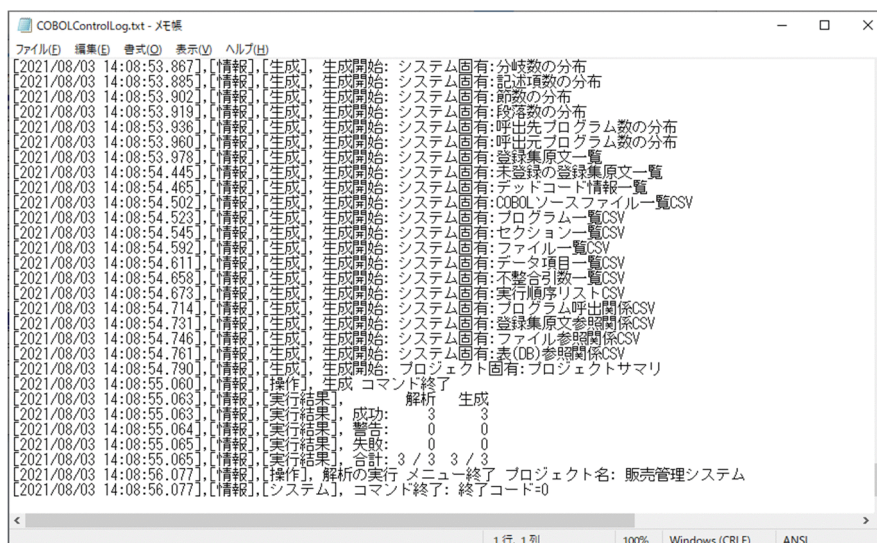
エラーメッセージをダブルクリックすると、COBOL ソースファイルが表示され、該当行にカーソルが移動します。エラーの個所を COBOL エディタで修正します。



この例では、手続き部に「DIVISION.」がないことが警告の原因なので、「DIVISION.」を入力します。上書き保存して、COBOL ソースファイルを閉じます。

- 操作ログを参照する

操作ログ (COBOLControlLog.txt) を選択すると、テキストエディタが起動します。操作ログのエラーを調査し、操作内容を確認してください。



問題を解決したあとで、COBOL ソース解析を再実行します。エラーについては、「[9. エラー時の対処](#)」を参照してください。

2.2 変更した COBOL ソースファイルだけを解析する

変更した COBOL ソースファイルだけを解析する手順です。

いったん解析した COBOL ソースファイルを COBOL エディタで変更したあとに、解析を再実行する場合、変更した COBOL ソースファイルを自動的に検出し、変更したファイルだけを解析・生成できます。

作業の前に確認すること

- 再解析する解析対象プロジェクトを、すでに解析している。
- 変更した COBOL ソースファイルを用いて再コンパイルしている。

図 2-2 COBOL ソース解析の操作の手順（変更した COBOL ソースファイルだけを解析する場合）



各手順について説明します。手順の番号は図中の番号と対応しています。

作業の手順

1. 再解析するプロジェクトマスタを開きます。

変更した COBOL ソースファイルを含むプロジェクトマスタを開発マネージャで開きます。

2. ソース解析プロジェクトの使用を開始します。

〔ツール〕－〔COBOL ソース解析〕－〔ソース解析プロジェクトの使用開始〕を選択します。

ソース解析プロジェクト名を設定するダイアログが表示されます。

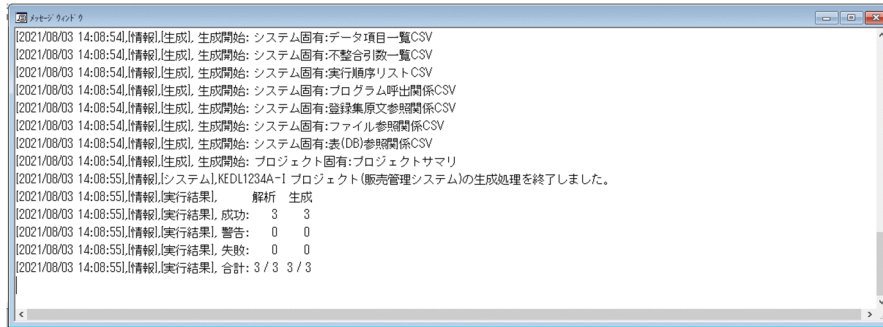


再解析するソース解析プロジェクト名を設定し、[OK] ボタンをクリックします。

3. COBOL ソース解析を実行します。

[ツール] - [COBOL ソース解析] - [解析の実行] を選択します。

4. 実行結果を確認します。



開発マネージャのメッセージウィンドウにエラーメッセージが表示されていないかを確認します。エラーがある場合、エラー原因を調査し、対策後、COBOL ソース解析を再実行してください。

5. 解析結果を参照します。

開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] または [解析結果(CSV/TAG)の表示] を選択します。

The screenshot shows the "プロジェクトサマリ - 販売管理システム" (Project Summary - Sales Management System) report. The report includes a table of overview information, a table of project details, and a distribution status table.

一覧情報			
項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	3	3	0
登録集原文ファイル	2	0	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	3	3	0	0	3
登録集原文	2	0	2	0	0
ファイル	1	0	1	0	0
表(DB)	0	0	0	0	0

分布状況 対象プログラム数=3本			
分布全体表示			
項目	平均(プログラム)	最小	最大
全行行数	31	28	38
記述項数	2	0	6
実行文の数	2	1	5
分岐の数	0	0	0
ループの数	1	1	1
段落数	1	1	1
呼出元プログラム数	0	0	0

ポイント

解析結果は、エクスプローラーなどで出力先フォルダを表示することで、ソース解析プロジェクトの使用を終了したあとでも参照できます。

6. ソース解析プロジェクトの使用を終了します。

[ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用終了] を選択します。

7. ログを参照します。

開発マネージャのメッセージウィンドウで解析状態および生成状態に、エラーまたは未生成が表示されている場合、ソース解析情報が正しく生成されていないことが考えられます。解析ログまたは操作ログを確認して、解析に失敗した原因を調査してください。

• 解析ログ

解析または生成でエラーになった場合に、コンパイルエラーを確認するときに参照します。次に示すファイルに格納されます。

Windows の環境変数 TEMP で定義されているフォルダ¥COBOLAnalyzeLog.tag

• 操作ログ

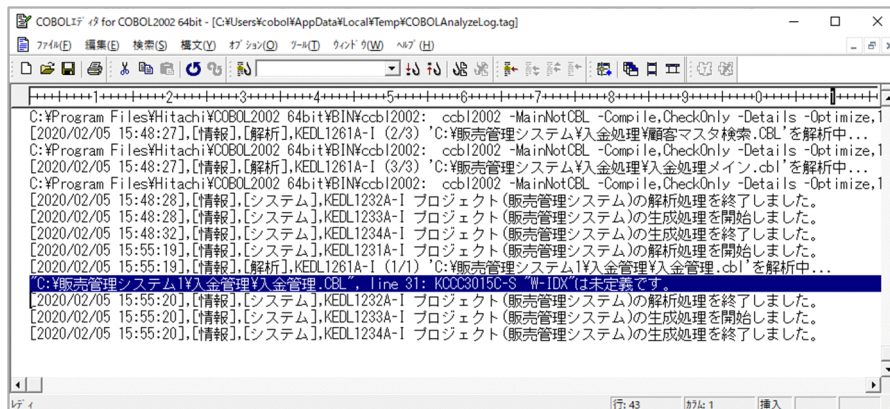
[解析生成オプション設定] 画面に設定した内容および操作の誤りを確認するときに参照します。次に示すファイルに格納されます。

Windows の環境変数 TEMP で定義されているフォルダ¥COBOLControlLog.txt

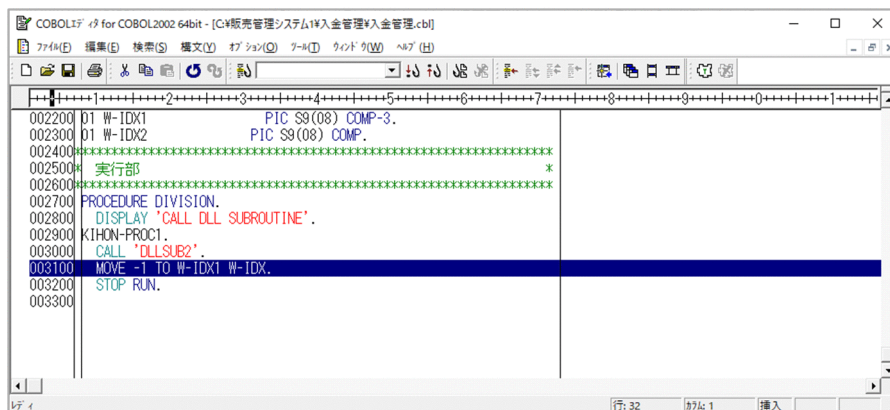
参照方法は、ログの種類によって異なります。

• 解析ログを参照する

解析ログ (COBOLAnalyzeLog.tag) を選択すると、COBOL エディタが起動します。



エラーメッセージをダブルクリックすると、COBOL ソースファイルが表示され、該当行にカーソルが移動します。エラーの個所を COBOL エディタで修正します。

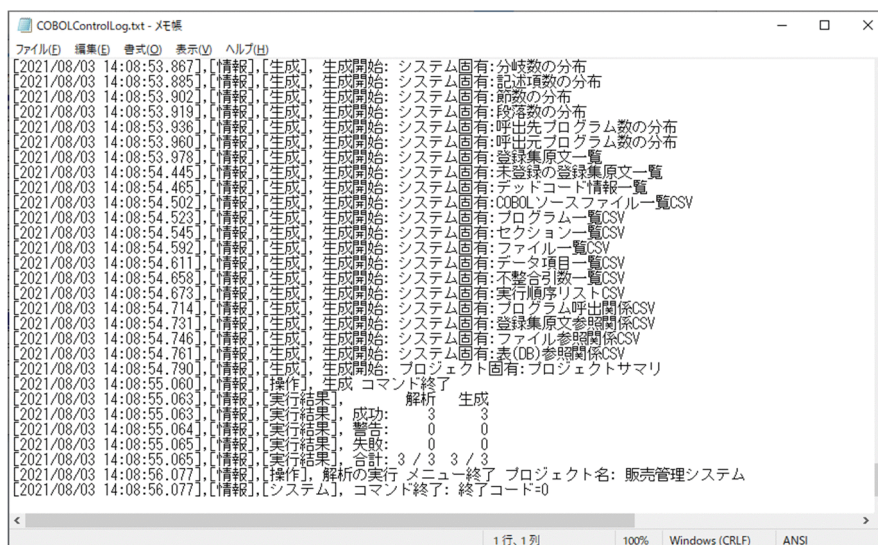


この例では、手続き部に「DIVISION.」がないことが警告の原因なので、「DIVISION.」を入力します。上書き保存して、COBOL ソースファイルを閉じます。

- 操作ログを参照する

操作ログ (COBOLControlLog.txt) を選択すると、テキストエディタが起動します。

操作ログのエラーを調査し、操作内容を確認してください。



問題を解決したあとで、COBOL ソース解析を再実行します。エラーについては、「[9. エラー時の対処](#)」を参照してください。

2.3 解析した COBOL ソースファイルの結果を参照する

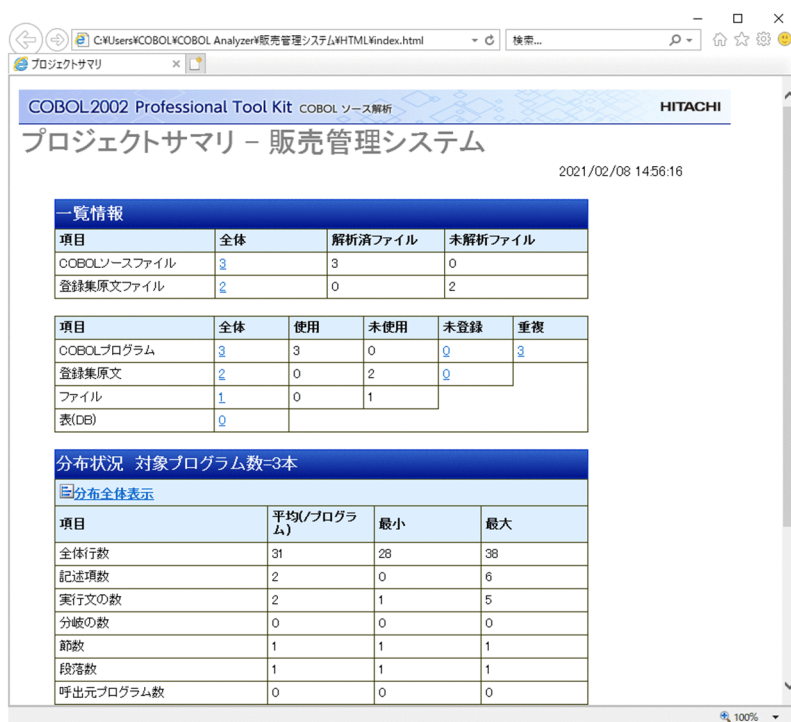
COBOL ソース解析で解析した結果を参照する手順です。

COBOL ソース解析の実行結果には、次に示す情報があります。

- 解析結果（HTML ファイル形式、CSV ファイル形式）
- 解析結果のログ（解析ログ、操作ログ）

HTML ファイル形式のソース解析情報を表示する場合

1. 参照したい解析結果を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択します。
HTML ファイル形式の解析情報が表示されます。



COBOL2002 Professional Tool Kit COBOL ソース解析				
プロジェクトサマリ - 販売管理システム				
2021/02/08 14:56:16				
一覧情報				
項目	全体	解析済ファイル	未解析ファイル	
COBOLソースファイル	3	3	0	
登録集原文ファイル	2	0	2	
項目	全体	使用	未使用	未登録
COBOLプログラム	3	3	0	0
登録集原文	2	0	2	0
ファイル	1	0	1	
表(DB)	0			
分布状況 対象プログラム数=3本				
分布全体表示				
項目	平均(ノプログラム)	最小	最大	
全体行数	31	28	38	
記述項数	2	0	6	
実行文の数	2	1	5	
分岐の数	0	0	0	
節数	1	1	1	
段落数	1	1	1	
呼出元プログラム数	0	0	0	

CSV ファイル形式のソース解析情報を表示する場合

1. 参照したい解析結果を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(CSV/TAG)の表示] を選択します。

CSV ファイル形式の解析情報が格納されているフォルダが表示されます。


```
[2021/08/03 14:08:53.867].[情報].[生成]生成開始: システム固有: 分岐数の分布
[2021/08/03 14:08:53.885].[情報].[生成]生成開始: システム固有: 記述項数の分布
[2021/08/03 14:08:53.902].[情報].[生成]生成開始: システム固有: 行数の分布
[2021/08/03 14:08:53.919].[情報].[生成]生成開始: システム固有: 段番号の分布
[2021/08/03 14:08:53.936].[情報].[生成]生成開始: システム固有: 呼出しプログラム数の分布
[2021/08/03 14:08:53.960].[情報].[生成]生成開始: システム固有: 呼出元プログラム数の分布
[2021/08/03 14:08:53.978].[情報].[生成]生成開始: システム固有: 登録集原文一覧
[2021/08/03 14:08:54.445].[情報].[生成]生成開始: システム固有: 未登録の登録集原文一覧
[2021/08/03 14:08:54.465].[情報].[生成]生成開始: システム固有: デッドコード情報一覧
[2021/08/03 14:08:54.502].[情報].[生成]生成開始: システム固有: COBOLソースファイル一覧CSV
[2021/08/03 14:08:54.523].[情報].[生成]生成開始: システム固有: プログラム一覧CSV
[2021/08/03 14:08:54.545].[情報].[生成]生成開始: システム固有: セクション一覧CSV
[2021/08/03 14:08:54.592].[情報].[生成]生成開始: システム固有: ファイル一覧CSV
[2021/08/03 14:08:54.611].[情報].[生成]生成開始: システム固有: データ項目一覧CSV
[2021/08/03 14:08:54.658].[情報].[生成]生成開始: システム固有: 不整合行数一覧CSV
[2021/08/03 14:08:54.673].[情報].[生成]生成開始: システム固有: 実行順序リストCSV
[2021/08/03 14:08:54.714].[情報].[生成]生成開始: システム固有: プログラム呼出関係CSV
[2021/08/03 14:08:54.731].[情報].[生成]生成開始: システム固有: 登録集原文参照関係CSV
[2021/08/03 14:08:54.746].[情報].[生成]生成開始: システム固有: ファイル参照関係CSV
[2021/08/03 14:08:54.781].[情報].[生成]生成開始: システム固有: 表(DB)参照関係CSV
[2021/08/03 14:08:54.790].[情報].[生成]生成開始: プロジェクト固有: プロジェクトサマリ
[2021/08/03 14:08:55.080].[情報].[操作]生成 コマンド終了
[2021/08/03 14:08:55.063].[情報].[実行結果]解析 生成
[2021/08/03 14:08:55.063].[情報].[実行結果]成功: 3 3
[2021/08/03 14:08:55.064].[情報].[実行結果]警告: 0 0
[2021/08/03 14:08:55.065].[情報].[実行結果]失敗: 0 0
[2021/08/03 14:08:55.065].[情報].[実行結果]合計: 3 / 3 3 / 3
[2021/08/03 14:08:56.077].[情報].[操作]解析の実行メニュー終了 プロジェクト名: 販売管理システム
[2021/08/03 14:08:56.077].[情報].[システム], コマンド終了: 終了コード=0
```

関連項目

- ・「[2.4 HTML ファイル形式のソース解析情報を参照する](#)」
- ・「[2.6 エラーがある COBOL プログラムを修正する](#)」
- ・「[2.7 CSV ファイル形式のソース解析情報を加工する](#)」
- ・「[2.8 COBOL ソース解析で操作した内容を参照する](#)」

2.4 HTML ファイル形式のソース解析情報を参照する

解析結果を参照する手順です。

作業の手順

1. 参照したい解析結果を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの「ツール」－「COBOL ソース解析」－「解析結果(HTML)の表示」を選択します。
HTML ファイル形式のプロジェクトサマリ画面 (index.html) が表示されます。

ポイント

ソース解析情報が出力されるフォルダは、「解析生成オプション設定」画面で出力先フォルダとして指定したフォルダです。

プロジェクトサマリ画面の各項目については、「7. 解析結果のリファレンス (HTML ファイル形式)」を参照してください。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ - 販売管理システム

2021/02/08 14:56:16

一覧情報			
項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	3	3	0
登録集原文ファイル	2	0	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	3	3	0	0	3
登録集原文	2	0	2	0	
ファイル	1	0	1		
表(DB)	0				

分布状況 対象プログラム数=3本

分布全体表示

項目	平均(ノプログラム)	最小	最大
全行行数	31	28	38
記述行数	2	0	6
実行文の数	2	1	5
分岐の数	0	0	0
節数	1	1	1
段落数	1	1	1
呼出元プログラム数	0	0	0

プロジェクトサマリ画面から、COBOL プログラムの次に示す情報がわかります。

[一覧情報]

解析したファイル数の内訳が「COBOL ソースファイル」「登録集原文ファイル」に分類して表示されます。

以降、項目ごとに件数が表示されます。

- [COBOL プログラム]

[全体]：解析したすべてのプログラムの個数です。[未登録] の個数は含まれません。

[使用]：主プログラム，または主プログラムから直接的もしくは間接的に呼び出されるプログラムの個数です。

[未使用]：主プログラムでないプログラムで，かつ主プログラムから直接的にも間接的にも呼び出されないプログラムの個数です。

[未登録]：CALL 文で呼出先プログラムとして指定されているが，COBOL ソースファイルがプロジェクトに登録されていないプログラムまたは登録されている COBOL ソースファイルにエラーがあって解析できていないプログラムの個数です。

[重複]：重複するプログラム名 (PROGRAM-ID) を持つプログラムの個数です。

- [登録集原文]

[全体]：解析したすべての登録集原文の個数です。

[使用]：COBOL ソースファイル中で実際に参照されている登録集原文の個数です。

[未使用]：COBOL ソースファイル中で参照されていない登録集原文の個数です。

[未登録]：COBOL ソースファイル中で参照されているが，参照されている登録集原文が見つからなくて解析されなかった登録集原文の個数です。

- [ファイル]

[全体]：解析したファイルの個数です。

- [表(DB)]

[全体]：解析した表の個数です。

リンクをクリックすると，該当する項目の一覧が表示されます。

[分布状況]

プログラムを構成する要素の個数の平均・最小・最大が表示されます。[分布全体表示] のリンクをクリックすると，分布情報が棒グラフで表示されます。

[デッドコード情報]

使用されていない可能性がある COBOL プログラムの個数が表示されます。リンクをクリックすると，該当する項目の一覧が表示されます。

関連項目

- [「7. 解析結果のリファレンス \(HTML ファイル形式\)」](#)

2.4.1 解析したプログラムの情報とソースを参照する

解析したプログラムの情報とソースを参照する手順です。

作業の手順

1. 参照したい解析結果を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択します。
HTML ファイル形式のプロジェクトサマリ画面 (index.html) が表示されます。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ - 販売管理システム 2021/02/08 14:56:16

一覧情報			
項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	3	3	0
登録集原文ファイル	2	0	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	3	3	0	0	3
登録集原文	2	0	2	0	
ファイル	1	0	1		
表(DB)	0				

分布状況 対象プログラム数=3本

[分布全体表示](#)

項目	平均(ノプログラム)	最小	最大
全体行数	31	28	38
記述項数	2	0	6
実行文の数	2	1	5
分岐の数	0	0	0
節数	1	1	1
段落数	1	1	1
呼出元プログラム数	0	0	0

4. プロジェクトサマリ画面の [一覧情報] に表示された各リンクをクリックします。

該当する項目の一覧情報が表示されます。

プログラム個別の情報とソースを参照する手順を次に示します。

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	9	1	0	0
登録集原文	12	12	1	0	
ファイル					
表(DB)					

1. プロジェクトサマリ画面で

1. プロジェクトサマリ画面で
[COBOLプログラム] 行の [全体] 列の件数をクリックすると、
[プログラム一覧] 画面が表示されます。

2. [プログラム一覧] 画面のプログラム名をクリック

2. [プログラム一覧] 画面のプログラム名をクリックすると、
[プログラム詳細] 画面の「プログラム概要」が表示されます。

The screenshot shows the 'COBOL2002 Professional Tool Kit' interface. At the top, there's a menu bar with options like 'ファイル', '編集', 'ツール', etc. Below it, a toolbar contains icons for file operations. The main window title is 'AC000-MENU'. The left sidebar has tabs for 'プロジェクト・サマリ', 'プログラム一覧', and 'プログラム詳細'. The 'プログラム詳細' tab is active, showing a tree view with 'プログラム階層図' and 'セクション一覧'. The right side of the screen displays the 'プログラム概要' (Program Overview) section, which includes fields for Program Name (AC000-MENU), Source File Name (ZMENU-画面1), Base Name (C/N入出管理システム), Execution Statements (Zメニュー画面), and other execution statistics.

3. [プログラム詳細] 画面の
右上の [ソース表示] をクリックすると、
[ソース表示 (COPY展開後)] 画面が表示されます。

ツール表示(COPY展開後) - Internet Explorer

ソース表示(COPY展開後)

プログラム名	ソースファイル名
AC000-MENU	デモニ画面.cbt

パス名 C:\入出管理システム

```

1 001000*T      デモニ画面
2 001001*      COMMENT 2
3 001002*      IDENTIFICATION
4 001010 IDENTIFICATION
5 001020 PROGRAM-ID AC000-MENU
6 001030 ENVIRONMENT
7 001070 SORT SEQUENTIAL
8 001080 SPECIAL-NAMES
9 001090 CONSOLE IS CRT
10 001100 CURSOR IS CRTA00R
11 001110 DATA
12 001120 WORKING-STORAGE
13 001130 77 CRTA00R
14 001140 77 ENT-CODE
15 001150 01 FIN-FLG
16 001160 08 CONT VALUE "I"
17 001170 08 FIN VALUE "Z"
18 001180 BOREN
19 001190 01 GYOWNG-ENTRY
20 001200 05 TITLE
21 001210 10 LINE 2 COLUMN 5 UNDERLINE
22 001220 05 OS CODE-LINE
23 001230 10 LINE 5 COLUMN 4 VALUE "業務デモニ"
24 001240 10 LINE 5 COLUMN 4 VALUE "業務"
25 001250 10 LINE 5 COLUMN 10 FOREGROUND-COLOR 1
26 001260 10 LINE 5 COLUMN 10 USING ENT-CODE 1
27 001270 10 LINE 5 COLUMN 10 USING ENT-CODE 2
  
```

4. [プログラム詳細] 画面の「セクション一覧」の
[実行順序] 列の [表示] をクリックすると、
節のソースコードの実行順序を示す
「実行順序表示」画面が表示されます。

実行順序表示 - Internet Explorer

【実行順序】列の「表示」
節のソースコードの実行
【実行順序表示】画面が

実行順序表示		
プログラム名	ソースファイル名	メニュー画面id
プログラム名	ACCO00.MENU	
パス名	C:\入出庫管理システム	
節名	ACCO00-MENU-MAIN	
節コメント	メニュー画面	

```

ACCO00-MENU-MAIN_SECTION.
ACCO00-MENU-MAIN-001.
MOVE "P" TO FIN-FLG
PERFORM UNTIL FIN
  DISPLAY SPGE
:
  EVALUATE FIN-CODE
  WHEN "01"
    CALL "RC34D_INPUT"
  WHEN "02"
    CALL "RC34D_UPD"
  WHEN "03"
    CALL "RC34D_SND"
  WHEN "99"
    MOVE "9" TO FIN-FLG
  WHEN OTHER
    CONTINUE
END-PERFORM
CONTINUE.
ACCO00-MENU-MAIN-999.
STOP RUN ---- (プログラム終了)
        
```

```

35: 001350 ACCO00-MENU-MAIN SECTION.
36: 0013600 MOVE "P" TO FIN-FLG
37: 001370 ACCO00-MENU-MAIN-001.
38: 001380 MOVE "1" TO FIN-FLG
39: 001390 PERFORM UNTIL FIN
40: 001400 DISPLAY SPGE
41: 001410 END-DISPLAY
42: 001420 DISPLAY GROUND-ENTRY
43: 001430 WHEN "01"
44: 001440 MOVE 0518 TO CRTADDR
45: 001450 ACCEPT GROUND-ENTRY
46: 001460 EVALUATE END-CODE
47: 001470 WHEN "01"
48: 001480 CALL "RC34D_INPUT"
49: 001490 WHEN "02"
50: 001500 CALL "RC34D_UPD"
51: 001510 WHEN "03"
52: 001520 CALL "RC34D_SND"
53: 001530 WHEN "99"
54: 001540 MOVE "9" TO FIN-FLG
55: 001550 WHEN OTHER
56: 001560 CONTINUE
57: 001570 END-EVALUATE
58: 001580 END-PERFORM
59: 001590 CONTINUE.
60: 001600 ACCO00-MENU-MAIN-999.
61: 001610 STOP RUN.
        
```

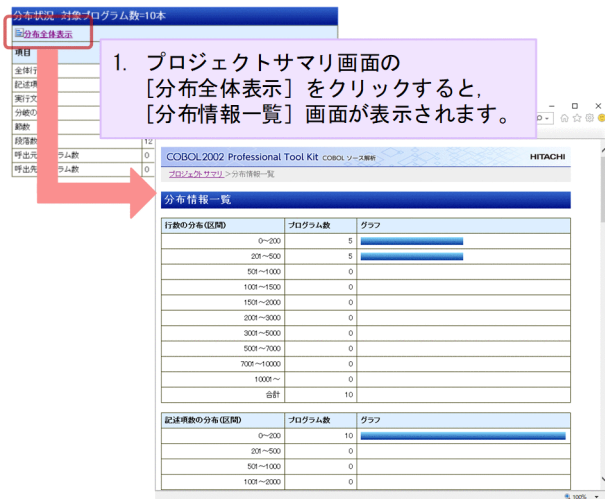
2.4.2 各種の分布情報を参照する

解析した COBOL ソースファイルの分布情報を参照する手順です。

2. COBOL ソース解析の基本的な使い方

作業の手順

1. 参照したい解析結果を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの【ツール】－【COBOL ソース解析】－【解析結果(HTML)の表示】を選択します。
HTML ファイル形式のプロジェクトサマリ画面 (index.html) が表示されます。
4. プロジェクトサマリ画面の【分布全体表示】をクリックします。
【分布情報一覧】画面が表示されます。分布情報一覧を参照する手順を次に示します。



分布情報から、解析したプログラムの規模と複雑性がわかります。

2.4.3 使われていないソースコードの候補を確認する

プログラム実行時に実行される可能性がない文、プログラムなど、プログラムの実行結果に影響を与えないソースコードのことをデッドコードといいます。

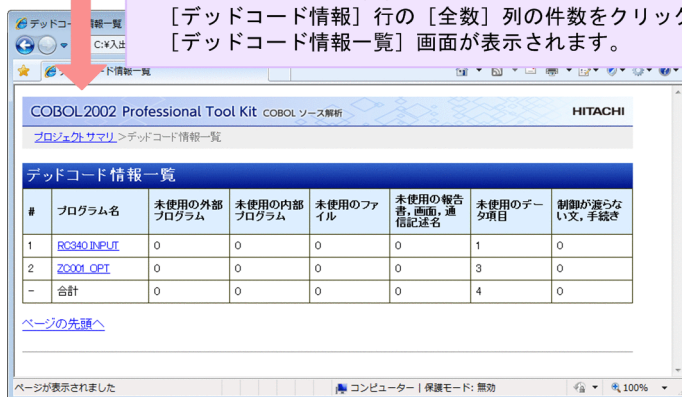
デッドコードの候補から、テストが不要なプログラムの選定や、プログラムの処理の誤りを発見できます。

作業の手順

1. 参照したい解析結果を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの【ツール】－【COBOL ソース解析】－【解析結果(HTML)の表示】を選択します。
HTML ファイル形式のプロジェクトサマリ画面 (index.html) が表示されます。
4. プロジェクトサマリ画面の【デッドコード】行の【全体】列の件数をクリックします。
デッドコード候補の一覧が表示されます。デッドコード情報の一覧を参照する手順を次に示します。

デッドコード情報 対象プログラム数=4本							
項目	全数	未使用の外部プログラム	未使用の内部プログラム	未使用のファイル	未使用の報告書、画面、通信記述名	未使用のデータ項目	制約が渡らない文、手続き
デッドコード	4	0	0	0	0	4	0

1. プロジェクトサマリ画面で
[デッドコード情報] 行の [全数] 列の件数をクリックします。
[デッドコード情報一覧] 画面が表示されます。



使われていないソースコードの候補を COBOL ソース解析が検知する方法については、「10.6 デッドコード候補の検知方法」を参照してください。

2.5 解析前に前回のソース解析情報を残す

ソース解析情報は、[解析生成オプション設定] 画面の [生成ドキュメント] タブで指定した出力先に出力されます。

解析する条件を変更する場合に、前の解析結果を残すときは、いったん出力先のフォルダを退避して解析を実行するか、[解析生成オプション設定] 画面の [生成ドキュメント] タブの出力先を変更してください。

■ 参考

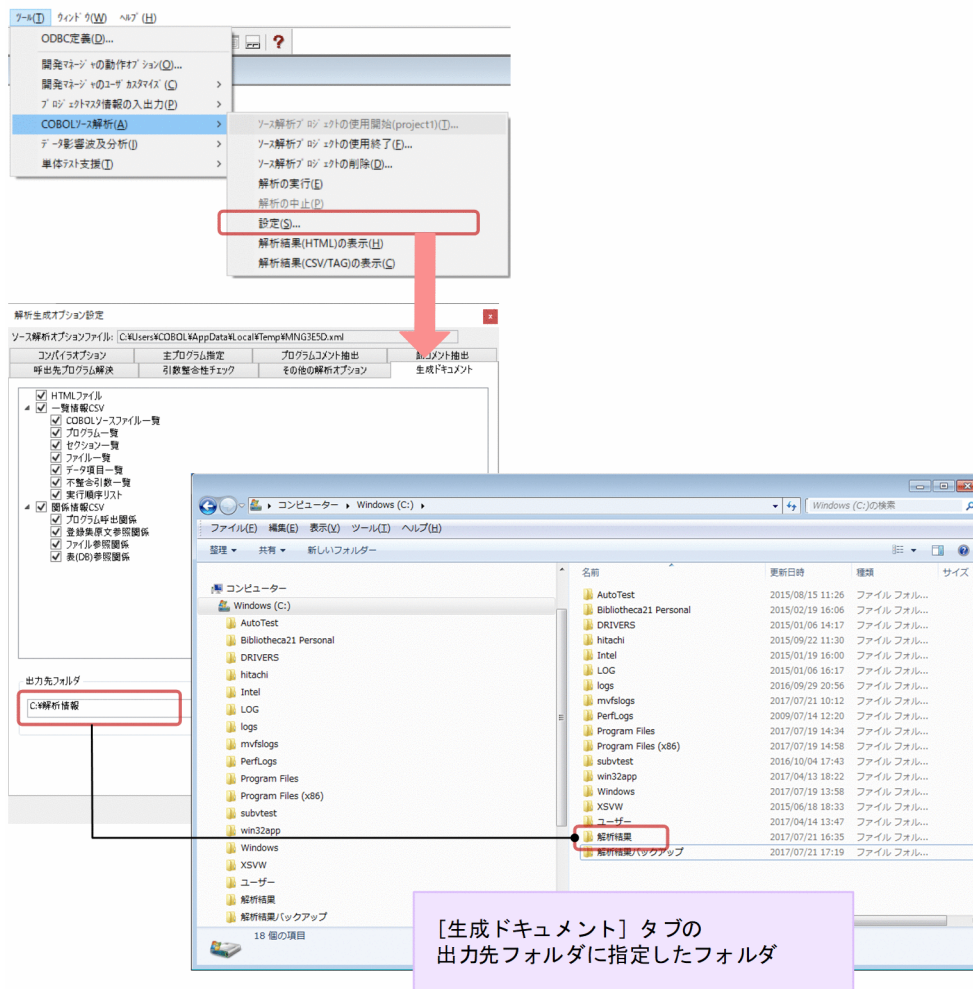
COBOL ソース解析の解析環境（環境設定情報）を保存する場合は、組み込みデータベースのコマンドを使用します。環境設定情報の保存と読み込みについては、マニュアル「COBOL2002 Professional 製品 導入ガイド」の解析環境の操作に関する説明を参照してください。

2.5.1 ソース解析情報のフォルダを退避する

ソース解析情報を退避する手順です。

作業の手順

1. 退避したいソース解析情報を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
4. [生成ドキュメント] タブを選択し、指定した出力先フォルダを確認します。



5. エクスプローラーで、出力先フォルダをコピーして、ソース解析情報をほかの場所に保存します。
6. 開発マネージャに戻り、COBOL ソース解析を実行します。

2.5.2 ソース解析情報の出力先を変更する

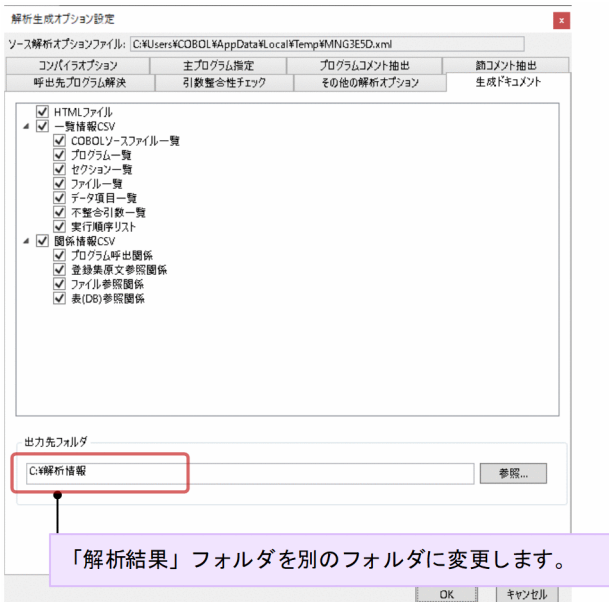
ソース解析情報の出力先を変更する手順です。

作業の手順

1. 出力先を変更したいソース解析情報を含むプロジェクトマスタを開発マネージャで開きます。
2. ソース解析プロジェクトの使用を開始します。
3. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。



4. [生成ドキュメント] タブを選択し、[出力先フォルダ] を変更します。



5. 開発マネージャに戻り、COBOL ソース解析を実行します。

2.6 エラーがある COBOL プログラムを修正する

COBOL ソース解析のログ情報（解析ログ、操作ログ）のうち、解析ログを参照して COBOL プログラムを修正する手順です。

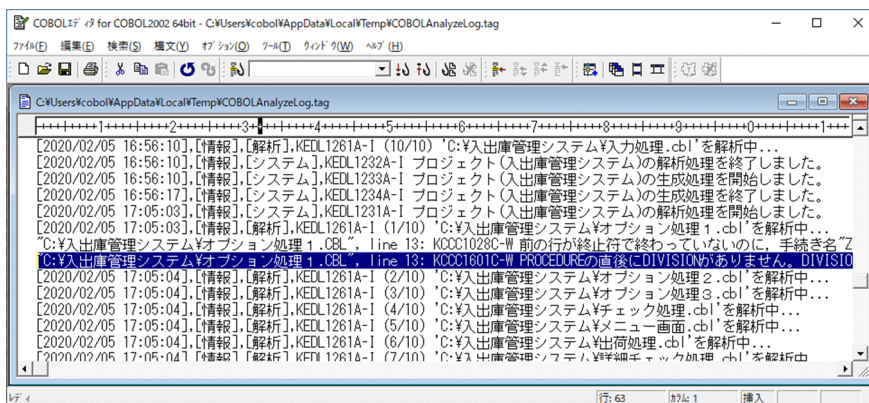
COBOL ソースファイルの解析時に COBOL2002 のコンパイルエラーが発生すると、解析処理がエラーになります。コンパイルエラーのレベルに応じて、実行結果は解析失敗または解析警告になります。解析ログには、COBOL コンパイラが出力したコンパイルエラー情報が出力されます。

解析ログはタグファイルのため、COBOL エディタで開きます。

作業の手順

1. 次に示す解析ログ（COBOLAnalyzeLog.tag）を選択します。

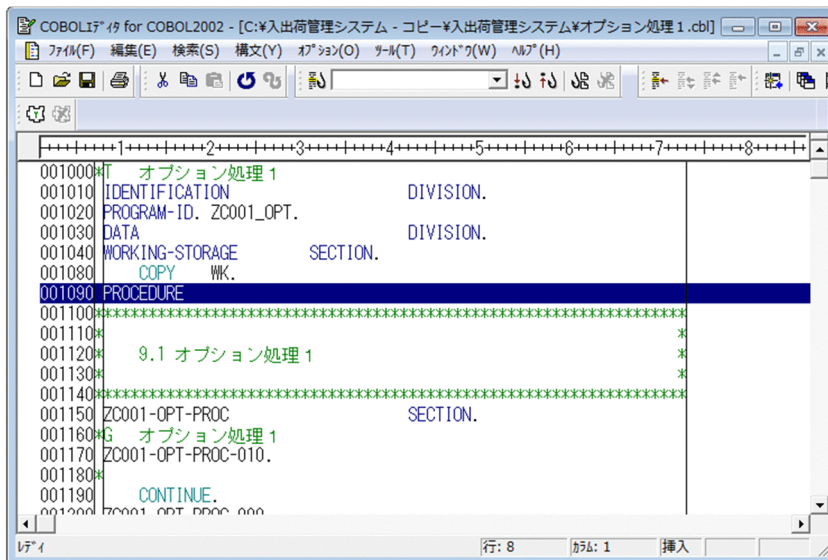
Windows の環境変数 *TEMP* で定義されているフォルダ¥COBOLAnalyzeLog.tag
解析ログの情報が COBOL エディタに表示されます。



2. 解析ログのエラーメッセージをダブルクリックします。

COBOL ソースファイルが表示され、該当行にカーソルが移動します。

3. COBOL エディタで該当する個所を修正します。



この例では、手続き部に「DIVISION.」がないことが警告の原因なので、「DIVISION.」を入力します。上書き保存して、COBOL ソースファイルを閉じます。

注意事項

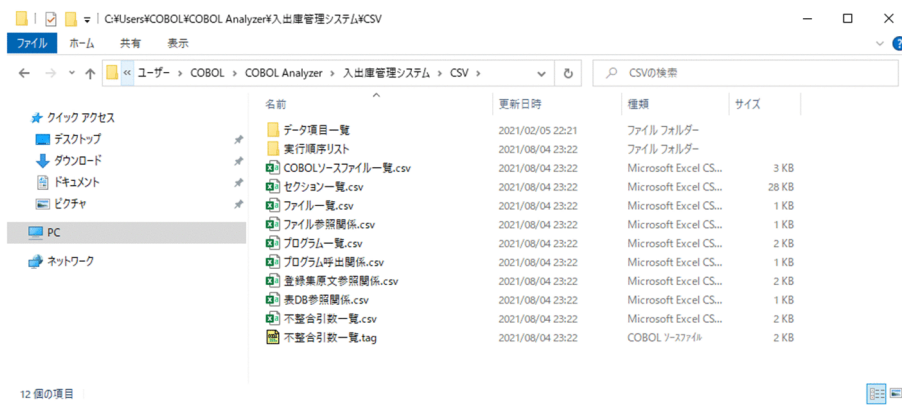
解析ログを表示したあとで、再度 COBOL ソース解析を実行する場合は、いったん解析ログを閉じてください。解析ログを開いたまま COBOL ソース解析を実行すると、解析ログが正しく取得できません。

2.7 CSV ファイル形式のソース解析情報を加工する

CSV ファイル形式のソース解析情報を使用する例として、CSV ファイルを加工する手順です。

作業の手順

1. 加工したいソース解析情報を含むプロジェクトマスタを開発マネージャで開きます。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(CSV/TAG)の表示] を選択します。
CSV ファイル形式のソース解析情報 (CSV ファイルおよびタグファイル) が格納されたフォルダが表示されます。



ポイント

ソース解析情報が出力されるフォルダは、[解析生成オプション設定] 画面の [生成ドキュメント] タブで出力先フォルダとして指定したフォルダです。

3. ExcelなどでCSV ファイルの内容を確認します。
4. CSV ファイルを入力として、任意の開発環境で加工して活用します。
それぞれのCSV ファイルの用途を次に示します。

一覧情報 CSV

CSV ファイル名と格納されている情報	用途
COBOL ソースファイル一覧.csv プロジェクトに登録された COBOL ソースファイルと登録集原文ファイルの一覧	<ul style="list-style-type: none">プロジェクト管理者が資産の棚卸しで、COBOL ソースファイルと登録集原文の本数からシステムの規模を把握する。プログラマーまたはテスト担当者が、COBOL ソースファイル単位にシステムに影響しない COBOL ソースファイルや登録集原文をシステム改修作業やテスト対象から除外する。
プログラム一覧.csv 解析が成功した COBOL ソースファイルに含まれるプログラムの一覧	<ul style="list-style-type: none">プロジェクト担当者が、プロジェクト内のプログラム数、使用プログラムや未使用プログラム数から登録したプロジェクトの稼働率を把握する。プロジェクト管理者が、プログラムの [実行文の数]、[分岐の数]、[呼出先プログラム数] などの情報から、プログラムを修正するために掛かる時間を見積もる。

CSV ファイル名と格納されている情報	用途
	<ul style="list-style-type: none"> プログラマーまたはテスト担当者が、行数情報からプログラムの規模を把握する。 プロジェクト担当者が共通サブルーチン化を検討するために、プロジェクト内で重複しているプログラム名を把握する。 SI 担当者またはプログラム管理者/担当者が、プログラム名の指定に重複するプログラムがないかを事前にチェックする。 プロジェクト担当者が資産の棚卸しで、コピーまたは新規で作成する COBOL ソースファイルを整理する。
セクション一覧.csv プロジェクトのプログラムの手続き部に記述されている節や段落の一覧	<ul style="list-style-type: none"> プロジェクト管理者/担当者が節や段落の粒度で使用状況や規模を把握する。
ファイル一覧.csv プロジェクトのプログラムで宣言されているファイル名と入出力文情報の一覧	<ul style="list-style-type: none"> プロジェクト管理者/担当者がプロジェクトのファイルの使用状況を把握する。 プロジェクト管理者/担当者がファイルの CRUD 情報作成のための素材データとして使用する。
データ項目一覧_プログラム名.csv プログラムで定義されているデータ項目の一覧	<ul style="list-style-type: none"> プロジェクト管理者またはプロジェクト担当者がプログラムデータの規模や使用状況を把握する。 プロジェクト担当者がデバッグでデータ項目の属性、サイズや位置を把握する。
不整合引数一覧.csv CALL 文によるプログラム呼び出しで引数の個数、受渡方法、データ属性やサイズが一致していない実引数と仮引数の情報の一覧	<ul style="list-style-type: none"> プロジェクト管理者またはプロジェクト担当者がシステムの更改や移植の契機で修正が必要な引数不整合の規模を把握する。 プロジェクト担当者がテストや障害調査で引数に問題がないことを検証する。
実行順序リスト.csv プログラムの節見出し、段落見出し、実行文、文の条件指定の実行順序をツリー形式で表現したもの	<ul style="list-style-type: none"> プロジェクト担当者が開発前の母体調査や障害の原因調査で効率良くかつ効果的にソースコードリーディングを実施するために使用する。

関係情報 CSV

CSV ファイル名と格納されている情報	用途
プログラム呼出関係.csv プログラムごとの、呼び出し元と呼び出し先のプログラムの関係	<ul style="list-style-type: none"> プロジェクト担当者が、特定のプログラムから直接または間接的に呼び出しているプログラムの一覧を確認する。 プロジェクト担当者が、あるプログラムを直接または間接的に呼び出しているプログラムを調査する。
登録集原文参照関係.csv COBOL ソースファイルと登録集原文内で参照する登録集原文の関係の一覧	<ul style="list-style-type: none"> プロジェクト担当者が、特定のプログラムから直接または間接的に COPY 文で指定している登録集原文の一覧を確認する。 プロジェクト担当者が、ある登録集原文を直接または間接的に参照しているプログラムや登録集原文を調査する。
ファイル参照関係.csv プログラムごとの、プログラム内で使用するファイルの関係の一覧	<ul style="list-style-type: none"> プロジェクト担当者が特定のファイルを使用するプログラムを把握する。 プロジェクト担当者が特定のプログラム内で使用しているファイルを把握する。

CSV ファイル名と格納されている情報	用途
表 DB 参照関係.csv プログラムごとの、プログラム内で使用するデータベースの表の参照関係	<ul style="list-style-type: none"> プロジェクト担当者が特定の表を使用するプログラムを把握する。 プロジェクト担当者が特定のプログラム内で使用している表を把握する。

関連項目

- ・「8. 解析結果のリファレンス（CSV ファイル形式）」

2.8 COBOL ソース解析で操作した内容を参照する

COBOL ソース解析のログ情報（解析ログ、操作ログ）のうち、操作ログを参照して COBOL ソース解析の操作履歴を参照する手順です。

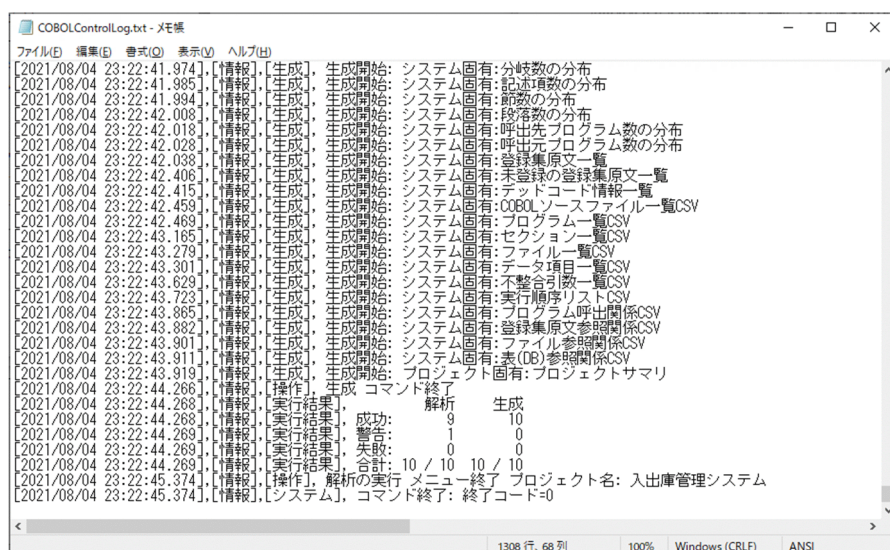
操作ログを参照すると、COBOL ソース解析の操作でエラーになった個所を追跡できます。

作業の手順

1. 次を示す操作ログ（COBOLControlLog.txt）を開きます。

Windows の環境変数 TEMP で定義されているフォルダ¥COBOLControlLog.txt

操作ログの情報が拡張子 txt に関連づけられたエディタに表示されます。



2. 操作ログを参照し、エラーが起こった原因を調査します。

関連項目

- ・「9. エラー時の対処」

3

業務目的別で使う

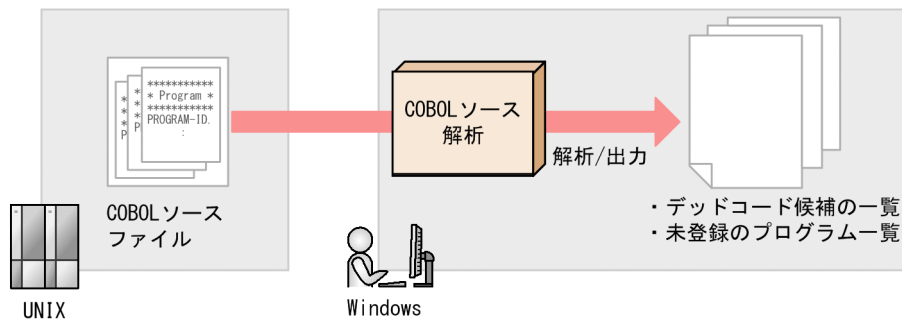
COBOL ソース解析の操作例について説明します。

3.1 COBOL 資産を棚卸する

UNIX 環境などのプラットフォームで使用していた COBOL ソースファイルを Windows 環境で使う場合、プログラムの保守書が整備されていないと、不要な COBOL ソースファイルが混在していないか、必要なソースファイルが不足していないかの調査に時間が掛かります。

COBOL ソース解析を使用すると、COBOL ソースファイル群と関係ない COBOL ソースファイルや、不足している COBOL ソースファイルの一覧をソース解析情報として生成できます。

図 3-1 COBOL 資産を棚卸する



注
UNIX環境からWindowsへのCOBOLソースファイルの転送には、FTPなどのファイル転送プログラムを使用します。

作業の手順

1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。
4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
5. コンパイラオプション、解析結果の出力先フォルダなどを設定します。
6. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析の実行] を選択します。
7. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択します。
HTML ファイル形式の解析情報が表示されます。

プロジェクトサマリ - 入在庫管理システム

2015/04/10 14:18:18

項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	10	10	0
登録集原文ファイル	14	12	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	10	0	0	0
登録集原文	14	12	2	0	0
ファイル	3	3	0	0	0
表(DB)	3				

分布状況 対象プログラム数=10本

[分布全体表示](#)

項目	平均(ノプログラム)	最小	最大
全行行数	155	21	335
記述項数	48	4	102

COBOL ソースファイル群と関係ない COBOL ソースファイルの件数を知りたいとき

[未使用] 列の件数を確認します。主プログラムでないプログラムで、どのプログラムからも呼び出されていないプログラムを確認するときは、「デッドコード情報」の [デッドコード] 行の [全数] 列をクリックします。

COBOL ソースファイル群に不足している COBOL ソースファイルを知りたいとき

[未登録] 列の件数をクリックします。[未登録のプログラム一覧] 画面が表示されます。未登録プログラムとは、プログラムから呼び出されているが、COBOL ソース解析の解析対象として登録されていないプログラムです。

未登録のプログラム一覧

#	プログラム名	引数の個数	呼出元プログラム名
1	CBLADTRM	2	ADDTERM2

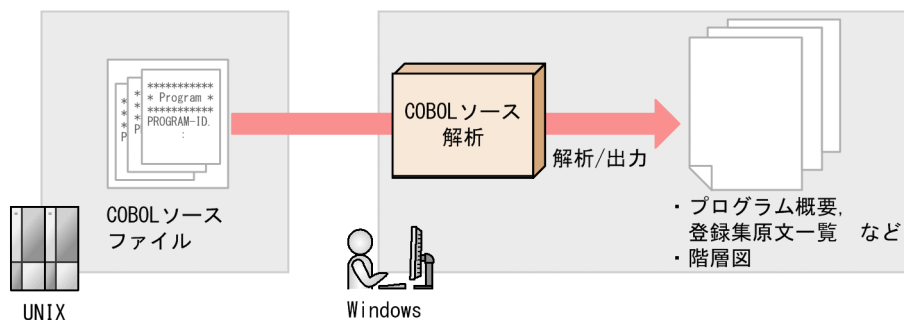
[ページの先頭へ](#)

3.2 プログラムの処理を調査する

UNIX 環境などのプラットフォームで使用していた COBOL ソースファイルの処理を調査する場合、プログラムの保守書が整備されていないと、COBOL ソースの調査に時間が掛かります。

COBOL ソース解析を使用すると、プログラム名、作業場所節データ数などをソース解析情報として出力できます。また、プログラムの処理の概要を視覚的に理解できるように、プログラムの階層図を出力できます。

図 3-2 プログラムの処理の概要を調査する



注
UNIX環境からWindowsへのCOBOLソースファイルの転送には、FTPなどのファイル転送プログラムを使用します。

作業の手順

1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。
4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
5. コンパイラオプション、解析結果の出力先フォルダなどを設定します。
6. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析の実行] を選択します。
7. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択します。
HTML ファイル形式の解析情報が表示されます。

プロジェクトサマリ - 入在庫管理システム

2015/04/10 14:18:18

一覧情報

項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	10	10	0
登録集原文ファイル	14	12	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	10	0	0	0
登録集原文	14	12	2	0	0
ファイル	3	3	0		
表(DB)	3				

分布状況 対象プログラム数=10本

分布全体表示

項目	平均(プログラム)	最小	最大
全行行数	155	21	335
記述行数	48	4	102

[COBOL プログラム] 行の [全体] 列の件数をクリックすると、[プログラム一覧] 画面が表示されます。

プログラム一覧

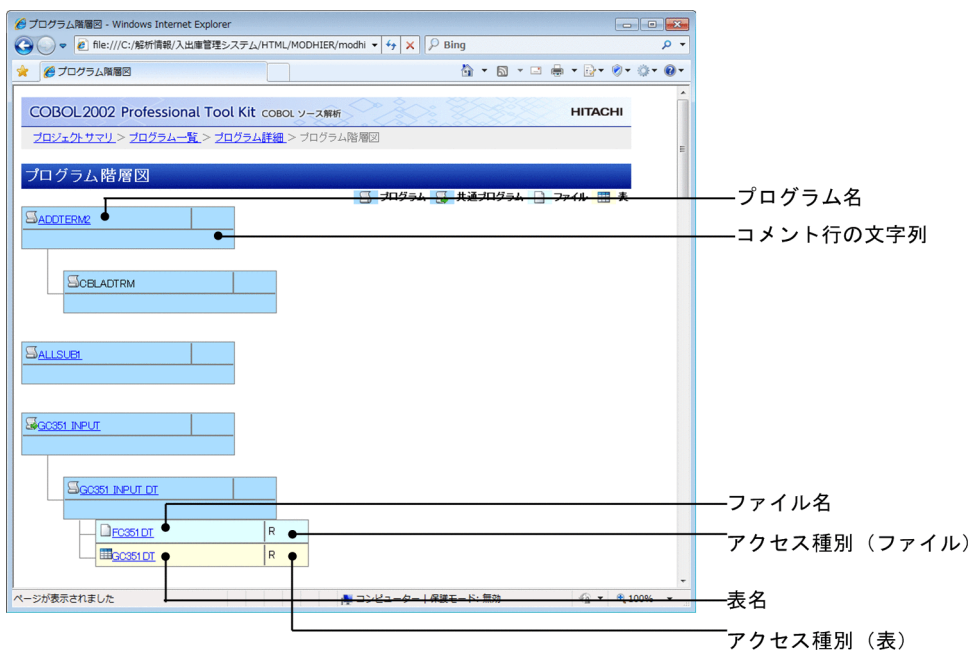
プロジェクトサマリ > プログラム一覧

#	プログラム名	使用有無	総行数	引数の個数	実行文の数	分岐の数	呼出先プログラム数	コメント行の文字列
1	AC000.MENU	使用	61	0	14	7	3	
2	AC00TERM	使用	94	0	24	10	1	
3	ALLUSER	使用	36	0	8	2	0	
4	DISPLAY	使用	24	0	2	0	0	
5	GC351.INPUT	使用	115	2	45	18	1	
6	GC351.INPUT.DT	使用	211	2	55	16	0	
7	MAIN	使用	69	0	24	0	3	
8	RC340.INPUT	使用	210	2	68	16	2	
9	RC340.SET	使用	335	1	107	31	1	
10	RC340.SND	使用	283	0	76	22	1	
11	RC340.UPD	使用	269	0	77	24	1	
12	SUER	使用	20	3	2	0	0	
13	SUER2	使用	14	3	2	0	0	
14	SUER3	使用	13	3	2	0	1	
15	SUER4	使用	11	1	1	0	0	

[プログラム一覧] 画面で、任意のプログラム名をクリックすると、[プログラム詳細] 画面が表示されます。[プログラム詳細] 画面では、プログラム名、コメント行の文字列、作業場所節データ数などが表示されるので、プログラムの処理の概要を調査できます。



[プログラム詳細] 画面でプログラム階層図へのリンクをクリックすると、[プログラム階層図] 画面が表示されて、視覚的にプログラムの構造を調査できます。

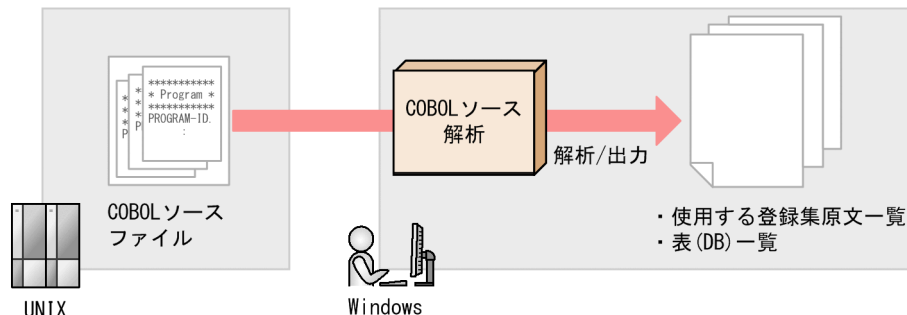


3.3 プログラムの入出力情報を調査する

UNIX 環境などのプラットフォームで使用していた COBOL ソースファイルの入出力情報を調査する場合、プログラムの保守書が整備されていないと、データベースの SQL 表を参照しているプログラムは幾つあるか、登録集原文とプログラムの関連から登録集原文を変更したときの影響を調査するのに時間が掛かります。

COBOL ソース解析を使用すると、SQL 表にアクセスしているプログラムの一覧、登録集原文を使用しているプログラムの一覧などをソース解析情報として出力できます。

図 3-3 入出力情報の影響を調査する



注
UNIX環境からWindowsへのCOBOLソースファイルの転送には、FTPなどのファイル転送プログラムを使用します。

作業の手順

1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。
4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
5. コンパイラオプション、解析結果の出力先フォルダなどを設定します。
6. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析の実行] を選択します。
7. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択します。
HTML ファイル形式の解析情報が表示されます。

プロジェクトサマリ - Windows Internet Explorer

C:\解析情報\出入庫管理システム\HTML\index.html

プロジェクトサマリ

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ - 入出庫管理システム

2015/04/10 14:18:18

一覧情報

項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	10	10	0
登録集原文ファイル	14	12	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	10	0	0	0
登録集原文	14	12	2	0	
ファイル	3	3	0		
表(DB)	3				

分布状況 対象プログラム数=10本

分布全体表示

項目	平均(プログラム)	最小	最大
全体行数	155	21	335
記述項数	48	4	102

ページが表示されました

登録集原文を使用しているプログラムの一覧を知りたいとき

[登録集原文] 行の [全体] 列の件数をクリックして、[登録集原文一覧] 画面を表示します。

登録集原文一覧 - Windows Internet Explorer

C:\解析情報\出入庫管理システム\HTML\copylist.html

登録集原文一覧

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ > 登録集原文一覧

登録集原文一覧

#	登録集原文名	使用有無	COPY元プログラム数	パス名
1	AOCTOD	使用	1	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
2	CBUXMLRC	未使用	0	C:\Program Files\Hitachi\COBOL2002\WCOPY
3	FC351.DT	使用	1	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
4	FC370	使用	3	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
5	FC380	使用	1	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
6	FGC301	使用	1	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
7	GC351	使用	4	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
8	GC351.DT	使用	1	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
9	GC351.ERR	使用	4	C:\入出庫管理システム\入出庫管理システム\WCOPY登録集
10	KEYLIST	使用	1	C:\Program Files\Hitachi\COBOL2002\SAMPLES\COBOL2002\WADDERM2

ページが表示されました

[登録集原文一覧] 画面で、任意の登録集原文名をクリックすると、[登録集原文詳細] 画面が表示されます。「COPY 元プログラム」とは、登録集原文を使用しているプログラムの一覧です。



SQL 表にアクセスしているプログラムの一覧を知りたいとき

[表(DB)] 行の [全体] 列の件数をクリックして、[表(DB)一覧] 画面を表示します。



[表(DB)一覧] 画面で、任意の表名をクリックすると、[表(DB)詳細] 画面が表示されます。



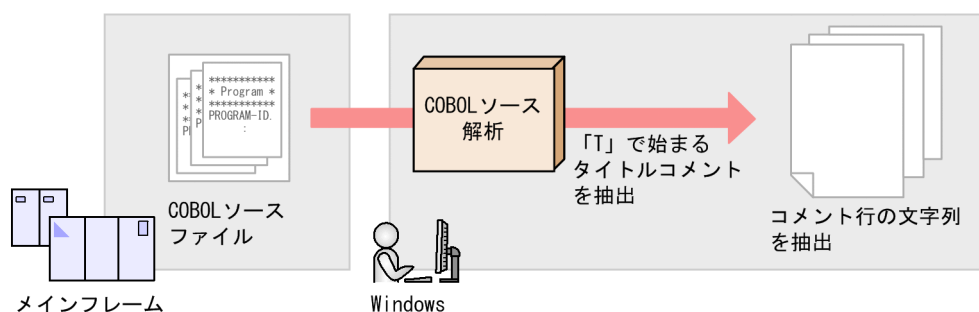
3.4 コメント行の文字列を抽出する

メインフレームで使用していた COBOL ソースファイルは、自動生成の規則に沿ってコーディングされている場合があります。COBOL ソース解析では、条件を付けてコメント文を抽出できます。なお、ここで説明する自動生成のコメント規則とは、次のとおりです。

表 3-1 自動生成のコメント規則

機能コメント	種類	概要
T	タイトルコメント	プログラム名を記述します。
D	データ定義コメント	データセグメント名を記述します。また、データセグメントの範囲を記述します。
0		データの意味、および使い方を記述します。
1		
2		
3		
P	プログラムコメント	プログラムの機能の概要を記述します。
G	処理部コメント	セクションの処理概要を記述します。また、処理のまとまりを記述します。
L		
S0	制御コメント	プログラムおよびセクション間の制御の流れを記述します。また、プログラム関連図での表示範囲を記述します。
S1		
S2		
S3		
U	ユーザーコメント	自動生成ツールから、ユーザー独自の日本語変換ルーチンへ渡す情報を記述します。

図 3-4 自動生成したタイトルコメントを抽出する



注

メインフレーム環境からWindowsへのCOBOLソースファイルの転送には、FTPなどのファイル転送プログラムを使用します。

作業の手順

1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。
4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
5. [解析生成オプション設定] 画面で抽出条件に T を指定します。
[解析生成オプション設定] 画面の [プログラムコメント抽出] タブで、[プログラムコメントを抽出する] にチェックします。[抽出位置] - [プログラムコメント行の検索条件] - [抽出キー文字列] が活性化されるので、空欄に半角英字の「T」を入力します。

解析生成オプション設定

ソース解析オプションファイル: C:\Users\%COBOL%\AppData\Local\Temp\%MNG59E6.xml

呼出先プログラム解決 コンパイルオプション	引数整合性チェック 主プログラム指定	その他の解析オプション プログラムコメント抽出	生成ドキュメント 節コメント抽出
<input checked="" type="checkbox"/> プログラムコメントを抽出する			

抽出位置

プログラムコメント行の検索場所

☒ プログラム名脱落の前
☐ プログラム名脱落の後

プログラムコメント行の検索条件

☒ 抽出キー文字列で始まるコメント行 (プログラムコメントは抽出キーの後の文字列)
☐ 抽出キー文字列で始まるコメント行の次行
☐ プログラム名脱落から指定行目の位置にあるコメント行

抽出キー文字列: T

0 行目(1~99)

OK キャンセル

6. コンパイラオプション、解析結果の出力先フォルダなどを設定します。
7. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析の実行] を選択します。
8. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(HTML)の表示] を選択します。
HTML ファイル形式の解析情報が表示されます。

プロジェクトサマリ - 入在庫管理システム

2015/04/10 14:18:18

一覧情報

項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	10	10	0
登録集原文ファイル	14	12	2

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	10	0	0	0
登録集原文	14	12	2	0	0
ファイル	3	3	0		
表(DB)	3				

分布状況 対象プログラム数=10本

分布全体表示

項目	平均(プログラム)	最小	最大
全体行数	155	21	335
記述項数	48	4	102

[COBOL プログラム] 行の [全体] 列の件数をクリックすると、[プログラム一覧] 画面が表示されます。T で始まるコメント欄の内容が表示されます。ただし、T は表示されません。

プログラム一覧

プロジェクトサマリ > プログラム一覧

#	プログラム名	使用有無	総行数	引数の個数	実行文の数	分岐の数	呼出先プログラム数	コメント行の文字列
1	AO000.MENU	使用	64	0	14	7	3	メニュー画面
2	GC351.INPUT	使用	115	2	45	18	1	チェック処理
3	GC351.INPUT.DT	使用	211	2	55	16	0	詳細チェック処理
4	BC340.INPUT	使用	218	2	68	16	2	入力処理
5	BC340.SET	使用	335	1	107	31	1	転送処理
6	BC340.SND	使用	283	0	76	22	1	出荷処理
7	BC340.LPD	使用	269	0	77	24	1	制動処理
8	ZC001.CPT	使用	22	0	2	0	0	
9	ZC002.CPT	使用	24	1	2	0	0	
10	ZC003.CPT	使用	21	1	2	0	0	
-	合計	-	1562	-	448	134	9	-

ページの先頭へ

[プログラム一覧] 画面で表示されるプログラム一覧から任意のプログラム名をクリックすると、[プログラム詳細] 画面が表示されます。[プログラム詳細] 画面の [コメント行の文字列] に、T で始まるコメント欄の内容が表示されます。ただし、T は表示されません。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ > プログラム一覧 > プログラム詳細

プログラム階層図 セクション階層図 ソース表示

プログラム概要

プログラム名	AC000,MENU		
COBOLソースファイル名	メニュー画面 cob		
パス名	C¥入出庫管理システム		
コメント行の文字列	メニュー画面		
総行数	64	有効行数	51
コメント行数	13	登録集	0
参照する登録集原文数	0	ファイル数	0
データ項目数	4	引数の個数	0
分岐の数	7	節数	1
表(DD)数	0	呼出元プログラム数	0
未使用の内部プログラム数	0	呼出先プログラム数	3
制約が渡らない文、手続きの数	0	未使用のデータ項目数	0
		未使用の報告書、画面、通信記述名の数	0

呼出元プログラム一覧

#	呼出元プログラム名
---	-----------

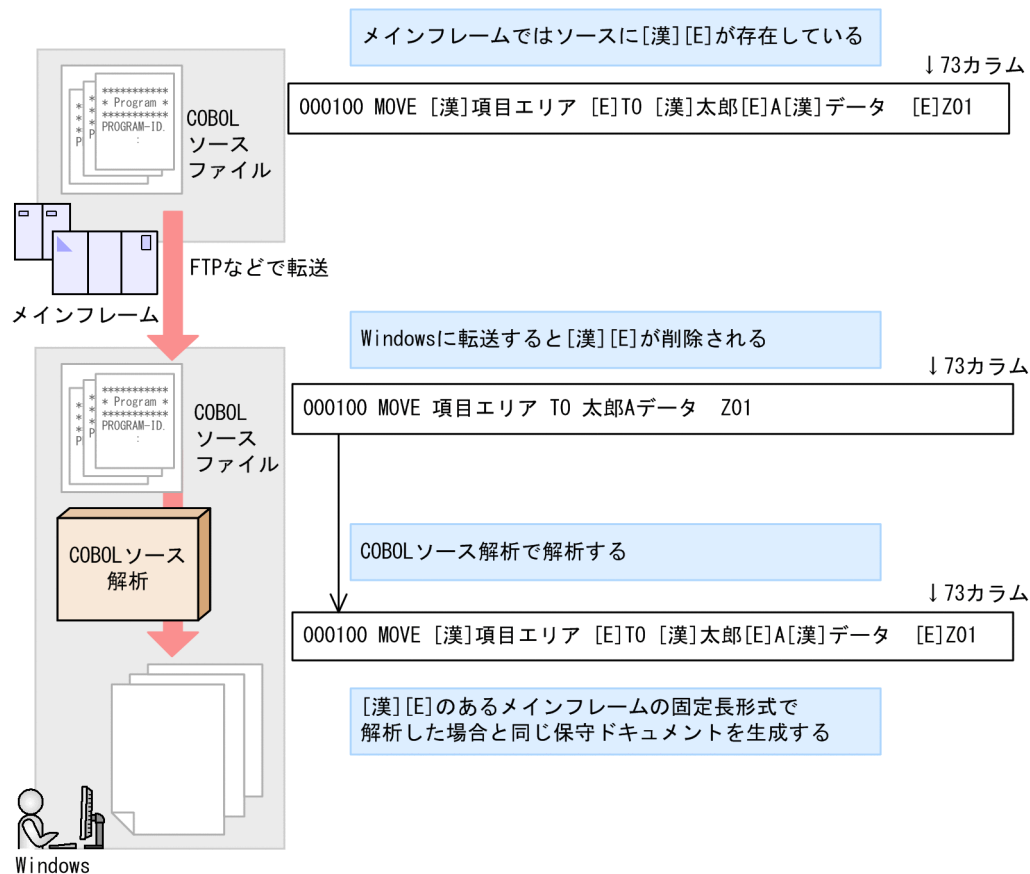
呼出先プログラム一覧

3.5 メインフレームの固定長形式がある COBOL ソースファイルを解析する

メインフレーム向けに開発した COBOL ソースファイルでは、機能キャラクタによって記述がずれることがあります。メインフレームの固定長形式の機能キャラクタを考慮して、COBOL ソースファイルを解析できるように修正する必要があります。

COBOL ソース解析では、コンパイラオプションを設定して、メインフレーム向けの COBOL ソースファイルを解析できます。COBOL ソースファイルを修正する必要はありません。

図 3-5 メインフレームの固定長形式の COBOL ソースファイルを解析する



(凡例)

- [漢]: 機能キャラクタ (X'0A42') を示します。
- [E]: 機能キャラクタ (X'0A41') を示します。

作業の手順

1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。

4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。

[解析生成オプション設定] 画面が表示されます。

5. [解析生成オプション設定] 画面で [-CompatiV3 -V3Rec,Fixed] コンパイラオプションを指定します。

[解析生成オプション設定] 画面の [コンパイラオプション] タブで, [オプション・環境変数を設定する] を選択します。[コンパイラオプションの設定] が活性化されるので, テキストボックスに [-CompatiV3 -V3Rec,Fixed] を入力します。

解析生成オプション設定

ソース解析オプションファイル: C:\Users\COBOL\AppData\Local\Temp\MNGA87F.xml

呼出先プログラム解決	引数整合性チェック	その他の解析オプション	生成ドキュメント
コンパイラオプション	主プログラム指定	プログラムコメント抽出	読コメント抽出

☐ 開発マネージャの設定を引き継ぐ

☒ オプション・環境変数を設定する

コンパイラオプションの設定

-CompatiV3 -V3Rec,Fixed

開発マネージャのコンパイラオプション設定内容

-NumAccept
-DebugLine

コンパイラ環境変数の設定

CBLCOPT2002=-StdVersion.1 -DebugData -Switch,EBCDIK
CBLLIB=c:\販売管理システム04\伝票作成\work\C:\Program Files\Hitachi\COBOL2002 64bit\copy
CBLPIDIR=c:\販売管理システム04\伝票作成\work

開発マネージャの環境変数設定内容

CBLCOPT2002=-StdVersion.1 -DebugData -Switch,EBCDIK
CBLLIB=c:\販売管理システム04\伝票作成\work\C:\Program Files\Hitachi\COBOL2002 64bit\copy
CBLPIDIR=c:\販売管理システム04\伝票作成\work

OK キャンセル

6. 解析結果の出力先フォルダなどを設定します。

7. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析の実行] を選択します。

機能キャラクタ (X'0A42'／X'0A41') があるメインフレーム形式として解析された結果が, ソース解析情報に出力されます。

3.6 プログラム呼び出しの実引数と仮引数の整合性をチェックする

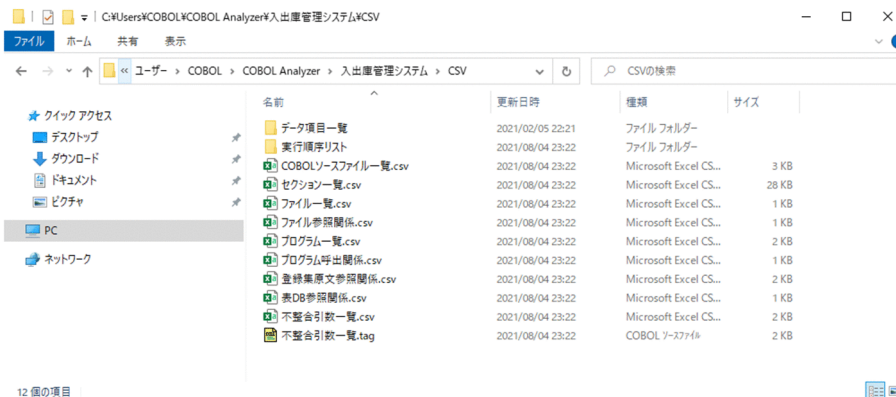
プログラム呼び出しの実引数と仮引数の不整合は、反復データ項目の添字の範囲外参照と並んで、運用時のプログラム異常終了の主要要因となっています。引数整合性チェックの機能では、プロジェクト中のすべての CALL 文の実引数と呼出先プログラムの仮引数の組み合わせに対して、次の項目の整合性をチェックします。

1. 実引数と仮引数の個数
2. 実引数と仮引数の受渡方法 (BY REFERENCE, BY CONTENT, BY VALUE, BY ATTRIBUTE)
3. 実引数と仮引数のデータ型の適合性
4. 実引数と仮引数のサイズ
5. 実引数と仮引数の集団項目の構造

この機能を使うことで、マイグレーションや障害発生を契機とした引数整合性の見直し作業を短期間で効率良く、かつ効果的に実施できます。

作業の手順

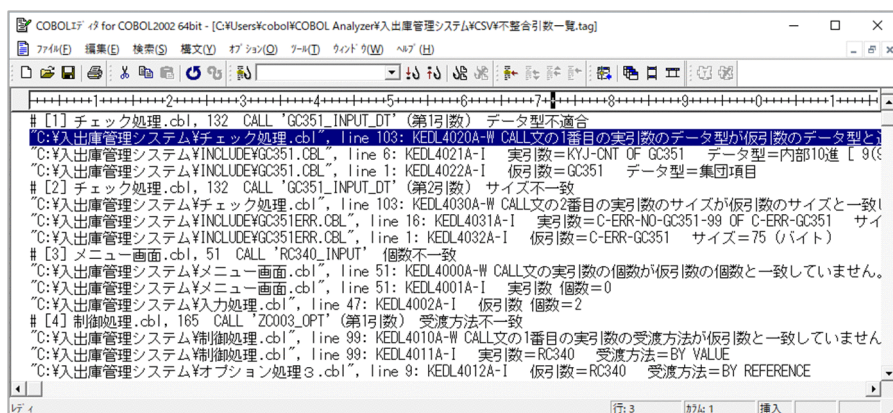
1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。
4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
5. [解析生成オプション設定] 画面で引数整合性チェックを有効にします。
[解析生成オプション設定] 画面の [引数整合性チェック] タブで、実施する引数整合性チェックを選択します。また、[生成ドキュメント] タブで、[不整合引数一覧] をチェックします。
6. コンパイラオプション、解析結果の出力先フォルダなどを設定します。
7. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析の実行] を選択します。
8. 開発マネージャの [ツール] - [COBOL ソース解析] - [解析結果(CSV/TAG)の表示] を選択します。
CSV ファイル形式のソース解析情報が格納されたフォルダが表示されます。



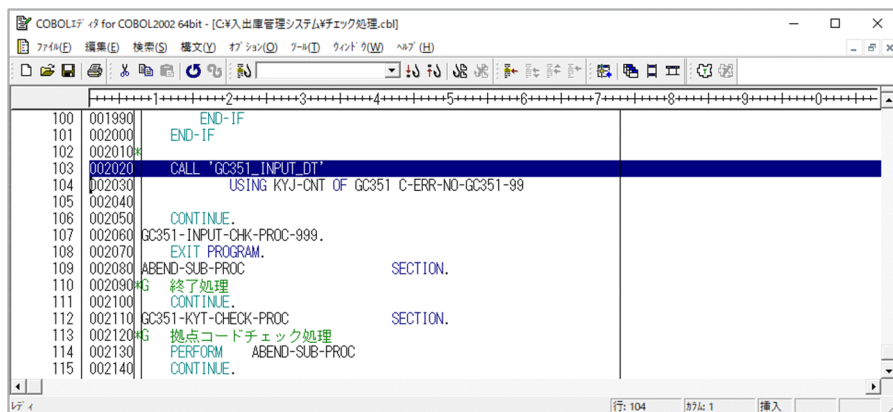
9. 不整合引数一覧を参照して、不整合のある引数を確認します。

- 不整合引数の警告メッセージを確認する

不整合引数一覧タグファイル（不整合引数一覧.tag）を選択すると、COBOL エディタが起動します。

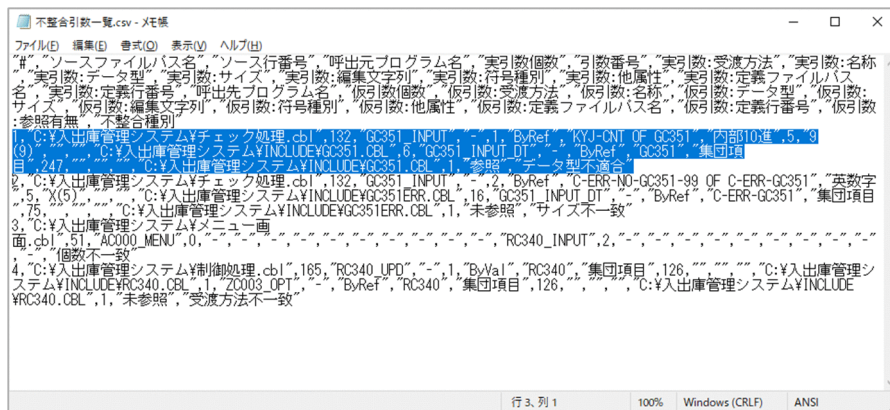


警告メッセージをダブルクリックすると、COBOL ソースファイルが表示され、該当行にカーソルが移動します。引数不整合のある CALL 文、実引数または仮引数のソースコードを COBOL エディタで修正します。



- 不整合引数の組み合わせを確認する

CSV ファイルが編集できるソフトウェア（CSV 編集ソフト、表計算ソフト、テキストエディタ）で不整合引数一覧 CSV ファイル（不整合引数一覧.csv）を開いて、引数不整合のある実引数と仮引数の組み合わせを確認します。



関連項目

- ・「8.2.6 不整合引数一覧」
- ・「10.11 引数整合性チェック」

3.7 処理ロジックが複雑なプログラムを調査する

処理ロジックを複雑にする主な要因には、次のものがあります。

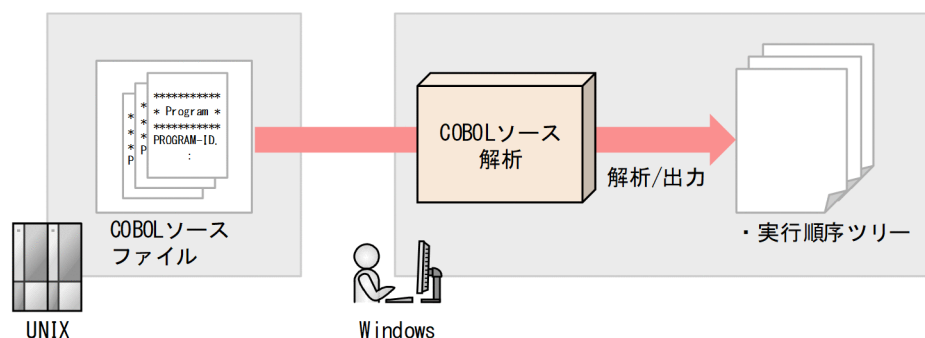
- 条件文やループ構文のネストが深い。
- 分岐や手続き呼び出しが多用されて、処理ロジックが入り組んでいる。

このような処理ロジックの場合、頭の中で大量の実行経路の組み立てが必要となります。また、分岐先や呼出先を探す手間で集中力が中断されるため、ソースコードリーディングが難しくなります。

実行順序表示機能を使用すると、すべての経路が分かりやすくツリー形式で表示され、分岐先や呼出先も直後に表示されるので探す手間が不要になります。

この機能を使うことで、処理ロジックが複雑なプログラムでも、ソースコードリーディングを効率的かつ効果的に実施できます。

図 3-6 処理ロジックが複雑なプログラムを調査する



注
UNIX環境からWindowsへのCOBOLソースファイルの転送には、FTPなどのファイル転送プログラムを使用します。

作業の手順

1. 開発マネージャで、解析用のプロジェクトを作成し、COBOL ソースファイルを登録します。
2. 開発マネージャの [ツール] - [COBOL ソース解析] - [ソース解析プロジェクトの使用開始] を選択します。
ソース解析プロジェクト名を設定するダイアログが表示されます。
3. 任意のプロジェクト名を設定します。
4. 開発マネージャの [ツール] - [COBOL ソース解析] - [設定] を選択します。
[解析生成オプション設定] 画面が表示されます。
5. [解析生成オプション設定] 画面で実行順序表示機能を使用するための設定をします。
[解析生成オプション設定] 画面の [その他の解析オプション] タブで、[プログラムの制御フロー情報 (実行順序表示機能で使用する情報) を抽出する] をチェックします。

解析生成オプション設定

ソース解析オプションファイル: C:\Users\%COBOL%\AppData\Local\Temp\MNGF599.xml

コンパイラオプション	主プログラム指定	プログラムコメント抽出	節コメント抽出
呼出先プログラム解決	引数整合性チェック	その他の解析オプション	生成ドキュメント

☒ レベル番号が02～49,66のデータ項目の情報を抽出する
☒ プログラムの制御フロー情報（実行順序表示機能で使用する情報）を抽出する

OK キャンセル

6. コンパイラオプション、解析結果の出力先フォルダなどを設定します。

7. 開発マネージャの［ツール］－［COBOL ソース解析］－［解析の実行］を選択します。

8. 開発マネージャの［ツール］－［COBOL ソース解析］－［解析結果(HTML)の表示］を選択します。
HTML ファイル形式の解析情報が表示されます。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ - 入在庫管理システム

2021/08/04 23:52:37

一覧情報

項目	全体	解析済ファイル	未解析ファイル
COBOLソースファイル	10	10	0
登録集原文ファイル	15	12	3

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	9	1	0	0
登録集原文	15	12	3	0	0
ファイル	3	3	0		
表(DB)	3				

分布状況 対象プログラム数=10本

[分布全体表示](#)

項目	平均(プログラム)	最小	最大
全体行数	153	19	332
記述項数	48	4	102
実行文の数	44	2	107
分岐の数	13	0	31
節数	6	1	14
段落数	11	0	28
呼出元プログラム数	0	0	2

[COBOL プログラム] 行の [全体] 列の件数をクリックすると、[プログラム一覧] 画面が表示されます。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ > プログラム一覧

プログラム一覧

#	プログラム名	使用有無	総行数	引数の個数	実行文の数	分岐の数	呼出先プログラム数	コメント行の文字列
1	AO000.MENU	使用	61	0	14	7	3	メニュー画面
2	GC351.INPUT	使用	112	2	45	18	1	チェック処理
3	GC351.INPUT.DT	使用	208	2	55	16	0	詳細チェック処理
4	RC340.INPUT	使用	218	2	68	16	2	入力処理
5	RC340.SET	使用	332	1	107	31	1	転送処理
6	RC340.SND	使用	280	0	76	22	1	出荷処理
7	RC340.UPD	使用	266	0	77	24	1	制御処理
8	ZC001.OPT	使用	19	0	2	0	0	オプション処理1
9	ZC002.OPT	未使用	21	1	2	0	0	オプション処理2
10	ZC003.OPT	使用	21	1	2	0	0	オプション処理3
-	合計	-	1538	-	448	134	9	-

[ページの先頭へ](#)

[プログラム一覧] 画面で、任意のプログラム名をクリックすると、[プログラム詳細] 画面が表示されます。

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ > プログラム一覧 > プログラム詳細

プログラム階層図 セクション階層図 ソース表示

プログラム概要

プログラム名	AC000.MENU		
COBOLソースファイル名	メニュー画面.cbl		
パス名	C:\入出庫管理システム		
コメント行の文字列	メニュー画面		
総行数	61	有効行数	51
コメント行数	10		
参照する登録集原文数	0	ファイル数	0
データ項目数	4	引数の個数	0
分岐の数	7	節数	1
表(DB)数	0	呼出元プログラム数	0
未使用の内部プログラム数	0	未使用のデータ項目数	0
制御が渡らない文、手続きの数	0	記述項数	9
		実行文の数	14
		段落数	2
		呼出先プログラム数	3
		未使用の報告書、画面、通信記述名の数	0

...

セクション一覧

#	手続き名(節名/段落名)	種別	呼出有無	制御の移行先の手続き名	コメント行の文字列	実行順序
1	AC000-MENU-MAIN	節	×		メニュー画面	表示
2	AC000-MENU-MAIN-010	段落	×			
3	AC000-MENU-MAIN-999	段落	×			

[プログラム詳細] 画面の「セクション一覧」で「表示」をクリックすると、[実行順序表示] 画面が表示されます。左ペインの実行順序ツリーをトレースすることで、節のソースコードの実行経路を調査できます。

実行順序表示 - Internet Explorer

実行順序表示

プログラム名: AC000.MENU ソースファイル名: メニュー画面.cbl

パス名: C:\入出庫管理システム

節名: AC000-MENU-MAIN

節コメント: メニュー画面

```

AC000-MENU-MAIN SECTION.
AC000-MENU-MAIN-010.
  MOVE "1" TO FIN-FLG
  PERFORM UNTIL FIN
    DISPLAY SPACE
  :
  EVALUATE ENT-CODE
    WHEN "01"
      CALL "RC340_INPUT"
    WHEN "02"
      CALL "RC340_UPD"
    WHEN "03"
      CALL "RC340_SND"
    WHEN "99"
      MOVE "2" TO FIN-FLG
  WHEN OTHER
    CONTINUE
  END-PERFORM
AC000-MENU-MAIN-999.
  STOP RUN ---- (プログラム終了)

```

実行順序ツリー

```

35 : 001350 AC000-MENU-MAIN SECTION.
36 : 001360 *6 メニュー画面
37 : 001370 AC000-MENU-MAIN-010.
38 : 001380 MOVE "1" TO FIN-FLG
39 : 001390 PERFORM UNTIL FIN
40 : 001400 DISPLAY SPACE
41 : 001410 END-DISPLAY
42 : 001420 DISPLAY GYOUNMU-ENTRY
43 : 001430 END-DISPLAY
44 : 001440 MOVE 0518 TO CRTADDR
45 : 001450 ACCEPT GYOUNMU-ENTRY
46 : 001460 EVALUATE ENT-CODE
47 : 001470 WHEN "01"
48 : 001480 CALL "RC340_INPUT"
49 : 001490 WHEN "02"
50 : 001500 CALL "RC340_UPD"
51 : 001510 WHEN "03"
52 : 001520 CALL "RC340_SND"
53 : 001530 WHEN "99"
54 : 001540 MOVE "2" TO FIN-FLG
55 : 001550 WHEN OTHER
56 : 001560 CONTINUE
57 : 001570 END-EVALUATE
58 : 001580 END-PERFORM
59 : 001590 CONTINUE.
60 : 001600 AC000-MENU-MAIN-999.
61 : 001610 STOP RUN.

```


参考

「解析生成オプション設定」画面の「生成ドキュメント」タブで「実行順序リスト」をチェックすることで、プログラム全体の実行順序ツリーの詳細を CSV ファイル（実行順序リストといいます）に出力できます。実行順序リストの格納フォルダは、開発マネージャの「ツール」－「COBOL ソース解析」－「解析結果(CSV/TAG)の表示」を選択すると表示できます。

関連項目

- 「10.16 制御フロー解析による実行順序表示機能」

4

バッチモードで実行する

COBOL ソース解析をバッチモードで実行する手順について説明します。

4.1 バッチモードで COBOL ソース解析を実行する

COBOL ソース解析の GUI を使用しないで、コマンドラインから COBOL ソース解析を実行するバッチモードの手順です。

バッチモードでは、解析生成オプションの設定を除き、コマンドプロンプト上で COBOL ソース解析の操作を実行できます。GUI で操作できない次の作業や、大量の COBOL プログラムを定期的に解析する、COBOL プログラムとソース解析情報を常に一致させるなど、GUI での操作をしないで一括で実行するときに使用します。

- ソース解析プロジェクトの名称の変更
- ソース解析プロジェクトの解析生成オプションのインポート／エクスポート
- ソース解析プロジェクト内のすべての COBOL ソースファイルの解析／生成
- ソース解析用データベースの初期化

作業の手順

1. Windows のコマンドプロンプトを起動します。
2. 次に示す cblcsa コマンドで、COBOL ソース解析の実行の対象となるソース解析プロジェクトを作成します。

```
cblcsa -Project ソース解析プロジェクト名 -Create
```

3. 解析生成オプションを設定します。

次の手順で解析生成オプションを設定します。ソース解析プロジェクトが開発マネージャ上で作成されている場合は、開発マネージャの [ツール] - [COBOL ソース解析] - [設定] でも設定できます。

1. ソース解析プロジェクトからソース解析オプションファイルを取得する。

```
cblcsa -Project ソース解析プロジェクト名 -ExportOption ソース解析オプションファイル名
```

2. [解析生成オプション設定] 画面を開いて、ソース解析オプションファイルの内容を編集する。

```
cblcsaSetOptions -OptionFile ソース解析オプションファイル名
```

3. ソース解析オプションファイルの変更をソース解析プロジェクトに登録する。

```
cblcsa -Project ソース解析プロジェクト名 -ImportOption ソース解析オプションファイル名
```

4. 解析対象の COBOL ソースファイルを登録します。

```
cblcsa -Project ソース解析プロジェクト名 -Add {-Source ソースファイル名 [;ソースファイル名] ... | -SourceFolder ソースフォルダ名 [;ソースフォルダ名] ...}
```

5. COBOL ソース解析を実行します。

```
cblcsa -Project ソース解析プロジェクト名 [-Analyzer {ALL | AUTO} ] [-LogFolder 解析ログ  
および操作ログの出力先フォルダ]
```

6. 実行後の結果を確認します。

バッチモードでは、標準エラー出力に次の情報が表示されます。

- 解析状況
- 生成状況
- 実行結果

コマンドラインオプションに-LogFolder を指定した場合は、指定した出力先フォルダに解析ログおよび操作ログが出力されます。

■ 注意事項

バッチモードで作成したソース解析プロジェクトを開発マネージャから操作する場合、ソース解析プロジェクトの使用開始をしても、解析結果を参照する開発マネージャのメニューは活性化しません。開発マネージャでソース解析プロジェクトの使用開始、解析生成オプションの設定、および COBOL ソース解析の実行を行ってください。

関連項目

- [「cblcsa コマンド - バッチモードで COBOL ソース解析を実行する」](#)

5

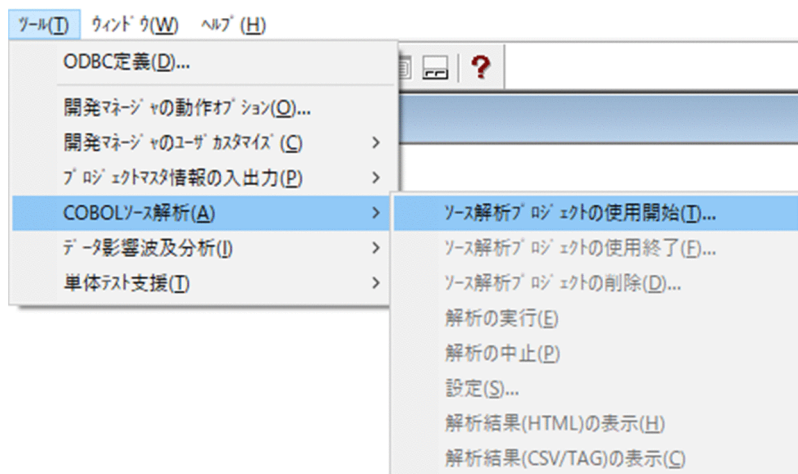
画面のリファレンス

COBOL ソース解析で解析対象プロジェクトを設定する画面の内容について説明します。

5.1 開発マネージャでの COBOL ソース解析の操作メニュー項目と設定画面

COBOL ソース解析に関して設定する、開発マネージャでの COBOL ソース解析の操作メニュー項目と設定画面です。

図 5-1 開発マネージャの [ツール] - [COBOL ソース解析] のメニュー



項目の説明

[ソース解析プロジェクトの使用開始]

ソース解析プロジェクト名を指定する [ソース解析プロジェクト] ダイアログボックスを表示します。指定したソース解析プロジェクトを開始します。

ソース解析プロジェクトが使用中の場合、メニュー名の後ろに、使用中のソース解析プロジェクト名を表示します。

例) 使用中のソース解析プロジェクト名が「販売管理システム」の場合のメニュー表示

ソース解析プロジェクトの使用開始(販売管理システム)

[ソース解析プロジェクトの使用終了]

使用中のソース解析プロジェクトを終了します。

[ソース解析プロジェクトの削除]

使用中のソース解析プロジェクトを削除します。

[解析の実行]

COBOL ソースファイルの解析を実行します。

[解析の中止]

実行中の COBOL ソースファイルの解析を中止します。

[設定]

[解析生成オプション設定] 画面を表示します。

[解析結果(HTML)の表示]

HTML 形式の解析結果のプロジェクトサマリ画面 (index.html) を表示します。

[解析結果(CSV/TAG)の表示]

CSV 形式の解析結果 (CSV ファイルおよびタグファイル) を格納したフォルダを表示します。

図 5-2 [ソース解析プロジェクト] ダイアログボックス



項目の説明

[ソース解析プロジェクト]

任意のソース解析プロジェクト名を指定します。ソース解析プロジェクト未作成のプロジェクトマスタの場合、プロジェクトマスタ名の先頭 30 文字が、それ以外の場合、前回設定した値が初期値として表示されます。

ソース解析プロジェクト名は、次の文字を含まない名前を 30 文字以下で指定します。英小文字と英大文字は区別されません。

「¥」「/」「:」「*」「?」「"」「<」「>」「|」「@」「;」、半角空白、全角空白、タブ

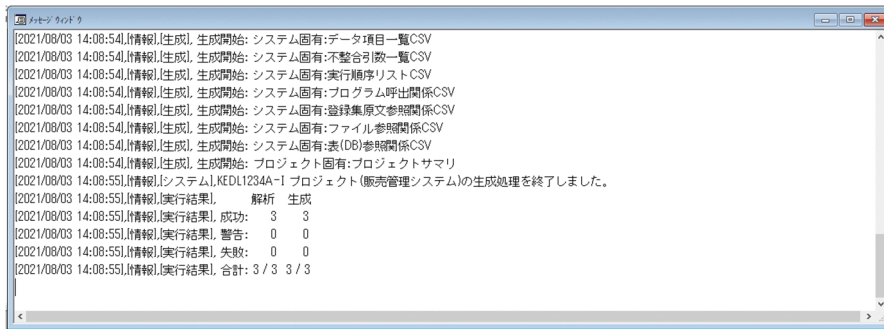
また、先頭文字にハイフン(-)を指定しないでください。

注意事項

- ソース解析プロジェクト未作成のプロジェクトマスタの場合、プロジェクトマスタ名の「;」、半角空白、全角空白、タブ、および先頭の「-」は、「_」に置換されて表示されます。また、先頭 30 文字が [ソース解析プロジェクト] に表示されるため、プロジェクトマスタ名が異なっても、同一のソース解析プロジェクト名として表示されることがあります。プロジェクトマスタ名を利用してソース解析プロジェクト名を作成する場合は、同一名称にならないように注意してください。
- 前回指定したソース解析プロジェクト名と異なるソース解析プロジェクト名を指定すると、ソース解析用データベースに新しくソース解析プロジェクトが作成されます。前回指定したソース解析プロジェクトはソース解析用データベースから削除されません。次のどちらかの方法で削除してください。
 - 前回指定したソース解析プロジェクトを使用開始し、ソース解析プロジェクトの削除を実施する。
 - cblcsa コマンドを使用する。
- ソース解析プロジェクトの使用を開始すると、ソース解析用データベースに対して排他制御を行います。ほかの開発マネージャからの起動も含めて、COBOL ソース解析を実行できません。COBOL ソース解析を実行するには、次のどれかを実施して排他を解除してください。
 - ソース解析プロジェクトの使用終了
 - ソース解析プロジェクトの削除
 - プロジェクトマスタを閉じる

- 開発マネージャの終了

図 5-3 開発マネージャのメッセージウィンドウ



COBOL ソース解析の実行状況のログ，エラーメッセージ，および実行結果のログが表示されます。

実行状況および実行結果のログの表示内容については、「[cblcsa コマンド - バッチモードで COBOL ソース解析を実行する](#)」の出力例を参照してください。

5.2 [解析生成オプション設定] 画面

[解析生成オプション設定] 画面では、解析に必要なコンパイラオプションと主プログラム指定、ソース解析情報として生成するドキュメントとその出力先フォルダを設定します。また、必要に応じて、コメント行の抽出オプションや引数整合性チェックなどのオプション機能を設定します。

5.2.1 [コンパイラオプション] タブ

[解析生成オプション設定] 画面の [コンパイラオプション] タブには、COBOL ソースファイルのコンパイルに必要なコンパイラオプションとコンパイラ環境変数を指定します。COBOL2002 コンパイラによるコンパイルで構文エラーとならないように過不足なく指定してください。

開発マネージャ上で [解析生成オプション設定] 画面を開いている場合は、開発マネージャに設定されている内容を引き継ぐか、[コンパイラオプションの設定] や [コンパイラ環境変数の設定] に指定するかを選択できます。

コンパイラオプションについては、マニュアル「COBOL2002 ユーザーズガイド」を参照してください。

図 5-4 [解析生成オプション設定] 画面の [コンパイラオプション] タブ

解析生成オプション設定

ソース解析オプションファイル: C:\Users\COBOL\AppData\Local\Temp\MNGA87F.xml

呼出先プログラム解決	引数整合性チェック	その他の解析オプション	生成ドキュメント
コンパイラオプション	主プログラム指定	プログラムコメント抽出	節コメント抽出

☒ 開発マネージャの設定を引き継ぐ
☐ オプション・環境変数を設定する

コンパイラオプションの設定

開発マネージャのコンパイラオプション設定内容

```
-NumAccept  
-DebugLine
```

コンパイラ環境変数の設定

```
CBLCOPT2002=-StdVersion.1 -DebugData -Switch,EBCDIK  
CBLLIB=C:\Program Files\Hitachi\COBOL2002 64bit\copy
```

開発マネージャの環境変数設定内容

```
CBLCOPT2002=-StdVersion.1 -DebugData -Switch,EBCDIK  
CBLLIB=c:\販売管理システム04\仮裏作成\work\C:\Program Files\Hitachi\COBOL2002 64bit\copy  
CBLPIDIR=c:\販売管理システム04\仮裏作成\work
```

OK キャンセル

[開発マネージャの設定を引き継ぐ]

開発マネージャで設定されているコンパイラオプションとコンパイラ環境変数を、ソース解析プロジェクトまたはソース解析オプションファイルに登録します。開発マネージャで設定されているコンパイラ

オプションとコンパイラ環境変数は、[開発マネージャのコンパイラオプション設定内容] と [開発マネージャのコンパイラ環境変数設定内容] に表示されます。

[オプション・環境変数を設定する]

[コンパイラオプションの設定] にコンパイラオプション、[コンパイラ環境変数の設定] にコンパイラ環境変数を指定して、ソース解析プロジェクトまたはソース解析オプションファイルに登録します。

[コンパイラオプションの設定]

COBOL ソースファイルのコンパイルに必要なコンパイラオプションを指定します。

複数のコンパイラオプションを指定する場合は、空白文字で区切って指定します。

指定できないコンパイラオプションは、指定してもコンパイル時に無視されます。コンパイラオプションの使用可否については、「[10.4 COBOL ソース解析で利用できるコンパイラオプション](#)」を参照してください。

[開発マネージャのコンパイラオプション設定内容]

開発マネージャで設定されているコンパイラオプションが表示されます。

表示内容は、コピー&ペーストで [コンパイラオプションの設定] に貼り付けられます。

[コンパイラ環境変数の設定]

COBOL ソースファイルのコンパイルに必要なコンパイラ環境変数を指定します。

1 行に 1 つの環境変数を「環境変数名=環境変数値」の形式で指定します。イコール (=) のない行は無視されます。

イコール (=) の後ろに何も指定しなかった場合は、空文字列が設定されます。環境変数値が空文字列の環境変数は「環境変数未定義」を意味します。この指定は、解析／生成の実行時に Windows の環境変数を適用しないで、未定義扱いにする場合に使用します。

同じ名前の環境変数を複数指定した場合は、最後に指定した値が有効になります。

ここに指定したコンパイラ環境変数は、解析／生成の開始時に Windows の環境変数に登録されて、COBOL ソースファイルの解析 (COBOL2002 コンパイラによるコンパイル) で有効になります。Windows に同じ環境変数がすでに登録されている場合は上書きとなるので、[コンパイラ環境変数の設定] に指定した値が優先されます。

コンパイラ環境変数の使用可否については、「[10.5 COBOL ソース解析で利用できるコンパイラ環境変数](#)」を参照してください。

[開発マネージャの環境変数設定内容]

開発マネージャで有効となっている次のコンパイラ環境変数が表示されます。

- 開発マネージャのプロジェクトに登録されているコンパイラ環境変数
- Windows の環境変数で定義されているコンパイラ環境変数

表示内容は、コピー&ペーストで [コンパイラ環境変数の設定] に貼り付けられます。

注意事項

環境変数 CBLLIB には、開発マネージャのプロジェクトで設定した作業フォルダのパスが追加されます。

関連項目

- ・「10.4 COBOL ソース解析で使えるコンパイラオプション」
- ・「10.5 COBOL ソース解析で使えるコンパイラ環境変数」

5.2.2 [主プログラム指定] タブ

[解析生成オプション設定] 画面の [主プログラム指定] タブには、解析する COBOL ソースファイルの主プログラムを指定します。

図 5-5 [解析生成オプション設定] 画面の [主プログラム指定] タブ

The screenshot shows the 'Main Program Specification' tab of the 'COBOL Source Analysis Options' dialog box. The dialog has a title bar '解析生成オプション設定' and a close button. Below the title bar is a text field for 'ソース解析オプションファイル' with the path 'C:\Users\%COBOL%\AppData\Local\Temp\MNGA87F.xml'. There are four tabs: '呼出先プログラム解決', '引数整合性チェック', 'その他の解析オプション', and '生成ドキュメント'. The '引数整合性チェック' tab is selected, showing sub-tabs for 'コンパイラオプション', '主プログラム指定', 'プログラムコメント抽出', and '節コメント抽出'. The '主プログラム指定' sub-tab is active, displaying the '主プログラムの自動検出' section. This section has a checked checkbox '主プログラムをCOBOLソースファイル先頭の最外側プログラムの中から検出する'. Below it are two groups of conditions: 'USING検出条件' and 'RETURNING検出条件'. The 'USING検出条件' group has four checkboxes: '引数なし' (checked), '引数1個で英数字である' (unchecked), '引数1個で可変長文字列型の集団項目である' (checked), and '引数2個で2進整数とポインタである' (checked). The 'RETURNING検出条件' group has one checked checkbox 'RETURNING指定なし'. At the bottom is the '主プログラム名の指定' section, which is currently empty. At the very bottom of the dialog are 'OK' and 'キャンセル' buttons.

[主プログラムの自動検出]

主プログラムを自動検出する場合は、[主プログラムを COBOL ソースファイル先頭の最外側プログラムの中から検出する] のチェックボックスをオンにします。このチェックボックスをオンにすると、[USING 検出条件] と [RETURNING 検出条件] の両方の条件を満たす COBOL ソースファイルの先頭の最外側プログラムが主プログラムとして扱われます。

[USING 検出条件]

主プログラムとして検出する COBOL ソースファイルの先頭の最外側プログラムが満たす USING 指定の条件を選択します。USING 検出条件は OR 条件です。チェックボックスがすべてオフの場合は、USING 検出条件は常に満たす扱いになります。

- ・ 引数なし

COBOL ソースファイルの先頭の最外側プログラムで、かつ引数なしのものを主プログラムとします。

- 引数 1 個で英数字である

COBOL ソースファイルの先頭の最外側プログラムで、かつ引数 1 個で参照渡し of 英数字項目であるものを主プログラムとします。

- 引数 1 個で可変長文字列型の集団項目である

COBOL ソースファイルの先頭の最外側プログラムで、かつ引数 1 個で参照渡し of 可変長文字列の集団項目であるものを主プログラムとします。可変長文字列の集団項目とは、長さの領域を示す 2 バイトの 2 進整数と、文字列の領域を示す英数字項目または集団項目の従属項目だけで構成される項目のことです。

- 引数 2 個で 2 進整数とポインタである

COBOL ソースファイルの先頭の最外側プログラムで、かつ引数 2 個で第 1 引数が値渡し of 2 進整数項目、第 2 引数が値渡し of ポインタ項目（アドレス項目）または参照渡し of ポインタ項目（アドレス項目）だけで構成される集団項目であるものを主プログラムとします。

[RETURNING 検出条件]

主プログラムとして検出する COBOL ソースファイル先頭の最外側プログラムが満たす RETURNING 指定の条件を選択します。チェックボックスがすべてオフの場合は、RETURNING 検出条件は常に満たす扱いになります。

- RETURNING 指定なし

COBOL ソースファイルの先頭の最外側プログラムで、かつ RETURNING 指定がないものを主プログラムとします。

[主プログラム名の指定]

主プログラムを明示的にプログラム名（PROGRAM-ID 段落に記述した名前）で指定する場合に使用します。プログラム名は「[10.9 解析結果に出力される COBOL ソースファイル中の名称](#)」で示すプログラム名の変換規則に従って変換した名前を指定します。また、複数のプログラム名を指定する場合は、コンマまたは改行で区切って指定します。

関連項目

- 「[10.12 主プログラムの指定方法](#)」

5.2.3 [プログラムコメント抽出] タブ

[解析生成オプション設定] 画面の [プログラムコメント抽出] タブでは、COBOL ソースファイルからプログラムコメントを抽出するかどうかを指定します。

プログラムコメントを抽出しない場合は、設定する必要はありません。

図 5-6 「解析生成オプション設定」画面の「プログラムコメント抽出」タブ

解析生成オプション設定

ソース解析オプションファイル: C:\Users\COBOL\AppData\Local\Temp\MMNGA87F.xml

呼出先プログラム解決	引数整合性チェック	その他の解析オプション	生成ドキュメント
コンパイルオプション	主プログラム指定	プログラムコメント抽出	節コメント抽出

☒ プログラムコメントを抽出する

抽出位置

プログラムコメント行の検索場所

☒ プログラム名段落の前

☐ プログラム名段落の後

プログラムコメント行の検索条件

☐ 抽出キー-文字列で始まるコメント行（プログラムコメントは抽出キー-の後の文字列）

☒ 抽出キー-文字列で始まるコメント行の次行

抽出キー-文字列

☐ プログラム名段落から指定行目の位置にあるコメント行

行目(1~99)

OK キャンセル

[プログラムコメントを抽出する]

COBOL ソースファイルからプログラムコメントを抽出するかどうかを選択します。チェックボックスにチェックすると、検索場所と検索条件を選択できます。

検出対象となるプログラムコメント行を次に示します。

- 固定形式正書法で、7 カラムに「*」、[/] がある行
- 自由形式正書法で、最初の文字に「*」、[/] がある行
- 「*>」で始まる行

[プログラムコメント行の検索場所]

プログラム名段落（PROGRAM-ID）の前と後ろのどちらのプログラムコメント行（プログラムコメントを含むコメント行）を検索するか選択します。

- [プログラム名段落の前]

プログラム名段落（PROGRAM-ID）の 1 行前の行から COBOL ソースファイルの先頭まで検索条件を満たすプログラムコメント行を検索するときに選択します。COBOL ソースファイルの先頭の最外側プログラムでない場合は、前のプログラムの有効行に到達した時点で検索を終了します。

- [プログラム名段落の後]

プログラム名段落（PROGRAM-ID）の 1 行後ろの行から見出し部の末尾まで検索条件を満たすプログラムコメント行を検索するときに選択します。

[プログラムコメント行の検索条件]

プログラムコメント行（プログラムコメントを含むコメント行）の検索条件を選択します。

- [抽出キー文字列で始まるコメント行（プログラムコメントは抽出キーの後の文字列）]

次に示す注記行をプログラムコメント行として検出するときに選択します。

- ・ プログラム名段落に最も近い注記行
- ・ 先頭に特定のマーカ（目印）が付いている注記行

[抽出キー文字列] には、プログラム名段落に最も近い注記行のときは空文字列、先頭に特定のマーカ（目印）が付いている注記行のときは特定のマーカ（目印）を指定します。

- [抽出キー文字列で始まるコメント行の次行]

先頭に特定のマーカ（目印）が付いている注記行の次の行をプログラムコメント行として検出するときに選択します。[抽出キー文字列] には特定のマーカ（目印）を指定します。

- [抽出キー文字列]

注記の先頭に付いている特定のマーカ（目印）の文字列、または空文字列を指定します。特定のマーカ（目印）の文字列は 65 バイト以内で指定します。

プログラムコメント行として、プログラム名段落に最も近い注記行を検出する場合は空文字列を、先頭に特定のマーカ（目印）が付いている注記行またはその次の行を検出する場合は特定のマーカ（目印）の文字列を指定してください。

全角空白およびタブコードを使用できません。使用すると、特定のマーカ（目印）を持つプログラムコメント行が検出できないため、プログラムコメントは抽出されません。

- [プログラム名段落から指定行目の位置にあるコメント行]

プログラム名段落（PROGRAM-ID）から特定の行目にある注記行をプログラムコメント行として検出するときに選択します。プログラム名段落（PROGRAM-ID）の前か後ろかは、[プログラムコメント行の検索場所] で指定します。[行目] には、プログラム名段落からの相対行数を 1～99 の範囲で指定します。1～99 以外を指定すると、1 が假定されます。相対行数は、注記行、空白行、デバッグ行および翻訳指令行の行数だけをカウントして指定してください。ほかの行はカウントしないでください。

なお、この項目を選択すると [抽出キー文字列] は指定できません。

関連項目

- 「[3.4 コメント行の文字列を抽出する](#)」
- 「[10.13 プログラムコメントの抽出](#)」

5.2.4 [節コメント抽出] タブ

[解析生成オプション設定] 画面の [節コメント抽出] タブでは、COBOL ソースファイルから節コメントを抽出するかどうかを指定します。

節コメントを抽出しない場合は、設定する必要はありません。

図 5-7 [解析生成オプション設定] 画面の [節コメント抽出] タブ

解析生成オプション設定

ソース解析オプションファイル: C:\Users\COBOL\AppData\Local\Temp\MNGA87F.xml

呼出先プログラム解決	引数整合性チェック	その他の解析オプション	生成ドキュメント
コンパイラオプション	主プログラム指定	プログラムコメント抽出	節コメント抽出

☒ 節コメントを抽出する

抽出位置

節コメント行の検索場所

☒ 節見出しの前

☐ 節見出しの後

節コメント行の検索条件

☐ 抽出キーワードで始まるコメント行（節コメントは抽出キーワードの後の文字列）

☒ 抽出キーワードで始まるコメント行の次行

抽出キーワード:

☐ 節見出しから指定行目の位置にあるコメント行

行目(1~99)

OK キャンセル

[節コメントを抽出する]

COBOL ソースファイルから節コメントを抽出するかどうかを選択します。チェックボックスにチェックすると、検索場所と検索条件を選択できます。ただし、節の見出しがない暗黙的な節（{PROCEDURE_START}）の節コメントは抽出できません。

検出対象となる節コメント行を次に示します。

- 固定形式正書法で、7 カラムに「*」、[/] がある行
- 自由形式正書法で、最初の文字に「*」、[/] がある行
- 「*>」で始まる行

[節コメント行の検索場所]

節見出しの前と後ろのどちらの節コメント行（節コメントを含むコメント行）を検索するか選択します。

- 節見出しの前
節見出しの 1 行前の行から COBOL ソースファイルの先頭まで検索条件を満たす節コメント行を検索するときに選択します。有効行に到達した時点で検索を終了します。
- 節見出しの後
節見出しの 1 行後ろの行から節の末尾まで検索条件を満たす節コメント行を検索するときに選択します。

[節コメント行の検索条件]

節コメント行（節コメントを含むコメント行）の検索条件を選択します。

- [抽出キー文字列で始まるコメント行（節コメントは抽出キーの後の文字列）]
次に示す注記行を節コメント行として検出するときに選択します。
 - ・節見出しに最も近い注記行
 - ・先頭に特定のマーカ（目印）が付いている注記行
 [抽出キー文字列] には、節見出しに最も近い注記行のときは空文字列、先頭に特定のマーカ（目印）が付いている注記行のときは特定のマーカ（目印）を指定します。
- [抽出キー文字列で始まるコメント行の次行]
先頭に特定のマーカ（目印）が付いている注記行の次の行を節コメント行として検出するときに選択します。[抽出キー文字列] には特定のマーカ（目印）を指定します。
- [抽出キー文字列]
注記の先頭に付いている特定のマーカ（目印）の文字列、または空文字列を指定します。特定のマーカ（目印）の文字列は 65 バイト以内で指定します。
節コメント行として、節見出しに最も近い注記行を検出する場合は空文字列を、先頭に特定のマーカ（目印）が付いている注記行またはその次の行を検出する場合は特定のマーカ（目印）の文字列を指定してください。
全角空白およびタブコードを使用できません。使用すると、特定のマーカ（目印）を持つ節コメント行が検出できないため、節コメントは抽出されません。
- [節見出しから指定行目の位置にあるコメント行]
節見出しから特定の行目にある注記行を節コメント行として検出するときに選択します。節見出しの前か後ろかは、[節コメント行の検索場所] で指定します。[行目] には、節見出しからの相対行数を 1～99 以外を指定します。1～99 以外を指定すると、1 が仮定されます。相対行数は、注記行、空白行、デバッグ行および翻訳指令行の行数だけをカウントして指定してください。ほかの行はカウントしないでください。
なお、この項目を選択すると [抽出キー文字列] は指定できません。

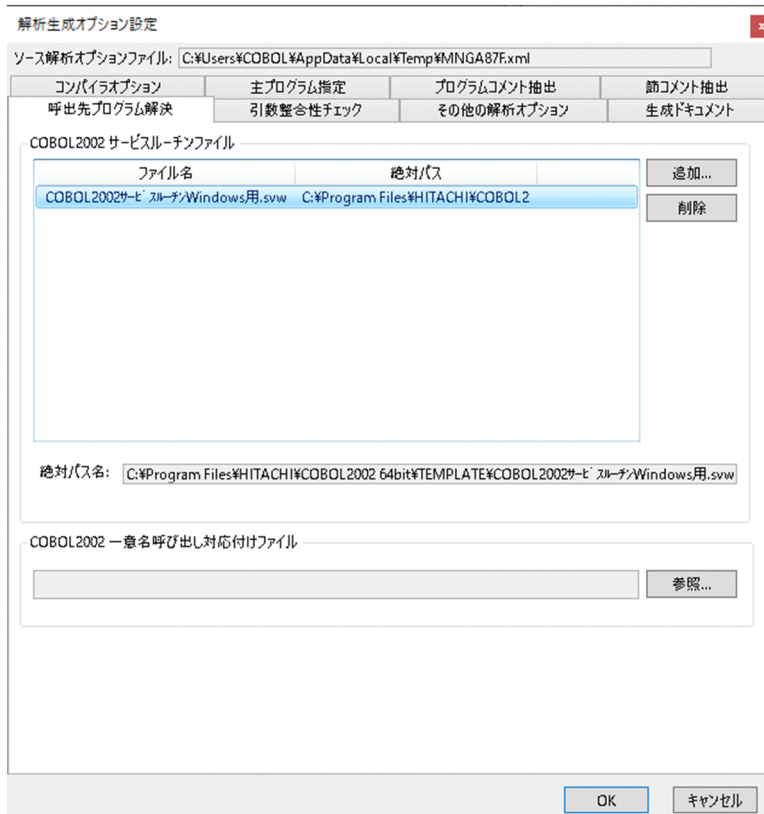
関連項目

- [3.4 コメント行の文字列を抽出する]
- [10.14 節コメントの抽出]

5.2.5 [呼出先プログラム解決] タブ

[解析生成オプション設定] 画面の [呼出先プログラム解決] タブでは、プログラムの呼び出し先を解決させるためのファイルを指定します。プロジェクトにないプログラムの呼び出し（サービスルーチン呼び出しなど）や一意名によるプログラム呼び出しによってプログラムの呼び出し先が解決できないで、解析結果が意図したとおりでない場合に使用します。

図 5-8 「解析生成オプション設定」画面の「呼出先プログラム解決」タブ



「COBOL2002 サービスルーチンファイル」

サービスルーチンファイル（svw ファイル）を指定します。プログラム内で使用するサービスルーチンやサードベンダが提供するライブラリを、「未登録プログラム一覧」から除外できます。初期表示では、何も表示されません。

指定できる COBOL サービスルーチンファイルを次に示します。これらのサービスルーチンファイルは、COBOL2002 でテンプレートとして提供されます。

- COBOL85 サービスルーチン VOS3 用.svw
- COBOL2002XML 連携機能サービスルーチン.svw
- COBOL2002 サービスルーチン Windows 用.svw
- COBOL2002Java プログラム呼び出し機能サービスルーチン.svw

また、ユーザー独自の COBOL サービスルーチンファイル（svw ファイル）を作成して、解析オプションに登録することもできます。ファイル名（パス名を除いたもの）と拡張子の長さの合計値は、100 バイト以内にしてください。

「追加」ボタン

「COBOL サービスルーチンファイルの参照」ダイアログが表示されます。「未登録プログラム一覧」から除外するサービスルーチンが指定された COBOL サービスルーチンファイルを選択します。

「削除」ボタン

「COBOL2002 サービスルーチンファイル」で選択されている COBOL サービスルーチンファイルが削除されます。

[絶対パス名]

[COBOL2002 サービスルーチンファイル] で選択されている COBOL2002 サービスルーチンファイルの絶対パス名が表示されます。

[COBOL2002 一意名呼び出し対応付けファイル]

一意名によるプログラム呼び出しの呼び出し先をコンパイラで完全に解決できない場合に、一意名呼び出し対応付けファイル (idc ファイル) を作成し、絶対パス名で指定します。一意名呼び出し対応付けファイルの作成方法については、「[付録 B.1 一意名呼び出し対応付けファイル](#)」を参照してください。

■ 注意事項

一意名によるプログラムの呼び出し先をコンパイラで完全に解決できない場合は、警告メッセージが出力されます。この場合、解析結果のドキュメント中のプログラムの使用有無に関する情報は確定しません。警告メッセージが出力された場合は、一意名呼び出し対応付けファイルを作成して、必ず一意名の呼び出し先を解決してください。

[参照] ボタン

[COBOL 一意名呼び出し対応付けファイルの参照] ダイアログが表示されます。

関連項目

- 「[付録 B.1 一意名呼び出し対応付けファイル](#)」
- 「[付録 B.2 サービスルーチンファイル](#)」

5.2.6 [引数整合性チェック] タブ

[解析生成オプション設定] 画面の [引数整合性チェック] タブでは、引数整合性チェックで実施する検査項目を設定します。

図 5-9 「解析生成オプション設定」画面の「引数整合性チェック」タブ



「引数の個数を検査する」

実引数と仮引数の個数が一致しているかどうかを検査する場合にチェックします。

「実引数個数 > 仮引数個数の場合は検出しない」

実引数個数 > 仮引数個数の場合、不整合引数情報の出力を抑止するときにチェックします。

「引数の受渡方法を検査する」

実引数と仮引数の受け渡し方法が一致しているかどうかを検査する場合にチェックします。

「引数のデータ型を検査する」

実引数と仮引数のデータ型が適合しているかどうかを検査する場合にチェックします。

「引数のサイズを検査する」

実引数と仮引数のサイズが一致しているかどうかを検査する場合にチェックします。

「次の条件をすべて満たすケースを検出しない」

実引数と仮引数が次の条件を満たす場合、引数のサイズの検査対象から除外するときにチェックします。

- BY REFERENCE 指定、または BY CONTENT 指定である
- 一方または両方が英数字集団項目である
- 実引数サイズ > 仮引数サイズである

[一方が可変長集団項目の場合はサイズ検査対象外とする(チェックがない場合は、最大長で検査する)]
実引数または仮引数の一方が可変長集団項目の場合、引数のサイズの検査対象から除外するときに
チェックします。チェックがないとき、可変長集団項目は最大長で検査されます。

[引数の集団項目の構造を検査する]

実引数と仮引数が集団項目で、その種別とサイズが同じである場合に、集団項目に属するデータ項目の
構造が一致しているかどうかを検査するときにチェックします。

この検査項目をチェックして解析生成オプションをプロジェクトに登録すると、[解析生成オプション
設定] 画面の [その他の解析オプション] タブの [レベル番号が 02~49,66 のデータ項目の情報を抽出
する] も一緒にチェックされているものとして登録されます。

[対応する仮引数のない実引数を検出する]

実引数個数>仮引数個数の場合に、対応する仮引数のない実引数の情報を出力するときにチェックしま
す。

[対応する実引数のない仮引数を検出する]

実引数個数<仮引数個数の場合に、対応する実引数のない仮引数の情報を出力するときにチェックしま
す。

関連項目

- [\[5.2.7 \[その他の解析オプション\] タブ\]](#)
- [\[10.11.2 引数整合性チェックの検査内容\]](#)

5.2.7 [その他の解析オプション] タブ

[解析生成オプション設定] 画面の [その他の解析オプション] タブでは、解析時に、従属項目情報を抽出
するかどうかや、実行順序表示機能で使用する情報を抽出するかどうかを指定できます。

図 5-10 「解析生成オプション設定」画面の「その他の解析オプション」タブ



「レベル番号が 02～49,66 のデータ項目を抽出する」

COBOL ソースの解析でレベル番号が 02～49 および 66 のデータ項目の情報を抽出対象とする場合に有効（チェックボックスをオン）にします。このオプションを有効にすると、次のドキュメントに従属項目関連の情報が出力されるようになります。

- データ項目一覧（CSV 形式だけ）

集団項目に属する従属項目（レベル番号が 02～49 および 66 のデータ項目）の情報まで出力されます。

- 不整合引数一覧

実引数のデータ名が修飾子付きデータ名に変更されて出力されます。

注意事項

- このオプションを有効（チェックボックスをオン）にすると、集団項目の従属項目数に比例して解析時間（データ項目情報のデータベース格納時間）が増加します。集団項目の従属項目（レベル番号が 02～49 および 66 のデータ項目）の情報が不要な場合は、このオプションを無効（チェックボックスをオフ）にしてください。デフォルトでは有効（チェックボックスがオン）になっています。
- 「解析生成オプション設定」画面の「引数整合性チェック」タブの「引数の集団項目の構造を検査する」が有効（チェックボックスがオン）の場合、このオプションを無効（チェッ

クボックスをオフ)にしても、その指定は無視され、有効(チェックボックスがオン)になります。

[プログラムの制御フロー情報(実行順序表示機能で使用する情報)を抽出する]

プログラムの制御フロー情報を抽出する場合に有効(チェックボックスをオン)にします。プログラムの制御フロー情報は、実行順序表示機能([実行順序表示]画面および実行順序リスト)を使用する場合に必要です。プログラムの制御フロー情報が抽出されていない場合、[実行順序表示]画面および実行順序リストの実行順序ツリーを使用したソースコードリーディングは利用できません。

注意事項

このオプションを有効(チェックボックスをオン)にすると、すべての COBOL ソースファイルにあるソースコードの制御フローをすべて解析し、大量の解析データをデータベースに登録するため、解析時間が増加します。[実行順序表示]画面および実行順序リストの実行順序ツリーを使用したソースコードリーディングを利用しない場合、このオプションは無効(チェックボックスをオフ)にしてください。デフォルトでは無効(チェックボックスがオフ)になっています。

関連項目

- ・[5.2.6 [引数整合性チェック] タブ]
- ・[10.16 制御フロー解析による実行順序表示機能]

5.2.8 [生成ドキュメント] タブ

[解析生成オプション設定]画面の[生成ドキュメント]タブでは、ソース解析情報として生成するドキュメント(HTML ファイル形式または CSV ファイル形式)を選択します。また、ソース解析情報のドキュメントを格納する出力先フォルダを指定します。

注意事項

- ・出力先フォルダのデフォルト値は%USERPROFILE%COBOL Analyzer です。このフォルダの下に、解析対象プロジェクト名のフォルダが生成され、ソース解析情報が生成されます。
- ・出力先フォルダは、空のフォルダを設定してください。過去に生成したソース解析情報が出力先フォルダに存在していると、生成済みのソース解析情報が上書きされます。また、古いドキュメントが残ってしまうことがあります。
- ・出力先フォルダには、空白で終わる名称のフォルダを指定しないでください。正しく解析できないことがあります。
- ・出力先フォルダが存在しない場合でも、自動的にフォルダが作成されます。ただし、存在しないドライブで始まる絶対パス名を指定すると、生成処理でエラーになります。

- [生成オプション設定] 画面ですべてのチェックボックスをオフにしても、出力先フォルダに「index.html」ファイル、「Stylesheet」フォルダおよび「Files」フォルダが生成されます。

図 5-11 [解析生成オプション設定] 画面の [生成ドキュメント] タブ

解析生成オプション設定

ソース解析オプションファイル: C:\Users\COBOL\AppData\Local\Temp\MNG2058.xml

コンパイルオプション 主プログラム指定 プログラムコメント抽出 簡コメント抽出

呼出先プログラム解決 引数整合性チェック その他の解析オプション 生成ドキュメント

☒ HTMLファイル

☒ 一覧情報CSV

☒ COBOLソースファイル一覧

☒ プログラム一覧

☒ セクション一覧

☒ ファイル一覧

☒ データ項目一覧

☒ 不整合引数一覧

☒ 実行順序リスト

☒ 関係情報CSV

☒ プログラム呼出関係

☒ 登録集原文参照関係

☒ ファイル参照関係

☒ 表(DB)参照関係

出力先フォルダ

C:\Users\COBOL\COBOL Analyzer 参照...

OK キャンセル

[生成ドキュメント]

生成するソース解析情報のドキュメントを選択します。

指定できる生成ドキュメントを次の表に示します。

表 5-1 指定できる生成ドキュメント

項番	生成ドキュメントオプション	説明
1	HTML ファイル	HTML ドキュメントのファイル一式を生成します。
2	一覧情報 CSV	一覧情報 CSV ファイル（子項目）の一括操作
3	COBOL ソースファイル一覧	COBOL ソースファイル一覧 CSV ファイルを生成します。
4	プログラム一覧	プログラム一覧 CSV ファイルを生成します。
5	セクション一覧	セクション一覧 CSV ファイルを生成します。
6	ファイル一覧	ファイル一覧 CSV ファイルを生成します。
7	データ項目一覧	データ項目一覧 CSV ファイルを生成します。
8	不整合引数一覧	不整合引数一覧 CSV ファイルを生成します。

項番	生成ドキュメントオプション	説明
9	実行順序リスト	実行順序リスト CSV ファイルを生成します。
10	関係情報 CSV	関係情報 CSV ファイル（子項目）の一括操作
11	プログラム呼出関係	プログラム呼出関係 CSV ファイルを生成します。
12	登録集原文参照関係	登録集原文参照関係 CSV ファイルを生成します。
13	ファイル参照関係	ファイル参照関係 CSV ファイルを生成します。
14	表(DB)参照関係	表(DB)参照関係 CSV ファイルを生成します。

■ 注意事項

〔解析生成オプション設定〕画面の〔生成ドキュメント〕タブで HTML ファイルの生成を無効（チェックボックスをオフ）にして解析／生成を実行した場合、すべての COBOL ソースファイルが生成対象外になります。COBOL ソースファイルの生成状態や生成日付も変更されません。ただし、〔解析生成オプション設定〕画面の〔生成ドキュメント〕タブで生成が有効（チェックボックスがオン）になっている CSV ファイルは生成されます。

〔出力先フォルダ〕

ソース解析情報の出力先フォルダを指定します。ここで指定する出力先フォルダは、HTML ファイル形式と CSV ファイル形式で共通です。

デフォルト値は%USERPROFILE%\COBOL Analyzer です。ここで指定したフォルダの下に解析対象プロジェクト名のフォルダが生成され、ソース解析情報の HTML フォルダと CSV フォルダが生成されます。

〔参照〕ボタン

〔フォルダの参照〕ダイアログが表示されます。〔出力先フォルダ〕に指定するフォルダを選択します。

6

コマンドリファレンス

COBOL ソース解析のコマンドの文法について説明します。

6.1 COBOL ソース解析で使用するコマンド

COBOL ソース解析で使用するコマンドの説明です。

cblcsa コマンド – バッチモードで COBOL ソース解析を実行する

cblcsa コマンドは、COBOL ソース解析をコマンドプロンプトから操作するときに使います。

コマンドの実行が完了すると、解析状況、生成状況、および実行結果が標準エラー出力に出力されます。

形式

ソース解析プロジェクトの操作（作成、削除、参照）

ソース解析プロジェクトを作成する場合

```
cblcsa -Project ソース解析プロジェクト名 -Create
```

ソース解析プロジェクトの名称を変更する場合

```
cblcsa -Project ソース解析プロジェクト名 -Rename 変更後ソース解析プロジェクト名
```

ソース解析プロジェクトを削除する場合

```
cblcsa -Project ソース解析プロジェクト名 -Delete
```

ソース解析プロジェクトの登録情報を取得する場合

```
cblcsa -Project ソース解析プロジェクト名 -GetStatus
```

ソース解析プロジェクトの一覧を表示する場合

```
cblcsa -ListProject
```

データベースの初期化

ソース解析用データベースを初期化する場合

```
cblcsa -InitializeDatabase
```

ソース解析プロジェクトの COBOL ソースファイルの操作（登録、削除、参照）

ソースファイル名を登録する場合

```
cblcsa -Project ソース解析プロジェクト名 -Add {-Source ソースファイル名 [;ソースファイル名] ... | -SourceFolder ソースフォルダ名 [;ソースフォルダ名] ... [-r] }
```

登録されたソースファイル名を削除する場合

```
cblcsa -Project ソース解析プロジェクト名 -Remove -Source {ソースファイル名 [;ソースファイル名] ... | ALL}
```

登録されたソースファイル名の状態を一覧表示する場合

```
cblcsa -Project ソース解析プロジェクト名 -ListSrcStatus
```

ソース解析プロジェクトの解析生成オプションの操作（取り出し，登録）

登録された解析生成オプションをソース解析オプションファイルに出力する場合

```
cblcsa -Project ソース解析プロジェクト名 -ExportOption ソース解析オプションファイル名
```

オプションファイルから解析生成オプションを登録する場合

```
cblcsa -Project ソース解析プロジェクト名 -ImportOption ソース解析オプションファイル名
```

ソース解析プロジェクトの解析生成の実行

COBOL ソース解析を実行する場合

```
cblcsa -Project プロジェクト名  
      [-Analyzer {AUTO | ALL} ]  
      [-LogFolder 出力先フォルダパス]
```

コマンドヘルプを表示する場合

```
cblcsa または  
cblcsa -? または  
cblcsa -Help
```

オプション

-Project ソース解析プロジェクト名

ソース解析プロジェクト名を指定します。

ソース解析プロジェクト名には，次の文字を含まない名前を 30 文字以下で指定します。英小文字と英大文字は区別されません。

```
「'」 「¥」 「/」 「:」 「*」 「?」 「"」 「<」 「>」 「|」 「@」 「;」，半角空白，全角空白，制御文字
```

また，先頭文字にハイフン(-)を指定しないでください。

このオプションを-Create オプションと同時に指定する場合，同じ名称のソース解析プロジェクトが COBOL ソース解析用データベースに存在してはいけません。-Create 以外のオプションと同時に指定する場合，ソース解析プロジェクト名と同じ名称のソース解析プロジェクトが COBOL ソース解析用データベースに存在している必要があります。

-Project オプションだけを指定し，ほかのオプションを指定しない場合は，-Analyzer AUTO が仮定されます。

-Create

COBOL ソース解析用データベースにソース解析プロジェクトを新規作成します。

同じ名称のソース解析プロジェクトが COBOL ソース解析用データベースに存在する場合は、エラーメッセージを出力してエラー終了します。

-Rename 変更後プロジェクト名

COBOL ソース解析用データベースのソース解析プロジェクトの名称を指定した変更後ソース解析プロジェクト名に変更します。変更後プロジェクト名は、-Project オプションのプロジェクト名の命名規則に従った名前指定してください。

-Delete

COBOL ソース解析用データベースのソース解析プロジェクトを削除します。

-GetStatus

COBOL ソース解析用データベースのソース解析プロジェクトの登録情報を表示（標準出力に出力）します。次に示す内容を表示します。

- ・ ソース解析プロジェクト名
- ・ 解析オプション情報、生成オプション情報
- ・ ソース一覧（解析状態、生成状態、ソースファイル名、解析時のソースファイル更新日時、解析日時、生成日時）

出力例

```
・ ソース解析プロジェクト名
  販売管理システム

・ 解析オプション情報
[コンパイラオプション]
コンパイラオプション :
  -Comp5 -Compativ3 -Compati85, All
コンパイラ環境変数 :
  CBLERRMAX=100
  CBLFIX=.cob .xxx
  CBLLIB=C:¥Program Files¥HITACHI¥COBOL2002 64bit¥COPY;C:¥temp¥COPY

[主プログラム指定]
主プログラムを自動検出 : ON
USING検出条件
  引数なし : ON
  引数1個で英数字である : OFF
  引数1個で可変長文字列型の集団項目である : ON
  引数2個で2進整数とポインタである : ON
RETURNING検出条件
  RETURNING指定なし : ON
主プログラム名の一覧の文字列 :

[プログラムコメント抽出]
```

プログラムコメントを抽出する : OFF
プログラムコメント行の検索場所
プログラム名段落の前 : ON
プログラム名段落の後 : OFF
プログラムコメント行の検索条件
抽出キー文字列で始まるコメント行(プログラムコメントは抽出キーの後の文字列) : ON
抽出キー文字列で始まるコメント行の次行 : OFF
抽出キー文字列 : ''
プログラム名段落から [] 行目(1~99)の位置にあるコメント行 : OFF

[節コメント抽出]

節コメントを抽出する : OFF
節コメント行の検索場所
節見出しの前 : ON
節見出しの後 : OFF
節コメント行の検索条件
抽出キー文字列で始まるコメント行(節コメントは抽出キーの後の文字列) : ON
抽出キー文字列で始まるコメント行の次行 : OFF
抽出キー文字列 : ''
節見出しから [] 行目(1~99)の位置にあるコメント行 : OFF

[呼出先プログラム解決]

COBOL2002 サービスルーチンファイル :

COBOL2002 一意名呼び出し対応付けファイル :

[引数整合性チェック]

引数の個数を検査する : ON
実引数個数>仮引数個数の場合は検出しない : OFF
引数の受渡方法を検査する : ON
引数のデータ型を検査する : ON
引数のサイズを検査する : ON
次の条件をすべて満たすケースを検出しない : OFF
・ BY REFERENCEまたはBY CONTENTである
・ 一方または両方が英数字集団項目である
・ 実引数サイズ>仮引数サイズである
一方が可変長集団項目の場合はサイズ検査対象外とする : ON
引数の集団項目の構造を検査する : OFF
対応する仮引数のない実引数を検出する : OFF
対応する実引数のない仮引数を検出する : OFF

[その他]

レベル番号が02~49,66のデータ項目の情報を抽出する : ON
プログラムの制御フロー情報を抽出する : ON

・ 生成オプション情報

出力先フォルダ :
D:¥Temp¥COBOL Analyzer

[HTML]

すべてのHTMLドキュメント : ON

[CSV]

COBOLソースファイル一覧 : ON
プログラム一覧 : ON
セクション一覧 : ON
ファイル一覧 : ON
データ項目一覧 : ON
不整合引数一覧 : ON

実行順序リスト : ON
プログラム呼出関係 : ON
登録集原文参照関係 : ON
ファイル参照関係 : ON
表(DB)参照関係 : ON

・ソース一覧 (解析状態, 生成状態, ソースファイル名, ソースファイル更新日時, 解析日時, 生成日時)
解析済, 生成済, C:¥販売管理システム¥伝票作成¥伝票作成メイン.cbl, 2002/11/28 15:41:22, 2021/08/04 22:40:15, 2021/08/04 22:40:17
解析済, 生成済, C:¥販売管理システム¥入金処理¥顧客マスタ検索.CBL, 1999/10/29 17:14:20, 2021/08/04 22:40:15, 2021/08/04 22:40:17
解析済(古い), 生成済(古い), C:¥販売管理システム¥入金処理¥入金処理メイン.cbl, 1999/11/15 21:22:52, 2021/08/04 22:40:15, 2021/08/04 22:40:17

(凡例)

解析オプション情報, および生成オプション情報

オプションの指定有無を示します。

- ON: オプションが指定されている。
- OFF: オプションが指定されていない。

ソース一覧の [解析状態]

COBOL ソースファイルの解析状態を示します。

- 解析済: 解析済み。
- 解析済 (古い): 解析済みで「解析済」または「警告」の状態だったが, 前回解析時点から COBOL ソースファイル (取り込んでいる登録集原文ファイルを含む) が更新されている。再解析を行う必要がある。
- 未解析: 解析されていない。
- 警告: 解析済みだが, 警告エラーが検出された。
- エラー: 解析できないエラーが検出された。

ソース一覧の [生成状態]

COBOL ソースファイルのドキュメントの生成状態を示します。

- 生成済: 生成済み。
- 生成済 (古い): 生成済みで「生成済」または「警告」の状態だったが, 前回解析時点から COBOL ソースファイル (取り込んでいる登録集原文ファイルを含む) が更新されている。または解析/生成の中断で解析は完了しているが生成はされていない。再生成を行う必要がある。
- 未生成: 生成されていない。
- 警告: 生成済みだが, 警告エラーが検出された。
- エラー: 生成できないエラーが検出された。

ソース一覧の [ソースファイル名]

解析または生成対象の COBOL ソースファイル名を絶対パス名で表示します。

-ListProject

COBOL ソース解析用データベースのソース解析プロジェクトの一覧を表示（標準出力に出力）します。
次に示す内容を表示します。

- ソース解析プロジェクト名
- プロジェクトの解析状態
- 最終解析日時
- プロジェクトの生成状態
- 最終生成日時

このオプションを指定した場合は、-Help(-?), -InitializeDatabase, -LogFolder 以外のほかのオプションはすべて無視されます。

出力例

解析状態	最終解析日時	生成状態	最終生成日時	ソース解析プロジェクト名
解析済み	2017/08/28 16:34	生成済み	2017/08/28 16:34	ProjectAlpha
未解析あり	2017/08/28 16:34	未生成		ProjectBeta
解析済み	2017/08/28 16:34	未生成あり	2017/08/28 16:34	ProjectGamma
未解析		未生成		ProjectDelta

(凡例)

[解析状態]

ソース解析プロジェクトの解析状態を示します。

- 解析済み：すべての COBOL ソースファイルが解析済み。
- 未解析あり：一部の COBOL ソースファイルが未解析。
- 未解析：すべての COBOL ソースファイルが未解析。

[最終解析日時]

最後に解析または生成を実行したときの解析処理の終了日時を [yyyy/mm/dd hh:mm] 形式で示します（yyyy：西暦年，mm：月，dd：日，hh：時，mm：分）。

[生成状態]

ソース解析プロジェクトの生成状態を示します。

- 生成済み：すべての COBOL ソースファイルが生成済み。
- 未生成あり：一部の COBOL ソースファイルが未生成。
- 未生成：すべての COBOL ソースファイルが未生成。

[最終生成日時]

最後に解析または生成を実行したときの生成処理の終了日時を [yyyy/mm/dd hh:mm] 形式で示します（yyyy：西暦年，mm：月，dd：日，hh：時，mm：分）。

-InitializeDatabase

COBOL ソース解析用データベースを初期化します。-Delete オプションよりも高速にデータベースの内容を削除できます。このオプションを指定した場合は、-Help(-?), -LogFolder 以外のほかのオプションはすべて無視されます。

このオプションは [Ctrl] + [C] でキャンセルできません。データベースを初期化しても問題ないことを入念に確認してから、-InitializeDatabase オプションを使用してください。

-Add

ソース解析プロジェクトに COBOL ソースファイル名を登録します。

-Remove

ソース解析プロジェクトからソースファイル名を削除します。

-Source {ソースファイル名 [:ソースファイル名] ... | ALL}

-Add オプションの場合

ソース解析プロジェクトに登録する COBOL ソースファイル（COBOL ソースファイルの拡張子をもつファイル）を指定します。COBOL ソースファイル以外は登録されません。

-Add オプションのときは ALL を指定できません。ワイルドカード「*」「?」を使用できます。ただし、フォルダ名の部分にはワイルドカードは指定できません。

-Remove オプションの場合

ソース解析プロジェクトから削除する COBOL ソースファイルを絶対パス名で指定します。ソース解析プロジェクトのすべての COBOL ソースファイルを削除するときは ALL を指定します。

-SourceFolder ソースフォルダ名 [:ソースフォルダ名] ...

COBOL ソースファイルを格納したフォルダを指定します。指定されたソースフォルダ内の COBOL ソースファイル（COBOL ソースファイルの拡張子をもつファイル）が登録対象となります。COBOL ソースファイル以外は登録されません。

-r

-SourceFolder オプションで指定したソースフォルダの、すべての下位フォルダの COBOL ソースファイルをソース解析プロジェクトへの登録対象にします。

-ListSrcStatus

ソース解析プロジェクトに登録されたソースファイルの解析生成状態の一覧を表示（標準出力に出力）します。次の内容がコンマ区切りで出力されます。

- 解析状態
- 生成状態
- ソースファイル名（絶対パス名）

- ・ ソースファイルの更新日時, 解析日時, 生成日時

出力例

```
解析状態, 生成状態, ソースファイル名, ソースファイル更新日時, 解析日時, 生成日時
解析済, 生成済, C:\temp\test01.cbl, 2017/08/28 14:31:24, 2017/08/28 17:03:24, 2017/08/28 17:03:27
エラー, 未生成, C:\temp\test02.cbl, 2017/08/28 15:56:02, ,
解析済, 生成済, C:\temp\test03.cbl, 2017/08/28 14:32:28, 2017/08/28 17:03:26, 2017/08/28 17:03:27
解析済(古い), 生成済(古い), C:\temp\test04.cbl, 2017/08/28 17:00:50, 2017/08/28 17:03:26, 2017/08/28 17:03:27
```

(凡例)

[解析状態]

COBOL ソースファイルの解析状態を示します。

- ・ 解析済：解析済み。
- ・ 解析済 (古い)：解析済みで「解析済」または「警告」の状態だったが、前回解析時点から COBOL ソースファイル（取り込んでいる登録集原文ファイルを含む）が更新されている。再解析を行う必要がある。
- ・ 未解析：解析されていない。
- ・ 警告：解析済みだが、警告エラーが検出された。
- ・ エラー：解析できないエラーが検出された。

[生成状態]

COBOL ソースファイルのプログラム固有ドキュメント (HTML) の生成状態を示します。

- ・ 生成済：生成済み。
- ・ 生成済 (古い)：生成済みで「生成済」または「警告」の状態だったが、前回解析時点から COBOL ソースファイル（取り込んでいる登録集原文ファイルを含む）が更新されている。または解析／生成の中断で解析は完了しているが生成はされていない。再生成を行う必要がある。
- ・ 未生成：生成されていない。
- ・ 警告：生成済みだが、警告エラーが検出された。
- ・ エラー：生成できないエラーが検出された。

[ソースファイル名]

解析または生成対象の COBOL ソースファイル名を絶対パス名で表示します。

[ソースファイル更新日時]

COBOL ソースファイルの更新日時を [yyyy/mm/dd hh:mm:ss] 形式で表示します (yyyy：西暦年, mm：月, dd：日, hh：時, mm：分, ss：秒)。

[解析日時]

COBOL ソースファイルの解析が完了した日時を [yyyy/mm/dd hh:mm:ss] 形式で表示します (yyyy：西暦年, mm：月, dd：日, hh：時, mm：分, ss：秒)。

解析状態が「未解析」または「エラー」の場合、解析日時は出力されません。

[生成日時]

COBOL ソースファイルのプログラム固有ドキュメント (HTML) の生成が完了した日時を [yyyy/mm/dd hh:mm:ss] 形式で表示します (yyyy: 西暦年, mm: 月, dd: 日, hh: 時, mm: 分, ss: 秒)。

生成状態が「未生成」または「エラー」の場合、生成日時は出力されません。

-ExportOption ソース解析オプションファイル名

解析生成オプションをソース解析オプションファイル (.xml) に出力します。

拡張子が.xml 以外の場合、エラーが出力されて、エラー終了します。

-ImportOption ソース解析オプションファイル名

ソース解析オプションファイル (.xml) の解析生成オプションをソース解析プロジェクトに登録します。

-Analyzer {AUTO | ALL}

COBOL ソースファイルの解析および生成の対象範囲を指定します。

AUTO: 解析および生成が必要な COBOL ソースファイルを自動的に選出して解析および生成の対象とします。

次の COBOL ソースファイルが解析対象となります。

- 解析オプションを変更したあとで、再解析していない COBOL ソースファイル
- 前回の解析でエラーが出力された COBOL ソースファイル
- COBOL ソースファイルの更新日時、または直接的もしくは間接的に取り込んでいる登録集原文ファイルの更新日時が、前回解析時の更新日時と異なる COBOL ソースファイル

次の COBOL ソースファイルが生成対象となります。

- 生成オプションを変更したあとで、再生成していない COBOL ソースファイル
- 前回の生成でエラーが出力された COBOL ソースファイル
- 解析対象となった COBOL ソースファイル
- COBOL ソースファイルのうち、前回の生成日時が前回の解析日時より古いもの。

ALL: すべての COBOL ソースファイル

-LogFolder 出力先フォルダパス

解析ログおよび操作ログの出力先フォルダを絶対パスまたはカレントフォルダからの相対パスで指定します。

省略すると、COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) に定義されている解析ログおよび操作ログの出力先フォルダに出力されます。

フォルダ名をダブルクォーテーション (") で囲んで指定する場合は、フォルダ名の末尾に円記号 (¥) は指定しないでください。Windows の仕様で円記号 (¥) がエスケープ文字の扱いになり、ダブルクォーテーションがフォルダ名の一部になってエラーになります。

解析ログおよび操作ログの詳細は、それぞれ、「9.2.1 解析ログを参照する」および「9.2.2 操作ログを参照する」を参照してください。

-? または -Help

コマンドヘルプが表示されます。ほかのオプションはすべて無視されます。cblcsa コマンドの引数をすべて省略した場合も、コマンドヘルプが表示されます。

出力例

実行状況表示

解析または生成の実行状況をコンソールに表示（標準エラー出力に出力）します。

解析状況

```
[2017/08/02 19:54:46],[情報],[システム],KEDL1231A-I プロジェクト(PROJECT1)の解析処理を開始しました。
[2017/08/02 19:54:46],[情報],[解析],KEDL1261A-I (1/3) 'D:¥SAMPLE¥ソースファイル1.cbl'を解析中...
[2017/08/02 19:54:47],[情報],[解析],KEDL1261A-I (2/3) 'D:¥SAMPLE¥ソースファイル2.cbl'を解析中...
[2017/08/02 19:54:47],[情報],[解析],KEDL1261A-I (3/3) 'D:¥SAMPLE¥ソースファイル3.cbl'を解析中...
[2017/08/02 19:54:48],[情報],[システム],KEDL1232A-I プロジェクト(PROJECT1)の解析処理を終了しました。
```

生成状況

```
[2017/08/02 19:54:48],[情報],[システム],KEDL1233A-I プロジェクト(PROJECT1)の生成処理を開始しました。
[2017/08/02 19:54:48],[情報],[生成],生成開始: システム固有: プログラム階層図
[2017/08/02 19:54:49],[情報],[生成],生成開始: プログラム固有: プログラム名=プログラム名1 (進捗: 1/3)
[2017/08/02 19:54:50],[情報],[生成],生成開始: プログラム固有: プログラム名=プログラム名2 (進捗: 2/3)
[2017/08/02 19:54:51],[情報],[生成],生成開始: プログラム固有: プログラム名=プログラム名3 (進捗: 3/3)
[2017/08/02 19:54:53],[情報],[生成],生成開始: システム固有: ドキュメント名
[2017/08/02 19:54:54],[情報],[生成],生成開始: プロジェクト固有: プロジェクトサマリ
[2017/08/02 19:54:55],[情報],[システム],KEDL1234A-I プロジェクト(PROJECT1)の生成処理を終了しました。
```

実行結果表示

解析または生成の成功、警告、失敗の集計結果をコンソールに表示（標準エラー出力に出力）します。

[2017/08/02 19:55:55],[情報],[実行結果],	解析	生成
[2017/08/02 19:55:55],[情報],[実行結果], 成功:	4	4
[2017/08/02 19:55:55],[情報],[実行結果], 警告:	0	0

[2017/08/02 19:55:55],[情報],[実行結果], 失敗:	0	0
[2017/08/02 19:55:55],[情報],[実行結果], 合計:	4 / 4	4 / 4

解析および生成の「成功」「失敗」「警告」「合計」の数の意味を次に示します。

種別	解析	生成
成功	解析状態が「解析済」で解析を完了した COBOL ソースファイル数	生成状態が「生成済」で生成を完了した COBOL ソースファイル数
警告	解析状態が「警告」で解析を完了した COBOL ソースファイル数	生成状態が「警告」で生成を完了した COBOL ソースファイル数
失敗	解析状態が「エラー」で解析を終了した COBOL ソースファイル数	生成状態が「エラー」で生成を終了した COBOL ソースファイル数
合計	分子は解析の成功、警告、失敗の合計数 分母は解析対象の COBOL ソースファイル数	分子は生成の成功、警告、失敗の合計数 分母は生成対象の COBOL ソースファイル数

注意事項

- システム全体の解析（ソースファイル間で参照されるプログラムや共用される登録集原文の解析）でエラーを検出した場合、またはシステム全体のドキュメント生成で警告またはエラーを検出した場合、次のように解析や生成の「警告」と「失敗」にカウントされます。このため、合計の分子は分母より 1 大きくなる場合があります。
 - システム全体の解析でエラーを検出した場合は、解析の「失敗」の欄に 1 を加算します。
 - システム全体のドキュメント生成が警告メッセージのみで終了した場合は、生成の「警告」の欄に 1 を加算します。
 - システム全体のドキュメント生成でエラーを検出した場合は、生成の「失敗」の欄に 1 を加算します。
- システム全体の解析でエラーとなっている場合、解析情報が正しくデータベースに抽出できていないことがあります。エラーメッセージのエラー情報を基にエラー要因を排除した上で、コマンドプロンプト上で cblcsa コマンドの -Analyzer ALL オプションを使用して、すべての COBOL ソースファイルを再解析してください。

cblcsa コマンドのコンソール出力例

```
> cblcsa
[2017/08/02 19:56:46],[情報],[システム], KEDL1231A-I プロジェクト(新しいプロジェクト)の解析処理を開始しました。
[2017/08/02 19:56:46],[情報],[解析], KEDL1261A-I (1/10) 'D:¥SAMPLE¥オプション処理 1.cbl' を解析中...
[2017/08/02 19:56:47],[情報],[解析], KEDL1261A-I (2/10) 'D:¥SAMPLE¥オプション処理 2.cbl' を解析中...
[2017/08/02 19:56:47],[情報],[解析], KEDL1261A-I (3/10) 'D:¥SAMPLE¥オプション処理 3.cbl' を解析中...
...
[2017/08/02 19:56:57],[情報],[システム], KEDL1232A-I プロジェクト(新しいプロジェクト)の解析処理を終了しました。
[2017/08/02 19:56:57],[情報],[システム], KEDL1233A-I プロジェクト(新しいプロジェクト)の生成処理を開始しました。
```



```

[2017/08/02 19:56:57],[情報],[生成], 生成開始: システム固有:プログラム階層図
[2017/08/02 19:56:58],[情報],[生成], 生成開始: プログラム固有:プログラム名=OPTPRO0C1 (進捗: 1 /10)
[2017/08/02 19:56:58],[情報],[生成], 生成開始: プログラム固有:プログラム名=OPTPRO0C2 (進捗: 2 /10)
[2017/08/02 19:56:58],[情報],[生成], 生成開始: プログラム固有:プログラム名=OPTPRO0C3 (進捗: 3 /10)
...
[2017/08/02 19:57:14],[情報],[生成], 生成開始: システム固有:表(DB)参照関係CSV
[2017/08/02 19:57:14],[情報],[生成], 生成開始: プロジェクト固有:プロジェクトサマリ
[2017/08/02 19:57:15],[情報],[システム], KEDL1234A-I プロジェクト(新しいプロジェクト)の生成処理を終了しました。
[2017/08/02 19:57:15],[情報],[実行結果],          解析      生成
[2017/08/02 19:57:15],[情報],[実行結果], 成功:      10      10
[2017/08/02 19:57:15],[情報],[実行結果], 警告:       0       0
[2017/08/02 19:57:15],[情報],[実行結果], 失敗:       0       0
[2017/08/02 19:57:15],[情報],[実行結果], 合計: 10 / 10  10 / 10

```

終了コード

終了コード	意味	メッセージとの対応
0	正常終了しました。	重要度 E のエラーはありません。
1	エラーが発生して終了しました。	重要度 E のエラーが 1 つ以上あります。

注意事項

- 同じオプションが複数指定された場合、あとに指定されたものだけ有効になります。
- オプション名の大文字小文字は区別しません。
- オプションの引数に空白文字、半角アンパサンド文字 (&), 半角キャレット文字 (^), 半角右丸括弧文字 (cblcsa コマンド名の左に, 半角左丸括弧文字を書いたとき) を含む場合は, ダブルクォーテーション (") で囲む必要があります。
- オプションの引数にハイフン (-) で始まる文字列は指定できません。
- オプション内を区切るために使用するセミコロン (;) の前後に空白文字を指定しないでください。
- パス区切り文字として, 斜線 (/) を使用したパス名は使用できません。パス区切り文字には円記号 (¥) を使用してください。パス名が斜線 (/) を含む場合の動作は保証しません。
- DB ユティリティまたは COBOL ソース解析が組み込みデータベースにアクセスしている場合, およびほかのユーザが組み込みデータベースにアクセスしている場合, エラーメッセージを出力してコマンドが終了します。
- cblcsa コマンドは同時に複数起動できません。複数起動した場合, エラーメッセージが出力されてコマンドが停止することがあります。cblcsa コマンドを実行する場合は, 並列で動作しないようにバッチファイルなどを使用して実行してください。
- cblcsa コマンドの実行中に DB ユティリティや COBOL ソース解析を起動した場合, エラーメッセージが出力されてコマンドが停止することがあります。このため, cblcsa コマンドを実行する場合は, DB ユティリティや COBOL ソース解析を起動しないでください。

- オプションの指定順序は問いません。
- Windows の環境変数 TEMP が定義されていない場合、エラーメッセージが出力されます。
- ソースファイル名およびソースフォルダ名に、セミコロン (;), アポストロフィ ('), コンマ (,) を含むパス名は指定できません。
- -Analyzer オプション以外のオプションの実行は、[Ctrl] + [C] キーで強制終了できません。
- [Ctrl] + [C] キーを使ってコマンドの実行を強制終了した場合、解析中であれば解析中のソースファイルの解析処理がロールバックされ、その直前の COBOL ソースファイルの解析情報までがデータベースに格納されます。また、生成中の場合は、生成中のドキュメントはエラー扱いとして出力が中断され、コマンドが終了します。再度、解析または生成を実施してエラー状態が解消されるまでは、システム固有ドキュメントの妥当性は保証しません。
- -Analyzer ALL による解析および生成を [Ctrl] + [C] キーを使って中断した場合、その解析および生成を中断した位置から再開することはできません。-Analyzer ALL で再実行して、最初から解析および生成をやり直す必要があります。
- -Analyzer AUTO による解析および生成を [Ctrl] + [C] キーを使って中断した場合、その解析および生成は -Analyzer AUTO による再実行で中断した COBOL ソースファイルから再開できます。
- -Analyzer AUTO による解析および生成では、プロジェクトの解析オプションや生成オプションが変更されていても、すべての COBOL ソースファイルが自動的に解析対象や生成対象になることはありません。解析オプションや生成オプションを変更した場合は、必ず -Analyzer ALL ですべての COBOL ソースファイルを解析対象と生成対象としてください。
- コマンドプロンプトで COBOL2002 のコンパイラ環境変数を設定しても、プロジェクトの解析生成オプションに同じコンパイラ環境変数が登録されている場合は、コマンドプロンプトの環境変数の設定値は無視され、解析生成オプションに登録されている環境変数の設定値が採用されます。
- コマンドプロンプトで COBOL2002 のコンパイラ環境変数にフォルダ名やファイル名を指定する場合、必ず絶対パス名で指定してください。
- バッチモードで作成したソース解析プロジェクトを開発マネージャから操作する場合、ソース解析プロジェクトの使用開始をしても、開発マネージャ画面に解析結果が表示されません。開発マネージャでソース解析プロジェクトの使用開始、解析生成オプションの設定、および COBOL ソース解析の実行を行ってください。
- 解析または生成でデータベース管理システムや ODBC のエラーが発生した場合、エラーメッセージを出力してコマンドを終了します。この場合、解析状態や生成状態をデータベースに格納できないため、次の現象が発生します。
 - エラーとなった COBOL ソースファイルの解析状態や生成状態が「エラー」と表示されない。
 - COBOL ソースファイルの解析状態および生成状態と、プロジェクトの解析状態および生成状態が不一致になる。

エラーメッセージのエラー情報を基にエラー要因を排除した上で、コマンドプロンプト上で cblcsa コマンドの -Analyzer ALL オプションを使用して、すべての COBOL ソースファイルを再解析してください。

cblcsaSetOptions コマンド –ソース解析オプションファイルを編集する

cblcsaSetOptions コマンドは、[解析生成オプション設定] 画面を起動して、ソース解析オプションファイルに設定されている解析生成オプションを表示・編集するときに使用します。

編集した内容は指定したソース解析オプションファイルに上書きします。

形式

```
cblcsaSetOptions -OptionFile ソース解析オプションファイル
```

オプション

-OptionFile ソース解析オプションファイル

ソース解析オプションファイルのパス名を指定します。

ソース解析オプションファイルは、cblcsa コマンドの-ExportOption オプションで出力します。

-Help

コマンドヘルプが表示されます。ほかのオプションはすべて無視されます。

注意事項

- 異なる環境から同一のソース解析オプションファイル名が指定された場合は、複数起動できません。
- オプション名の太文字小文字は区別しません。
- オプションの引数に空白文字を含む場合は、ダブルクォーテーション (") で囲む必要があります。
- 開発マネージャで作成したソース解析プロジェクトに対して、バッチモードで解析結果の出力先を変更した場合、開発マネージャには変更前の解析結果が表示されます。ソース解析プロジェクトの使用開始をしたあとに、解析生成オプションの設定、および COBOL ソース解析の実行を行ってください。

関連項目

- [「5.2 \[解析生成オプション設定\] 画面」](#)
- [「cblcsa コマンド – バッチモードで COBOL ソース解析を実行する」](#)

7

解析結果のリファレンス（HTML ファイル形式）

COBOL ソース解析で出力する、ソース解析情報（HTML ファイル形式）の内容について説明します。

7.1 ソース解析情報の画面遷移

ソース解析情報を参照するときは、Web ブラウザで解析結果のプロジェクトサマリ画面 (index.html) を表示して、参照する項目のリンクをクリックします。

ソース解析情報の画面遷移を次に示します。

図 7-1 ソース解析情報の画面遷移 (システム全体の情報)

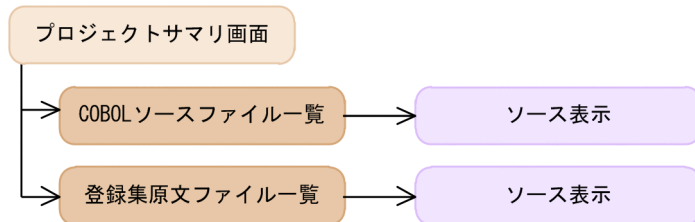


図 7-2 ソース解析情報の画面遷移 (プログラム固有の情報)

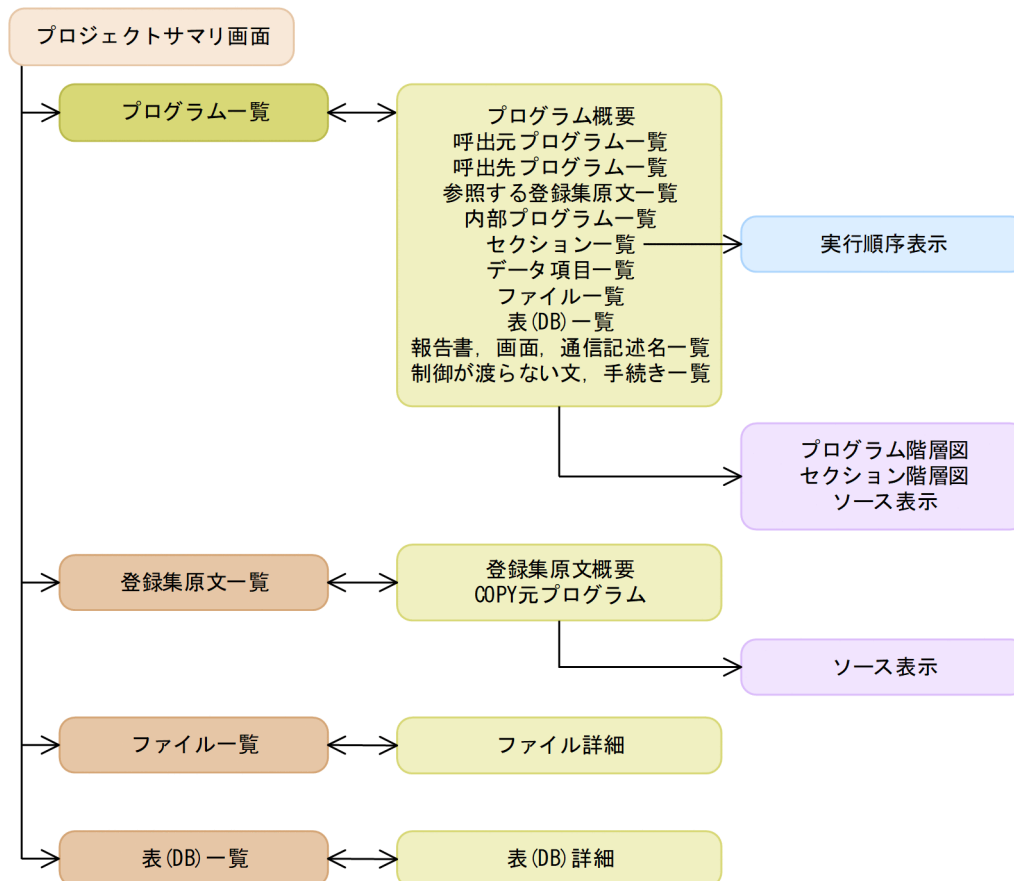


図 7-3 ソース解析情報の画面遷移（各種の分布情報）

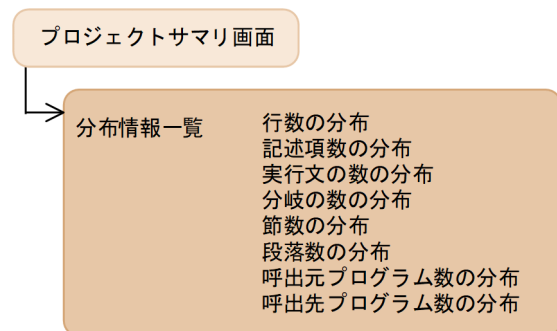
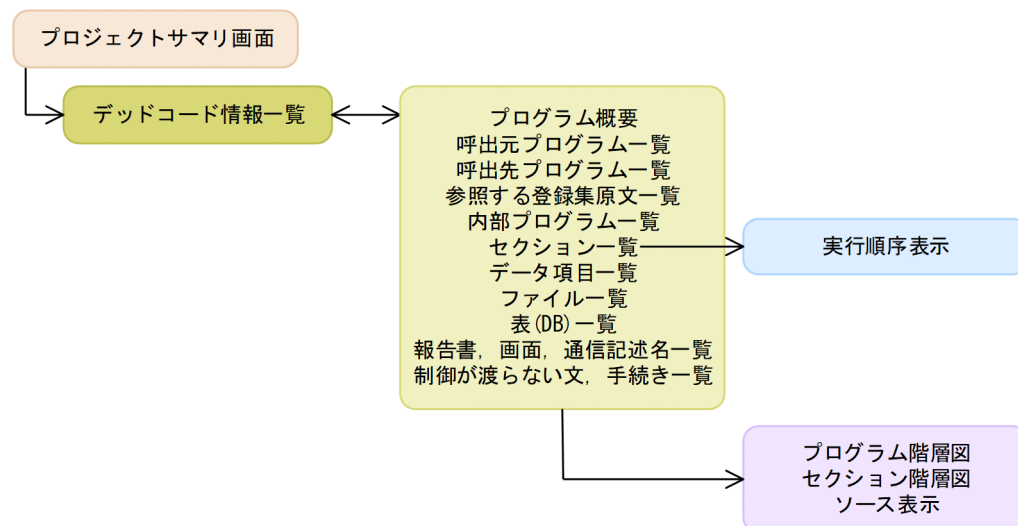


図 7-4 ソース解析情報の画面遷移（デッドコード情報）



7.2 プロジェクトサマリ画面

プロジェクトサマリ画面は、ソース解析情報（HTML ドキュメント）の出力先フォルダにある index.html をクリックすることで表示されるソース解析情報の表示画面のトップページです。この画面には各種解析情報の個数が表示され、その個数は各種解析情報の画面へのリンクとなっています。プロジェクトサマリ画面は、各種解析情報の目次としての役割を担う画面です。画面のタイトルには、「プロジェクトサマリ - プロジェクト名」の形式でプロジェクト名が表示されます。

図 7-5 プロジェクトサマリ画面

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ - 入在庫管理システム

2021/02/09 01:30:30

一覧情報					
項目	全体	解析済ファイル	未解析ファイル		
COBOLソースファイル	10	10	0		
登録集原文ファイル	13	12	1		

項目	全体	使用	未使用	未登録	重複
COBOLプログラム	10	9	1	0	0
登録集原文	13	12	1	0	
ファイル	3	3	0		
表(DB)	3				

分布状況 対象プログラム数=10本

[分布全体表示](#)

項目	平均(ノプログラム)	最小	最大
全体的行数	156	21	335
記述項数	48	4	102
実行文の数	44	2	107
分岐の数	13	0	31
節数	6	1	14
段落数	11	0	28
呼出元プログラム数	0	0	2
呼出先プログラム数	0	0	3

デッドコード情報 対象プログラム数=4本

項目	全数	未使用の外部プログラム	未使用の内部プログラム	未使用のファイル	未使用の報告書、画面、通信記述名	未使用のデータ項目	制御が渡らない文、手続き
デッドコード	5	1	0	0	0	4	0

[ページの先頭へ](#)

100%

「一覧情報」の項目の説明

[COBOL ソースファイル]

解析した COBOL ソースファイルの個数です。

・[全体]

全体の COBOL ソースファイルの個数です。クリックすると、[COBOL ソースファイル一覧] 画面に遷移します。

・[解析済ファイル]

解析済み COBOL ソースファイルの個数です。

- [未解析ファイル]

未解析の COBOL ソースファイルの個数です。

[登録集原文ファイル]

登録集原文ファイルの個数です。

- [全体]

全体の登録集原文ファイルの個数です。クリックすると、[登録集原文ファイル一覧] 画面に遷移します。

- [解析済ファイル]

解析済みの登録集原文ファイルの個数です。

- [未解析ファイル]

未解析の登録集原文ファイルの個数です。

[COBOL プログラム]

解析した COBOL プログラムの個数です。

- [全体]

解析したすべてのプログラムの個数です。クリックすると、[プログラム一覧] 画面に遷移します。なお、[未登録] の個数は含まれません。

- [使用]

使用されているプログラムの個数です。

使用されているプログラムとは、主プログラム、または主プログラムから直接的もしくは間接的に呼び出されるプログラムのことです。

- [未使用]

使用されていないプログラムの個数です。

使用されていないプログラムとは、主プログラムでないプログラムで、かつ主プログラムから直接的にも間接的にも呼び出されないプログラムのことです。

- [未登録]

CALL 文で呼出先プログラムとして指定されているが、COBOL ソースファイルがプロジェクトに登録されていないプログラムまたは登録されている COBOL ソースファイルにエラーがあって解析できていないプログラムの個数です。クリックすると、[未登録のプログラム一覧] 画面に遷移します。

- [重複]

重複するプログラム名 (PROGRAM-ID) を持つプログラムの個数です。個数をクリックすると、[重複するプログラム一覧] 画面に遷移します。

[登録集原文]

解析した登録集原文の個数です。

- [全体]

解析したすべての登録集原文の個数です。クリックすると、[登録集原文一覧] 画面に遷移します。
[未登録] の個数は含まれません。

- [使用]

COBOL ソースファイル中で実際に参照されている登録集原文の個数です。

- [未使用]

COBOL ソースファイル中で参照されていない登録集原文の個数です。

- [未登録]

COBOL ソースファイル中で参照されているが、参照されている登録集原文が見つからなくて解析されなかった登録集原文の個数です。クリックすると、[未登録の登録集原文一覧] 画面に遷移します。

[ファイル]

解析したファイルの個数です。

- [全体]

解析したすべてのファイルの個数です。クリックすると、[ファイル一覧] 画面に遷移します。

- [使用]

COBOL ソースファイル中で使用されているファイルの個数です。

- [未使用]

COBOL ソースファイル中で使用されていないファイルの個数です。

[表(DB)]

埋め込み SQL 文で使用しているすべての表の個数です。

- [全体]

解析したすべての表の個数です。クリックすると、[表(DB)一覧] 画面に遷移します。

「分布状況 対象プログラム数=使用プログラム数と未使用プログラム数の合計値 本」の項目の説明

[分布全体表示]

クリックすると各種の分布情報が棒グラフで表示されます。

[全体行数] [記述項数] [実行文の数] [分岐の数] [節数] [段落数] [呼出元プログラム数] [呼出先プログラム数]

次に示すプログラムの各種の分布情報が表示されます。

- [平均(/プログラム)]

全プログラムの平均値です。

- [最小]

最小のプログラムの値です。

- [最大]

最大のプログラムの値です。

[デッドコード]

次の個数が表示されます。

- [全数]
すべてのデッドコードの個数です。クリックすると、[デッドコード情報一覧] 画面に遷移します。
- [未使用の外部プログラム]
主プログラムではないプログラムで、どのプログラムからも呼び出されていないプログラムの個数です。
- [未使用の内部プログラム]
未使用の内部プログラムの個数です。
- [未使用のファイル]
未使用のファイルの個数です。
- [未使用の報告書、画面、通信記述名]
未使用の報告書、画面、および通信記述名の個数です。
- [未使用のデータ項目]
未使用のデータ項目の個数です。
- [制御が渡らない文、手続き]
制御が渡らない文、および手続きの個数です。

7.3 システム固有画面

システム固有の情報は、プロジェクトサマリ画面（index.html）にあるリンクから参照できます。

7.3.1 [COBOL ソースファイル一覧] 画面

COBOL ソース解析に登録された COBOL ソースファイルの一覧です。プロジェクトサマリ画面の [COBOL ソースファイル] 行の [全体] 列の個数をクリックするとパス名とファイル名でソートされて表示されます。

画面の「COBOL ソースファイル名」をクリックすると、各ファイルの [ソース表示] 画面に遷移します。

図 7-6 COBOL ソースファイル一覧



#	ファイル名	解析状態	パス名
1	オプション処理1.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
2	オプション処理2.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
3	オプション処理3.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
4	チェック処理.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
5	メニュー画面.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
6	出荷処理.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
7	詳細チェック処理.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
8	制御処理.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
9	転送処理.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl
10	入力処理.cbl	解析済	C:\¥入出荷管理システム¥入出荷管理システム¥HTML¥cl

[ファイル名]

COBOL ソース解析に登録された COBOL ソースファイル名です。

解析対象プロジェクト内で COBOL ソースファイル名が重複する場合は、同じ COBOL ソースファイルが連続して表示されます。重複する COBOL ソースファイルのそれぞれの格納場所は、[パス名] で確認してください。

[解析状態]

COBOL ソースファイルの解析状態です。表示される文字の意味を次に示します。

- 解析済

COBOL ソースファイルは解析済みです。

- 未解析

COBOL ソースファイルは未解析です。

[パス名]

COBOL ソースファイルが格納されているパスです。

7.3.2 「登録集原文ファイル一覧」画面

COBOL ソース解析に登録された登録集原文ファイルの一覧です。プロジェクトサマリ画面の「登録集原文ファイル」行の「全体」列の個数をクリックすると表示されます。

画面の「登録集原文ファイル名」をクリックすると、各ファイルの「ソース表示」画面に遷移します。

図 7-7 登録集原文ファイル一覧

#	ファイル名	解析状態	パス名
1	ACCTCD.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
2	FC351.DT.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
3	FC370.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
4	FC380.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
5	FC390.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
6	GC351.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
7	GC351.DT.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
8	GC351.ERR.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
9	PRCCD.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
10	RC340.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
11	SQLCA.COB	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
12	WK.CBL	解析済	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集
13	CBLXMLRC.CBL	未解析	C:\Program Files\Hitachi\COBOL2002\WCOPY
14	XDMDCMSG.COB	未解析	C:\出入荷管理システム\出入荷管理システム\WCOPY\登録集

[ファイル名]

登録集原文ファイル名です。

解析対象プロジェクト内で登録集原文ファイル名が重複する場合は、同じ登録集原文ファイル名が連続して表示されます。重複する登録集原文ファイルのそれぞれの格納場所は、[パス名]で確認してください。

[解析状態]

登録集原文ファイルの解析状態です。表示される文字の意味を次に示します。

- 解析済

登録集原文ファイルは解析済みです。

- 未解析

登録集原文ファイルは未解析です。

[パス名]

登録集原文ファイルが格納されているパスです。

7.3.3 【プログラム一覧】 画面

使用されているプログラムと使用されていないプログラムの一覧です。プロジェクトサマリ画面の [COBOL プログラム] 行の [全体] 列の個数をクリックすると表示されます。

画面の「プログラム名」をクリックすると、各プログラムの [プログラム詳細] 画面の「プログラム概要」に遷移します。

図 7-8 プログラム一覧

#	プログラム名	使用有無	総行数	引数の個数	実行文の数	分岐の数	呼出先プログラムの数	コメント行の文字列
1	AC000.MENU	使用	61	0	14	7	3	
2	GC351.INPUT	使用	115	2	45	18	1	
3	GC351.INPUT.DT	使用	211	2	55	16	0	
4	RC340.INPUT	使用	218	2	68	16	2	
5	RC340.SET	使用	335	1	107	31	1	
6	RC340.SND	使用	283	0	76	22	1	
7	RC340.UPD	使用	269	0	77	24	1	
8	ZC001.OPT	使用	22	0	2	0	0	
9	ZC002.OPT	使用	24	1	2	0	0	
10	ZC003.OPT	使用	21	1	2	0	0	
-	合計	-	1559	-	448	134	9	-

[プログラム名]

COBOL ソースファイルの見出し部の PROGRAM-ID です。

[使用有無]

表示される文字の意味を次に示します。

- 使用

主プログラム、または主プログラムから直接的もしくは間接的に呼び出されるプログラムです。

- 未使用

主プログラムでないプログラムで、かつ主プログラムから直接的にも間接的にも呼び出されないプログラムです。デッドコード内の CALL 文の場合、呼び出しはないものとして扱います。

[総行数]

プログラムの総行数です。

[引数の個数]

手続き部の見出しの引数の個数です。なお、RETURNING の指定は引数として表示されません。

[実行文の数]

プログラムの実行文の数です。

[分岐の数]

プログラムの分岐の数です。

[呼出先プログラム数]

プログラムから呼び出しているプログラムの数です。

[コメント行の文字列]

プログラムコメント（プログラムの機能を要約するコメント）の文字列です。

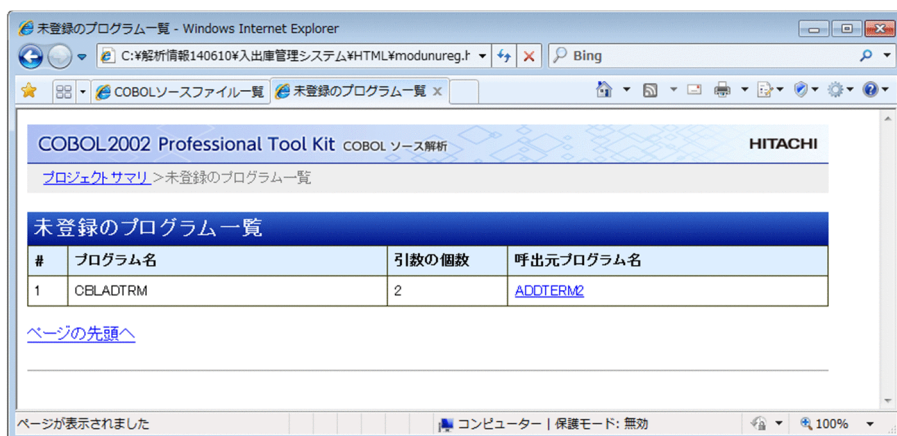
[解析生成オプション設定] 画面の [プログラムコメント抽出] タブで [プログラムコメントを抽出する] をチェックすると、条件に従って抽出されたコメントが表示されます。[プログラムコメント抽出] タブについては、[5.2.3 [プログラムコメント抽出] タブ] を参照してください。

(1) [未登録のプログラム一覧] 画面

プログラムから呼び出されているが、COBOL ソース解析の解析対象として登録されていないプログラムの一覧です。プロジェクトサマリ画面の [COBOL プログラム] 行の [未登録] 列の個数をクリックすると表示されます。

画面の「呼出元プログラム名」をクリックすると、各プログラムの [プログラム詳細] 画面の「プログラム概要」に遷移します。

図 7-9 未登録のプログラム一覧



[プログラム名]

COBOL ソース解析に登録されていない COBOL ソースファイルの見出し部の PROGRAM-ID です。

[引数の個数]

未登録プログラムを呼び出すときの、CALL 文の引数の個数です。RETURNING の指定は引数として表示されません。

[呼出元プログラム名]

未登録プログラムを呼び出しているプログラム名です。

(2) [重複するプログラム一覧] 画面

重複するプログラム名 (PROGRAM-ID) を持つプログラムの一覧です。

COBOL ソースファイルのファイル名が異なっていて、同じプログラム名 (PROGRAM-ID) が使われている場合、プログラム名 (PROGRAM-ID) の名前を変更し、解析します。[重複するプログラム一覧] 画面では、COBOL ソース解析が名前を変更したプログラム名 (PROGRAM-ID) が表示されます。

プロジェクトサマリ画面の [COBOL プログラム] 行の [重複] 列の個数をクリックすると表示されます。

画面の「リネームプログラム名」をクリックすると、各プログラムの [プログラム詳細] 画面の「プログラム概要」に遷移します。

COBOL ソースファイルを追加したときに、重複プログラム数が 999 を超えたときは、解析エラー (KEDL1265A-E) が出力されます。解析中に重複プログラム数が 999 を超えたときは、解析エラー (KEDL1266A-E) が出力されます。

図 7-10 重複するプログラム一覧

#	プログラム名	リネームプログラム名	総行数	実行文の数	分岐の数	呼出先プログラム数	引数の個数	コメント行の文字列
1	ZC001_OPT	ZC001_OPT(1)	22	2	0	0	0	T オプション処理1
2	ZC001_OPT	ZC001_OPT(2)	22	2	0	0	0	T オプション処理6
3	ZC002_OPT	ZC002_OPT(1)	24	2	0	0	1	T オプション処理2
4	ZC002_OPT	ZC002_OPT(2)	21	2	0	0	1	T オプション処理5
5	ZC002_OPT	ZC002_OPT(3)	24	2	0	0	1	T オプション処理7
6	ZC002_OPT	ZC002_OPT(4)	21	2	0	0	1	T オプション処理9
7	ZC003_OPT	ZC003_OPT(1)	21	2	0	0	1	T オプション処理3
8	ZC003_OPT	ZC003_OPT(2)	21	2	0	0	1	T オプション処理4
9	ZC003_OPT	ZC003_OPT(3)	21	2	0	0	1	T オプション処理8

[ページの先頭へ](#)

[プログラム名]

重複する COBOL ソースファイルの見出し部の PROGRAM-ID です。

[リネームプログラム名]

複数の同じプログラム名の解析結果を判断するため、重複しないように付けられたプログラム名です。名称の規則は、「プログラム名(*nnn*)」*nnn*：一連番号（1～999）です。一連番号は同じプログラム名に割り振られます。

名称が重複したプログラムが解析されたときの、リネームプログラム名の表示例を次に示します。「ZC001_OPT」という名称の3種類のプログラムが解析された例です。

```
ZC001_OPT(1)
ZC001_OPT(2)
ZC001_OPT(3)
```

解析対象プロジェクト内でプログラム名が重複するときは、重複するプログラム名にはすべて一連番号が付きます。

[総行数]

プログラムの総行数です。

[実行文の数]

プログラムの実行文の数です。

[分岐の数]

プログラムの分岐の数です。

[呼出先プログラム数]

プログラムから呼び出しているプログラムの数です。

[引数の個数]

手続き部の見出しの引数の個数です。なお、RETURNING 指定は引数として表示されません。

[コメント行の文字列]

「解析生成オプション設定」画面の「プログラムコメント抽出」タブで「プログラムコメントを抽出する」をチェックすると、条件に従って抽出されたコメントが表示されます。

7.3.4 「登録集原文一覧」画面

使用されている登録集原文と使用されていない登録集原文の一覧です。プロジェクトサマリ画面の「登録集原文」行の「全体」列の個数をクリックすると表示されます。

画面の「登録集原文名」をクリックすると、各プログラムの「登録集原文詳細」画面に遷移します。なお、「使用有無」が「使用」になっている場合だけ、「登録集原文詳細」画面に遷移できます。

登録集原文名は、昇順にソートされます。

図 7-11 登録集原文一覧

#	登録集原文名	使用有無	コピー元プログラム数	パス名
1	ACCTCD	使用	1	C:\入出荷管理システム調査\入出荷管理システム\WCOPY登録集
2	CBLXMLRC	未使用	0	C:\Program Files\Hitachi\COBOL2002\WCOPY
3	FC351.DT	使用	1	C:\入出荷管理システム調査\入出荷管理システム\WCOPY登録集
4	FC370	使用	3	C:\入出荷管理システム調査\入出荷管理システム\WCOPY登録集
5	FC380	使用	1	C:\入出荷管理システム調査\入出荷管理システム\WCOPY登録集
6	FC390I	使用	1	C:\入出荷管理システム調査\入出荷管理システム\WCOPY登録集
7	FC351	使用	4	C:\入出荷管理システム調査\入出荷管理システム\WCOPY登録集

[登録集原文名]

解析対象の COBOL ソースファイルに含まれる、すべての登録集原文名です。

解析対象プロジェクト内で登録集原文ファイル名が重複する場合は、同じ登録集原文名が連続して表示されます。重複する登録集原文ファイルのそれぞれの格納場所は、[パス名] で確認してください。

[使用有無]

表示される文字の意味を次に示します。

- 使用

COBOL ソースファイルで参照されている登録集原文です。間接的に参照されている登録集原文も含まれます。

- 未使用

COBOL ソースファイルで参照されていない登録集原文です。間接的に参照されていない登録集原文も含まれます。解析生成オプションの環境変数 CBLLIB に指定されたパスの中から、参照されない登録集原文が未使用の登録集原文として検索されます。

[コピー元プログラム数]

この登録集原文を使用しているプログラム数です。

[パス名]

登録集原文のファイルが格納されているパス名です。

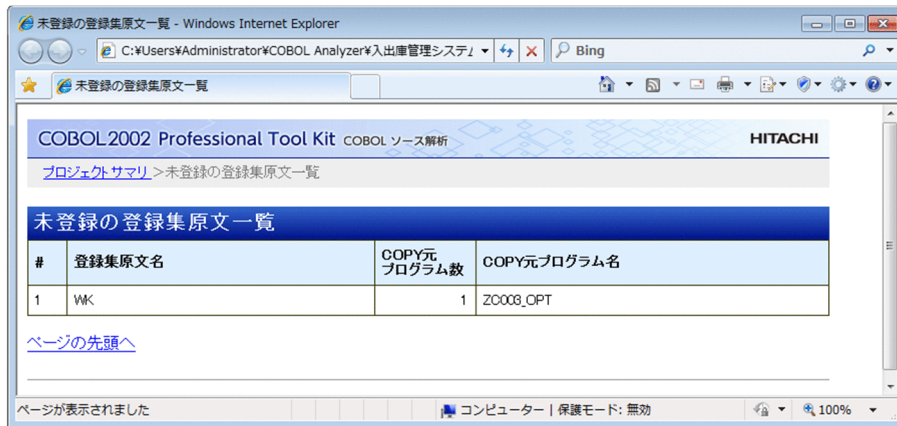
(1) [未登録の登録集原文一覧] 画面

COBOL ソースファイル中で参照されているが、参照されている登録集原文が見つからなくて解析できなかった登録集原文の一覧です。プロジェクトサマリ画面の [登録集原文] 行の [未登録] 列の個数をクリックすると表示されます。登録集原文名は、昇順にソートされます。

この画面の「登録集原文名」をクリックすると、「登録集原文詳細」画面に遷移します。

なお、手続き部（PROCEDURE DIVISION）の COPY 文はソース表示に反映されるだけで、使用している登録集原文の対象外となります。

図 7-12 未登録の登録集原文一覧



【登録集原文名】

COPY 文で指定された登録集原文名です。登録集原文名に「¥」を含む場合は解析エラーになります。登録集原文名の拡張子は省略して表示されます。

解析対象プロジェクト内で登録集原文ファイル名が重複する場合は、同じ登録集原文名が連続して表示されます。

【COPY 元プログラム数】

この登録集原文を使用しているプログラム数です。

【COPY 元プログラム名】

この登録集原文を使用しているプログラム名です。使用しているプログラムが複数ある場合、プログラム名は改行で区切られて表示されます。

(2) 【登録集原文詳細】 画面

COBOL ソースファイル内で使用されている登録集原文の詳細情報です。【登録集原文詳細】画面には、「登録集原文概要（登録集原文名）」と「COPY 元プログラム一覧」が表示されます。

「登録集原文ファイル名」をクリックすると、登録集原文ファイルの【ソース表示】画面に遷移します。「プログラム名」をクリックすると、【プログラム詳細】画面の「プログラム概要」に遷移します。「最上位の COPY 元登録集原文名」をクリックすると、最上位の登録集原文の【登録集原文詳細】画面に遷移します。

図 7-13 登録集原文詳細



「登録集原文概要（登録集原文名）」の項目の説明

「登録集原文名」

COPY 文で指定された登録集原文名です。登録集原文名に「¥」を含む場合は解析エラーになります。登録集原文名の拡張子は省略して表示されます。

「登録集原文ファイル名」

登録集原文のファイル名です。ファイル名をクリックすると、登録集原文ファイルの「ソース表示」画面が表示されます。

「コピー元プログラム数」

この登録集原文を使用しているプログラム数です。

「パス名」

登録集原文のファイルが格納されているパス名です。

「COPY 元プログラム一覧」の項目の説明

「プログラム名」

COBOL ソースファイルの見出し部の PROGRAM-ID です。

「展開場所」

登録集原文が COPY 文で展開された、次に示す節や部の名称が表示されます。複数の場所で展開されている場合、「ファイル節,作業場所節」のように「,」で区切って表示されます。

- ファイル節 (FILE SECTION)
- 作業場所節 (WORKING-STORAGE SECTION)
- 連絡節 (LINKAGE SECTION)
- データ部 (DATA DIVISION)

- その他の節
- 手続き部 (PROCEDURE DIVISION)

[直接参照]

COPY 元プログラムから直接参照されている場合は「○」が表示されます。

[最上位の COPY 元登録集原文名]

COPY 元プログラムから間接参照されている場合、最上位の COPY 元登録集原文名が表示されます。最上位の COPY 元登録集原文名が複数あるときは複数行表示されます。

7.3.5 [ファイル一覧] 画面

ファイル管理記述項の SELECT 句で宣言されたファイル名の一覧です。プロジェクトサマリ画面の [ファイル] 行の [全体] 列の個数をクリックすると表示されます。

[ファイル名] をクリックすると、[ファイル詳細] 画面に遷移します。

図 7-14 ファイル一覧

#	ファイル名	編成	入出力文	使用プログラム数
1	FC3S1.DT	SEQUENTIAL	READ	1
2	FC370	SEQUENTIAL	READ	2
3	FC380	SEQUENTIAL	READ	1

[ファイル名]

ファイル管理記述項の SELECT 句で指定したファイル名です。

[編成]

ファイルの論理的構成 (ORGANIZATION 句) の指定です。ORGANIZATION 句を省略した COBOL ソースファイルの場合は、「SEQUENTIAL」が表示されます。ただし、整列併合ファイルのときは何も表示されません。

[入出力文]

ファイルを使用している入出力文です。表示される文字を次に示します。複数の入出力文で使用的是場合は、READ/WRITE のように「/」で区切って表示されます。

- READ

- WRITE
- REWRITE
- DELETE

[使用プログラム数]

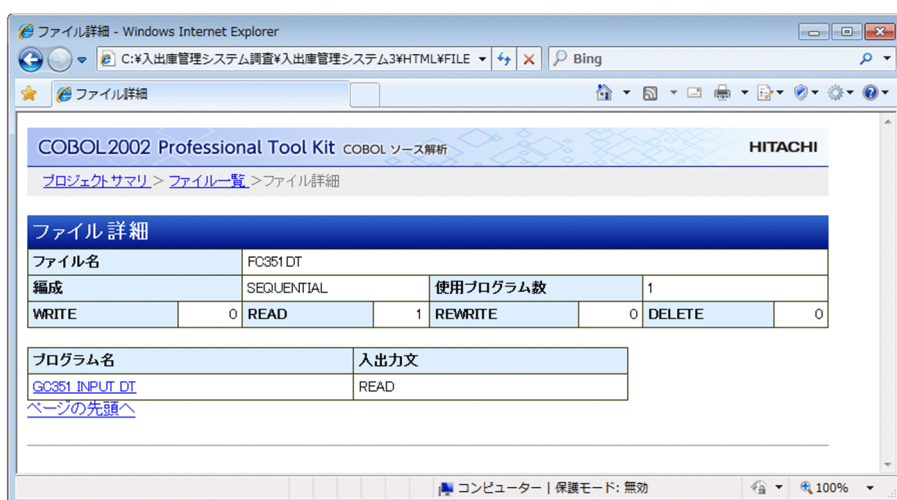
ファイルを使用しているプログラム数です。

(1) [ファイル詳細] 画面

ファイルの詳細情報です。[ファイル一覧] 画面の「ファイル名」をクリックすると表示されます。

この画面の「プログラム名」をクリックすると、[プログラム詳細] 画面の「プログラム概要」に遷移します。

図 7-15 ファイル詳細



[ファイル名]

ファイル管理記述項の SELECT 句で指定したファイル名です。

[編成]

ファイルの論理的構成 (ORGANIZATION 句) の指定です。ORGANIZATION 句を省略した COBOL ソースファイルの場合は、「SEQUENTIAL」が表示されます。ただし、整列併合ファイルのときは何も表示されません。

[使用プログラム数]

ファイルを使用しているプログラム数です。

[WRITE] [READ] [REWRITE] [DELETE]

入出力文を使用している回数です。

[プログラム名]

COBOL ソースファイルの見出し部の PROGRAM-ID です。

[入出力文]

ファイルを使用している入出力文です。

7.3.6 [表(DB)一覧] 画面

埋め込み SQL 文で使用しているすべての SQL 表の一覧です。プロジェクトサマリ画面の [表(DB)] 行の [全体] 列の個数をクリックすると表示されます。

この画面の「表名」をクリックすると、[表(DB)詳細] 画面に遷移します。

図 7-16 表(DB)一覧

#	表名	入出力種別(*1)	使用プログラム数
1	FGC301	RUD	1
2	GC3S1_DT	R	1
3	RC340	CRUD	4

(*1)入出力種別: C=INSERT, R=SELECT/FETCH/問合せを実行するSQL文, U=UPDATE, D=DELETE

[ページの先頭へ](#)

[表名]

埋め込み SQL 文で使用されている SQL 表名です。

[入出力種別]

表名が呼び出されるときの入出力種別（モード）です。

[使用プログラム数]

表名を使用しているプログラム数です。

(1) [表(DB)詳細] 画面

SQL 表の詳細情報です。[表(DB)一覧] 画面の「表名」をクリックすると表示されます。

この画面の「プログラム名」をクリックすると、[プログラム詳細] 画面の「プログラム概要」に遷移します。

図 7-17 表名(DB)詳細



[表名]

埋め込み SQL 文で使用されている SQL 表名です。

[使用プログラム数]

表名を使用しているプログラム数です。

[C] [R] [U] [D]

表名が呼び出されるときの入出力種別（モード）ごとのプログラム数が表示されます。モードの意味を次に示します。

- C : INSERT
- R : SELECT, FETCH, または問合せを実行する SQL 文
- U : UPDATE
- D : DELETE

[プログラム名]

表名を使用しているプログラム名です。

[入出力種別]

表名が呼び出されるときの入出力種別（モード）です。

7.3.7 【プログラム階層図】 画面

プログラムの階層を表示する画面です。[プログラム一覧] 画面から [プログラム詳細] 画面を表示して「プログラム階層図」をクリックすると表示されます。

表示されるのは、プログラムから呼び出されているプログラム、ファイル、および表名です。表示の対象になるのは、主プログラムと指定されたプログラムと未使用の外部プログラムです。プログラム階層図では、呼び出し元の CALL 文がデッドコードであるときは、使用する外部プログラムとして扱います。

プログラムからさらにプログラムを呼び出していて、かつ、共通プログラムではない（呼び出し元のプログラム数が1個）場合は、プログラムが階層表示されます。共通プログラム（呼び出し元のプログラム数が2個以上）の場合は、プログラムが階層表示されません。表示の対象となるプログラム数の分だけ、すべての階層図が表示されます。

各ボックスの「プログラム名」をクリックすると、[プログラム詳細] 画面の「プログラム概要」に遷移します。「ファイル名」をクリックすると、[ファイル詳細] 画面に遷移します。「表名」をクリックすると、[表(DB)詳細] 画面に遷移します。階層図のルートにある「共通プログラム名」をクリックすると、[プログラム詳細] 画面の「プログラム概要」に遷移します。ルート以外の「共通プログラム名」をクリックすると次に示す画面に遷移します。

- 主プログラムが [共通プログラム] のときは、主プログラムをルートにするプログラム階層図に遷移します。
- 主プログラム以外が [共通プログラム] のときは、共通プログラムをルートにする別のプログラム階層図に遷移します。

図 7-18 プログラム階層図

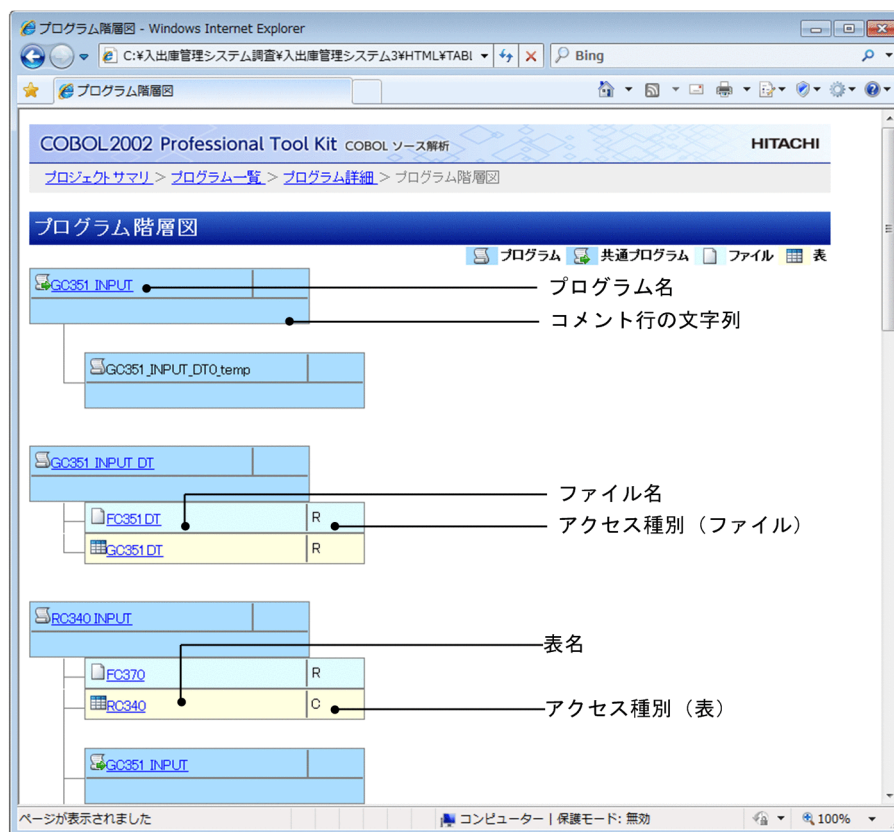
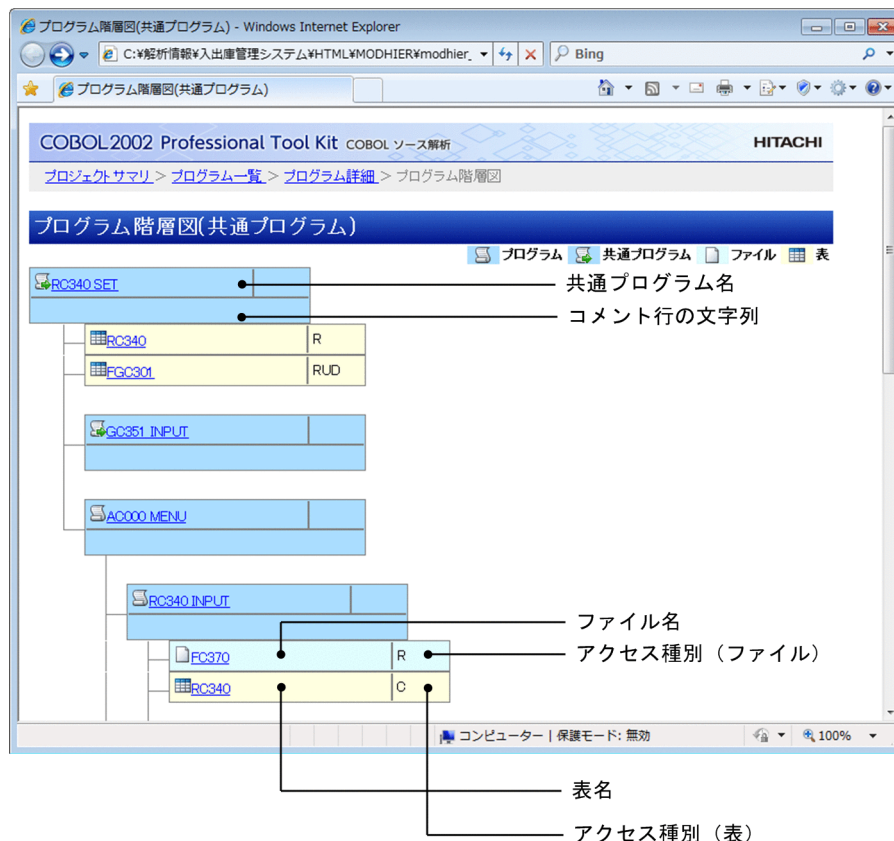


図 7-19 プログラム階層図（共通プログラム）



[プログラム名]

COBOL ソースファイルの見出し部の PROGRAM-ID です。

[コメント行の文字列]

プログラムコメント（プログラムの機能を要約するコメント）の文字列です。

[解析生成オプション設定] 画面の [プログラムコメント抽出] タブで [プログラムコメントを抽出する] にチェックすると、条件に従って抽出されたコメントが表示されます。[プログラムコメント抽出] タブについては、「5.2.3 [プログラムコメント抽出] タブ」を参照してください。

[ファイル名]

FD（ファイル記述項）または SD（整列併合用ファイル記述項）に記述されたファイル名です。

[アクセス種別（ファイル）]

ファイルを使用している入出力文です。表示される文字を次に示します。複数の入出力文で使っている場合は、R/RW のように「/」で区切って表示されます。

- R : READ
- W : WRITE
- RW : REWRITE
- D : DELETE

[表名]

埋め込み SQL 文に記述されている SQL 表です。

[アクセス種別 (表)]

SQL 表を呼び出すときのモードです。モードの意味を次に示します。

- C : INSERT
- R : SELECT, FETCH, または問合せを実行する SQL 文
- U : UPDATE
- D : DELETE

複数の埋め込み SQL 文を使用しているときは、表名ごとに、アクセス種別表示名の順に SQL 表が表示されます。

アクセス種別は C, R, U, D の順序で表示されます。

例えば、UPDATE 文と SELECT 文が定義されている場合、「RU」と表示されます。

[共通プログラム名]

共通プログラムに対応する COBOL ソースファイルの見出し部の PROGRAM-ID です。

7.3.8 【デッドコード情報一覧】 画面

デッドコード情報の一覧です。プロジェクトサマリ画面の [デッドコード] 行の [全数] 列の個数をクリックすると表示されます。

この画面の「プログラム名」をクリックすると、[プログラム詳細] 画面の「プログラム概要」に遷移します。

図 7-20 デッドコード情報一覧



#	プログラム名	未使用の外部プログラム	未使用の内部プログラム	未使用のファイル	未使用の報告書、画面、通信記述名	未使用のデータ項目	制御が渡らない文、手続き
1	RC340.INPUT	0	0	0	0	1	0
2	ZC001.OPT	0	0	0	0	3	0
-	合計	0	0	0	0	4	0

[プログラム名]

COBOL ソースファイルの見出し部の PROGRAM-ID です。

[未使用の外部プログラム]

未使用の外部プログラムの個数です。

[未使用の内部プログラム]

未使用の内部プログラムの個数です。

[未使用のファイル]

未使用のファイルの個数です。

[未使用の報告書, 画面, 通信記述名]

未使用の報告書, 画面, 通信記述名の個数です。

[未使用のデータ項目]

未使用のデータ項目の個数です。なお, 表示されるのはデッドコード候補として検知された未使用のデータ項目の個数です。このため, [プログラム詳細] 画面の「データ項目一覧」で表示されている未使用の個数と一致しない場合があります。

[制御が渡らない文, 手続き]

制御が渡らない文, 手続きの個数です。

7.4 プログラム固有の情報

プログラム固有の情報は、プロジェクトサマリ画面 (index.html) からは参照できません。[プログラム一覧] から任意のプログラムを選択して、[プログラム詳細] 画面の「プログラム概要」を表示してリンクをたどってください。

プログラム固有の情報のソース解析情報を次の表に示します。

表 7-1 プログラム固有の情報のソース解析情報

項番	解析内容	説明
1	プログラム概要	プログラムの概要です。
2	呼出元プログラム一覧	基準になるプログラムを呼び出しているプログラムの一覧です。
3	呼出先プログラム一覧	基準になるプログラムから呼び出しているプログラムの一覧です。
4	参照する登録集原文一覧	プログラム内で参照している登録集原文の一覧です。
5	内部プログラム一覧	内部プログラムの一覧です。
6	セクション一覧	手続き部の節と段落の一覧です。
7	データ項目一覧	プログラムの 01, 77 レベルのデータ項目の一覧です。
8	ファイル一覧	ファイル管理記述項に記述されているすべてのファイルの一覧です。
9	表(DB)一覧	表(DB)の一覧です。
10	報告書、画面、通信記述名一覧	報告書、画面、および通信記述名の一覧です。
11	制御が渡らない文、手続き一覧	制御が渡らない文、および手続きの一覧です。

7.4.1 [プログラム詳細] 画面

プログラムの詳細を表示する画面です。[プログラム詳細] 画面には、「プログラム概要」や「呼出元プログラム一覧」など固有の情報のソース解析情報が表示されます。

この画面の「プログラム階層図」をクリックすると、該当する[プログラム階層図]画面に遷移します。ただし、使用されていない内部プログラムを表示している場合は、[プログラム階層図]にリンクは付きません。「セクション階層図」をクリックすると、プログラムデータの[セクション階層図]画面に遷移します。「ソース表示」をクリックすると、[ソース表示]画面に遷移します。

図 7-21 プログラム詳細

COBOL2002 Professional Tool Kit COBOL ソース解析 HITACHI

プロジェクトサマリ > プログラム一覧 > プログラム詳細

プログラム階層図 セクション階層図 ソース表示

プログラム概要				
プログラム名	AC000.MENU			
COBOLソースファイル名	メニュー画面.cbl			
パス名	C:\入出庫管理システム			
コメント行の文字列	メニュー画面			
総行数	64	有効行数	51	登録集
コメント行数	13			
参照する登録集原文数	0	ファイル数	0	記述項数
データ項目数	4	引数の個数	0	実行文の数
分岐の数	7	節数	1	段落数
表(DB)数	0	呼出元プログラム数	0	呼出先プログラム数
未使用の内部プログラム数	0	未使用のデータ項目数	0	未使用の報告書、画面、通信記述名の数
制御が渡らない文、手続きの数	0			

呼出元プログラム一覧	
#	呼出元プログラム名

「プログラム概要」の項目の説明

【プログラム名】

COBOL ソースファイルの見出し部の PROGRAM-ID です。

【COBOL ソースファイル名】

COBOL ソースファイル名です。「COBOL ソースファイル名」をクリックすると、「ソース表示」画面に遷移します。

【パス名】

COBOL ソースファイルが格納されているパス名です。

【コメント行の文字列】

プログラムコメント（プログラムの機能を要約するコメント）の文字列です。

「解析生成オプション設定」画面の「プログラムコメント抽出」タブで「プログラムコメントを抽出する」にチェックすると、条件に従って抽出されたコメントが表示されます。「プログラムコメント抽出」タブについては、「5.2.3 「プログラムコメント抽出」タブ」を参照してください。

【総行数】

プログラムの総行数です。

【有効行数】

プログラムの総行数からコメント行数を引いた値です。

【登録集】

登録集原文で展開したプログラムの行数です

[コメント行数]

COBOL ソースファイルまたは登録集原文ファイルのコメント行と注記項の総数です（COPY 文展開前）。コメント行の行数カウントについては、「[7.6.2 ソース表示の色と行数カウント](#)」を参照してください。

[参照する登録集原文数]

COPY 文で展開している登録集原文の数です。

[ファイル数]

ファイルの個数です。

[記述項数]

データ部で定義している 01～49, 66, 77 レベルのデータ名の個数です。

[データ項目数]

データ部で定義している 01, 77 レベルのデータ項目の数です。同じ名称のデータ項目は、1 個ではなく、定義の数だけカウントします。

[引数の個数]

手続き部（PROCEDURE DIVISION）の見出しの引数の個数です。ただし、RETURNING 指定は引数として扱いません。

[実行文の数]

手続き部（PROCEDURE DIVISION）の文の数です。ただし、「文」には ELSE, および END 動詞は含みません。また、EXEC SQL～END-EXEC は 1 文として数えます。対象となる文については、「[10.8 実行文として数える文](#)」を参照してください。

[分岐の数]

プログラムの分岐の数です。分岐の数は、COBOL2002 カバレッジ情報の C1 メジャーの数で示します。詳細については、マニュアル「COBOL2002 操作ガイド」のカバレッジ情報の説明を参照してください。

[節数]

手続き部の節の個数です。

[段落数]

手続き部の段落の個数です。

[表(DB)数]

埋め込み SQL 文で使用している表の個数です。

[呼出元プログラム数]

呼び出し元プログラムの個数です。

[呼出先プログラム数]

呼び出し先プログラムの個数です。

[未使用の内部プログラム数]

未使用の内部プログラムの個数です。

[未使用のデータ項目数]

未使用のデータ項目の個数です。

なお、表示されるのはデッドコード候補として検知された未使用のデータ項目の個数です。このため、[プログラム詳細] 画面の「データ項目一覧」で表示されている未使用の個数と一致しない場合があります。

[未使用の報告書、画面、通信記述数の数]

未使用の報告書、画面、および通信記述数の個数です。

[制御が渡らない文、手続きの数]

制御が渡らない文、および手続きの個数です。

「呼出元プログラム一覧」の項目の説明

[呼出元プログラム名]

基準になるプログラムを呼び出しているプログラムの名称です。「呼出元プログラム名」をクリックすると、[プログラム詳細] 画面に遷移します。

「呼出先プログラム一覧」の項目の説明

[呼出先プログラム名]

COBOL ソースファイルの見出し部の PROGRAM-ID です。

[種別]

表示される文字の意味を次に示します。

- 外部
外部プログラム
- 内部
内部プログラム
- —
一意名指定の CALL 文で呼び出し先プログラム名が解決できていない

[引数の個数]

CALL 文の引数の個数です。

「参照する登録集原文一覧」の項目の説明

[登録集原文名]

COPY 文で指定された登録集原文名です。登録集原文名に「¥」を含む場合は解析エラーになります。登録集原文名の拡張子は省略して表示されます。

解析対象プロジェクト内で登録集原文ファイル名が重複する場合は、同じ登録集原文名が連続して表示されます。

[直接参照]

COPY 元プログラムから直接参照されている登録集原文の場合は「○」が表示されます。

[接頭辞]

COPY 文 PREFIXING 指定の語です。

[接尾辞]

COPY 文 SUFFIXING 指定の語です。

「内部プログラム一覧」の項目の説明

[内部プログラム名]

内部プログラムの名前です。「内部プログラム名」をクリックすると、[プログラム詳細] 画面の「プログラム概要」に遷移します。

[使用有無]

表示される文字の意味を次に示します。

- 使用
内部プログラムを使用しています。
- 未使用
内部プログラムを使用していません。

[引数の個数]

内部プログラムの引数の個数です。

「セクション一覧」の項目の説明

[手続き名（節名／段落名）]

手続き部の節または段落の見出しに記述された名称です。

[種別] が節の場合、「節名」がリンク付きで表示されます。「節名」をクリックすると、プログラムデータの [セクション階層図] 画面の該当するセクションに遷移します。

[種別] が段落の場合、「段落名」がインデントを付けてリンクなしで表示されます。

注意事項

手続き部の先頭が節の見出しで始まっていないとき、手続き部の先頭に「{PROCEDURE_START}」という名前の節が仮定されます。

[種別]

表示される文字の意味を次に示します。

- 節
手続きは節です。
- 段落
手続きは段落です。

[呼出有無]

表示される文字の意味を次に示します。

- 手続き（節／段落）は PERFORM 文に呼出先として指定されています。
- ×
手続き（節／段落）は PERFORM 文に呼出先として指定されていません。
ただし、THROUGH 指定または THRU 指定を使った範囲指定の場合、範囲内に節見出しがある節は呼出先として指定されているものとして扱います。

[制御の移行先の手続き名]

節内または段落内の PERFORM 文に指定した呼出先（制御の移行先）の手続き名を斜線区切りで並べた文字列です。段落の場合、手続き名は「段落名 [節名]」の形式で表示されます。ただし、THROUGH 指定または THRU 指定で出口手続きが指定されている場合は、「呼出先手続き名:出口手続き名」の形式で表示されます。表示される節名や段落名については、「[10.15 そと PERFORM 文からの呼出先情報の抽出規則](#)」を参照してください。

[コメント行の文字列]

[種別] が節の場合は、節コメント（手続き部の節の機能を要約するコメント）の文字列です。[解析生成オプション設定] 画面の [節コメント抽出] タブで [節コメントを抽出する] にチェックすると、条件に従って抽出されたコメントが表示されます。[節コメント抽出] タブについては、「[5.2.4 \[節コメント抽出\] タブ](#)」を参照してください。

[種別] が段落の場合は、常に空文字列が表示されます。

[実行順序]

[種別] が節の場合に、[解析生成オプション設定] 画面の [その他の解析オプション] タブで [プログラムの制御フロー情報(実行順序表示機能で使用する情報)を抽出する] を有効（チェックボックスをオン）にしているときは、「表示」がリンク付きで表示されます。「表示」をクリックすると、その節の [実行順序表示] 画面に遷移します。プログラム全体の [実行順序表示] 画面を表示する場合は、手続き部先頭の節の「表示」をクリックします。[実行順序表示] 画面については、「[7.4.5 \[実行順序表示\] 画面](#)」を参照してください。

[種別] が段落の場合は、常に空文字列が表示されます。

「データ項目一覧」の項目の説明

[レベル番号]

データ記述項のレベル番号です。

[データ項目名]

ファイル節、作業場所節、または連絡節で定義している 01, 77 レベルのデータ項目の名称です。「データ項目名」をクリックすると、[プログラムデータ] 画面の「ファイル節のデータ定義」、「作業場所節のデータ定義」、または「連絡節のデータ定義」に遷移します。

[参照有無]

データ項目名（データ名）のプログラム内での参照有無です。

- 参照

プログラム内で参照されています。

データ項目が集団項目の場合、集団項目に属するどれかのデータ項目が参照されているときも「参照」とします。データ項目が条件変数である場合、条件名が参照されているときも「参照」とします。制御が渡らない文（デッドコードの文）から参照されているデータ項目も「参照」とします。

ただし、データ項目名が手続き部（PROCEDURE DIVISION）の ENTRY 文の USING 指定や、REDEFINES 句、RENAMES 句または SAME AS 句にだけ指定されているときは、参照されていないもの（未参照）とします。

- 未参照

プログラム内で参照されていません。データ項目が集団項目の場合、集団項目に属するすべてのデータ項目が未参照のときだけ「未参照」とします。

[定義場所節]

データ項目を定義している節の名前です。

[サイズ]

データ項目のバイト長（内部ブール項目の場合はビット長）です。

「ファイル一覧」の項目の説明

[ファイル名]

ファイル管理記述項の SELECT 句で指定したファイル名です。

[使用有無]

表示される文字の意味を次に示します。

- 使用

プログラム中でファイルを使用しています。

- 未使用

プログラム中でファイルを使用していません。

[入出力文]

ファイルを使用している入出力文です。表示される文字を次に示します。複数の入出力文で使用している場合は、READ/WRITE のように「/」で区切って表示されます。

- READ

- WRITE

- REWRITE

- DELETE

「表(DB)一覧」の項目の説明

[表名]

埋め込み SQL 文で使用している表名です。「表名」をクリックすると、[表(DB)詳細] 画面に遷移します。

[入出力種別]

表名が呼び出されるときの入出力種別（モード）です。

「報告書、画面、通信記述名一覧」の項目の説明

[報告書、画面、通信記述名]

報告書、画面、および通信記述の名称です。それぞれの名称をクリックすると、[ソース表示] 画面の該当行に遷移します。

[使用有無]

表示される文字の意味を次に示します。

- 使用
報告書、画面、および通信記述名を使用しています。
- 未使用
報告書、画面、および通信記述名を使用しません。

[種類]

「報告書名」、 「画面名」、 および 「通信記述名」 のどれかを表示します。

「制御が渡らない文、手続き一覧」の項目の説明

[該当行]

デッドコードの行番号です。該当行をクリックすると、別ウィンドウで [ソース表示] 画面が表示され、該当行に遷移します。

[内容]

デッドコードの内容です。次に示す内容が表示されます。

内容に表示されるメッセージテキスト	説明
制御が渡ることがない文があります。 (行番号="開始行番号"~"終了行番号")	制御が渡らない文の先頭行から終了行を示します。 [ソース表示] 画面の該当行をグレー表示します。
手続き"名称 1"に制御が渡ることはありません。	制御が渡らない手続きの定義位置を示します。 [ソース表示] 画面の該当手続きの先頭行をグレー表示します。
手続き"名称 1"に制御が渡ることはありません。ただし、手続き名"名称 2"を参照する文があるため確認してください。	制御が渡らない手続きの定義位置を示します。ただし、手続きを参照する文（ENTRY 文など）があるため確認してください。 [ソース表示] 画面の該当手続きの先頭行をグレー表示します。

7.4.2 【プログラムデータ】 画面

01, 77 レベルのデータ定義の一覧です。[プログラム詳細] 画面の [データ項目一覧] からデータ項目名を選択すると表示されます。

[プログラムデータ] 画面に表示される画面と内容を次に示します。

- [ファイル節のデータ定義]
ファイル節 (FILE SECTION) で宣言されているデータ定義です。
- [作業場所節のデータ定義]
作業場所節 (WORKING-STORAGE SECTION) で宣言されているデータ定義です。
- [連絡節のデータ定義]
連絡節 (LINKAGE SECTION) で宣言されているデータ定義です。

図 7-22 プログラムデータ

#	レベル番号	データ名	属性	用途	サイズ	参照
ファイル節のデータ定義						
作業場所節のデータ定義						
1	01	WK	GROUP		60	○
連絡節のデータ定義						
1	01	GC351	GROUP		247	○
2	01	C-FRR-GC351	GROUP		75	○

[レベル番号]

データ記述項のレベル番号です。

[データ名]

データ記述項のデータ名です。

「データ名」のリンクをクリックすると、[ソース表示(COPY 展開後)] 画面の該当するデータ記述項に遷移します。

[属性]

データ記述項の PICTURE 句の文字列※です。集団項目の場合は「GROUP」を表示されます。
数字編集項目の小数部のゼロ抑制文字 (*, Z) は、「9」が表示されます。

(例)

PIC *.** → *.99

PIC Z.ZZ → Z.99

動的長基本項目で LIMIT が指定されている場合は、「X(LIMIT=n)」または「N(LIMIT=n)」の形式で表示されます。

注※

PICTURE 句の文字列のままではなく、「(文字数)」で正規化したものが表示されます。

[用途]

データ記述項の USAGE 句で指定した形式が表示されます。

動的長基本項目、および GROUP-USAGE IS NATIONAL で宣言された集団項目の場合は、次の情報が表示されます。

- 動的長基本項目の場合：DYNAMIC
- GROUP-USAGE IS NATIONAL で宣言された集団項目の場合：NATIONAL

[サイズ]

データ項目のバイト長（内部ブール項目の場合はビット長）です。

[参照]

データ記述項のデータ名がプログラム内で参照されている場合、「○」が表示されます。○（参照）が表示される条件は、[プログラム詳細] 画面の「データ項目一覧」の「参照有無」の「参照」と同じです。

関連項目

- [「7.4.1 \[プログラム詳細\] 画面」](#)

7.4.3 [ソース表示] 画面

プログラムのソースコードが表示されます。

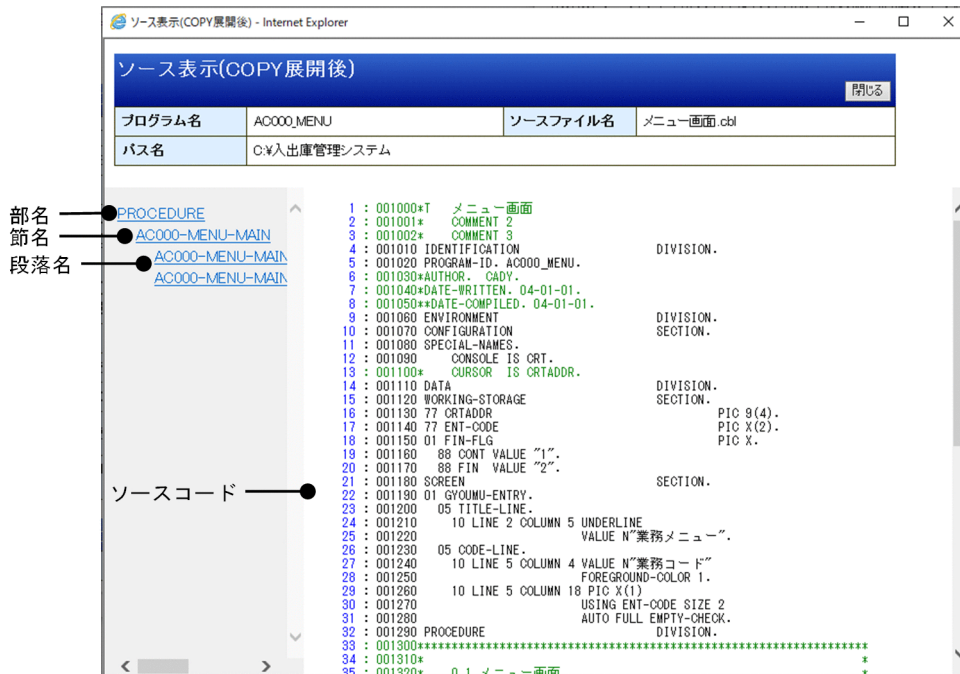
ソースコードを表示する画面の種類を次に示します。

- [ソース表示] 画面
「COBOL ソースファイル名」または「登録集原文ファイル名」のリンクをクリックすると表示されます。COBOL ソースファイルや登録集原文ファイルの中のソースコードがそのまま表示されます。
- [ソース表示(COPY 展開後)] 画面
[プログラム詳細] 画面右上の「ソース表示」のリンク、[プログラムデータ] 画面の「データ名」のリンク、または[セクション階層図] 画面の「節名」のリンクをクリックすると表示されます。ソースコード内の COPY 文の位置には、COPY 文で取り込まれた登録集原文が展開されて表示されます。この画面の左ペインには、部名・節名・段落名の目次（しおり）が表示されます。部名・節名・段落名をクリックすると、該当する部見出し、節見出し、段落見出しが最上行に表示されるようにスクロールします。

注意事項

行終端の制御文字は、X'0A'またはX'0D0A'です。X'1A'(EOF)は、ファイルの終端と見なします。COBOL ソースファイルに入力できない制御文字を含む場合には、エラーメッセージが出力されます。

図 7-23 ソース表示



ソースコードの文字色の意味を次に示します。ただし、条件によって色が異なる個所があります。

- ・ コメント行は、緑色で表示する。
- ・ コメント行以外の行は黒色で表示する。

[プログラム名]

COBOL ソースファイルの見出し部の PROGRAM-ID です。

[ソースファイル名]

COBOL ソースファイル名です。

[パス名]

COBOL ソースファイルが格納されているフォルダの絶対パス名です。

左ペインの表示

[ソース表示(COPY 展開後)] 画面の場合、部名・節名・段落名の目次（しおり）が表示されます。部名・節名・段落名をクリックすると、該当する部見出し、節見出し、段落見出しをソースコードの最上行に位置付けることができます。手続き部の先頭が節の見出しで始まっていない場合、「{PROCEDURE_START}」という節名が表示されます。

ソースコード

COBOL ソースファイルのソースコードが表示されます。ソース表示で表示されるソース行の最大長は 255 バイトです。ただし、コンパイルオプションの指定によって、ソース行の最大長が 80 バイトになることがあります。詳細は、「[7.6.3 ソース表示で表示されるソース行の最大長](#)」を参照してください。プログラム名に記号 (¥, @, #) を含む場合も、COBOL2002 の変換規則は適用されないで、そのまま表示されます。

デッドコードの対象になる行は、背景がグレーで表示されます。1 行に複数の文があり、一部の文だけがデッドコードの場合でも行全体の背景がグレーになります。

COPY 文に PREFIXING 指定/SUFFIXING 指定があっても、接頭辞/接尾辞が付加された表示になりません。

関連項目

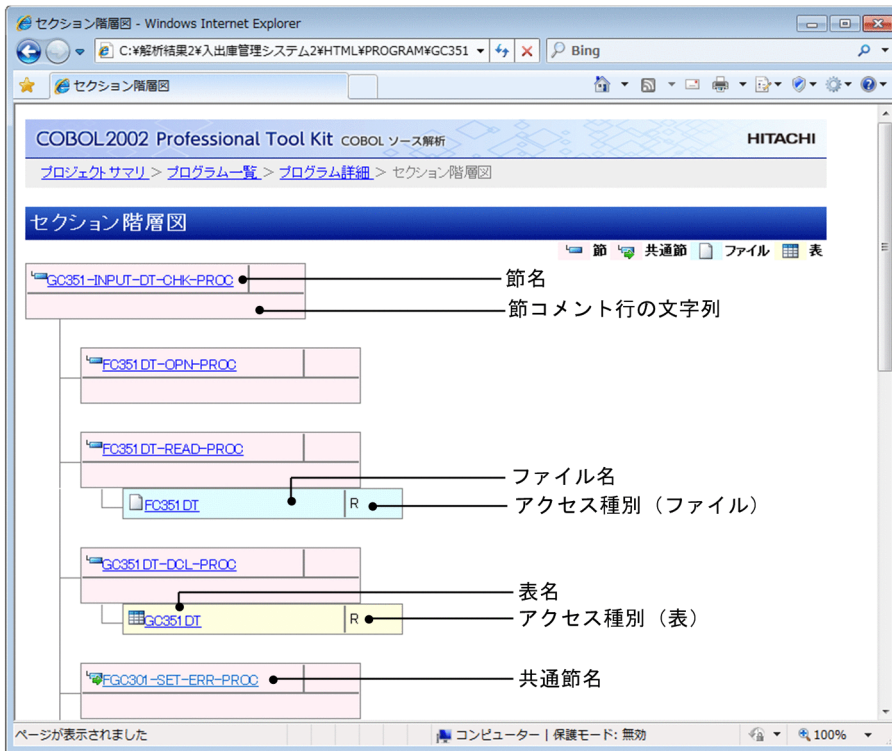
- 「[7.6.2 ソース表示の色と行数カウント](#)」

7.4.4 【セクション階層図】 画面

プログラムのすべての節の階層図が表示されます。表示するのは、ファイルおよび表名です。節からさらに PERFORM 文で節を呼び出している場合や、共通節ではない場合（呼び出し元の節が 1 個）は節が階層表示されます。共通節（呼び出し元節が 2 個以上）のときは節が階層表示されません。

各ボックスの「節名」をクリックすると、[ソース表示(COPY 展開後)] 画面の節見出しの行に遷移します。「ファイル名」をクリックすると、[ファイル詳細] 画面に遷移します。「表名」をクリックすると、[表(DB)詳細] 画面に遷移します。階層図のルートにある「共通節名」をクリックすると、[ソース表示(COPY 展開後)] 画面の節見出しの行に遷移します。ルート以外の「共通節名」をクリックすると、共通節をルートにする別の【セクション階層図】画面に遷移します。

図 7-24 セクション階層図



[節名]

手続き部の節の見出しに記述された名称です。手続き部の先頭が節の見出しで始まっていない場合、「{PROCEDURE_START}」という節名が表示されます。「節名」をクリックすると、プログラムデータの「セクション階層図」画面の該当するセクションに遷移します。

[節コメント行の文字列]

節コメント（手続き部の節の機能を要約するコメント）の文字列です。

「解析生成オプション設定」画面の「節コメント抽出」タブで「節コメントを抽出する」にチェックすると、条件に従って抽出されたコメントが表示されます。「節コメント抽出」タブについては、「[5.2.4 \[節コメント抽出\] タブ](#)」を参照してください。

[ファイル名]

ファイル管理記述項の SELECT 句に記述されたファイル名です。

[アクセス種別 (ファイル)]

ファイルを使用している入出力文です。表示される文字を次に示します。複数の入出力文で使っている場合は、R/RW のように「/」で区切って表示されます。

- R : READ
- W : WRITE
- RW : REWRITE
- D : DELETE

[表名]

埋め込み SQL 文に記述されている SQL 表です。

[アクセス種別 (表)]

SQL 表が呼び出されるときのもードです。モードの意味を次に示します。

- C : INSERT
- R : SELECT または FETCH
- U : UPDATE
- D : DELETE

複数の埋め込み SQL 文を使用しているときは、SQL 表名とアクセス種別表示名の順に表示されます。
アクセス種別は C, R, U, D の順序で表示されます。

例えば、UPDATE 文と SELECT 文が定義されている場合、「RU」と表示されます。

[共通節名]

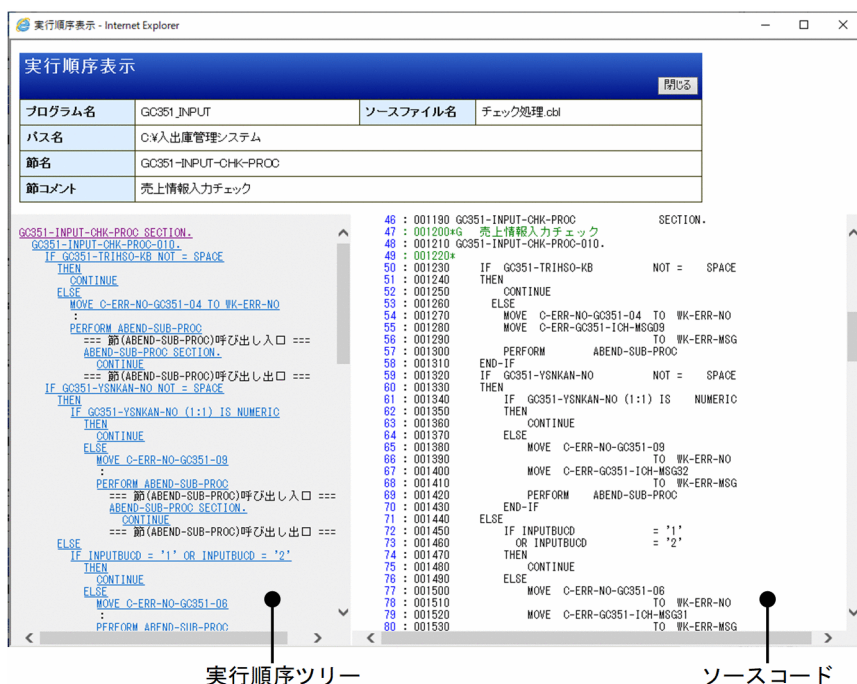
共通節名です。

7.4.5 【実行順序表示】 画面

ソースコードの実行順序を表す実行順序ツリーと、ソースプログラムのソースコードが表示されます。

[プログラム詳細] 画面にある「セクション一覧」の「実行順序」の「表示」リンクをクリックすると表示されます。この画面の左ペインには、節またはプログラム全体に対する実行順序ツリーが目次（しおり）として表示されます。右ペインには、[ソース表示(COPY 展開後)] 画面の右ペインと同じように、COBOL ソースファイルのソースコードが表示されます。

図 7-25 実行順序表示



[プログラム名]

節が属するプログラム名です。

[ソースファイル名]

節が属するプログラムを含む COBOL ソースファイル名です。

[パス名]

COBOL ソースファイルが格納されているフォルダの絶対パス名です。

[節名]

実行順序が表示されている節の名称です。

[節コメント]

節コメント（手続き部の節の機能を要約するコメント）の文字列です。

左ペインの表示

節見出し、段落見出し、実行文および条件指定のソースコードが実行順にツリー形式で表示されます。

ソースコード

COBOL ソースファイルのソースコードが表示されます。ソース表示で表示されるソース行の最大長は 255 バイトです。ただし、コンパイルオプションの指定によって、ソース行の最大長が 80 バイトになることがあります。詳細は、「7.6.3 ソース表示で表示されるソース行の最大長」を参照してください。プログラム名に記号（¥, @, #）を含む場合も、COBOL2002 の変換規則は適用されないで、そのまま表示されます。

デッドコードの対象になる行は、背景がグレーで表示されます。1 行に複数の文があり、一部の文だけがデッドコードの場合でも行全体の背景がグレーになります。

COPY 文に PREFIXING 指定/SUFFIXING 指定があっても、接頭辞/接尾辞が付加された表示になりません。

関連項目

- [「7.4.1 \[プログラム詳細\] 画面」](#)
- [「7.6.2 ソース表示の色と行数カウント」](#)
- [「10.16 制御フロー解析による実行順序表示機能」](#)

7.5 [分布情報一覧] 画面

次の分布を棒グラフで表示する画面です。

- 行数の分布
- 記述項数の分布
- 実行文の数の分布
- 分岐の数の分布
- 節数の分布
- 段落数の分布
- 呼出元プログラム数の分布
- 呼出先プログラム数の分布

プロジェクトサマリ画面の「分布全体表示」のリンクをクリックすると表示されます。

図 7-26 分布情報一覧



棒グラフの長さは次の計算式で算出されます。

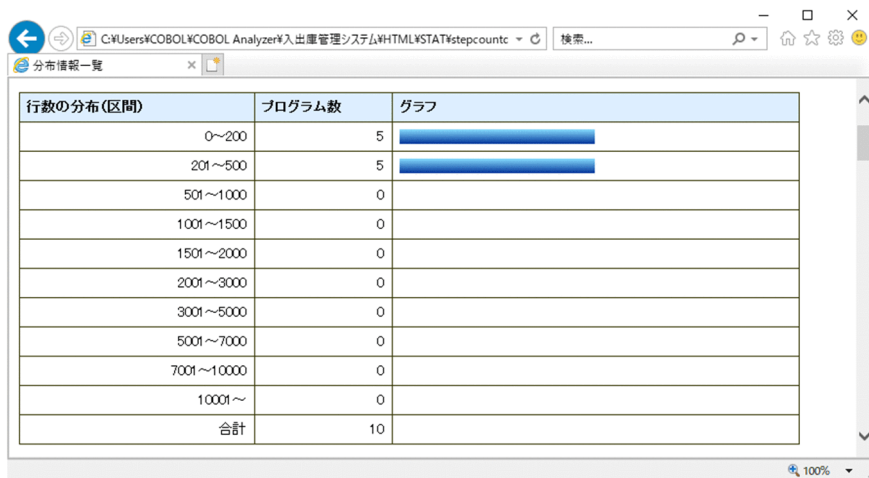
区間の棒の長さ（ピクセル）＝区間のプログラム数÷全区間のプログラム数の合計×400

分布情報一覧の分布区間の範囲はカスタマイズできます。カスタマイズの方法については、「[10.10 COBOL ソース解析の環境定義ファイル（CBLANALYZER.INI）](#)」を参照してください。

[行数の分布]

プログラムの行数がどのように分布しているかを示す棒グラフです。

図 7-27 行数の分布



[行数の分布（区間）]

定義域（プログラムの行数）を分割した区間の範囲です。

[プログラム数]

行数が区間の範囲内であるプログラムの数です。

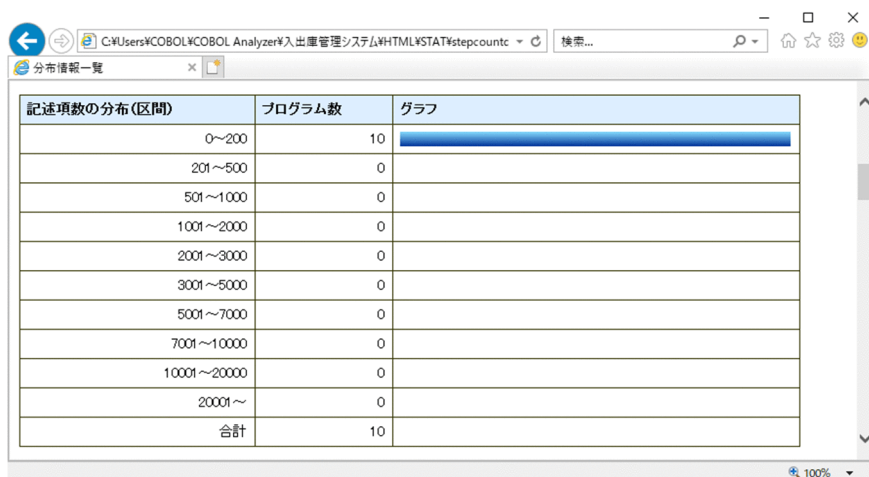
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[記述項数の分布]

プログラムのデータ記述項数がどのように分布しているかを示す棒グラフです。

図 7-28 記述項数の分布



[記述項数の分布（区間）]

定義域（プログラムのデータ記述項数）を分割した区間の範囲です。

[プログラム数]

データ記述項数が区間の範囲内であるプログラムの数です。

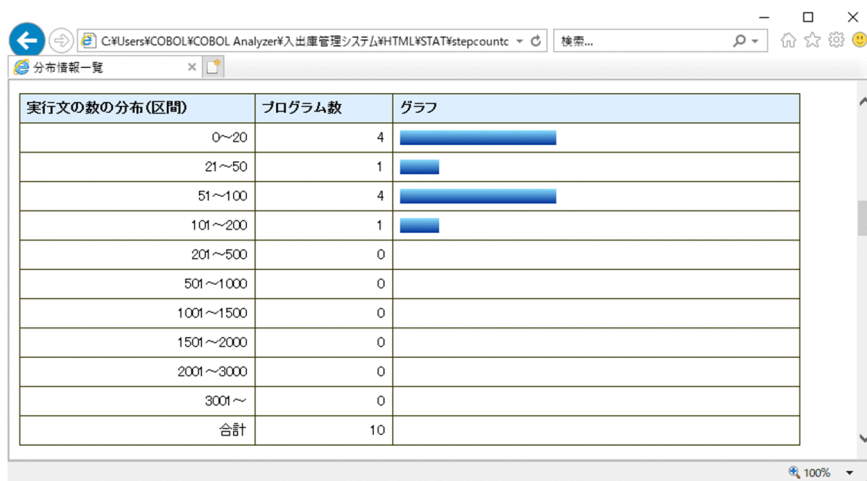
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[実行文の数の分布]

プログラムの実行文の数がどのように分布しているかを示す棒グラフです。

図 7-29 実行文の数の分布



[実行文の数の分布（区間）]

定義域（プログラムの実行文の数）を分割した区間の範囲です。

[プログラム数]

実行文の数が区間の範囲内であるプログラムの数です。

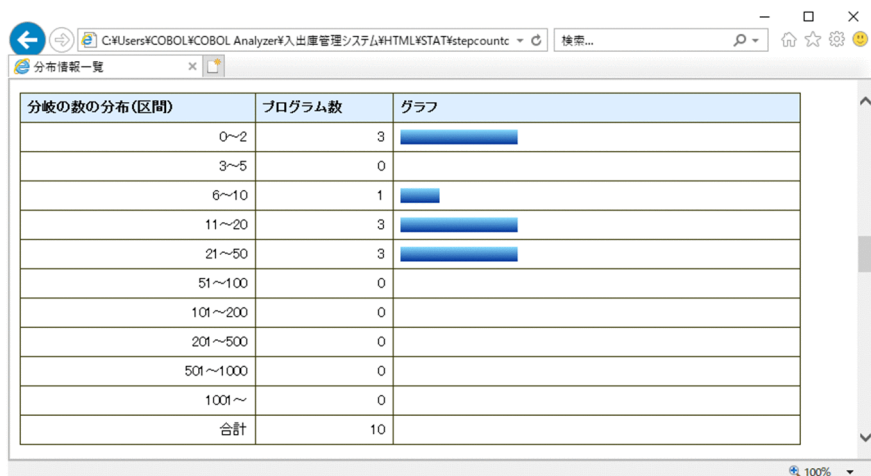
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[分岐の数の分布]

プログラムの分岐の数がどのように分布しているかを示す棒グラフです。

図 7-30 分岐の数の分布



[分岐の数の分布（区間）]

定義域（プログラムの分岐の数）を分割した区間の範囲です。

[プログラム数]

分岐の数が区間の範囲内であるプログラムの数です。

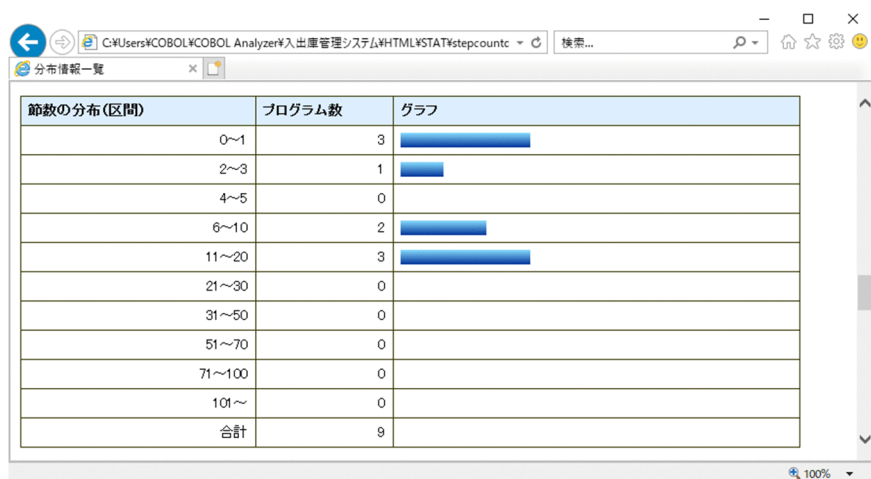
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[節数の分布]

プログラムの手続き部の節数がどのように分布しているかを示す棒グラフです。

図 7-31 節数の分布



[節数の分布（区間）]

定義域（プログラムの節数）を分割した区間の範囲です。

[プログラム数]

節数が区間の範囲内であるプログラムの数です。

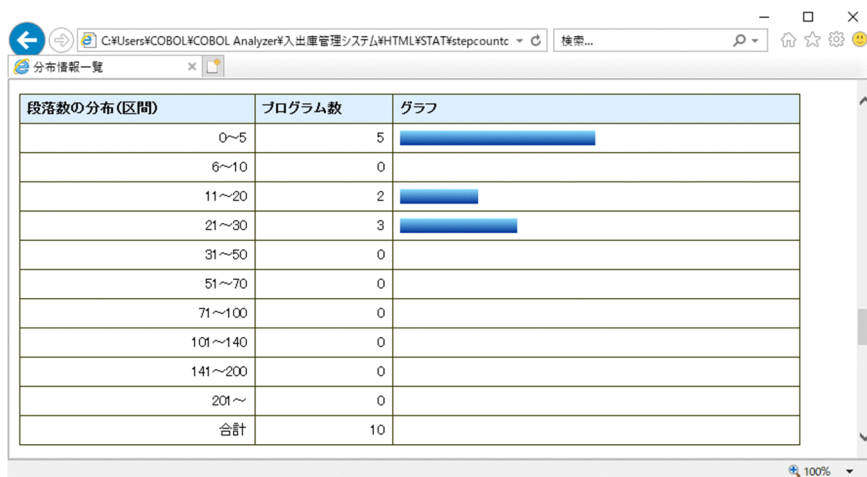
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[段落数の分布]

プログラムの手続き部の段落数がどのように分布しているかを示す棒グラフです。

図 7-32 段落数の分布



[段落数の分布 (区間)]

定義域（プログラムの段落数）を分割した区間の範囲です。

[プログラム数]

段落数が区間の範囲内であるプログラムの数です。

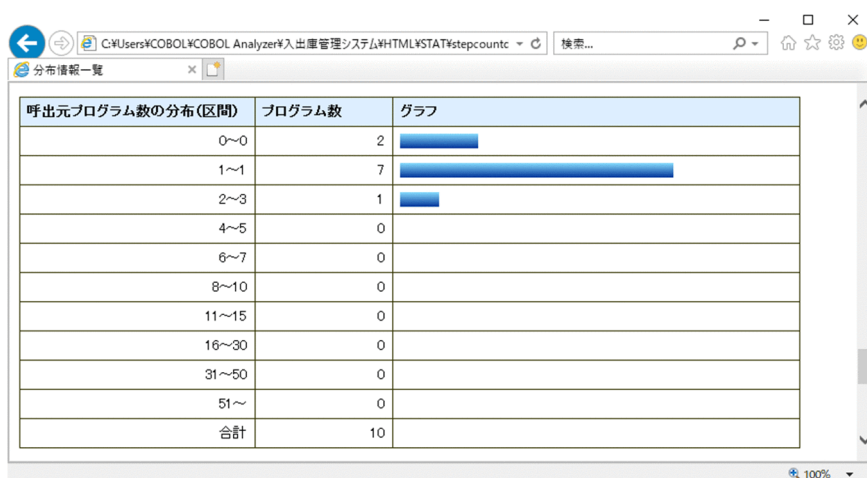
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[呼出元プログラム数の分布]

プログラムの呼出元プログラム数がどのように分布しているかを示す棒グラフです。

図 7-33 呼出元プログラム数の分布



[呼出元プログラム数の分布（区間）]

定義域（プログラムの呼出元プログラム数）を分割した区間の範囲です。

[プログラム数]

呼出元プログラム数が区間の範囲内であるプログラムの数です。

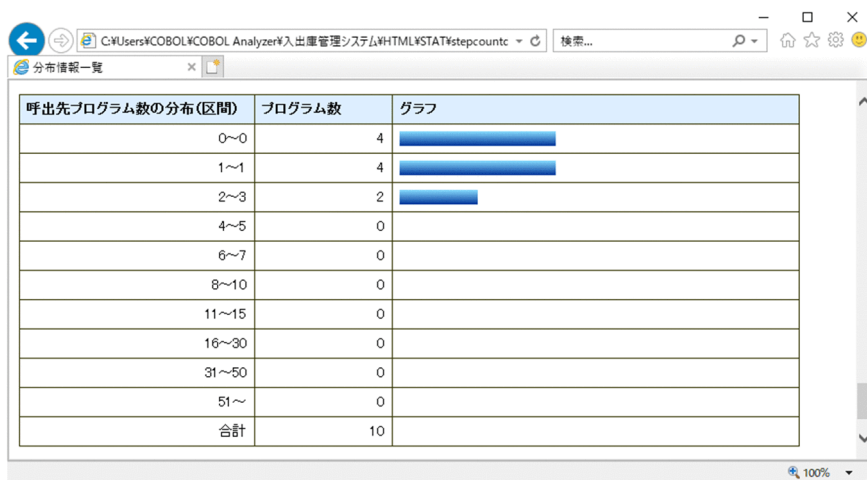
[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

[呼出先プログラム数の分布]

プログラムの呼出先プログラム数がどのように分布しているかを示す棒グラフです。

図 7-34 呼出先プログラム数の分布



[呼出先プログラム数の分布（区間）]

定義域（プログラムの呼出先プログラム数）を分割した区間の範囲です。

[プログラム数]

呼出先プログラム数が区間の範囲内であるプログラムの数です。

[グラフ]

区間のプログラム数の全体の割合に応じた長さの棒グラフの棒です。

7.6 ソース解析情報の表示規則

COBOL ソースファイルの内容と差異があるソース解析情報の項目、およびソース表示の色と行数カウントの説明です。

7.6.1 COBOL ソースファイルの内容と差異があるソース解析情報の項目

ソース解析情報の項目のうち、COBOL ソースファイルの内容と差異がある項目の説明です。

変換して表示されるソース解析情報の項目

COBOL ソースファイルの内容から変換して表示される項目を次の表に示します。

表 7-2 変換して表示されるソース解析情報の項目

変換して表示される項目	変換規則
[プログラム名]	<p>プログラム名は、HTML ドキュメント「ソース表示」および「ソース表示(COPY 展開後)」内では、COBOL ソースファイルに記述された名前が表示されます。それ以外のドキュメントでは、「10.9 解析結果に出力される COBOL ソースファイル中の名称」に示すプログラム名の変換規則に従って変換された名前が表示されます。</p> <p>CALL 一意名を使用しているプログラム名の規則と、CALL 一意名を使用しているプログラムが CALL 定数と同じ名称のプログラム名の規則を次に示します。</p> <ul style="list-style-type: none">CALL 一意名の自動解決でプログラム名が認識されたときは「プログラム名(AUTO)」と表示されます。一意名呼び出し対応付けファイルでプログラム名が認識されたときは「プログラム名(FILE)」と表示されます。CALL 一意名の自動解決と一意名呼び出し対応付けファイルでプログラム名が認識されたときは「プログラム名(AUTO)」と表示されます。定数指定の CALL 文で指定しているプログラム名と、CALL 一意名の認識機能で使用されているプログラム名が同じ名称の場合は、上記の規則で表示されます。
[呼出先プログラム名]	<p>CALL 文のプログラム名は、定数指定の場合は定数の値が表示されます。</p> <p>一意名指定の場合は次のように表示されます。</p> <ul style="list-style-type: none">CALL 一意名の自動認識で呼び出し先プログラム名が解決できたとき「呼び出し先プログラム名(AUTO)」を表示します。一意名呼び出し対応付けファイル (idc ファイル) で呼び出し先プログラム名が解決できたとき「呼び出し先プログラム名(FILE)」を表示します。CALL 一意名の自動認識と一意名呼び出し対応付けファイル (idc ファイル) で呼び出し先プログラム名を解決できたとき「呼び出し先プログラム名(AUTO)」を表示します。呼び出し先プログラム名が解決できないとき「(一意名)」を表示します。
[プログラムコメント行の文字列]	<p>COBOL ソースファイル上に複数の半角空白がある場合、1 つで表示されます。</p>

変換して表示される項目	変換規則
[節コメント行の文字列] [作成者]	
[用途]	<p>USAGE 句の指定が次のように変換されます。</p> <ul style="list-style-type: none"> • COMPUTATIONAL : COMP • COMPUTATIONAL-1 : COMP-1 • COMPUTATIONAL-2 : COMP-2 • COMPUTATIONAL-3 : COMP-3 • COMPUTATIONAL-4 : COMP • COMP-4 : COMP • COMPUTATIONAL-5 : COMP-5 • COMPUTATIONAL-X : COMP-X • PACKED-DECIMAL : COMP-3 • BINARY : COMP • POINTER : ADDRESS <p>オブジェクト参照データ項目には「OBJECT REFERENCE」が表示されます。 OLE オブジェクト参照データ項目には「OBJECT REFERENCE OLE」が表示されます。 また、DISPLAY、NATIONAL は表示されません。 上記以外は、USAGE 句の指定がそのまま表示されます。</p>

行数、行番号に関する項目

ソース解析情報のうち、行数と行番号に関する項目の説明を次の表に示します。

表 7-3 行数、行番号に関するソース解析情報の項目

項目	表示規則
[開始行番号] [該当行] [呼出行番号] [行番号] [使用行番号] [終了行番号] [節内行番号]	「行番号」は COBOL の一連番号ではなく、COBOL ソースファイルの行番号です。
[行数]	プログラム一覧やプログラム概要、各種の分布情報で出力する行数は、COPY 文展開前の COBOL ソースファイルの行数です。そのため、行数はソース表示の情報と異なることがあります。

7.6.2 ソース表示の色と行数カウント

ソース表示のソースコードの文字色と、行数カウント（有効行数とコメント行数）の説明です。

次に示すドキュメントのソース表示の色と行数カウントについて次の表に示します。

- HTML ドキュメント「ソース表示(COPY 展開後)」および「実行順序表示」のソースコードの文字色
- HTML ドキュメント「プログラム詳細のプログラム概要」の有効行数とコメント行数
- CSV ドキュメント「プログラム一覧 CSV」の有効行数とコメント行数
- CSV ドキュメント「セクション一覧 CSV」の有効行数とコメント行数

表 7-4 ソース表示の色と行数カウント（その 1）

項番	行の種類	色	カウント
1	固定形式正書法で、7 カラムが「*」または「/」の行	行全体を緑色	コメント行数としてカウント
2	自由形式正書法で、1 カラムが「*」または「/」の行	行全体を緑色	コメント行数としてカウント
3	固定形式正書法で、7 カラムが「\$」の行	-CmDol オプションあり	コメント行数としてカウント
4		-CmDol オプションなし	ほかの規則に従う※
5	固定形式正書法で、1 カラムが「*」の行	-CmAster オプションあり	コメント行数としてカウント
6		-CmAster オプションなし	ほかの規則に従う※
7	固定形式正書法で、7 カラムが「D」または「d」の行	-DebugLine オプションあり	ほかの規則に従う※
8		-DebugLine オプションなし	コメント行数としてカウント
9	自由形式正書法で、1 カラムから「D△」または「d△」で始まる行	-DebugLine オプションあり	ほかの規則に従う※
10		-DebugLine オプションなし	コメント行数としてカウント
11	デバッグ指示子「>>D」で始まる行	-DebugLine オプションあり	ほかの規則に従う※
12		-DebugLine オプションなし	コメント行数としてカウント
13	空白行	行全体を緑色	コメント行数としてカウント
14	翻訳指令行（「>>」で始まる行）	行全体を緑色	コメント行数としてカウント
15	条件翻訳の無効行	行全体を緑色	コメント行数としてカウント

項番	行の種類		色	カウント
16	注記指示子	「*>」で始まる行	行全体を緑色	コメント行数としてカウント
17		有効な構文の後の「*>」	行全体を黒色	有効行数としてカウント
18	注記項の行（AUTHOR などの注記項段落の行も含む）		行全体を緑色	コメント行数としてカウント
19	有効行（上記以外の行）		行全体を黒色	有効行数としてカウント

（凡例）

△：1 個の半角空白，全角空白，タブまたは改行を示します。

注※

指定された条件だけでは，行種別が決定しません。そのため，ほかの項番（行の種類）の規則に従います。

例)

「*>」で始まる行のとき，色は「行全体を緑色」，カウントは「コメント行数としてカウント」です。注記項の行のとき，色は「行全体を緑色」，カウントは「コメント行数としてカウント」です。有効な構文（DISPLAY 文など）があるとき，色は「行全体を黒色」，カウントは「有効行数としてカウント」になります。

次に示すドキュメントのソース表示の色と行数カウントについて次の表に示します。

- HTML ドキュメント「ソース表示」のソースコードの文字色
- CSV ドキュメント「COBOL ソースファイル一覧 CSV」の有効行数とコメント行数

表 7-5 ソース表示の色と行数カウント（その 2）

項番	行の種類		色	カウント
1	固定形式正書法で，7 カラムが「*」または「/」の行		行全体を緑色	コメント行数としてカウント
2	自由形式正書法で，1 カラムが「*」または「/」の行		行全体を緑色	コメント行数としてカウント
3	固定形式正書法で，7 カラムが「\$」の行	-CmDol オプションあり	ほかの規則に従う※	
4		-CmDol オプションなし		
5	固定形式正書法で，1 カラムが「*」の行	-CmAster オプションあり	ほかの規則に従う※	
6		-CmAster オプションなし		
7	固定形式正書法で，7 カラムが「D」または「d」の行	-DebugLine オプションあり	行全体を緑色	コメント行数としてカウント
8		-DebugLine オプションなし	行全体を緑色	コメント行数としてカウント

項番	行の種類		色	カウント
9	自由形式正書法で、1 カラムから「D△」または「d△」で始まる行	-DebugLine オプションあり	行全体を緑色	コメント行数としてカウント
10		-DebugLine オプションなし	行全体を緑色	コメント行数としてカウント
11	デバッグ指示子「>>D」で始まる行	-DebugLine オプションあり	行全体を黒色 「>>D」以降を緑色	コメント行数としてカウント
12		-DebugLine オプションなし	行全体を黒色 「>>D」以降を緑色	コメント行数としてカウント
13	空白行		行全体を黒色	コメント行数としてカウント
14	翻訳指令行（「>>」で始まる行）		行全体を黒色 「>>」以降を緑色	コメント行数としてカウント
15	条件翻訳の無効行		行全体を黒色	有効行数としてカウント
16	注記指示子	「*>」で始まる行	行全体を黒色 「*>」以降を緑色	コメント行数としてカウント
17		有効な構文の後の「*>」	行全体を黒色 「*>」以降を緑色	有効行数としてカウント
18	注記項の行（AUTHOR などの注記項段落の行も含む）		行全体を黒色	有効行数としてカウント
19	有効行（上記以外の行）		行全体を黒色	有効行数としてカウント

（凡例）

△：1 個の半角空白，全角空白，タブまたは改行を示します。

注

行の種類の固定形式正書法または自由形式正書法は，ファイルの拡張子で判断します。判断できない場合は，固定形式正書法と見なしてドキュメントを処理します。

注※

指定された条件だけでは，行種別が決定しません。そのため，ほかの項番（行の種類）の規則に従います。

例）

「*>」で始まる行のとき，色は「「*>」以降を緑色」，カウントは「コメント行数としてカウント」です。注記項の行のとき，色は「行全体を黒色」，カウントは「有効行数としてカウント」です。

有効な構文（DISPLAY 文など）があるとき，色は「行全体を黒色」，カウントは「有効行数としてカウント」になります。

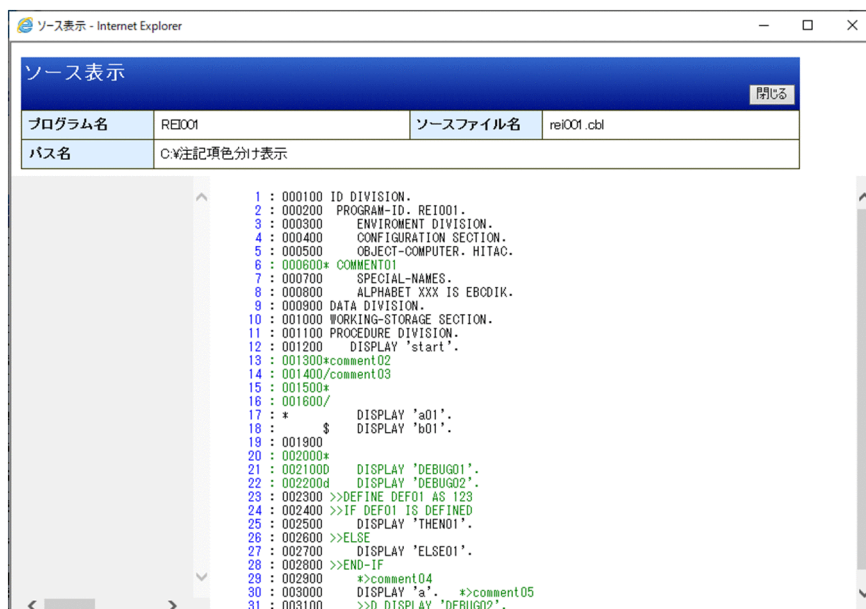
ソース表示のソースコードの文字色の表示例

コンパイラオプション-Compat85,All を指定して解析・生成したソース表示とソース表示(COPY 展開後)のソースコードの文字色を次に示します。

解析対象の COBOL プログラムは次のとおりです。なお、6 行を除く 3～8 行は注記項の行です。3 行は「ENVIRONMENT」の途中の「N」がなくて「ENVIROMENT」になっているため、部の見出しになりません。A 領域から始まる部の見出しが現れる 9 行の直前の 8 行までを注記項の行とします。

1 : 000100 ID DIVISION.	
2 : 000200 PROGRAM-ID. REI001.	
3 : 000300 ENVIROMENT DIVISION.	※ 「ENVIRONMENT」の途中の「N」がないため、部の見出しになりません。
4 : 000400 CONFIGURATION SECTION.	
5 : 000500 OBJECT-COMPUTER. HITAC.	
6 : 000600* COMMENT01	注記項の行
7 : 000700 SPECIAL-NAMES.	
8 : 000800 ALPHABET XXX IS EBCDIK.	
9 : 000900 DATA DIVISION.	
10 : 001000 WORKING-STORAGE SECTION.	
11 : 001100 PROCEDURE DIVISION.	
12 : 001200 DISPLAY 'start'.	
13 : 001300*comment02	
14 : 001400/comment03	
15 : 001500*	
16 : 001600/	
17 : * DISPLAY 'a01'.	
18 : \$ DISPLAY 'b01'.	
19 : 001900	
20 : 002000a	
21 : 002100D DISPLAY 'DEBUG01'.	
22 : 002200d DISPLAY 'DEBUG02'.	
23 : 002300 >>DEFINE DEF01 AS 123	
24 : 002400 >>IF DEF01 IS DEFINED	
25 : 002500 DISPLAY 'THEN01'.	
26 : 002600 >>ELSE	
27 : 002700 DISPLAY 'ELSE01'.	
28 : 002800 >>END-IF	
29 : 002900 *>comment04	
30 : 003000 DISPLAY 'a'. *>comment05	
31 : 003100 >>D DISPLAY 'DEBUG02'.	

例 1 「ソース表示」のソースコードの文字色



例 2 「ソース表示(COPY 展開後)」のソースコードの文字色

17 行は、-CmAster オプションありのときは、行全体が緑色になります。

18 行は、-CmDol オプションありのときは、行全体が緑色になります。

21, 22, 31 行は、-DebugLine オプションありのときは、行全体が黒色になります。

23 行の DEFINE 指令がないとき、25 行は行全体が緑色になり、27 行は行全体が黒色になります。



7.6.3 ソース表示で表示されるソース行の最大長

COBOL ソース解析の生成ドキュメントのソース表示で表示される、ソース行の最大長を次の表に示します。

表 7-6 ソース表示で表示されるソース行の最大長

COBOL ソースファイルの正書法	ソース行の最大長（カラム）	
	[ソース表示] 画面	[ソース表示(COPY 展開後)] 画面
固定形式正書法	255	255 または 80※
自由形式正書法	255	255

注※

次のコンパイラオプションのどれかが有効である場合、ソース行の最大長を 80 とします。

-V3Spec -V3Rec,Fixed

8

解析結果のリファレンス（CSV ファイル形式）

COBOL ソース解析で出力する、ソース解析情報（CSV ファイル形式）の内容について説明します。

8.1 CSV ファイルの種類

CSV ファイルの種類とファイル共通フォーマットの説明です。

CSV ファイルの種類

各 CSV ファイルは、1 つの解析対象プロジェクトで 1 個生成されます。それぞれの CSV ファイルの種類を次に示します。

- 一覧情報 CSV

解析対象プロジェクト単位に各情報を一覧形式でレコード出力します。一覧情報 CSV の種類を次に示します。

- COBOL ソースファイル一覧
- プログラム一覧
- セクション一覧
- ファイル一覧
- データ項目一覧
- 不整合引数一覧
- 実行順序リスト

- 関係情報 CSV

解析対象プロジェクト単位にプログラムと各情報の関係を二次元マトリクス形式でレコード出力します。関係情報 CSV の種類を次に示します。

- プログラム呼出関係
- 登録集原文参照関係
- ファイル参照関係
- 表(DB)参照関係

CSV ファイルの形式

CSV ファイルの形式を次に示します。

- 1 行目に「項目名 1～n」、2 行目以降に「データ 1～n」を出力します。文字列データは引用符「"」で囲みます。文字列データに引用符「"」を含む場合には 2 個続けて出力します。
- 文字列データが空文字列（長さゼロの文字列）の場合には、引用符「"」を 2 個続けて「""」を出力します。
- 数値データは、引用符なしでそのまま数値を出力します。
- 数値データがゼロの場合には、引用符なしで 0 を出力します。
- 項目間は、コンマ「,」で区切って出力します。

8.2 一覧情報 CSV

COBOL ソース解析に登録された COBOL ソースファイル，登録集原文ファイル，および解析対象の COBOL ソースファイルに含まれるプログラムの一覧を CSV ファイル形式で出力します。

8.2.1 COBOL ソースファイル一覧

COBOL ソース解析に登録された COBOL ソースファイルと，登録集原文ファイルの一覧です。

出力ファイル名を次に示します。

COBOLソースファイル一覧.csv

フォーマットを次に示します。

"プロジェクト名","ファイル種別","COBOLソースファイル名","使用有無","総行数","有効行数","コメント行数","パス名"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[ファイル種別]

ファイルの種別です。出力される文字の意味を次に示します。

- プログラム
COBOL ソースファイル
- 登録集
登録集原文ファイル

[COBOL ソースファイル名]

COBOL ソースファイル名，または，登録集原文ファイル名です。

解析対象プロジェクト内でファイル名が重複する場合は，同じファイル名が連続して表示されます。重複するファイルのそれぞれの格納場所は，[パス名]で確認してください。

[使用有無]

COBOL ソースファイルまたは登録集原文ファイルの使用有無です。出力される文字の意味を次に示します。

- 使用
COBOL ソースファイルまたは登録集原文ファイルは使用されています。
- 未使用
COBOL ソースファイルの場合

COBOL ソースファイル内のすべてのプログラムが未使用です。デッドコードの CALL 文だけから呼ばれるプログラムは未使用として扱います。

登録集原文ファイルの場合

プロジェクトのどの COBOL ソースファイルからも参照されていません。

解析生成オプションの環境変数 CBLLIB に指定された登録集原文ファイルの検索フォルダの中から、COBOL ソースファイルで参照されない登録集原文ファイルが未使用の登録集原文ファイルとして抽出されます。

- 主

[解析生成オプション設定] 画面の [主プログラム指定] タブの指定によって主プログラムと認識されたプログラムを含む COBOL ソースファイルです。

- エラー

エラーが検出された COBOL ソースファイルです。または、登録集原文の取り込み失敗で警告エラーが検出された COBOL ソースファイルです。

[総行数]

COBOL ソースファイルまたは登録集原文ファイルの総行数 (COPY 文展開前) です。

[有効行数]

総行数からコメント行数を引いた値 (COPY 文展開前) です。

[コメント行数]

COBOL ソースファイルまたは登録集原文ファイルのコメント行と注記項の総数です (COPY 文展開前)。コメント行の行数カウントについては、「[7.6.2 ソース表示の色と行数カウント](#)」を参照してください。

[パス名]

COBOL ソースファイルまたは登録集原文ファイルが格納されているフォルダのパスです。

CSV ファイルの出力例

```
"プロジェクト名","ファイル種別","COBOLソースファイル名","使用有無","総行数","有効行数","コメント行数","パス名"
"入在庫管理システム","プログラム","オプション処理 1 .cbl","主",22,11,11,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","オプション処理 2 .cbl","主",24,13,11,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","オプション処理 3 .cbl","主",21,13,8,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","チェック処理.cbl","主",115,100,15,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","メニュー画面.cbl","主",64,51,13,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","出荷処理.cbl","主",283,201,82,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","詳細チェック処理.cbl","主",211,152,59,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","制御処理.cbl","主",269,199,70,"C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","プログラム","転送処理.cbl","主",335,256,79,"C:¥入出荷管理システム¥入出荷管理システム"
```

荷管理システム”
 ”入出庫管理システム”, ”プログラム”, ”入力処理. cbl”, ”主”, 218, 159, 59, ”C:¥入出荷管理システム¥入出荷管理システム”
 ”入出庫管理システム”, ”登録集”, ”ACCTCD. CBL”, ”使用”, 2, 2, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”FC351DT. CBL”, ”使用”, 3, 3, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”FC370. CBL”, ”使用”, 15, 15, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”FC380. CBL”, ”使用”, 4, 4, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”FGC301. CBL”, ”使用”, 17, 17, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”GC351. CBL”, ”使用”, 9, 9, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”GC351DT. CBL”, ”使用”, 3, 3, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”GC351ERR. CBL”, ”使用”, 16, 16, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”PRCCD. CBL”, ”使用”, 5, 5, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”RC340. CBL”, ”使用”, 17, 17, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”SQLCA. COB”, ”使用”, 12, 11, 1, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”WK. CBL”, ”使用”, 4, 4, 0, ”C:¥入出荷管理システム¥入出荷管理システム¥COPY登録集”
 ”入出庫管理システム”, ”登録集”, ”XDMDMSG. COB”, ”未使用”, 13, 10, 3, ”C:¥Program Files¥Hitachi¥COBOL2002¥COPY”

注意事項

- 未使用の COBOL ソースファイルからだけ COPY 文展開される登録集原文は、デッドコードと見なして【使用有無】に「未使用」と出力されます。
- デッドコードとなるスタックコンパイルのプログラムからだけ COPY 文展開される登録集原文であっても、展開元の COBOL ソースファイルが使用されていれば、その登録集原文は【使用有無】に「使用」と出力されます。
- どの COBOL ソースファイルからも COPY 文展開されない登録集原文は、【使用有無】に「未使用」と出力されます。
- COBOL ソースファイル一覧は、【ファイル種別】と【COBOL ソースファイル名】でソートされて出力されます。【ファイル種別】は、「プログラム」「登録集」の順に、【COBOL ソースファイル名】は、昇順にソートされます。

関連項目

- 「7.6.2 ソース表示の色と行数カウント」

8.2.2 プログラム一覧

解析対象の COBOL ソースファイルに含まれるプログラムの一覧です。

出力ファイル名を次に示します。

プログラム一覧.csv

フォーマットを次に示します。

"プロジェクト名","プログラム種別","プログラム名","リネーム後","使用有無","ソースファイル名","パス名","コメント行の文字列","総行数","有効行数","コメント行数","参照する登録集原文数","ファイル数","記述項数","データ項目数","実行文の数","分岐の数","節数","段落数","表(DB)数","呼出元プログラム数","呼出先プログラム数"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[プログラム種別]

プログラム構造の種別です。

- 外部プログラム
外部プログラムを示します。
- 内部プログラム
内部プログラムを示します。

[プログラム名]

プログラム名です。

[リネーム後]

重複するプログラム名が存在するため、解析時に割り当てられたプログラム名です。

[使用有無]

プログラムの使用有無です。

- 使用
主プログラムから直接的または間接的に呼び出されるプログラムです。
- 未使用
プログラムから直接的にも間接的にも呼び出されないプログラムです。
- 主
[解析生成オプション設定] 画面の [主プログラム指定] タブの指定によって主プログラムと認識されたプログラムです。

[ソースファイル名]

COBOL ソースファイル名です。

[パス名]

COBOL ソースファイルが格納されているフォルダのパスです。

[コメント行の文字列]

抽出したプログラムコメント行の文字列です。

[総行数]

プログラムの総行数（COPY 文展開前）です。

[有効行数]

総行数からコメント行数を引いた値（COPY 文展開前）です。

[コメント行数]

COBOL ソースファイルまたは登録集原文ファイルのコメント行と注記項の総数です（COPY 文展開前）。コメント行の行数カウントについては、「[7.6.2 ソース表示の色と行数カウント](#)」を参照してください。

[参照する登録集原文数]

直接参照する登録集原文の数です。

[ファイル数]

プログラム内で使用するファイルの数です。

[記述項数]

01～49, 66, 77 レベルのデータ名の数です。

[データ項目数]

01, 77 レベルのデータ項目の数です。

[実行文の数]

プログラム内の実行文の数です。対象となる実行文については、「[10.8 実行文として数える文](#)」を参照してください。

[分岐の数]

プログラム内の分岐の数です。分岐の数は、COBOL2002 カバレッジ情報の C1 メジャーの数で示します。詳細については、マニュアル「COBOL2002 操作ガイド」のカバレッジ情報の説明を参照してください。

[節数]

手続き部の節の数です。

[段落数]

手続き部の段落の数です。

[表(DB)数]

埋め込み SQL 文で使用している表の数です。

[呼出元プログラム数]

このプログラムを呼び出している親プログラムの数です。

[呼出先プログラム数]

プログラム内で呼び出している子プログラムの数です。

CSV ファイルの出力例

```
"プロジェクト名","プログラム種別","プログラム名","リネーム後","使用有無","ソースファイル名",
"パス名","コメント行の文字列","総行数","有効行数","コメント行数","参照する登録集原文数",
"ファイル数","記述項数","データ項目数","実行文の数","分岐の数","節数","段落数","表(DB)数",
"呼出元プログラム数","呼出先プログラム数"
"入在庫管理システム","外部プログラム","AC000_MENU","","主","メニュー画面.cbl","C:¥入在庫管理
システム","メニュー画面",64,51,13,0,0,9,4,14,7,1,2,0,0,3
"入在庫管理システム","外部プログラム","GC351_INPUT","","使用","チェック処理.cbl","C:¥入在庫
管理システム","チェック処理",115,100,15,3,0,29,3,45,18,3,2,0,2,1
"入在庫管理システム","外部プログラム","GC351_INPUT_DT","","使用","詳細チェック処理.cbl","C:
¥入在庫管理システム","詳細チェック処理",211,152,59,6,1,68,10,55,16,9,18,1,1,0
"入在庫管理システム","外部プログラム","RC340_INPUT","","使用","入力処理.cbl","C:¥入在庫管理
システム","入力処理",221,162,59,6,1,95,9,68,16,10,18,1,1,2
"入在庫管理システム","外部プログラム","RC340_SET","","使用","転送処理.cbl","C:¥入在庫管理シ
ステム","転送処理",335,256,79,6,0,102,12,107,31,13,24,2,1,1
"入在庫管理システム","外部プログラム","RC340_SND","","主","出荷処理.cbl","C:¥入在庫管理シス
テム","出荷処理",283,201,82,4,1,64,8,76,22,14,28,1,1,1
"入在庫管理システム","外部プログラム","RC340_UPD","","主","制御処理.cbl","C:¥入在庫管理シス
テム","制御処理",269,199,70,4,1,76,9,77,24,11,22,1,1,1
"入在庫管理システム","外部プログラム","ZC001_OPT","","主","オプション処理1.cbl","C:¥入在庫
管理システム","オプション処理1",22,11,11,1,0,4,1,2,0,1,0,0,1,0
"入在庫管理システム","外部プログラム","ZC002_OPT","","未使用","オプション処理2.cbl","C:¥入
在庫管理システム","オプション処理2",24,13,11,2,0,21,2,2,0,1,2,0,0,0
"入在庫管理システム","外部プログラム","ZC003_OPT","","使用","オプション処理3.cbl","C:¥入出
庫管理システム","オプション処理3",21,13,8,2,0,21,2,2,0,1,2,0,1,0
```

注意事項

- 外部プログラム名が重複している場合、[プログラム名] には同じプログラム名が出力されて、[リネーム後] にリネームプログラム名が出力されます。
- プログラム一覧は、[プログラム名] で昇順にソートされて出力されます。

関連項目

- 「7.6.2 ソース表示の色と行数カウント」

8.2.3 セクション一覧

手続き部 (PROCEDURE DIVISION) にあるすべての節および段落の一覧です。

出力ファイル名を次に示します。

セクション一覧.csv

フォーマットを次に示します。

”プロジェクト名”, ”プログラム名”, ”節名”, ”段落名”, ”種別”, ”区分”, ”使用有無”, ”総行数”, ”有効行数”, ”コメント行数”, ”ソースファイル名”, ”パス名”, ”コメント”

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[プログラム名]

節が定義されているプログラム名です。

[節名]

節名です。手続き部の先頭が節の見出しで始まっていない場合, 「{PROCEDURE_START}」という節名が出力されます。

[段落名]

段落名です。種別が節の場合は空文字列が出力されます。

[種別]

情報の種別です。

- 節
節の情報です。
- 段落
段落の情報です。

[区分]

手続きの区分です。

- 宣言部分
宣言部分の節または段落です。
- 手続き部分
手続き部分の節または段落です。

[使用有無]

節または段落の使用有無です。

- 使用
使用されています（制御が渡ります）。
- 未使用
未使用です（制御は渡りません）。

注意事項

- PERFORM 文ではなく、節見出しまたは段落見出しの前の行から制御が移る場合は、節名や段落名は参照されていなくても、その節または段落は「使用」と扱います。

- 節に属する段落が1つでも使用されている場合は、節名は参照されていなくても、その節は「使用」と扱います。

[総行数]

節または段落の総行数（有効行数とコメント行数の合計）です。

[有効行数]

節または段落内の有効行数です（COPY 文展開行を含む）。

[コメント行数]

節または段落内のコメント行の総数です（COPY 文展開行を含む）。コメント行の行数カウントについては、「7.6.2 ソース表示の色と行数カウント」を参照してください。

[ソースファイル名]

COBOL ソースファイル名です。

[パス名]

COBOL ソースファイルが格納されているフォルダのパスです。

[コメント]

節コメントです。COBOL ソース解析で節コメントが抽出されていない場合や段落の場合は、空文字列を出力します。

CSV ファイルの出力例

```
"プロジェクト名","プログラム名","節名","段落名","種別","区分","使用有無","総行数","有効行数",
"コメント行数","ソースファイル名","パス名","コメント"
"入出庫管理システム","AC000_MENU","AC000-MENU-MAIN","","節","手続き部分","使用",27,26,1,"メ
ニュー画面.cbl","C:¥入出荷管理システム¥入出荷管理システム","メニュー画面"
"入出庫管理システム","AC000_MENU","AC000-MENU-MAIN","AC000-MENU-MAIN-010","段落","手続き部分",
"使用",23,23,0,"メニュー画面.cbl","C:¥入出荷管理システム¥入出荷管理システム",""
"入出庫管理システム","AC000_MENU","AC000-MENU-MAIN","AC000-MENU-MAIN-999","段落","手続き部分",
"使用",2,2,0,"メニュー画面.cbl","C:¥入出荷管理システム¥入出荷管理システム",""
"入出庫管理システム","GC351_INPUT","GC351-INPUT-CHK-PROC","","節","手続き部分","使用",89,85,
4,"チェック処理.cbl","C:¥入出荷管理システム¥入出荷管理システム","売上情報入力チェック"
"入出庫管理システム","GC351_INPUT","GC351-INPUT-CHK-PROC","GC351-INPUT-CHK-PROC-010","段落",
"手続き部分","使用",85,82,3,"チェック処理.cbl","C:¥入出荷管理システム¥入出荷管理システム",""
"入出庫管理システム","GC351_INPUT","GC351-INPUT-CHK-PROC","GC351-INPUT-CHK-PROC-999","段落",
"手続き部分","使用",2,2,0,"チェック処理.cbl","C:¥入出荷管理システム¥入出荷管理システム",""
"入出庫管理システム","GC351_INPUT","ABEND-SUB-PROC","","節","手続き部分","使用",3,2,1,
"チェック処理.cbl","C:¥入出荷管理システム¥入出荷管理システム","終了処理"
"入出庫管理システム","GC351_INPUT","GC351-KYT-CHECK-PROC","","節","手続き部分","使用",4,3,1,
"チェック処理.cbl","C:¥入出荷管理システム¥入出荷管理システム","拠点コードチェック処理"
"入出庫管理システム","RC340_INPUT","RC340-INPUT-PROC","","節","手続き部分","使用",39,33,6,
"入力処理.cbl","C:¥入出荷管理システム¥入出荷管理システム","受注DB入力処理"
"入出庫管理システム","RC340_INPUT","RC340-INPUT-PROC","RC340-INPUT-PROC-010","段落","手続き",
"部分","使用",32,30,2,"入力処理.cbl","C:¥入出荷管理システム¥入出荷管理システム",""
"入出庫管理システム","RC340_INPUT","RC340-INPUT-PROC","RC340-INPUT-PROC-999","段落","手続き",
"部分","使用",5,2,3,"入力処理.cbl","C:¥入出荷管理システム¥入出荷管理システム",""
"入出庫管理システム","RC340_INPUT","GC351-SET","","節","手続き部分","使用",6,2,4,"入力処理.c",
"bl","C:¥入出荷管理システム¥入出荷管理システム","チェック準備処理"
```

"入在庫管理システム", "RC340 INPUT", "RC340-INPUT-ERR-PROC", "", "節", "手続き部分", "使用", 6, 5, 1, "入力処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", "エラー処理"

"入在庫管理システム", "RC340 INPUT", "RC340-INPUT-ERR-PROC", "RC340-INPUT-ERR-PROC-010", "段落", "手続き部分", "使用", 2, 2, 0, "入力処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 INPUT", "RC340-INPUT-ERR-PROC", "RC340-INPUT-ERR-PROC-999", "段落", "手続き部分", "使用", 2, 2, 0, "入力処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "RC340-UPD-PROC-MAIN", "", "節", "手続き部分", "使用", 32, 25, 7, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", "受注DB更新処理"

"入在庫管理システム", "RC340 SND", "RC340-UPD-PROC-MAIN", "RC340-UPD-PROC-MAIN-010", "段落", "手続き部分", "使用", 25, 22, 3, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "RC340-UPD-PROC-MAIN", "RC340-UPD-PROC-MAIN-999", "段落", "手続き部分", "使用", 5, 2, 3, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "PRINT-BILL-PROC", "", "節", "手続き部分", "未使用", 9, 5, 4, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", "納品書・請求書発行処理"

"入在庫管理システム", "RC340 SND", "PRINT-BILL-PROC", "PRINT-BILL-PROC-010", "段落", "手続き部分", "未使用", 2, 2, 0, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "PRINT-BILL-PROC", "PRINT-BILL-PROC-999", "段落", "手続き部分", "未使用", 5, 2, 3, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "PRINT-VOUCHER-P-PROC", "", "節", "手続き部分", "使用", 9, 5, 4, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", "小包伝発行処理"

"入在庫管理システム", "RC340 SND", "PRINT-VOUCHER-P-PROC", "PRINT-VOUCHER-P-PROC-10", "段落", "手続き部分", "使用", 2, 2, 0, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "PRINT-VOUCHER-P-PROC", "PRINT-VOUCHER-P-PROC-999", "段落", "手続き部分", "使用", 5, 2, 3, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "PRINT-VOUCHER-T-PROC", "", "節", "手続き部分", "使用", 9, 5, 4, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", "宅急便伝票発行処理"

"入在庫管理システム", "RC340 SND", "PRINT-VOUCHER-T-PROC", "PRINT-VOUCHER-T-PROC-010", "段落", "手続き部分", "使用", 2, 2, 0, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

"入在庫管理システム", "RC340 SND", "PRINT-VOUCHER-T-PROC", "PRINT-VOUCHER-T-PROC-999", "段落", "手続き部分", "使用", 5, 2, 3, "出荷処理.cbl", "C:¥入出荷管理システム¥入出荷管理システム", ""

注意事項

- ・ セクション一覧は、[プログラム名] で昇順にソートされて出力されます。なお、同じプログラム内では、節や段落の出現順に出力されます。
- ・ 解析でエラーが出力された COBOL ソースファイルの場合、セクション一覧には節や段落情報が出力されないか、または古い COBOL ソースファイルの節や段落情報が出力されます。

関連項目

- ・ 「7.6.2 ソース表示の色と行数カウント」

8.2.4 ファイル一覧

ファイル管理記述項の SELECT 句で宣言されたファイル名の一覧です。

出力ファイル名を次に示します。

ファイル一覧.csv

フォーマットを次に示します。

"プロジェクト名","プログラム名","ファイル名","READ","WRITE","REWRITE","DELETE","使用有無","GLOBAL","EXTERNAL","定義元ソースファイル名","パス名"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[プログラム名]

プログラム名です。

[ファイル名]

ファイル管理記述項の SELECT 句で指定したファイル名です。

[READ]

ファイルを参照している入出力文が READ 文かどうかを「TRUE」または「FALSE」で示します。

- TRUE
READ 文で参照されています。
- FALSE
READ 文で参照されていません。

[WRITE]

ファイルを参照している入出力文が WRITE 文かどうかを「TRUE」または「FALSE」で示します。

- TRUE
WRITE 文で参照されています。
- FALSE
WRITE 文で参照されていません。

[REWRITE]

ファイルを参照している入出力文が REWRITE 文かどうかを「TRUE」または「FALSE」で示します。

- TRUE
REWRITE 文で参照されています。
- FALSE
REWRITE 文で参照されていません。

[DELETE]

ファイルを参照している入出力文が DELETE 文かどうかを「TRUE」または「FALSE」で示します。

- TRUE
DELETE 文で参照されています。
- FALSE

DELETE 文で参照されていません。

[使用有無]

ファイルの使用有無です。

- 使用
プログラムの実行でファイルまたはレコードが参照されています。
- 未使用
プログラムの実行でファイル、レコードの両方とも参照されていません。

[GLOBAL]

GLOBAL 句の指定有無を「TRUE」または「FALSE」で示します。

- TRUE
GLOBAL 句が指定されています。
- FALSE
GLOBAL 句は指定されていません。

[EXTERNAL]

EXTERNAL 句の指定有無を「TRUE」または「FALSE」で示します。

- TRUE
EXTERNAL 句が指定されています。
- FALSE
EXTERNAL 句は指定されていません。

[定義元ソースファイル名]

ファイル名の定義元の COBOL ソースファイル名です。

[パス名]

COBOL ソースファイルが格納されているフォルダのパスです。

CSV ファイルの出力例

```
"プロジェクト名","プログラム名","ファイル名","READ","WRITE","REWRITE","DELETE","使用有無","G
LOBAL","EXTERNAL","定義元ソースファイル名","パス名"
"入在庫管理システム","RC340_INPUT","FC370","TRUE","FALSE","FALSE","FALSE","使用","FALSE","FA
LSE","入力処理.cbl","C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","RC340_SND","FC360","FALSE","FALSE","FALSE","FALSE","未使用","FALSE","F
ALSE","出荷処理.cbl","C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","RC340_SND","FC380","TRUE","FALSE","FALSE","FALSE","使用","FALSE","TRUE
","出荷処理.cbl","C:¥入出荷管理システム¥入出荷管理システム"
"入在庫管理システム","RC340_UPD","FC370","TRUE","FALSE","FALSE","FALSE","使用","FALSE","FALS
E","制御処理.cbl","C:¥入出荷管理システム¥入出荷管理システム"
```


注意事項

- ファイル一覧は、[プログラム名] で昇順にソートされて出力されます。なお、同じプログラム内では、[ファイル名] で昇順にソートされて出力されます。
- デッドコード候補の制御が渡らない文や手続きだけから参照されている場合、および未使用プログラムのファイルの場合、[READ], [WRITE], [REWRITE] および [DELETE] には「参照」、[使用有無] には「未使用」が出力されます。
- 解析でエラーが出力された COBOL ソースファイルの場合、ファイル一覧にはファイル情報が出力されないか、または古い COBOL ソースファイルのファイル情報が出力されます。

8.2.5 データ項目一覧

プログラムごとの、ファイル節、作業場所節、局所場所節、または連絡節のデータ記述項で定義されたデータ項目の一覧です。また、[解析生成オプション設定] 画面の [その他の解析オプション] タブでレベル番号が 02～49 および 66 のデータ項目の抽出を有効（チェックボックスをオン）にしている場合は、集団項目の従属項目の情報も出力されます。

出力ファイル名を次に示します。

データ項目一覧_プログラム名.csv

フォーマットを次に示します。

"プロジェクト名","プログラム名","定義場所","ファイル名","オフセット","ビットオフセット","論理レベル","レベル番号","データ名","領域属性","データ型","サイズ","編集文字列","符号種別","他属性","反復回数","有効範囲","使用有無","参照有無","完全修飾名","変換前データ名","ファイルパス名","相対行番号","コメント文字列"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[プログラム名]

プログラム名です。

[定義場所]

データ記述項を記述している節の名前です。

- ファイル節
ファイル節で定義しているレコードです。
- 作業場所節
作業場所節で定義しているデータ項目です。

- 局所場所節
局所場所節で定義しているデータ項目です。
- 連絡節
連絡節で定義しているデータ項目です。

[ファイル名]

データ項目が属するファイル名です。

ファイル節のデータ記述項で定義しているデータ項目以外の場合は、空文字列が出力されます。

[オフセット]

01 レベルの集団項目の先頭からのバイトオフセットです。

01, 77 レベルのデータ項目の場合は、「0」が出力されます。また、オフセットが不定な場合 (DEPENDING ON 指定のある OCCURS 句を記述したデータ項目のあとのデータ項目の場合) は、「-1」が出力されます。

[ビットオフセット]

領域先頭のビットオフセットです。

内部ブール項目でない、または内部ブール項目がバイトオフセットから始まる場合は、「0」が出力されます。

[論理レベル]

01 レベルのデータ項目を 1 とした階層レベルです。

再命名項目の場合は、「2」が出力されます。

[レベル番号]

データ記述項のレベル番号です。

[データ名]

データ名です。

[領域属性]

データ項目の領域に関する属性情報です。

- REDEFINES データ名
REDEFINES 句の情報を示します。
- RENAMES データ名 [THRU データ名]
RENAMES 句の情報を示します。
- ADDRESSED BY アドレス名
ADDRESSED BY 句の情報を示します。

[データ型]

データ項目のデータ型です。

データ型	説明
集団項目	固定長の英数字集団項目を示します。

データ型	説明
集団項目(可変長)	可変長の英数字集団項目を示します。
日本語集団項目	日本語集団項目 (GROUP-USAGE IS NATIONAL) を示します。
英字	英字項目を示します。
英数字	英数字項目を示します。
動的長英数字	動的長英数字項目を示します。
英数字編集	英数字編集項目を示します。
外部 10 進	外部 10 進項目を示します。
内部 10 進	内部 10 進項目 (COMP-3 または PACKED-DECIMAL) を示します。
2 進	2 進項目 (COMP, COMP-4 または BINARY) を示します。
2 進(COMP-X)	COMP-X 項目を示します。
2 進(COMP-5)	COMP-5 項目を示します。
数字編集	数字編集項目を示します。
外部浮動	外部浮動小数点項目を示します。
内部浮動	内部浮動小数点項目 (COMP-1 または COMP-2) を示します。
日本語	日本語項目を示します。
動的長日本語	動的長日本語項目を示します。
日本語編集	日本語編集項目を示します。
外部ブール	外部ブール項目を示します。
内部ブール	内部ブール項目を示します。
アドレス	アドレス項目, またはポインタ項目を示します。
指標	指標項目を示します。
オブジェクト参照	オブジェクト参照項目を示します。
OLE 参照	OLE オブジェクト参照項目を示します。
OLE バリエント	バリエント項目を示します。

[サイズ]

データ項目のバイト長 (内部ブール項目の場合はビット長) です。

[編集文字列]

PICTURE 句の文字列です。

COBOL ソースに記述した PICTURE 句の文字列のままではなく、PICTURE 文字の繰り返し指定「(文字数)」で正規化されたものが出力されます。

[符号種別]

SIGN 句の符号種別の情報です。

- ""
SIGN 句の指定がないことを示します。
- LEADING
SIGN IS LEADING を示します。
- TRAILING
SIGN IS TRAILING を示します。
- LEADING SEPARATE
SIGN IS LEADING SEPARATE を示します。
- TRAILING SEPARATE
SIGN IS TRAILING SEPARATE を示します。

[他属性]

属性情報です。文字 (|) で連結して出力されます。

- OCCURS
OCCURS 句の指定があることを示します。
- SYNC
SYNCHRONIZED 句の指定があることを示します。
- JUST
JUSTIFIED RIGHT 句の指定があることを示します。
- BLANK
BLANK WHEN ZERO 句の指定があることを示します。

[反復回数]

OCCURS 句の反復回数です。

OCCURS 句の指定がない場合は、「0」が出力されます。

[有効範囲]

名前の有効範囲です。

- 大域外部名
大域 (GLOBAL) 属性を持つ外部名を示します。
- 外部名
外部 (EXTERNAL) 属性を持つ名前を示します。
- 大域名
大域 (GLOBAL) 属性を持つ名前を示します。
- 局所名

外部（EXTERNAL）属性と大域（GLOBAL）属性の両方を持たない名前を示します。

- 無名

FILLER 項目または FILLER が仮定された無名のデータ項目を示します。

注意事項

集団項目が大域（GLOBAL）属性を持つ場合、その従属項目も大域（GLOBAL）属性を持ちます。

[使用有無]

01 レベルデータ項目（01, 77 レベルのデータ項目）の領域の使用有無です。レベル番号が 02～49, 66 のデータ項目の場合は、「-」が出力されます。

- 使用

制御が渡る可能性がある文（デッドコードでない文）で領域の全体、または一部が参照されていることを示します。

- 未使用

制御が渡る可能性がある文（デッドコードでない文）で領域のどこも参照されていないことを示します。

- -

01 レベルデータ項目と同じであることを示します。

[参照有無]

データ名のプログラム内の参照有無です。

- 参照

データ名がプログラム内で参照されていることを示します。

集団項目の場合、集団項目のデータ名が参照されていなくても、従属項目のデータ名が参照されているときは「参照」とします。条件変数の場合、条件名が参照されているときは「参照」とします。制御が渡らない文（デッドコードの文）だけから参照されている場合も「参照」とします。

ただし、データ名が手続き部（PROCEDURE DIVISION）の ENTRY 文の USING 指定または RETURNING 指定や、REDEFINES 句、RENAMES 句、または SAME AS 句だけに指定されている場合は、参照されていないもの（未参照）とします。

- 未参照

データ名がプログラム内で参照されていないことを示します。

集団項目の場合、集団項目のデータ名が参照されていないだけでなく、すべての従属項目のデータ名も参照されていないときだけ「未参照」とします。

[完全修飾名]

修飾子付きデータ名です。次の形式で出力されます。

データ名 [0F 集団項目名]… [IN ファイル名]

[変換前データ名]

次に示す変換を実施する前の COBOL ソースファイルや、登録集原文ファイルに記述したままのデータ名です。

- COPY 文の PREFIXING 指定による変換
- COPY 文の SUFFIXING 指定による変換
- COPY 文の REPLACING 指定による変換
- REPLACE 文による変換
- 文字の等価変換

■ 注意事項

- [データ名] と同じデータ名の場合は、空文字列が出力されます。
- FILLER 項目の場合は、空文字列が出力されます。
- [データ名] のデータ名が複数の原文語を置換して作成されている場合、先頭の原文語が変換前データ名として出力されます。
- 変換前データ名の長さが 120 バイトを超える場合、120 バイトで切り捨てられます。

[ファイルパス名]

データ記述項を含む COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[相対行番号]

データ記述項を含む COBOL ソースファイルまたは登録集原文ファイル内の相対行番号です。

[コメント文字列]

注記行の文字列です。

データ記述項の直前にある連続した注記行（空白行やデバッグ行は含まない）の場合は、次の方法で一行にした文字列が出力されます。

1. プログラム原文領域の先頭が「*」「/」「#」「+」「>」の場合は、半角空白とみなします。
また、プログラム原文領域の先頭から連続した「*」「/」「#」「+」「-」「=」「>」の同一文字の並びも、1 文字の半角空白とみなします。
2. プログラム原文領域中の 4 つ以上連続した「*」「/」「#」「+」「-」「=」の同一文字の並びは、1 文字の半角空白とみなします。
3. プログラム原文領域の最初の文字列が 2 桁の数字の場合は、非注記行が現れたとみなします。
4. プログラム原文領域中の半角空白の 2 つ以上連続した並びは、1 文字の半角空白とみなします。
5. 1.~4.の結果、各行のコメント文字列の先頭から連続した空白文字は削除します。
6. 複数行を結合したあとのコメント文字列の先頭および末尾の連続した空白文字は削除します。
7. 1.~6.の結果、コメント文字列が 127 文字を超える場合は、127 文字で切り捨てます。

(例)

COBOL ソースファイルの例

```
* 01 AAA PIC X.  
*****  
** 氏名△  
** (名字△名前)  
*****  
01 BBB PIC N(10).
```

コメント文字列の出力例

"氏名△(名字△名前)"

(凡例)

△：半角空白を示します。

CSV ファイルの出力例

```
"プロジェクト名","プログラム名","定義場所","ファイル名","オフセット","ビットオフセット","論  
理レベル","レベル番号","データ名","領域属性","データ型","サイズ","編集文字列","符号種別","他  
属性","反復回数","有効範囲","使用有無","参照有無","完全修飾名","変換前データ名","ファイルパ  
ス名","相対行番号","コメント文字列"  
"入在庫管理システム","GC351_INPUT_DT","ファイル節","FC351DT",0,0,1,"01","FC351DT-REC","","","集  
団項目",12,"","","",0,"局所名","使用","参照","FC351DT-REC IN FC351DT","","","C:¥入在庫管理シ  
ステム¥INCLUDE¥FC351DT.CBL",1,""  
"入在庫管理システム","GC351_INPUT_DT","ファイル節","FC351DT",0,0,2,"03","FC351DT-COND-CD","","  
","英数字",2,"X(2)","","","",0,"局所名","-","参照","FC351DT-COND-CD OF FC351DT-REC IN FC351DT",  
","","C:¥入在庫管理システム¥INCLUDE¥FC351DT.CBL",2,""  
"入在庫管理システム","GC351_INPUT_DT","ファイル節","FC351DT",2,0,2,"03","FC351DT-COND","","  
","英数字",10,"X(10)","","","",0,"局所名","-","未参照","FC351DT-COND OF FC351DT-REC IN FC351DT",  
","","C:¥入在庫管理システム¥INCLUDE¥FC351DT.CBL",3,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",0,0,1,"01","GC351DT-REC","","","集  
団項目",12,"","","",0,"局所名","使用","参照","GC351DT-REC","","","C:¥入在庫管理シ  
ステム¥INCLUDE¥GC351DT.CBL",1,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",0,0,2,"03","GC351DT-COND-CD","","","英  
数字",2,"X(2)","","","",0,"局所名","-","参照","GC351DT-COND-CD OF GC351DT-REC","","","C:¥入在庫管理  
システム¥INCLUDE¥GC351DT.CBL",2,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",2,0,2,"03","GC351DT-YSNKAN-NO","","","英  
数字",10,"X(10)","","","",0,"局所名","-","未参照","GC351DT-YSNKAN-NO OF GC351DT-REC","","","C:¥入  
在庫管理システム¥INCLUDE¥GC351DT.CBL",3,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",0,0,1,"01","DB1-GC351DT-REC","","","集  
団項目",12,"","","",0,"局所名","使用","参照","DB1-GC351DT-REC","GC351DT-REC","C:¥入在庫管理シ  
ステム¥INCLUDE¥GC351DT.CBL",1,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",0,0,2,"03","DB1-GC351DT-COND-CD","","  
","英数字",2,"X(2)","","","",0,"局所名","-","未参照","DB1-GC351DT-COND-CD OF DB1-GC351DT-REC","GC  
351DT-COND-CD","C:¥入在庫管理システム¥INCLUDE¥GC351DT.CBL",2,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",2,0,2,"03","DB1-GC351DT-YSNKAN-NO","","  
","英数字",10,"X(10)","","","",0,"局所名","-","参照","DB1-GC351DT-YSNKAN-NO OF DB1-GC351DT-REC",  
","GC351DT-YSNKAN-NO","C:¥入在庫管理システム¥INCLUDE¥GC351DT.CBL",3,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",0,0,1,"01","FC370-REC","","","集  
団項目",100,"","","",0,"局所名","使用","参照","FC370-REC","","","C:¥入在庫管理シ  
ステム¥INCLUDE¥FC370.CBL",1,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",0,0,2,"03","FC370-YSNKAN-NO","","","英  
数字",10,"X(10)","","","",0,"局所名","-","参照","FC370-YSNKAN-NO OF FC370-REC","","","C:¥入在庫管理  
システム¥INCLUDE¥FC370.CBL",2,""  
"入在庫管理システム","GC351_INPUT_DT","作業場所節","","",10,0,2,"03","FC370-SYOR-KB","","","英数字
```


",2,"X(2)","",0,"局所名","-", "未参照", "FC370-SYOR-KB OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 3, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 12, 0, 2, "03", "FC370-HSEIMOTCD", "", "英数字", 2, "X(2)", "", 0, "局所名", "-", "未参照", "FC370-HSEIMOTCD OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 4, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 14, 0, 2, "03", "FC370-SHK-KB", "", "英数字", 2, "X(2)", "", 0, "局所名", "-", "未参照", "FC370-SHK-KB OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 5, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 16, 0, 2, "03", "FC370-SHK-NO", "", "英数字", 10, "X(10)", "", 0, "局所名", "-", "未参照", "FC370-SHK-NO OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 6, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 26, 0, 2, "03", "FC370-SYOZK-KGYCD", "", "英数字", 10, "X(10)", "", 0, "局所名", "-", "未参照", "FC370-SYOZK-KGYCD OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 7, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 36, 0, 2, "03", "FC370-YSIGNP-KG", "", "英数字", 2, "X(2)", "", 0, "局所名", "-", "未参照", "FC370-YSIGNP-KG OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 8, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 38, 0, 2, "03", "FC370-TORI-KB", "", "英数字", 2, "X(2)", "", 0, "局所名", "-", "未参照", "FC370-TORI-KB OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 9, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 40, 0, 2, "03", "FC370-SHN-CD", "", "英数字", 10, "X(10)", "", 0, "局所名", "-", "未参照", "FC370-SHN-CD OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 10, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 50, 0, 2, "03", "FC370-SHNSUB-CD", "", "英数字", 10, "X(10)", "", 0, "局所名", "-", "未参照", "FC370-SHNSUB-CD OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 11, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 60, 0, 2, "03", "FC370-TORI-SYYMD", "", "英数字", 8, "X(8)", "", 0, "局所名", "-", "未参照", "FC370-TORI-SYYMD OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 12, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 68, 0, 2, "03", "FC370-SYKSIHR-KG", "", "英数字", 15, "X(15)", "", 0, "局所名", "-", "未参照", "FC370-SYKSIHR-KG OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 13, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 83, 0, 2, "03", "FC370-SIHR-KB", "", "英数字", 2, "X(2)", "", 0, "局所名", "-", "未参照", "FC370-SIHR-KB OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 14, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 85, 0, 2, "03", "FC370-SIHR-KG", "", "英数字", 15, "X(15)", "", 0, "局所名", "-", "未参照", "FC370-SIHR-KG OF FC370-REC", "", "C:¥入出庫管理システム¥INCLUDE¥FC370.CBL", 15, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 0, 0, 1, "01", "SQLCA", "", "集団項目", 36, "", "", 0, "局所名", "使用", "参照", "SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 1, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 0, 0, 2, "02", "SQLCAID", "", "集団項目", 8, "", "", 0, "局所名", "-", "未参照", "SQLCAID OF SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 2, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 0, 0, 3, "03", "SQLCAIDC", "", "英数字", 5, "X(5)", "", 0, "局所名", "-", "未参照", "SQLCAIDC OF SQLCAID OF SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 3, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 5, 0, 3, "03", "SQLCAIDS", "", "英数字", 2, "X(2)", "", 0, "局所名", "-", "未参照", "SQLCAIDS OF SQLCAID OF SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 4, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 7, 0, 3, "03", "SQLCAIDE", "", "英数字", 1, "X(1)", "", 0, "局所名", "-", "未参照", "SQLCAIDE OF SQLCAID OF SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 5, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 8, 0, 2, "02", "SQLCABC", "", "外部10進", 4, "9(4)", "", 0, "局所名", "-", "未参照", "SQLCABC OF SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 6, ""
 "入出庫管理システム", "GC351 INPUT_DT", "作業場所節", "", 12, 0, 2, "02", "SQLCODE", "", "外部10進", 4, "9(4)", "", 0, "局所名", "-", "参照", "SQLCODE OF SQLCA", "", "C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB", 7, ""

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","16,0,2,"02","SQLCODE-N","","","日本語",8,"N(4)","","","",0,"局所名","-","参照","SQLCODE-N OF SQLCA","","","C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB",8,""

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","24,0,2,"02","SQLERRM","","","集団項目",12,"","","",0,"局所名","-","未参照","SQLERRM OF SQLCA","","","C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB",9,""

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","24,0,3,"03","SQLERRML","","","外部10進",4,"S9(4)","","","",0,"局所名","-","未参照","SQLERRML OF SQLERRM OF SQLCA","","","C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB",10,""

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","28,0,3,"03","SQLERRMC","","","英数字",8,"X(8)","","","",0,"局所名","-","未参照","SQLERRMC OF SQLERRM OF SQLCA","","","C:¥入出庫管理システム¥INCLUDE¥SQLCA.COB",11,""

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","0,0,1,"01","WORK-AREA","","","集団項目",100,"","","",0,"局所名","使用","参照","WORK-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",24,"ワーク エリア *"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","0,0,2,"03","WK-ERR-MSG","","","日本語",100,"N(50)","","","",0,"局所名","-","参照","WK-ERR-MSG OF WORK-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",26,"D エラ-メッセージ"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","0,0,1,"01","FC351DT-WORK-AREA","","","集団項目",11,"","","",0,"局所名","使用","参照","FC351DT-WORK-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",30,"FC351DT 作業領域 *"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","0,0,2,"03","WK-FC351DT-END-SW","","","英数字",3,"X(3)","","","",0,"局所名","-","参照","WK-FC351DT-END-SW OF FC351DT-WORK-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",32,"D EOF スイッチ"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","3,0,2,"03","WK-FC351DT-CNT","","","2進",8,"S9(11)","","","",0,"局所名","-","参照","WK-FC351DT-CNT OF FC351DT-WORK-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",35,"D カウント 領域"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","0,0,1,"01","CR-GC351DT-01-FETCH-AREA","","","集団項目",9,"","","",0,"局所名","使用","参照","CR-GC351DT-01-FETCH-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",40,"CR-GC351DT-01 作業領域 *"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","0,0,2,"03","CR-GC351DT-01-END-SW","","","英数字",3,"X(3)","","","",0,"局所名","-","参照","CR-GC351DT-01-END-SW OF CR-GC351DT-01-FETCH-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",42,"D EOF スイッチ"

"入出庫管理システム","GC351_INPUT_DT","作業場所節","","3,0,2,"03","CR-GC351DT-01-FE-CNT","","","内部10進",6,"S9(11)","","","",0,"局所名","-","参照","CR-GC351DT-01-FE-CNT OF CR-GC351DT-01-FETCH-AREA","","","C:¥入出庫管理システム¥詳細チェック処理.cbl",45,"D 入力カウント エリア"

"入出庫管理システム","GC351_INPUT_DT","連絡節","","0,0,1,"01","GC351","","","集団項目",247,"","","",0,"局所名","使用","参照","GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",1,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","0,0,2,"03","GC351-YSNKAN-NO","","","英数字",100,"X(10)","","","",0,"局所名","-","未参照","GC351-YSNKAN-NO OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",2,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","10,0,2,"03","GC351-KYT-CD","","","英数字",10,"X(10)","","","",0,"局所名","-","未参照","GC351-KYT-CD OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",3,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","20,0,2,"03","INPUTBUCD","","","英数字",10,"X(10)","","","",0,"局所名","-","未参照","INPUTBUCD OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",4,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","30,0,2,"03","GC351-TRIHSO-KB","","","英数字",2,"X(2)","","","",0,"局所名","-","未参照","GC351-TRIHSO-KB OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",5,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","32,0,2,"03","KYJ-CNT","","","内部10進",5,"9(9)","","","",0,"局所名","-","未参照","KYJ-CNT OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",6,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","37,0,2,"03","GC351-NM-KJ","","","日本語",100,"N(50)","","","",0,"局所名","-","未参照","GC351-NM-KJ OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",7,""

"入出庫管理システム","GC351_INPUT_DT","連絡節","","137,0,2,"03","GC351-NM-KN","","","日本語",100,"N(50)","","","",0,"局所名","-","未参照","GC351-NM-KN OF GC351","","","C:¥入出庫管理システム¥INCLUDE¥GC351.CBL",8,""

```

UDE%GC351.CBL",8,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","237,0,2,"03","GC351-OPT-COND","","英数字",
10,"X(10)","",0,"局所名","-","参照","GC351-OPT-COND OF GC351","","C:¥入出庫管理システム¥I
NCLUDE%GC351.CBL",9,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","0,0,1,"01","C-ERR-GC351","","集団項目",75,
"",0,"局所名","使用","未参照","C-ERR-GC351","","C:¥入出庫管理システム¥INCLUDE%GC351ERR
.CBL",1,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","0,0,2,"03","C-ERR-GC351-ICH-MSG01","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG01 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",2,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","5,0,2,"03","C-ERR-GC351-ICH-MSG02","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG02 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",3,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","10,0,2,"03","C-ERR-GC351-ICH-MSG04","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG04 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",4,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","15,0,2,"03","C-ERR-GC351-ICH-MSG09","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG09 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",5,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","20,0,2,"03","C-ERR-GC351-ICH-MSG21","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG21 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",6,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","25,0,2,"03","C-ERR-GC351-ICH-MSG31","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG31 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",7,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","30,0,2,"03","C-ERR-GC351-ICH-MSG32","","英
数字",5,"X(5)","",0,"局所名","-","未参照","C-ERR-GC351-ICH-MSG32 OF C-ERR-GC351","","C:
¥入出庫管理システム¥INCLUDE%GC351ERR.CBL",8,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","35,0,2,"03","C-ERR-N0-GC351-04","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-04 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",9,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","40,0,2,"03","C-ERR-N0-GC351-06","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-06 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",10,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","45,0,2,"03","C-ERR-N0-GC351-09","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-09 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",11,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","50,0,2,"03","C-ERR-N0-GC351-11","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-11 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",12,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","55,0,2,"03","C-ERR-N0-GC351-12","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-12 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",13,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","60,0,2,"03","C-ERR-N0-GC351-24","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-24 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",14,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","65,0,2,"03","C-ERR-N0-GC351-66","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-66 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",15,""
"入出庫管理システム","GC351 INPUT_DT","連絡節","","70,0,2,"03","C-ERR-N0-GC351-99","","英数字",
5,"X(5)","",0,"局所名","-","未参照","C-ERR-N0-GC351-99 OF C-ERR-GC351","","C:¥入出庫管
理システム¥INCLUDE%GC351ERR.CBL",16,""

```

注意事項

- データ項目一覧は、ソースファイル内での定義順に出力されます。

- 解析でエラーが出力された COBOL ソースファイルの場合、データ項目一覧にはデータ項目情報は出力されないか、または古い COBOL ソースファイルのファイル情報が出力されます。
- プログラム名が重複している場合は、リネームプログラム名が [プログラム名] として出力されます。
- プログラム名が英小文字を含む場合、出力ファイル名は、データ項目一覧_プログラム名.英小文字識別子コード.csv となります。英小文字識別コードは 16 進数で、英大文字と英小文字だけが異なるファイル名が同一とみなされて上書きされることを回避するために付加されます。

8.2.6 不整合引数一覧

プロジェクト内のすべての CALL 文に対する引数整合性チェックで検出された不整合引数情報の一覧です。引数整合性チェックは、[解析生成オプション設定] 画面の [生成ドキュメント] タブで [不整合引数一覧] をチェックしている場合、次に示す順序で実行されます。

1. 個数チェック
2. 引数ごとの個別チェック
 1. 受渡方法のチェック
 2. データ型の適合チェック
 3. サイズのチェック
 4. 集団項目の構造チェック

引数整合性チェックの結果として、引数不整合のある実引数と仮引数の組み合わせの情報が不整合引数一覧 CSV ファイルに、引数不整合の警告メッセージが不整合引数一覧タグファイルに出力されます。不整合引数一覧タグファイルに出力される警告メッセージについては、[「10.11.2 引数整合性チェックの検査内容」](#)を参照してください。

不整合引数 CSV ファイルの出力ファイル名を次に示します。

不整合引数一覧.csv

不整合引数 CSV ファイルのフォーマットを次に示します。

```
"#", "ソースファイルパス名", "ソース行番号", "呼出元プログラム名", "実引数個数", "引数番号", "実引数: 受渡方法", "実引数: 名称", "実引数: データ型", "実引数: サイズ", "実引数: 編集文字列", "実引数: 符号種別", "実引数: 他属性", "実引数: 定義ファイルパス名", "実引数: 定義行番号", "呼出先プログラム名", "仮引数個数", "仮引数: 受渡方法", "仮引数: 名称", "仮引数: データ型", "仮引数: サイズ", "仮引数: 編集文字列", "仮引数: 符号種別", "仮引数: 他属性", "仮引数: 定義ファイルパス名", "仮引数: 定義行番号", "仮引数: 参照有無", "不整合種別"
```

不整合引数 CSV ファイルのフォーマットの各項目に出力される内容を次に示します。

[#]

不整合引数情報の項番です。1 から出力されます。

[ソースファイルパス名]

COBOL ソースファイルが格納されているフォルダの絶対パス名です。

[ソース行番号]

CALL 文のソース行番号（COPY 展開前のソース原文内の行番号）です。

[呼出元プログラム名]

呼び出し元のプログラム名です。

[実引数個数]

CALL 文の USING 指定の実引数の個数です。個数チェックで個数不一致が検出された場合に出力されます。

個数チェック以外で不整合が検出された場合は、「-」が出力されます。また、個数不一致の不整合が検出された場合は、後続する実引数の情報には、すべて「-」が出力されます。

[引数番号]

実引数の通番です。

[実引数:受渡方法]

実引数の受け渡し方法です。

- ByRef
BY REFERENCE 指定での受け渡しを示します。
- ByCon
BY CONTENT 指定での受け渡しを示します。
- ByVal
BY VALUE 指定での受け渡しを示します。
- ByAtr
BY ATTRIBUTE 指定での受け渡しを示します。

[実引数:名称]

実引数として指定したデータ名、定数、ADDR 関数、ADDRESS OF 指定、LENGTH OF 指定、NULL 表意定数です。

【解析生成オプション設定】画面の【その他の解析オプション】タブで【レベル番号が 02～49,66 のデータ項目の情報を抽出する】をチェックしている場合は、次に示す形式で修飾子付きデータ名が出力されます。チェックしていない場合は、データ名だけが出力されます。

データ名 [OF 集団項目名] … [IN ファイル名]

[実引数:データ型]

実引数のデータ型です。実引数が部分参照の場合は、「部分参照(部分参照の結果の型)」が出力されます。

データ型	説明
集団項目	実引数が固定長の英数字集団項目であることを示します。
集団項目(可変長)	実引数が可変長の英数字集団項目であることを示します。
日本語集団項目	実引数が日本語集団項目 (GROUP-USAGE IS NATIONAL) であることを示します。
英字	実引数が英字項目であることを示します。
英数字	実引数が英数字項目または英数字定数であることを示します。 -JPN,Alnum または-JPN,V3JPN コンパイラオプションを指定すると、CALL 文の引数に日本語定数が指定できますが、英数字定数として扱われます。
動的長英数字	実引数が動的長英数字項目であることを示します。
英数字編集	実引数が英数字編集項目であることを示します。
外部 10 進	実引数が外部 10 進項目であることを示します。
内部 10 進	実引数が内部 10 進項目 (COMP-3 または PACKED-DECIMAL) であることを示します。
2 進	実引数が 2 進項目 (COMP, COMP-4 もしくは BINARY), 数字定数, または LENGTH OF であることを示します。
2 進(COMP-X)	実引数が COMP-X 項目であることを示します。
2 進(COMP-5)	実引数が COMP-5 項目であることを示します。
数字編集	実引数が数字編集項目であることを示します。
外部浮動	実引数が外部浮動小数点項目であることを示します。
内部浮動	実引数が内部浮動小数点項目 (COMP-1 もしくは COMP-2), または浮動小数点定数であることを示します。
日本語	実引数が日本語項目であることを示します。
動的長日本語	実引数が動的長日本語項目であることを示します。
日本語編集	実引数が日本語編集項目であることを示します。
外部ブール	実引数が外部ブール項目であることを示します。
内部ブール	実引数が内部ブール項目であることを示します。
アドレス	実引数がアドレス項目もしくはポインタ項目, ADDR 関数, ADDRESS OF, または NULL 表意定数であることを示します。
指標	実引数が指標項目であることを示します。
オブジェクト参照	実引数がオブジェクト参照項目であることを示します。
OLE 参照	実引数が OLE オブジェクト参照項目であることを示します。
OLE バリエーション	実引数がバリエーション項目であることを示します。
プロパティ	実引数がオブジェクトプロパティであることを示します。

[実引数:サイズ]

実引数のバイト長 (内部ブール項目の場合はビット長) です。

実引数が部分参照で長さの指定が定数の場合は、その定数値から換算されるサイズが出力されます。長さの指定が定数ではない場合は、部分参照されているデータ項目の全体サイズが出力されます。

[実引数:編集文字列]

実引数の PICTURE 句の文字列です。

[実引数:符号種別]

SIGN 句の符号種別の情報です。

- ""
SIGN 句の指定がないことを示します。
- LEADING
SIGN IS LEADING を示します。
- TRAILING
SIGN IS TRAILING を示します。
- LEADING SEPARATE
SIGN IS LEADING SEPARATE を示します。
- TRAILING SEPARATE
SIGN IS TRAILING SEPARATE を示します。

[実引数:他属性]

そのほか属性情報です。文字 (!) で連結して出力されます。

- OCCURS
OCCURS 句の指定があることを示します。
- SYNC
SYNCHRONIZED 句の指定があることを示します。
- JUST
JUSTIFIED RIGHT 句の指定があることを示します。
- BLANK
BLANK WHEN ZERO 句の指定があることを示します。

[実引数:定義ファイルパス名]

実引数のデータ項目を定義している COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。実引数がデータ項目でない場合は、「-」が出力されます。

[実引数:定義行番号]

実引数のデータ項目を定義している COBOL ソースファイルまたは登録集原文ファイル内の相対行番号です。実引数がデータ項目でない場合は、「-」が出力されます。

[呼出先プログラム名]

呼び出し先のプログラム名です。

[仮引数:個数]

手続き部見出しにある USING 指定の仮引数の個数です。個数チェックで個数不一致が検出された場合に出力されます。

個数チェック以外で不整合が検出された場合は、「-」が出力されます。また、個数不一致の不整合が検出された場合は、後続する仮引数の情報には、すべて「-」が出力されます。

[仮引数:受渡方法]

仮引数の受け渡し方法です。

- ByRef
BY REFERENCE 指定での受け渡しを示します。
- ByVal
BY VALUE 指定での受け渡しを示します。

[仮引数:名称]

仮引数として指定したデータ名です。

[仮引数:データ型]

仮引数のデータ項目のデータ型です。

出力される文字の意味は、[実引数:データ型]と同じです。

[仮引数:サイズ]

仮引数のバイト長（内部ブール項目の場合はビット長）です。

[仮引数:編集文字列]

仮引数の PICTURE 句の文字列です。

[仮引数:符号種別]

SIGN 句の符号種別の情報です。

出力される文字の意味は、[実引数:符号種別]と同じです。

[仮引数:他属性]

そのほかの属性情報です。文字 (!) で連結して出力されます。

出力される文字の意味は、[実引数:他属性]と同じです。

[仮引数:定義ファイルパス名]

仮引数のデータ項目を定義している COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[仮引数:定義行番号]

仮引数のデータ項目を定義している COBOL ソースファイルまたは登録集原文ファイル内の相対行番号です。

[仮引数:参照有無]

仮引数の参照有無です。

- 参照

仮引数のデータ名がプログラム内で参照されていることを示します。

- 未参照

仮引数のデータ名がプログラム内で参照されていないことを示します。

参照、未参照が出力される条件は、「データ項目一覧」の「参照有無」と同じです。

[不整合種別]

検出した不整合の種別です。

- 個数不一致

実引数と仮引数の個数が不一致であることを示します。

- 受渡方法不一致

実引数と仮引数の受け渡し方法が不一致であることを示します。

- データ型不適合

実引数と仮引数のデータ型が不適合であることを示します。

- サイズ不一致

実引数と仮引数のサイズが不一致であることを示します。

- 構造不一致

実引数と仮引数の集団項目の構造が不一致であることを示します。

- 対応する仮引数なし

対応する仮引数がない実引数の情報であることを示します。

- 対応する実引数なし

対応する実引数がない仮引数の情報であることを示します。

CSV ファイルの出力例

```
"#", "ソースファイルパス名", "ソース行番号", "呼出元プログラム名", "実引数個数", "引数番号", "実引数:受渡方法", "実引数:名称", "実引数:データ型", "実引数:サイズ", "実引数:編集文字列", "実引数:符号種別", "実引数:他属性", "実引数:定義ファイルパス名", "実引数:定義行番号", "呼出先プログラム名", "仮引数個数", "仮引数:受渡方法", "仮引数:名称", "仮引数:データ型", "仮引数:サイズ", "仮引数:編集文字列", "仮引数:符号種別", "仮引数:他属性", "仮引数:定義ファイルパス名", "仮引数:定義行番号", "仮引数:参照有無", "不整合種別"
1, "C:¥入出庫管理システム¥チェック処理.cbl", 129, "GC351_INPUT", "-", 1, "ByRef", "GC351-OPT-COND 0 F GC351", "英数字", 10, "X(10)", "", "", "C:¥入出庫管理システム¥INCLUDE¥GC351.CBL", 9, "GC351_INPUT_DT", "-", "ByRef", "GC351", "集団項目", 247, "", "", "", "C:¥入出庫管理システム¥INCLUDE¥GC351.CBL", 1, "参照", "サイズ不一致"
2, "C:¥入出庫管理システム¥チェック処理.cbl", 129, "GC351_INPUT", "-", 2, "ByRef", "C-ERR-N0-GC351-9 9 OF C-ERR-GC351", "英数字", 5, "X(5)", "", "", "C:¥入出庫管理システム¥INCLUDE¥GC351ERR.CBL", 16, "GC351_INPUT_DT", "-", "ByRef", "C-ERR-GC351", "集団項目", 75, "", "", "", "C:¥入出庫管理システム¥INCLUDE¥GC351ERR.CBL", 1, "未参照", "サイズ不一致"
3, "C:¥入出庫管理システム¥メニュー画面.cbl", 48, "AC000_MENU", 0, "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "RC340_INPUT", 2, "-", "-", "-", "-", "-", "-", "-", "-", "-", "個数不一致"
4, "C:¥入出庫管理システム¥制御処理.cbl", 162, "RC340_UPD", 0, "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "ZC003_OPT", 1, "-", "-", "-", "-", "-", "-", "-", "-", "-", "個数不一致"
```

注意事項

- 引数ごとの個別チェックで不整合を検出した場合、同じ引数に対して後続の個別チェックは実行されません。
- 対応する仮引数のない実引数の情報を出力する場合、仮引数の情報の列（[引数番号] を除く）と判定の列にはすべて「-」が出力されます。
- 対応する実引数のない仮引数の情報を出力する場合、実引数の情報の列（[引数番号] を除く）と判定の列にはすべて「-」が出力されます。
- 不整合引数一覧は、[ソース行番号]、[呼出元プログラム名] および [引数番号] で昇順にソートされて出力されます。
- プログラム名が重複している場合は、リネームプログラム名が [呼出元プログラム名]、[呼出先プログラム名] として出力されます。
- [実引数:名称] に出力するデータ名には、添字や部分参照子の情報は付加されません。
- 実引数の英数字定数や日本語定数（-JPN コンパイラオプション指定時）が非印字可能文字（制御文字や非シフト JIS 文字）を含む場合、[実引数:名称] には、それぞれ 16 進英数字定数、16 進日本語定数に変換したものが出力されます。
- 実引数の英数字定数や日本語定数（-JPN コンパイラオプション指定時）の長さが 160 バイトを超えている場合、[実引数:名称] には、160 バイトで切り捨てられ、末尾に「...」が付加されたものが出力されます。
- 実引数の 16 進英数字定数や 16 進日本語定数（-JPN コンパイラオプション指定時）の長さが 80 バイト（16 進桁数が 160 桁）を超えている場合、[実引数:名称] には、80 バイト（16 進桁数が 160 桁）で切り捨てられ、末尾に「...」が付加されたものが出力されます。
- 英数字定数の場合、[実引数:名称] には、英数字定数値をアポストロフィ（'）で囲んだものが出力されます。英数字定数値がアポストロフィ（'）を含んでいても、連続するアポストロフィ（''）には変換されません。

関連項目

- 「8.2.5 データ項目一覧」
- 「10.11 引数整合性チェック」

8.2.7 実行順序リスト

プログラムごとの、ソースコード（節見出し、段落見出し、文および条件指定）の実行順を示すリストです。
出力ファイル名を次に示します。

実行順序リスト_プログラム名.csv

フォーマットを次に示します。

```
#####  
#  
# 実行順序リスト  
#   ソースファイルパス名 :  
#   プログラム名 :  
#  
#####  
ソース行番号, [,] ...”ソースコードのテキスト”
```

フォーマットの各項目に出力される内容を次に示します。

[ソースファイルパス名]

COBOL ソースファイルの絶対パス名です。

[プログラム名]

プログラム名です。

[ソース行番号]

ソースコード（節見出し、段落見出し、文および条件指定）のソース行番号（COPY 展開後のソース原文内の行番号）です。ソース行番号は 6 桁で出力されます。6 桁に満たない場合、ソース行番号の前は半角空白で埋められます。

[ソースコードのテキスト]

ソースコード（節見出し、段落見出し、文および条件指定）のテキストです。ソースコードのテキストは、ソースコードの階層レベルを示すコンマ（,）でインデントを付けて出力されます。

CSV ファイルの出力例

COBOL ソースファイル「チェック処理.cbl」中のプログラム「GC351_INPUT」の実行順序リスト（実行順序リスト_GC351_INPUT.csv）の出力例を示します。

```
#####  
#  
# 実行順序リスト  
#   ソースファイルパス名 : C:¥入出庫管理システム¥チェック処理.cbl  
#   プログラム名 : GC351_INPUT  
#  
#####  
46,"GC351-INPUT-CHK-PROC SECTION."  
48,,,"GC351-INPUT-CHK-PROC-010."  
50,,,,"IF GC351-TRIHS0-KB NOT = SPACE"  
51,,,,,"THEN"  
52,,,,,"CONTINUE"  
53,,,,,"ELSE"  
54,,,,,"MOVE C-ERR-NO-GC351-04 TO WK-ERR-NO"  
55,,,,,"MOVE C-ERR-GC351-ICH-MSG09 TO WK-ERR-MSG"  
57,,,,,"PERFORM ABEND-SUB-PROC"  
    ,,,,,,"=== 節(ABEND-SUB-PROC)呼び出し入口 ==="  
135,,,,,"ABEND-SUB-PROC SECTION."  
137,,,,,"CONTINUE"
```

```

,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
59,,, "IF GC351-YSNKAN-NO NOT = SPACE"
60,,, "THEN"
61,,, "IF GC351-YSNKAN-NO (1:1) IS NUMERIC"
62,,, "THEN"
63,,, "CONTINUE"
64,,, "ELSE"
65,,, "MOVE C-ERR-NO-GC351-09 TO WK-ERR-NO"
67,,, "MOVE C-ERR-GC351-ICH-MSG32 TO WK-ERR-MSG"
69,,, "PERFORM ABEND-SUB-PROC"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,, "ABEND-SUB-PROC SECTION."
137,,,,, "CONTINUE"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
71,,, "ELSE"
72,,, "IF INPUTBUCD = '1' OR INPUTBUCD = '2'"
74,,, "THEN"
75,,, "CONTINUE"
76,,, "ELSE"
77,,, "MOVE C-ERR-NO-GC351-06 TO WK-ERR-NO"
79,,, "MOVE C-ERR-GC351-ICH-MSG31 TO WK-ERR-MSG"
81,,, "PERFORM ABEND-SUB-PROC"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,, "ABEND-SUB-PROC SECTION."
137,,,,, "CONTINUE"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
84,,, "IF GC351-NM-KJ NOT = SPACE"
85,,, "THEN"
86,,, "CONTINUE"
87,,, "ELSE"
88,,, "MOVE C-ERR-NO-GC351-11 TO WK-ERR-NO"
89,,, "MOVE C-ERR-GC351-ICH-MSG01 TO WK-ERR-MSG"
91,,, "PERFORM ABEND-SUB-PROC"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,, "ABEND-SUB-PROC SECTION."
137,,,,, "CONTINUE"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
93,,, "IF GC351-NM-KN NOT = SPACE"
94,,, "THEN"
95,,, "CONTINUE"
96,,, "ELSE"
97,,, "MOVE C-ERR-NO-GC351-12 TO WK-ERR-NO"
98,,, "MOVE C-ERR-GC351-ICH-MSG02 TO WK-ERR-MSG"
100,,, "PERFORM ABEND-SUB-PROC"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,, "ABEND-SUB-PROC SECTION."
137,,,,, "CONTINUE"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
102,,, "IF KYJ-CNT NOT = ZERO"
103,,, "THEN"
104,,, "CONTINUE"
105,,, "ELSE"
106,,, "MOVE C-ERR-NO-GC351-24 TO WK-ERR-NO"
107,,, "MOVE C-ERR-GC351-ICH-MSG04 TO WK-ERR-MSG"
109,,, "PERFORM ABEND-SUB-PROC"
,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,, "ABEND-SUB-PROC SECTION."
137,,,,, "CONTINUE"

```

```

,,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
111,,, "IF GC351-KYT-CD = '2' OR GC351-KYT-CD = SPACE"
113,,, "THEN"
114,,, "CONTINUE"
115,,, "ELSE"
116,,, "PERFORM GC351-KYT-CHECK-PROC"
,,,,,, "=== 節(GC351-KYT-CHECK-PROC)呼び出し入口 ==="
138,,,,,, "GC351-KYT-CHECK-PROC SECTION."
140,,,,,, "PERFORM ABEND-SUB-PROC"
,,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,,, "ABEND-SUB-PROC SECTION."
137,,,,,, "CONTINUE"
,,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
141,,,,,, "CONTINUE"
,,,,,, "=== 節(GC351-KYT-CHECK-PROC)呼び出し出口 ==="
117,,,,,, "IF WK-KYT-CHECK-RTN-CD NOT = ZERO"
118,,,,,, "THEN"
119,,,,,, "CONTINUE"
120,,,,,, "ELSE"
121,,,,,, "MOVE C-ERR-NO-GC351-66 TO WK-ERR-NO"
123,,,,,, "MOVE C-ERR-GC351-ICH-MSG21 TO WK-ERR-MSG"
125,,,,,, "PERFORM ABEND-SUB-PROC"
,,,,,, "=== 節(ABEND-SUB-PROC)呼び出し入口 ==="
135,,,,,, "ABEND-SUB-PROC SECTION."
137,,,,,, "CONTINUE"
,,,,,, "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
129,,, "CALL 'GC351_INPUT_DT' USING KYJ-CNT OF GC351 C-ERR-NO-GC351-99"
132,,, "CONTINUE"
133,,, "GC351-INPUT-CHK-PROC-999."
134,,, "EXIT PROGRAM ---> (プログラム終了)"

```

注意事項

- [解析生成オプション設定] 画面の [その他の解析オプション] タブで、[プログラムの制御フロー情報（実行順序表示機能で使用する情報）を抽出する] が無効（チェックボックスをオフ）になっている場合、ヘッダだけが出力されます。
- 解析でエラーが出力された COBOL ソースファイルの場合、実行順序リストは出力されないか、または古い COBOL ソースファイルの実行順序リストが出力されます。
- プログラム名が重複している場合は、リネームプログラム名が [プログラム名] として出力されます。
- プログラム名が英小文字を含む場合、出力ファイル名は、実行順序リスト_プログラム名.英小文字識別子コード.csv となります。英小文字識別コードは 16 進数で、英大文字と英小文字だけが異なるファイル名が同一とみなされて上書きされることを回避するために付加されます。

関連項目

- 「5.2.7 [その他の解析オプション] タブ」
- 「10.16 制御フロー解析による実行順序表示機能」

8.3 関係情報 CSV

呼び出し元と呼び出し先のプログラム関係、COBOL ソースファイルと登録集原文内で参照する登録集原文の関係、プログラム内で使用するファイルの関係一覧、およびプログラム内で使用するデータベースの表の参照関係を出力します。

8.3.1 プログラム呼出関係

プログラムごとの、呼び出し元と呼び出し先のプログラムの関係です。

出力ファイル名を次に示します。

プログラム呼出関係.csv

フォーマットを次に示します。

1 行目

"プロジェクト名","呼出元プログラム名","呼出先プログラム名1","呼出先プログラム名2"...呼出先プログラム名n"

2 行目以降

"プロジェクト名","呼出元プログラム名","呼出種別","呼出種別"...呼出種別"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[呼出元プログラム名]

呼び出し元のプログラム名です。

[呼出先プログラム名 1] [呼出先プログラム名 2] ... [呼出先プログラム名 n]

参照先の呼び出し先プログラム名です。

[呼出種別]

プログラムの呼び出し種別です。表示される記号の意味を次に示します。

- ：直接呼び出し。
- ：間接呼び出し。

CSV ファイルの出力例

- テキストの場合

"プロジェクト名","呼出元プログラム名","GC351_INPUT","GC351_INPUT_DT","RC340_INPUT","RC340_SET","RC340_SND","RC340_UPD","ZC001_OPT","ZC003_OPT"
"入在庫管理システム","AC000_MENU","○","○","◎","○","◎","◎","○","○"


```
"入出庫管理システム", "GC351_INPUT", "", "", "", "", "", "", "", "", ""
"入出庫管理システム", "RC340_INPUT", "", "", "", "", "", "", "", "", ""
"入出庫管理システム", "RC340_SET", "", "", "", "", "", "", "", "", ""
"入出庫管理システム", "RC340_SND", "", "", "", "", "", "", "", "", ""
"入出庫管理システム", "RC340_UPD", "", "", "", "", "", "", "", "", ""
```

- Excel の場合

[illegible]

注意事項

- 直接呼び出し、および間接呼び出しの両方の呼び出しをするプログラムの場合は、[呼出種別]は「◎」の直接呼び出しとして出力されます。
- プログラムの呼び出しをしないプログラムは、[呼出元プログラム名]には含まれません。
- 未使用のプログラムでもプログラムの呼び出しをする場合は、[呼出元プログラム名]に含まれます。
- CALL 一意名解決で呼び出し先プログラムを解決できなかったプログラムの場合でも、[呼出元プログラム名]に含まれます。この場合、[呼出種別]には何も出力されません。
- 外部プログラム名が重複している場合は、リネームプログラム名が[呼出元プログラム名]に出力されます。なお、内部プログラムが重複しているときは、同じ呼び出し元プログラム名が出力されます。
- プログラムから呼び出されないプログラムは、[呼出先プログラム名 1～n]には含まれません。
- 解析生成オプションで指定された COBOL2002 サービスルーチンファイルに指定されているサービスルーチンは、プログラムから呼び出されていれば[呼出先プログラム名 1～n]に含まれます。
- 未登録のプログラムは、[呼出先プログラム名 1～n]に含まれます。
- プログラム呼び出し関係は、[呼出元プログラム名]で昇順にソートされて出力されます。また、1行目の[呼出先プログラム名 1～n]も、昇順にソートされて出力されます。
- CSV ファイルを出力しているときにプログラム呼び出し階層数が制限を超えた場合、生成エラー (KEDL1271A-E) が出力され、CSV ファイルの出力は中止されます。プログラム呼び出しの階層の最大値は、COBOL ソース解析の環境設定定義ファイル (CBLANALYZER.INI) のプログラム呼び出し最大階層を指定する CALLDEPTHMAX パラメータで定義します。

8.3.2 登録集原文参照関係

COBOL ソースファイルと登録集原文ファイル内で参照する登録集原文ファイルの関係の一覧です。

出力ファイル名を次に示します。

登録集原文参照関係.csv

フォーマットを次に示します。

1 行目

"プロジェクト名","ファイル種別","参照元","登録集原文ファイル名1","登録集原文ファイル名2"
"..."登録集原文ファイル名n"

2 行目以降

"プロジェクト名","ファイル種別","参照元","参照種別","参照種別..."参照種別"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[ファイル種別]

ファイルの種別です。出力される文字の意味を次に示します。

- プログラム
COBOL ソースファイルです。
- 登録集
登録集原文ファイルです。

[参照元]

参照元の COBOL ソースファイル名または登録集原文名です。

[登録集原文ファイル名 1] [登録集原文ファイル名 2] … [登録集原文ファイル名 n]

参照先の登録集原文ファイル名です。

[参照種別]

登録集原文の参照種別です。表示される記号の意味を次に示します。

- ◎：直接参照。
- ○：間接参照。

CSV ファイルの出力例

- テキストの場合

"プロジェクト名","ファイル種別","参照元","ACCTCD.CBL","FC351DT.CBL","FC370.CBL","FC380.CBL",
"FGC301.CBL","GC351.CBL","GC351DT.CBL","GC351ERR.CBL","PRCCD.CBL","RC340.CBL","SQLCA.C
OB","WK.CBL"

```

"入出庫管理システム","プログラム","オプション処理 1.cbl","","","","","","","","","","","","","","",""
"入出庫管理システム","プログラム","オプション処理 2.cbl","","","","","","","","","","","","","",""
"入出庫管理システム","プログラム","オプション処理 3.cbl","","","","","","","","","","","","","",""
"入出庫管理システム","プログラム","チェック処理.cbl","","","","","","","","","","","","",""
"入出庫管理システム","プログラム","出荷処理.cbl","","","","","","","","","","","",""
"入出庫管理システム","プログラム","詳細チェック処理.cbl","","","","","","","","","","",""
"入出庫管理システム","プログラム","制御処理.cbl","","","","","","","","","",""
"入出庫管理システム","プログラム","転送処理.cbl","","","","","","","","",""
"入出庫管理システム","プログラム","入力処理.cbl","","","","","","","",""

```

• Excel の場合

登録集原文参照関係.csv - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	プロジェクト名	ファイル種別	参照元	ACCTCD.C	FC351 DT C	FC370.CBL	FC380.CBL	FGC301	CB	GC351	CBL	GC351 DT C	GC351 ERR (PRCCD.CBL	RC340
2	入出庫管理システム	プログラム	オプション処理1.cbl											
3	入出庫管理システム	プログラム	オプション処理2.cbl											
4	入出庫管理システム	プログラム	オプション処理3.cbl											
5	入出庫管理システム	プログラム	チェック処理.cbl						◎			◎		◎
6	入出庫管理システム	プログラム	出荷処理.cbl				◎							◎
7	入出庫管理システム	プログラム	詳細チェック処理.cbl		◎	◎			◎	◎	◎			◎
8	入出庫管理システム	プログラム	制御処理.cbl			◎							◎	◎
9	入出庫管理システム	プログラム	転送処理.cbl	◎				◎	◎		◎			◎
10	入出庫管理システム	プログラム	入力処理.cbl			◎			◎		◎			◎
11														
12														
13														

注意事項

- COBOL ソースファイルから参照されていない登録集原文ファイルは、1 行目の [登録集原文ファイル名 1~n] には含まれません。
- COPY 文が 1 個も存在しない COBOL ソースファイルや登録集原文ファイルは、[参照元] には含まれません。
- 未使用でも COPY 文展開する COBOL ソースファイルは、[参照元] に含まれます。
- 未使用の登録集原文ファイルは、[参照元] に含まれません。
- 解析対象プロジェクト内で COBOL ソースファイル名が重複している場合は、[参照元] として表示する COBOL ソースファイル名の後ろに「(パス名)」が付加されます。

(例)

C:¥TEST1¥チェック処理.CBL と C:¥TEST2¥チェック処理.CBL がある場合は、それぞれ "チェック処理.CBL(C:¥TEST1¥)", "チェック処理.CBL(C:¥TEST2¥)" と出力されます。

- 解析対象プロジェクト内で登録集原文ファイル名が重複している場合は、[登録集原文ファイル名 1～n] および [参照元] として表示する登録集原文ファイル名の後ろに「(パス名)」が付加されます。

(例)

C:¥COPY1¥FC370.CBL と C:¥COPY2¥FC370.CBL がある場合は、それぞれ "FC370.CBL(C:¥COPY1¥)", "FC370.CBL(C:¥COPY2¥)" と出力されます。

- 登録集原文参照関係は、[ファイル種別] と [参照元] でソートされて出力されます。[ファイル種別] は、「プログラム」「登録集」の順に、[参照元] は、COBOL ソースファイル名または登録集原文ファイル名で昇順にソートされます。

また、1 行目の [登録集原文ファイル名 1～n] も、昇順にソートされて出力されます。

8.3.3 ファイル参照関係

プログラムごとの、プログラム内で使用するファイルの関係の一覧です。

出力ファイル名を次に示します。

ファイル参照関係.csv

フォーマットを次に示します。

1 行目

"プロジェクト名","プログラム名","ファイル名1","ファイル名2"..."ファイル名n"

2 行目以降

"プロジェクト名","プログラム名","CRUD","CRUD"..."CRUD"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[プログラム名]

プログラム名です。

[ファイル名 1] [ファイル名 2] ... [ファイル名 n]

参照先のファイル名です。

[CRUD]

ファイル名の使用モードです。次の 4 つのモードを組み合わせで出力します。

- C : WRITE
- R : READ

- U : REWRITE
- D : DELETE

CSV ファイルの出力例

- テキストの場合

```
"プロジェクト名","プログラム名","FC351DT","FC370","FC380"
"入在庫管理システム","GC351_INPUT_DT","R","",""
"入在庫管理システム","RC340_INPUT","","","R",""
"入在庫管理システム","RC340_SND","","","R"
"入在庫管理システム","RC340_UPD","","","R",""
```

- Excel の場合

	A	B	C	D	E	F	G	H	I
1	プロジェクト名	プログラム名	FC351DT	FC370	FC380				
2	入在庫管理システム	GC351_INPUT_DT	R						
3	入在庫管理システム	RC340_INPUT		R					
4	入在庫管理システム	RC340_SND			R				
5	入在庫管理システム	RC340_UPD		R					

注意事項

- ファイルを1つも使っていないプログラムは、[プログラム名]には含まれません。
- 未使用のプログラムでもファイルを使用するプログラムは、[プログラム名]に含まれます。
- 未使用のプログラムからだけ使われているファイルであっても、1行目の[ファイル名1～n]に含まれます。
- 外部プログラム名が重複している場合は、リネームプログラム名が[プログラム名]として出力されます。なお、内部プログラムが重複している場合は、同じ[プログラム名]が出力されます。
- ファイル参照関係は、[プログラム名]で昇順にソートされて出力されます。
また、1行目の[ファイル名1～n]も、昇順にソートされて出力されます。

8.3.4 表(DB)参照関係

プログラムごとの、プログラム内で使用するデータベースの表の参照関係です。

出力ファイル名を次に示します。

表DB参照関係.csv

フォーマットを次に示します。

1 行目

"プロジェクト名","プログラム名","表名1","表名2"...表名n"

2 行目以降

"プロジェクト名","プログラム名","CRUD","CRUD"...CRUD"

フォーマットの各項目に出力される内容を次に示します。

[プロジェクト名]

解析対象のプロジェクト名です。

[プログラム名]

プログラム名です。

[表名 1] [表名 2] ... [表名 n]

参照先の表名です。

[CRUD]

表の使用モードです。次の 4 つのモードを組み合わせて出力します。

- C : INSERT
- R : SELECT または FETCH
- U : UPDATE
- D : DELETE

表の使用モードは C, R, U, D の順序で出力します。

例えば, UPDATE 文と SELECT 文が定義されている場合, 「RU」と出力します。

CSV ファイルの出力例

- テキストの場合

"プロジェクト名","プログラム名","FGC301","GC351DT","RC340"
"入在庫管理システム","GC351_INPUT_DT","","","R",""
"入在庫管理システム","RC340_INPUT","","","C"
"入在庫管理システム","RC340_SND","","","RUD"
"入在庫管理システム","RC340_UPD","","","RUD"

- Excel の場合



	A	B	C	D	E	F	G	H	I
1	プロジェクト名	プログラム名	FGC301	GC351 DT	RC340				
2	入在庫管理システム	GC351_INPUT_DT		R					
3	入在庫管理システム	RC340_INPUT			C				
4	入在庫管理システム	RC340_SND			RUD				
5	入在庫管理システム	RC340_UPD			RUD				

■ 注意事項

- CRUD 操作の SQL 文を 1 つも使っていないプログラムは、[プログラム名] には含まれません。
- 「== SQL 構文エラー ==」や「== カーソル名未定義 ==」は、1 行目の [表名 1~n] には含まれません。
- 未使用のプログラムでも表を使用するプログラムは、[プログラム名] に含まれます。
- 未使用のプログラムからだけ使われている表であっても、1 行目の [表名 1~n] に含まれます。
- 外部プログラム名が重複している場合は、リネームプログラム名が [プログラム名] に出力されます。なお、内部プログラムが重複している場合は、同じ [プログラム名] が出力されます。
- 表(DB)参照関係は、[プログラム名] で昇順にソートされて出力されます。
また、1 行目の [表名 1~n] も、昇順にソートされて出力されます。

9

エラー時の対処

COBOL ソース解析でエラーが起こったときの対処について説明します。

9.1 事例から探す

COBOL ソース解析のトラブルの事例と対処を説明します。該当する事例がない場合、「[9.3 メッセージから探す](#)」を参照してメッセージの内容と対処に従うか、「[9.4 そのほかの方法から探す](#)」を参照してください。

9.1.1 COBOL ソース解析を実行できない

COBOL ソース解析を実行できない事例への対処です。

次に示す事例について説明します。

- COBOL ソース解析の起動に失敗した
- COBOL ソース解析の操作で予期しないエラーが発生する
- 解析結果がエラーになる
- 解析ログ、操作ログのエラーが出力される

(1) COBOL ソース解析の起動に失敗した

COBOL ソース解析の起動に失敗する理由は次のことが考えられます。

1. 同じ Windows マシンで DB ユティリティを起動している。
 2. Windows Server の場合、ほかのユーザーが COBOL ソース解析を使用している。
1. の場合は、DB ユティリティをいったん終了してから、COBOL ソース解析を起動してください。
2. の場合は、ほかのユーザーが使用し終わるのを待ってから、COBOL ソース解析を起動してください。

(2) COBOL ソース解析の操作で予期しないエラーが発生する

次のことが考えられます。

- [解析生成オプション設定] 画面の [生成ドキュメント] タブで指定できる出力先フォルダのパス名の最大長（167 文字）を超えるパス名を指定すると、ソース解析情報の生成中に予期しないエラーが発生することがあります。
- 「KEDL1213A-E 予期しないエラーが発生しました」のエラーが発生した場合、ODBC データソースの作成で、hirdb.ini のパスに誤りがあることが考えられます。

正しいパス名を指定してください。

ODBC データソースの作成については、マニュアル「COBOL2002 Professional 製品 導入ガイド」を参照してください。

(3) 解析結果がエラーになる

COBOL2002 でコンパイルエラーがない COBOL ソースファイルを解析しているかどうかを確認してください。

COBOL ソースファイルの解析でエラーになった場合は、次のことが考えられます。

- COBOL ソース解析でサポートしていない COBOL2002 の機能を使用してコーディングされている
- 登録集原文を COBOL ソースファイルとして解析している

原因によって、エラーの結果が異なります。詳細は、「[10. COBOL ソース解析の仕様範囲のリファレンス](#)」を参照してください。

なお、機能キャラクタを含む COBOL ソースファイルを解析するため、コンパイラオプションを選択して解析エラーになった場合は、COBOL エディタの [オプション] - [カスタマイズ] メニューから、カスタマイズダイアログボックスの [表示] タブにある [機能キャラクタの設定] で [グラフィック] オプションを選択して、COBOL ソースが 72 カラム以内に収まっているかを確認してください。72 カラム以内に収まっていないときは、72 カラム以内に収まるように COBOL ソースを変更してください。

(4) 解析ログ、操作ログのエラーが出力される

解析ログファイル、操作ログファイルをエディタなどのほかのプログラムで開いていないか確認してください。

解析ログファイル、操作ログファイルを開いているプログラムを終了し、COBOL ソース解析を実行してください。

9.1.2 出力結果が意図したとおりではない

COBOL ソース解析を実行したが、出力した結果が意図したとおりではない事例への対処です。

次に示す事例について説明します。

- プログラムコメントまたは節コメントが出力されない
- プログラム名の記号（#, ¥, @, -）が表示されない
- 「ページが表示されません」と表示されるページがある
- 解析/生成対象にしていない COBOL ソースファイルまで解析/生成される
- 半角空白が正しく出力されない
- レイアウトが崩れる

(1) プログラムコメントまたは節コメントが出力されない

生成されたソース解析情報にプログラムコメントまたは節コメントが出力されない原因は次の 2 種類が考えられます。

1. 解析対象の COBOL ソースファイルにコメント行がない。

COBOL ソース解析では、COBOL ソースファイルのコメント行をソース解析情報に出力しています。コメント行がないと、プログラムコメントおよび節コメントはソース解析情報に出力されません。

2. 解析生成オプションの設定が不正である。

「解析生成オプション設定」画面の「プログラムコメント抽出」タブと「節コメント抽出」タブに正しい値を設定しないと、生成されたソース解析情報にプログラムコメントおよび節コメントが出力されません。

解析オプションの設定が正しいかどうか確認してください。

「解析生成オプション設定」画面については、「[5.2 「解析生成オプション設定」画面](#)」を参照してください。

「解析生成オプション設定」画面で入力する値の制限事項については、「[10. COBOL ソース解析の仕様範囲のリファレンス](#)」を参照してください。

(2) プログラム名の記号（#, ¥, @, -）が表示されない

COBOL ソース解析では、PROGRAM-ID および CALL 文の定数で指定するプログラム名の記号（#, ¥, @, -）を次に示す条件で変換して、ソース解析情報に出力します。

- 先頭の「#」「¥」「@」は、それぞれ「J」「A」「B」に変換。
- 2 文字目以降の「#」「¥」「@」は、それぞれ「0」「1」「2」に変換。
- 「-」は、「_」に変換。

また、モジュール名の先頭が数字の場合、「1」は「A」、「2」は「B」、「3」は「C」、「4」は「D」、「5」は「E」、「6」は「F」、「7」は「G」、「8」は「H」、「9」は「I」に変換されます。

(例)

プログラム名が「#PROGRAM@01」の場合は「JPROGRAM201」に変換されます。

(3) 「ページが表示されません」と表示されるページがある

次の理由で HTML ドキュメントが生成されていない場合、そのドキュメントへのリンクには、「ページが表示されません」が表示されます。ただし、以前に生成した古いドキュメントが存在する場合は、その古いドキュメントが表示されます。

- 「解析生成オプション設定」画面の「生成ドキュメント」タブで HTML ファイルの生成を無効（チェックボックスをオフ）になっている
- 解析エラーまたは生成エラーが発生している

これらの理由に該当するかどうかは、COBOL ソース解析の操作ログで確認してください。COBOL ソース解析の操作ログについては、「[9.2.2 操作ログを参照する](#)」および「[9.3 メッセージから探す](#)」を参照してください。

(4) 解析/生成対象にしていない COBOL ソースファイルまで解析/生成される

[解析生成オプション設定] 画面の [生成ドキュメント] タブで出力先フォルダとして指定したフォルダに、以前にソース解析情報を生成していたことが考えられます。前回の解析結果と今回の解析結果の両方がフォルダ内に格納され、解析または生成対象にしていない COBOL ソースファイルが解析または生成されているように見えることが考えられます。

出力先フォルダに誤りがないかどうかを確認してください。また、前回のソース解析情報を格納した出力先フォルダは削除または退避してください。

出力先フォルダについては、「[5.2.8 \[生成ドキュメント\] タブ](#)」を参照してください。

(5) 半角空白が正しく出力されない

COBOL プログラム上の複数の半角空白は、1 つの半角空白としてソース解析情報に出力されます。複数の半角空白が 1 つの半角空白として出力される値を次に示します。

- プログラムコメント行
- 節コメント行

(6) レイアウトが崩れる

Web ブラウザの文字のサイズを [大] または [最大] にすると、ソース解析情報のタイトルの幅より広く表示され、レイアウトが崩れることがあります。Web ブラウザの文字のサイズを [中] に設定してください。

9.1.3 ログが蓄積されない

解析ログファイル (COBOLAnalyzeLog.tag, COBOLAnalyzeLog2.tag, COBOLAnalyze_プロジェクト名_yyyymmddhhmmss_Log.tag) または操作ログファイル (COBOLControlLog.txt, COBOLControlLog2.txt, COBOLControl_プロジェクト名_yyyymmddhhmmss_Log.txt) に書き込み権限がないと、ログが蓄積されないことがあります。

ファイルのアクセス権を確認してください。

解析ログについては「[9.2.1 解析ログを参照する](#)」、操作ログについては「[9.2.2 操作ログを参照する](#)」を参照してください。

注意事項

ログを表示したあとで、再度 COBOL ソース解析を実行する場合は、いったんログを閉じてください。ログを開いたまま COBOL ソース解析を実行すると、ログが正しく取得できません。また、COBOL ソース解析の実行中は、ファイルを排他制御するエディタでログを開かないでください。

9.1.4 一時ファイルが削除されない

COBOL ソース解析で使用する一時ファイルは通常は削除されますが、解析処理または生成処理を途中で打ち切った場合には、次に示すフォルダに一時ファイルが蓄積されたままになります。

Windows の環境変数 *TEMP* で定義されているフォルダ¥COBOL Analyzer

一時ファイルがディスク容量を圧迫する場合は、上記のフォルダ内の不要なファイルを削除してください。

9.1.5 開発マネージャから解析結果を参照できない

バッチモードで作成したソース解析プロジェクトを開発マネージャから操作する場合、ソース解析プロジェクトの使用開始をしても、解析結果を参照する開発マネージャのメニューは活性化しません。開発マネージャでソース解析プロジェクトの使用開始、解析生成オプションの設定、COBOL ソース解析の実行を行ってください。

9.1.6 解析結果の出力先の変更が開発マネージャに反映されない

開発マネージャで作成したソース解析プロジェクトに対して、バッチモードで解析結果の出力先を変更した場合、開発マネージャには変更前の解析結果が表示されます。ソース解析プロジェクトの使用開始をしたあとに、解析生成オプションの設定、および COBOL ソース解析の実行を行ってください。

9.1.7 ソース解析プロジェクトの解析生成の実行で想定以上に時間が掛かる

大規模システムのソース解析プロジェクトでは解析生成の実行で時間が掛かる場合があります。

ソース解析プロジェクトの解析生成の実行中にスリープ機能（自動休止機能）が実行されると、Windows マシンが自動的に休止され、ソース解析プロジェクトの解析生成の実行で想定以上に時間が掛かることがあります。

このため、ソース解析の解析生成の実行中に Windows マシンがスリープ状態にならないように、スリープ機能をオフにしておくことを推奨します。

スリープ機能の設定は、次の手順で変更します。

設定手順は、インストールした Windows によって異なることがあります。

1. Windows の [コントロールパネル] で、[電源オプション] をクリックします。
2. [電源プランの選択] で、現在選択されているプランの「プラン設定の変更」をクリックします。
3. [次のプランの設定の変更] で、「コンピューターをスリープ状態にする」を「なし」に設定します。
4. [変更の保存] ボタンをクリックします。

9.2 ログを参照する

COBOL ソース解析のログ（解析ログ、操作ログ）の参照方法の説明です。

9.2.1 解析ログを参照する

解析ログは、解析または生成でエラーになった場合に、コンパイルエラーを基に COBOL プログラムの問題を確認するときに参照します。

解析ログは、COBOL ソース解析の解析または生成を実行するタイミングで出力されます。COBOL ソース解析が解析ログファイルの作成に失敗した場合、エラーメッセージを出力して、処理を終了します。

(1) 解析ログの出力先ファイル

解析ログは、次に示すファイルに格納されます。

- 開発マネージャの場合または cblcsa コマンドに -LogFolder オプションを指定しない場合

COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) で定義されているフォルダ¥COBOLAnalyze
Log.tag (現行ログ)
COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) で定義されているフォルダ¥COBOLAnalyze
Log2.tag (退避ログ)

COBOL ソース解析がメッセージを出力するたびに解析ログがファイルに追加書きされます。現行ログの最大容量を超えた場合、退避ログにコピーしたあとで、現行ログが初期化されてログが出力されます。退避ログにすでにログがある場合は上書きされ、古いログは消去されます。このため、解析ログを保持する場合は、適時、退避ログにある解析ログファイルを任意の場所にバックアップしてください。

- cblcsa コマンドに -LogFolder オプションを指定した場合

cblcsa コマンドの -LogFolder オプションに指定されたフォルダ¥COBOLAnalyze_プロジェクト名_yyyy
mmddhhmmss_Log.tag

-LogFolder オプションを省略したときは、COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) に定義されている解析ログの出力先フォルダに出力されます。

解析ログはタイムスタンプ付きのファイル名で出力されるため、不要な解析ログは定期的に削除することを推奨します。

同じプロジェクトを連続で解析するなど、cblcsa コマンドを起動した開始時刻が秒まで同じとなる場合、同じ解析ログファイルに複数の解析ログが出力されます。1 秒以上間隔をあけて実行してください。

出力先フォルダや解析ログファイルの最大サイズは、COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) で変更できます。

COBOL ソース解析が解析ログファイルの作成に失敗した場合、次のように動作します。

- 開発マネージャの場合

エラーメッセージを表示して、処理を続行します。

- cblcsa コマンドの場合

エラーメッセージを標準エラー出力に出力して、処理を終了します。

(2) 解析ログの表示方法

解析ログファイルを COBOL エディタなどのテキストを表示するプログラムで開きます。

注意事項

解析ログを表示したあとで、再度 COBOL ソース解析を実行する場合は、いったん解析ログファイルを閉じてください。解析ログファイルを開いたまま COBOL ソース解析を実行すると、解析ログを正しく取得できないことがあります。また、COBOL ソース解析の実行中は、ファイルを排他制御するエディタで解析ログを開かないでください。

(3) 解析ログの出力形式

形式

[yyyy/mm/dd hh:mm:ss] , [情報] , [AAAA] BBBB...

形式の説明

- [yyyy/mm/dd hh:mm:ss] : ログを採取した時刻です (yyyy : 西暦年, mm : 月, dd : 日, hh : 時, mm : 分, ss : 秒)。
- [情報] : ログの重要度です。解析の開始時間, 終了時間などの情報を示します。
- [AAAA] : ログの種別です。
「システム」: 解析制御と生成制御の開始/終了ログを示します。
「解析」: 各 COBOL ソースファイルの解析処理の開始ログを示します。
- BBBB... : メッセージの内容です。

(4) 解析ログの出力例

[2020/02/13 18:50:37], [情報], [システム], KEDL1231A-I プロジェクト(販売管理システム)の解析処理を開始しました。	1.
[2020/02/13 18:50:37], [情報], [解析], KEDL1261A-I (1/4) 'D:¥Temp¥CBLSOURCE¥伝票作成サブ.cb1' を解析中...	
[2020/02/13 18:50:37], [情報], [解析], KEDL1261A-I (2/4) 'D:¥Temp¥CBLSOURCE¥伝票作成メイン.cb1' を解析中...	
[2020/02/13 18:50:38], [情報], [解析], KEDL1261A-I (3/4) 'D:¥Temp¥CBLSOURCE¥入金処理サブ.cb1' を解析中...	
[2020/02/13 18:50:38], [情報], [解析], KEDL1261A-I (4/4) 'D:¥Temp¥CBLSOURCE¥入金処理メイン.cb1' を解析中...	
[2020/02/13 18:50:38], [情報], [システム], KEDL1232A-I プロジェクト(販売管理システム)の解析処理を終了しました。	2.
[2020/02/13 18:50:38], [情報], [システム], KEDL1233A-I プロジェクト(販売管理システム)の生成処理を開始しました。	
[2020/02/13 18:50:41], [情報], [システム], KEDL1234A-I プロジェクト(販売管理システム)の生成処理を終了しました。	

1. 解析に関するログ

- 解析処理の開始
- 各 COBOL ソースファイルの解析開始
- COBOL2002 コンパイラの構文解析メッセージ
- 解析処理の終了

2. 生成に関するログ

- 生成処理の開始
- 生成処理の終了

関連項目

- 「9.3.3 メッセージの一覧」
- 「10.10 COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI)」

9.2.2 操作ログを参照する

操作ログは、ソース解析プロジェクトの解析生成オプションの設定や COBOL ソースファイルの登録や削除に問題がないかどうか、また必要な解析や生成が行われているかを確認するときに参照します。

操作ログは、次に示すタイミングで出力されます。

- cblcsa コマンドの起動と終了
- 次のプロジェクト操作の処理の開始と終了
 - ソース解析プロジェクトの作成、名称変更および削除
 - 解析生成オプションの登録
 - COBOL ソースファイルの登録や削除
 - COBOL ソース解析用データベースの初期化
- 解析／生成の各処理の開始と終了
- エラー検出時

COBOL ソース解析が操作ログファイルの作成に失敗した場合、エラーメッセージを出力して、処理を終了します。

(1) 操作ログの出力先ファイル

操作ログは、次に示すファイルに格納されます。

- 開発マネージャの場合または cblcsa コマンドに -LogFolder オプションを指定しない場合

COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) で定義されているフォルダ¥COBOLControl Log.txt (現行ログ)

COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) で定義されているフォルダ¥COBOLControlLog2.txt (退避ログ)

COBOL ソース解析がメッセージを出力するたびに操作ログがファイルに追加書きされます。現行ログの最大容量を超えた場合、退避ログにコピーしたあとで、現行ログが初期化されてログが出力されます。退避ログにすでにログがある場合は上書きされ、古いログは消去されます。このため、操作ログを保持する場合は、適時、退避ログにある操作ログファイルを任意の場所にバックアップしてください。

- cblcsa コマンドに-LogFolder オプションを指定した場合

cblcsaコマンドの-LogFolderオプションに指定されたフォルダ¥COBOLControl_プロジェクト名_yyyymmddhhmmss_Log.txt

-LogFolder オプションを省略したときは、COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) に定義されている操作ログの出力先フォルダに出力されます。

操作ログはタイムスタンプ付きのファイル名で出力されるため、不要な操作ログは定期的に削除することを推奨します。

-InitializeDatabase オプションや-ListProject オプションを指定した場合、操作ログファイル名に"プロジェクト名_"は出力されません。

同じプロジェクトを連続で解析するなど、cblcsa コマンドを起動した開始時刻が秒まで同じとなる場合、同じ操作ログファイルに複数の操作ログが出力されます。1 秒以上間隔をあけて実行してください。

出力先フォルダや操作ログファイルの最大サイズは、COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) で変更できます。

COBOL ソース解析が操作ログファイルの作成に失敗した場合、次のように動作します。

- 開発マネージャ画面の場合
エラーメッセージを表示して、処理を続行します。
- cblcsa コマンドの場合
エラーメッセージを標準エラー出力に出力して、処理を終了します。

(2) 操作ログの表示方法

操作ログファイルをメモ帳などのテキストを表示するプログラムで開きます。

注意事項

操作ログを表示したあとで、再度 COBOL ソース解析を実行する場合は、いったん操作ログファイルを閉じてください。操作ログファイルを開いたまま COBOL ソース解析を実行すると、操作ログを正しく取得できないことがあります。また、COBOL ソース解析の実行中は、ファイルを排他制御するエディタで操作ログを開かないでください。

(3) 操作ログの出力形式

形式

[yyyy/mm/dd hh:mm:ss.fff] , [AA] , [BBBB] CCCC...DDDD,EEEE,FFFF,GGGG,HHHH

形式の説明

- [yyyy/mm/dd hh:mm:ss.fff] : ログを採取した時刻です (yyyy : 西暦年, mm : 月, dd : 日, hh : 時, mm : 分, ss : 秒, fff : ミリ秒)。
- [AA] : ログの重要度です。
「情報」: 解析の開始時間, 終了時間などの情報を示します。
「警告」: 解析は成功しましたが, 一部の情報を取得できていないことを示します。
「エラー」: プロジェクト操作, 解析または生成に失敗していることを示します。プロジェクト操作や解析の結果はデータベースに格納されていません。生成中のエラーでは, 生成中のドキュメントおよびそのドキュメントと関連するドキュメントが未完成です。
- [BBBB] : ログの種別です。
「システム」: COBOL ソース解析のシステムエラー, 実行ログの情報を示します。
「DB」: データベースアクセスエラーを示します。
「操作」: COBOL ソース解析の操作の実行ログを示します。
「追加」: 追加した COBOL ソースファイルを示します。
「削除」: 削除した COBOL ソースファイルを示します。
「選択方法」: 解析対象と生成対象の選択方法の情報を示します。
「解析オプション」: プロジェクトに設定された解析オプションの状態を示します。
「パース」: COBOL ソースファイルを読み込むとき, または COPY 文による登録集原文を展開するときのエラーを示します。
「解析」: COBOL ソースファイルの構文を解析するとき, または組み込みデータベースに登録するときの実行ログおよびエラーを示します。
「生成オプション」: プロジェクトに設定された生成オプションの状態を示します。
「生成」: 組み込みデータベースのデータから, ソース解析情報を生成するときの実行ログおよびエラーを示します。
「実行結果」: 解析および生成の実行結果を示します。
- CCCC... : メッセージです。
- DDDD : エラーを起こしたプログラム名です。
- EEEE : エラーを起こした行の番号です。ファイルの入出力エラー/DB アクセス時のエラーなどのシステムエラーや, リソース不足など, COBOL ソースファイルの行を特定できない場合は表示されません。
- FFFF : エラー番号です。
- GGGG : エラー詳細です。

- *HHHH*：エラーを検知した COBOL ソース解析のソースファイル名とプロシジャ名です。(ソースファイル名, プロシジャ名)の形式で出力されます。

注意事項

- 重要度が「情報」の場合は、出力項目は「CCCC」までです。「DDDD」以降は出力されません。末尾のコンマも出力されません。
- 重要度が「警告」または「エラー」の場合は、すべての項目が出力されます。
- 「DDDD」以降の項目については、検知したログの種別によって項目の数が増減します。「HHHH」が出力されるときは末尾のコンマは出力されません。「HHHH」が出力されないときは末尾にコンマが出力されます。

(4) 操作ログの出力例

[2021/08/05 00:51:52.446], [情報], [システム], -----	
[2021/08/05 00:51:52.471], [情報], [システム], 基本情報	
[2021/08/05 00:51:52.472], [情報], [システム], Program Product: P-CC2651-1234 04-40	1.
[2021/08/05 00:51:52.473], [情報], [システム], HOSTNAME: TOOLKIT-SERVER	
[2021/08/05 00:51:52.474], [情報], [システム], OS: Microsoft Windows [Version 6.2.9200]	
[2021/08/05 00:51:52.474], [情報], [システム], ODBC DSN: COBOL2002ToolKitDB	
[2021/08/05 00:51:52.475], [情報], [システム], コマンド開始:	
[2021/08/05 00:51:52.476], [情報], [操作], オプション登録の開始: プロジェクト名=販売管理システム	2.
[2021/08/05 00:51:53.522], [情報], [操作], オプション登録の終了: プロジェクト名=販売管理システム	
[2021/08/05 00:51:53.522], [情報], [システム], コマンド終了: 終了コード=0	5.
[2021/08/05 00:51:58.055], [情報], [システム], -----	
[2021/08/05 00:51:58.080], [情報], [システム], 基本情報	
[2021/08/05 00:51:58.081], [情報], [システム], Program Product: P-CC2651-1234 04-40	1.
[2021/08/05 00:51:58.081], [情報], [システム], HOSTNAME: TOOLKIT-SERVER	
[2021/08/05 00:51:58.082], [情報], [システム], OS: Microsoft Windows [Version 6.2.9200]	
[2021/08/05 00:51:58.083], [情報], [システム], ODBC DSN: COBOL2002ToolKitDB	
[2021/08/05 00:51:58.083], [情報], [システム], コマンド開始:	
[2021/08/05 00:51:58.085], [情報], [操作], プロジェクト同期の開始: プロジェクト名=販売管理システム	
[2021/08/05 00:51:58.602], [警告], [追加], KEDL1133A-W "C:\販売管理システム¥入金処理¥入金画面処理.CBL" はプログラム名が取得できないため登録されませんでした。	2.
[2021/08/05 00:51:58.612], [情報], [操作], プロジェクト同期の終了: プロジェクト名=販売管理システム	
[2021/08/05 00:51:58.613], [情報], [システム], コマンド終了: 終了コード=0	5.
[2021/08/05 00:51:59.147], [情報], [システム], -----	
[2021/08/05 00:51:59.172], [情報], [システム], 基本情報	
[2021/08/05 00:51:59.174], [情報], [システム], Program Product: P-CC2651-1234 04-40	1.
[2021/08/05 00:51:59.174], [情報], [システム], HOSTNAME: TOOLKIT-SERVER	
[2021/08/05 00:51:59.175], [情報], [システム], OS: Microsoft Windows [Version 6.2.9200]	
[2021/08/05 00:51:59.175], [情報], [システム], ODBC DSN: COBOL2002ToolKitDB	
[2021/08/05 00:51:59.176], [情報], [システム], コマンド開始:	
[2021/08/05 00:51:59.178], [情報], [操作], 解析の実行 メニュー開始 プロジェクト名: 販売管理システム	
[2021/08/05 00:51:59.912], [情報], [操作], 解析 コマンド開始	
[2021/08/05 00:51:59.920], [情報], [解析オプション], コンパイラオプション=-NumAccept -DebugLine	
[2021/08/05 00:51:59.921], [情報], [解析オプション], コンパイラ環境変数:	
[2021/08/05 00:51:59.921], [情報], [解析オプション], CBLCOPT2002=-StdVersion.1 - DebugData -Switch, EBCDIC	
[2021/08/05 00:51:59.922], [情報], [解析オプション], CBLLIB=c:\販売管理システム¥伝票作成¥work\C:\Program Files¥Hitachi¥COBOL2002 64bit¥copy	
[2021/08/05 00:51:59.922], [情報], [解析オプション], CBLPIDIR=c:\販売管理システム¥伝票作成¥work	
[2021/08/05 00:51:59.922], [情報], [解析オプション], 登録集原文フォルダ:	3.
[2021/08/05 00:51:59.923], [情報], [解析オプション], c:\販売管理システム¥伝票作成¥work	
[2021/08/05 00:51:59.923], [情報], [解析オプション], C:\Program Files¥Hitachi¥COBOL2002 64bit¥copy	
[2021/08/05 00:51:59.923], [情報], [解析オプション], 主プログラムの自動検出: する	
[2021/08/05 00:51:59.924], [情報], [解析オプション], USING検出条件:	
[2021/08/05 00:51:59.924], [情報], [解析オプション], 引数なし=ON	
[2021/08/05 00:51:59.925], [情報], [解析オプション], 引数1個で英数字である=OFF	
[2021/08/05 00:51:59.925], [情報], [解析オプション], 引数1個で可変長文字列型の集団項目である=ON	
[2021/08/05 00:51:59.925], [情報], [解析オプション], 引数2個で2進整数とポインタである=ON	
[2021/08/05 00:51:59.926], [情報], [解析オプション], RETURNING検出条件:	
[2021/08/05 00:51:59.926], [情報], [解析オプション], RETURNING指定なし=ON	

(続く)

(続き)

[2021/08/05 00:51:59.926], [情報], [解析オプション], 主プログラム名の指定=
[2021/08/05 00:51:59.927], [情報], [解析オプション], プログラムコメント抽出:
[2021/08/05 00:51:59.927], [情報], [解析オプション], プログラムコメントを抽出する=OFF
[2021/08/05 00:51:59.927], [情報], [解析オプション], 節コメント抽出:
[2021/08/05 00:51:59.928], [情報], [解析オプション], 節コメントを抽出する=OFF
[2021/08/05 00:51:59.928], [情報], [解析オプション], 引数整合性チェック:
[2021/08/05 00:51:59.929], [情報], [解析オプション], 引数の個数を検査する=ON
[2021/08/05 00:51:59.929], [情報], [解析オプション], 実引数個数>仮引数個数の場合は検
出ししない=OFF
[2021/08/05 00:51:59.929], [情報], [解析オプション], 引数の受渡方法を検査する=ON
[2021/08/05 00:51:59.930], [情報], [解析オプション], 引数のデータ型を検査する=ON
[2021/08/05 00:51:59.930], [情報], [解析オプション], 引数のサイズを検査する=ON
[2021/08/05 00:51:59.930], [情報], [解析オプション], 次の条件をすべて満たすケースを検
出ししない=OFF
[2021/08/05 00:51:59.931], [情報], [解析オプション], ・ BY REFERENCEまたはBY CONTENTで
ある
[2021/08/05 00:51:59.931], [情報], [解析オプション], ・ 一方または両方が英数字集団項目
である
[2021/08/05 00:51:59.932], [情報], [解析オプション], ・ 実引数サイズ>仮引数サイズであ
る
[2021/08/05 00:51:59.932], [情報], [解析オプション], 一方が可変長集団項目の場合はサイ
ズ検査対象外とする=ON
[2021/08/05 00:51:59.932], [情報], [解析オプション], 引数の集団項目の構造を検査する=ON
[2021/08/05 00:51:59.933], [情報], [解析オプション], 対応する仮引数のない実引数を検出す
る=OFF
[2021/08/05 00:51:59.933], [情報], [解析オプション], 対応する実引数のない仮引数を検出す
る=OFF
[2021/08/05 00:51:59.933], [情報], [解析オプション], その他:
[2021/08/05 00:51:59.934], [情報], [解析オプション], レベル番号が02~49, 66のデータ項目の
情報を抽出する=ON
[2021/08/05 00:51:59.934], [情報], [解析オプション], プログラムの制御フロー情報を抽出す
る=ON
[2021/08/05 00:51:59.935], [情報], [選択方法], 解析対象の選択方法=AUTO
[2021/08/05 00:51:59.938], [情報], [解析], 解析開始: 'C:¥販売管理システム¥伝票作成¥伝票作
成メイン.cbl'
[2021/08/05 00:52:00.186], [情報], [解析], 解析開始: 'C:¥販売管理システム¥入金処理¥顧客マ
スタ検索.CBL'
[2021/08/05 00:52:00.293], [情報], [解析], 解析開始: 'C:¥販売管理システム¥入金処理¥入金処
理メイン.cbl'
[2021/08/05 00:52:00.452], [情報], [解析], 解析開始: システム全体の解析
[2021/08/05 00:52:00.453], [情報], [解析], 解析開始: 重複プログラム名の名称変更
[2021/08/05 00:52:00.469], [情報], [解析], 解析開始: 未使用登録集原文の登録
[2021/08/05 00:52:00.484], [情報], [解析], 解析開始: 外部プログラムの呼出先解決
[2021/08/05 00:52:00.489], [情報], [解析], 解析開始: 実行されないCALL文の印付け
[2021/08/05 00:52:00.492], [情報], [解析], 解析開始: 未使用外部プログラムの登録
[2021/08/05 00:52:00.495], [情報], [解析], 解析開始: 未使用資産情報の登録
[2021/08/05 00:52:00.506], [情報], [操作], 解析 コマンド終了
[2021/08/05 00:52:00.507], [情報], [操作], 生成 コマンド開始
[2021/08/05 00:52:00.508], [情報], [生成オプション], -HTML=ON
[2021/08/05 00:52:00.509], [情報], [生成オプション], -一覧情報CSV
[2021/08/05 00:52:00.510], [情報], [生成オプション], -COBOLソースファイル一覧=ON
[2021/08/05 00:52:00.510], [情報], [生成オプション], -プログラム一覧=ON
[2021/08/05 00:52:00.511], [情報], [生成オプション], -セクション一覧=ON
[2021/08/05 00:52:00.511], [情報], [生成オプション], -ファイル一覧=ON
[2021/08/05 00:52:00.512], [情報], [生成オプション], -データ項目一覧=ON
[2021/08/05 00:52:00.512], [情報], [生成オプション], -不整合引数一覧=ON
[2021/08/05 00:52:00.513], [情報], [生成オプション], -実行順序リスト=ON

3.

(続く)

(続き)

```
[2021/08/05 00:52:00.513], [情報], [生成オプション], -関係情報CSV
[2021/08/05 00:52:00.513], [情報], [生成オプション],   |プログラム呼出関係=ON
[2021/08/05 00:52:00.514], [情報], [生成オプション],   |登録集原文参照関係=ON
[2021/08/05 00:52:00.514], [情報], [生成オプション],   |ファイル参照関係=ON
[2021/08/05 00:52:00.514], [情報], [生成オプション],   |表 (DB) 参照関係=ON
[2021/08/05 00:52:00.515], [情報], [生成オプション],   出力先フォルダ: C:\Users\¥COBOL¥COBOL
Analyzer
[2021/08/05 00:52:00.515], [情報], [選択方法], 生成対象の選択方法=AUTO
[2021/08/05 00:52:00.529], [情報], [生成], 生成開始: システム固有: プログラム階層図
[2021/08/05 00:52:01.015], [情報], [生成], 生成開始: プログラム固有: プログラム名
=COBOL85_PROGRAM (3)
[2021/08/05 00:52:01.370], [情報], [生成], 生成開始: プログラム固有: プログラム名
=COBOL85_PROGRAM (2)
[2021/08/05 00:52:01.585], [情報], [生成], 生成開始: プログラム固有: プログラム名
=COBOL85_PROGRAM (1)
[2021/08/05 00:52:01.797], [情報], [生成], 生成開始: システム固有: プログラム一覧
[2021/08/05 00:52:01.813], [情報], [生成], 生成開始: システム固有: COBOLソースファイル一覧
[2021/08/05 00:52:01.824], [情報], [生成], 生成開始: システム固有: 登録集原文ファイル一覧
[2021/08/05 00:52:01.834], [情報], [生成], 生成開始: システム固有: ファイル一覧
[2021/08/05 00:52:01.845], [情報], [生成], 生成開始: システム固有: ソース表示
[2021/08/05 00:52:01.908], [情報], [生成], 生成開始: システム固有: 未登録のプログラム一覧
[2021/08/05 00:52:01.918], [情報], [生成], 生成開始: システム固有: 重複するプログラム一覧
[2021/08/05 00:52:01.928], [情報], [生成], 生成開始: システム固有: ファイル詳細
[2021/08/05 00:52:01.947], [情報], [生成], 生成開始: システム固有: 表 (DB) 一覧
[2021/08/05 00:52:01.956], [情報], [生成], 生成開始: システム固有: 表 (DB) 詳細
[2021/08/05 00:52:01.959], [情報], [生成], 生成開始: システム固有: 登録集原文詳細の登録集原文概要
[2021/08/05 00:52:01.962], [情報], [生成], 生成開始: システム固有: 登録集原文詳細のCOPY元プログラム
[2021/08/05 00:52:01.966], [情報], [生成], 生成開始: システム固有: 行数の分布
[2021/08/05 00:52:01.978], [情報], [生成], 生成開始: システム固有: 実行文数の分布
[2021/08/05 00:52:01.988], [情報], [生成], 生成開始: システム固有: 分岐数の分布
[2021/08/05 00:52:01.997], [情報], [生成], 生成開始: システム固有: 記述項数の分布
[2021/08/05 00:52:02.006], [情報], [生成], 生成開始: システム固有: 節数の分布
[2021/08/05 00:52:02.016], [情報], [生成], 生成開始: システム固有: 段落数の分布
[2021/08/05 00:52:02.025], [情報], [生成], 生成開始: システム固有: 呼出先プログラム数の分布
[2021/08/05 00:52:02.035], [情報], [生成], 生成開始: システム固有: 呼出元プログラム数の分布
[2021/08/05 00:52:02.044], [情報], [生成], 生成開始: システム固有: 登録集原文一覧
[2021/08/05 00:52:02.231], [情報], [生成], 生成開始: システム固有: 未登録の登録集原文一覧
[2021/08/05 00:52:02.240], [情報], [生成], 生成開始: システム固有: デッドコード情報一覧
[2021/08/05 00:52:02.261], [情報], [生成], 生成開始: システム固有: COBOLソースファイル一覧
CSV
[2021/08/05 00:52:02.271], [情報], [生成], 生成開始: システム固有: プログラム一覧CSV
[2021/08/05 00:52:02.283], [情報], [生成], 生成開始: システム固有: セクション一覧CSV
[2021/08/05 00:52:02.329], [情報], [生成], 生成開始: システム固有: ファイル一覧CSV
[2021/08/05 00:52:02.342], [情報], [生成], 生成開始: システム固有: データ項目一覧CSV
[2021/08/05 00:52:02.376], [情報], [生成], 生成開始: システム固有: 不整合引数一覧CSV
[2021/08/05 00:52:02.391], [情報], [生成], 生成開始: システム固有: 実行順序リストCSV
[2021/08/05 00:52:02.412], [情報], [生成], 生成開始: システム固有: プログラム呼出関係CSV
[2021/08/05 00:52:02.424], [情報], [生成], 生成開始: システム固有: 登録集原文参照関係CSV
[2021/08/05 00:52:02.435], [情報], [生成], 生成開始: システム固有: ファイル参照関係CSV
[2021/08/05 00:52:02.442], [情報], [生成], 生成開始: システム固有: 表 (DB) 参照関係CSV
[2021/08/05 00:52:02.451], [情報], [生成], 生成開始: プロジェクト固有: プロジェクトサマリ
```

3.

(続く)

(続き)

```
[2021/08/05 00:52:02.656], [情報], [操作], 生成 コマンド終了
[2021/08/05 00:52:02.658], [情報], [実行結果],      解析      生成
[2021/08/05 00:52:02.658], [情報], [実行結果], 成功:      3      3
[2021/08/05 00:52:02.659], [情報], [実行結果], 警告:      0      0
[2021/08/05 00:52:02.659], [情報], [実行結果], 失敗:      0      0
[2021/08/05 00:52:02.659], [情報], [実行結果], 合計: 3 / 3  3 / 3
[2021/08/05 00:52:03.142], [情報], [操作], 解析の実行 メニュー終了 プロジェクト名: 販売管理システム
[2021/08/05 00:52:03.143], [情報], [システム], コマンド終了: 終了コード=0
```

3.

4.

5.

1. cblcsa コマンド起動

- 基本情報

次の項目が出力されます。

- 製品の形名バージョン
- Windows マシンのホスト名 + OS 名 (バージョン)
- 接続先の ODBC データソース名 (COBOL2002ToolKitDB)

- cblcsa コマンド開始

コマンド引数が出力されます。

2. COBOL ソース解析のプロジェクト操作の実行

次に示すプロジェクト操作の処理の開始／終了ログやエラーメッセージが出力されます。

- ソース解析プロジェクトの作成, 名称変更, および削除
- 解析生成オプションの登録
- COBOL ソースファイルの登録と削除
- COBOL ソース解析用データベースの初期化

3. 解析／生成の実行

解析または生成処理の開始／終了ログやエラーメッセージが出力されます。

4. 解析／生成の実行結果

解析または生成の成功, 警告, 失敗の集計結果が出力されます。

5. cblcsa コマンド終了

終了コードが出力されます。

関連項目

- [「9.3.3 メッセージの一覧」](#)
- [「10.10 COBOL ソース解析の環境定義ファイル \(CBLANALYZER.INI\)」](#)

9.3 メッセージから探す

COBOL ソース解析の操作で出力されるメッセージの説明です。

コンパイラ、テストデバグガ、および COBOL2002 の実行時のエラーメッセージ（KCCC で始まるメッセージ）については、マニュアル「COBOL2002 メッセージ」を参照してください。

組み込みデータベースの操作（DB ユティリティの操作）で出力されるメッセージについては、マニュアル「COBOL2002 Professional 製品 導入ガイド」を参照してください。

9.3.1 メッセージの形式

COBOL ソース解析のメッセージの形式です。

[yyyy/mm/dd hh:mm:ss], [yyyy/mm/dd hh:mm:ss.fff], [AA], [BBBB] 部分については、[\[9.2.1 解析ログを参照する\]](#) または [\[9.2.2 操作ログを参照する\]](#) を参照してください。

形式

コンソールだけに出力する場合

KEDLnnnnA-i XX..XX CCCC...

解析ログとコンソールに出力する場合

[yyyy/mm/dd hh:mm:ss], [AA], [BBBB] KEDLnnnnA-i XX..XX CCCC...

操作ログとコンソールに出力する場合

[yyyy/mm/dd hh:mm:ss.fff], [AA], [BBBB] KEDLnnnnA-i XX..XX CCCC...

(注) 操作ログメッセージをコンソールに出力する場合、fff は出力しません。

形式の説明

- KEDLnnnnA-i : メッセージ ID を示します。「i」はメッセージの重要度を示します。メッセージの重要度には次の 4 種類があります。
 - 「E」: エラーメッセージです。正常に動作しないトラブルが起こったことを示します。
 - 「W」: 警告メッセージです。リソースの使用状況などについての警告、またはコマンドの指定誤りに対して値を仮定して処理を続行することを示します。
 - 「I」: インフォメーションメッセージです。「E」および「W」に該当しない単純な動作状況を示します。
- XX..XX : メッセージテキストを示します。
- CCCC... : エラーが発生したプログラム名、ファイル名、エラー番号、エラー詳細、エラーが発生したファイル名、行番号、および対象行などが表示されます。該当する情報がない場合は表示されません。

メッセージの出力例

[2015/04/10 16:41:07],[警告],[生成],KEDL1237A-W 割り当てファイル呼出元の階層レベルが10を超えています。11以上の階層レベルは割り当てファイル呼出元階層図には表示しません。割り当てファイル名(*****),割り当てファイルアクセスプログラム名(*****),階層レベル超えプログラムの呼出先プログラム名(*****),階層レベル超えプログラム名(*****)

9.3.2 解析エラー発生時のエラー位置の表示

COBOL ソース解析を実行して解析エラーになったときに、COBOL ソースファイルのエラー位置を特定できるように、エラーメッセージの次行にエラー発生時の COBOL ソースファイルのパス名、行番号、および対象行が表示されることがあります。エラー発生時の COBOL ソースファイルのパス名、行番号、およびソース行の出力形式を次に示します。

形式

"COBOL ソースファイルのパス名", line *nnnn*: ソース行

説明

- COBOL ソースファイルのパス名：
エラーになった COBOL ソースファイルの絶対パス名です。
- line *nnnn* :
エラーが発生した行番号です。
- ソース行：
エラーが発生したソース行が表示されます。ただし、空白は削除して表示されることがあります。

9.3.3 メッセージの一覧

COBOL ソース解析の操作で出力されるメッセージと対処方法について説明します。

(1) COBOL ソース解析のメッセージ (KEDL0000~KEDL0999) と対処

表 9-1 COBOL ソース解析のメッセージ (KEDL0000~KEDL0999) と対処

メッセージ ID	メッセージテキスト	対処
KEDL0003A-E	メモリ不足が発生しました。詳細情報=*** 1 ***	メッセージ出力時に処理している COBOL ソースファイルを分割して再実行するか、またはソース解析プロジェクトから該当する COBOL ソースファイルを削除して再実行してください。
KEDL0004A-E	内部エラーが発生しました。詳細情報=*** 1 ***	当社保守員に連絡してください。
KEDL0005A-E	予期せぬエラーが発生しました。詳細情報=*** 1 ***	メッセージ中の詳細情報を基に原因を調査してください。

メッセージ ID	メッセージテキスト	対処
		原因がメッセージの詳細情報で特定できない場合は、当社保守員に連絡してください。
KEDL0101A-E	*** 1 *** オプションの引数が不正です。	オプションを修正して再実行してください。
KEDL0102A-E	*** 1 *** オプションに引数がありません。	オプションを修正して再実行してください。
KEDL0103A-E	*** 1 *** は不正なオプションです。	オプションを修正して再実行してください。
KEDL0104A-E	操作対象のプロジェクトを指定する-Project オプションが指定されていません。	-Project オプションを修正して再実行してください。
KEDL0105A-E	操作を指定するオプションが指定されていません。	操作オプションを指定して再実行してください。
KEDL0107A-E	*** 1 *** オプションと *** 2 *** オプションは同時に指定できません。	操作対象のオプションが1つになるように修正して再実行してください。
KEDL0108A-W	*** 1 *** は不正なソースファイル名です。このファイル名の指定を無視します。	ソースファイル名を修正して再実行してください。
KEDL0109A-W	ソースファイル *** 1 *** が見つかりません。このファイル名の指定を無視します。	指定したパスおよびファイル名を修正して再実行してください。
KEDL0110A-E	*** 1 *** は不正なソースフォルダ名です。	フォルダ名を修正して再実行してください。
KEDL0111A-E	ソースフォルダ *** 1 *** が見つかりません。	指定したパスおよびフォルダ名を修正して再実行してください。または、アクセス権を修正して再実行してください。
KEDL0112A-E	*** 1 *** は無効なコマンド引数です。	正しいコマンドラインで再実行してください。
KEDL0113A-E	*** 1 *** オプションにソースファイルを指定するオプションが指定されていません。	ソースファイルを指定するオプションを指定して再実行してください。
KEDL0114A-E	-Remove オプション指定時にソースファイル名が絶対パス名で指定されていません。	ソースファイルの絶対パス名を指定して再実行してください。
KEDL0115A-E	プロジェクト名 *** 1 *** が不正です。	プロジェクト名を修正して再実行してください。
KEDL0116A-E	*** 1 *** オプションは*** 2 *** オプション指定時だけ有効です。	オプションを修正して再実行してください。
KEDL0117A-E	ソース解析ソースリストファイル *** 1 *** が見つかりません。	ソース解析ソースリストファイルのパス名を修正して再実行してください。
KEDL0118A-E	ソース解析オプションファイル ***1*** が見つかりません。	ソース解析オプションファイルのパス名を修正して再実行してください。
KEDL0119A-E	有効なソースファイルの指定がありません。	正しいオプションに修正して再実行してください。
KEDL0122A-E	-LogFolder オプションに指定されたフォルダが見つかりません。	存在するフォルダの名称を指定してください。
KEDL0123A-E	-LogFolder オプションに指定されたフォルダの絶対パス名が長すぎます。	指定されたフォルダに対応する絶対パス名が190文字以下となるようにフォルダのパス名を指定してください。

メッセージID	メッセージテキスト	対処
KEDL0124A-E	-Source オプションおよび-SourceFolder オプションにアポストロフィ(')またはカンマ(,)を含むパス名は指定できません。	ソースファイルまたはソースフォルダのパス名を変更して再実行してください。

(2) COBOL ソース解析のメッセージ (KEDL1000~KEDL1999) と対処

表 9-2 COBOL ソース解析のメッセージ (KEDL1000~KEDL1999) と対処

メッセージID	メッセージテキスト	対処
KEDL1005A-E	COBOL2002 コンパイラの起動に失敗しました。 エラー番号=*** 1 ***,エラー詳細=*** 2 ***	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1015A-E	COBOL2002 コンパイラの解析情報の取得に失敗しました。	当社保守員に連絡してください。
KEDL1016A-W	節(*** 1 ***)のコメント行の文字列が取得できませんでした。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1017A-E	登録集原文の解析情報取得の SQL コマンド実行で予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1018A-E	COBOL ソースファイルの解析情報のデータベース登録処理で予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1020A-E	SQL コマンド実行で予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1021A-E	COBOL ソースファイルの解析情報削除の SQL コマンド実行で予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1024A-E	修飾パスの取得中に予期しないエラーが発生しました。	正しい修飾名に変更してください。
KEDL1025A-E	節情報の取得中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1052A-W	コメント行の文字列が取得できませんでした。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1056A-E	組み込みデータベースとの接続に失敗しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1057A-E	プロジェクトのオプション情報の読み込み中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1059A-E	ファイルへの保存に失敗しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1061A-E	XSLT ファイル(*** 1 ***)が存在しません。	COBOL2002 Professional Tool Kit を再インストールしてください。再インストールしても、同じメッセージが出るときは、当社保守員に連絡してください。

メッセージID	メッセージテキスト	対処
KEDL1062A-E	*** 1 ***への変換中に予期しないエラーが発生しました。XSLT ファイル：*** 2 ***	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1063A-E	ファイル（*** 1 ***）の保存中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1064A-E	XML の保存中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1071A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1072A-E	プロジェクトサマリの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1074A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1075A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1077A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1079A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1081A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1083A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1084A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1086A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1087A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1089A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1090A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1091A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1094A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1098A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。

メッセージID	メッセージテキスト	対処
KEDL1099A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1108A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1113A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1114A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1115A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1116A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1120A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1124A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1126A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1127A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1129A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1130A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1131A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1132A-E	*** 1 ***ドキュメントの生成中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1133A-W	*** 1 ***はプログラム名が取得できないため登録されませんでした。	—
KEDL1134A-E	プロジェクトの状態情報の書き込み中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1135A-E	ソースファイルの状態情報の書き込み中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1136A-E	状態フラグの取得中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1141A-E	ソースファイル情報の取得中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。

メッセージ ID	メッセージテキスト	対処
KEDL1142A-E	プロジェクトの読み込み中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1143A-E	プロジェクトのソースファイル情報の読み込み中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1146A-E	一連番号の取得中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1200A-E	オプション値の読み込み時に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1229A-E	エラーファイルのアクセス中に予期しないエラーが発生しました。	ディスク容量が十分あるかを確認してください。
KEDL1230A-E	コマンドファイルの作成中に予期しないエラーが発生しました。	ディスク容量が十分あるかを確認してください。
KEDL1231A-I	プロジェクト(** 1 **)の解析処理を開始しました。	—
KEDL1232A-I	プロジェクト(** 1 **)の解析処理を終了しました。	—
KEDL1233A-I	プロジェクト(** 1 **)の生成処理を開始しました。	—
KEDL1234A-I	プロジェクト(** 1 **)の生成処理を終了しました。	—
KEDL1235A-W	プログラムの階層レベルが 10 を超えています。11 以上の階層レベルはプログラム階層図には表示しません。基点となるプログラム名(** 1 **),階層レベル超えプログラムの呼出元プログラム名(** 2 **),階層レベル超えプログラム名(** 3 **)	—
KEDL1236A-W	セクションの階層レベルが 10 を超えています。11 以上の階層レベルはセクション階層図には表示しません。基点となる節名(** 1 **),階層レベル超え節の呼出元節名(** 2 **),階層レベル超え節名(** 3 **)	—
KEDL1256A-E	最大プロジェクト ID の取得中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1260A-E	COBOL ソースファイルの各種解析情報の合計が 400,000 個を超えています。この COBOL ソースファイルの解析を中止します。	<p>1 つの COBOL ソースファイル内の、次に示す項目の個数の合計が 400,000 個を超えています。COBOL ソースファイルを複数に分割してください。</p> <ul style="list-style-type: none"> • ソース行（約 256 行で 1 個） • プログラム • 部見出し • 節名 • 段落名 • ファイル管理記述項 • ファイル入出力文（READ, WRITE, REWRITE, DELETE）

メッセージID	メッセージテキスト	対処
		<ul style="list-style-type: none"> 登録集原文 COPY 文 01, 77 レベルデータ項目（記述項情報） 01 レベル集団項目（構成情報） 仮引数情報 PERFORM 文の呼出先情報 CALL 文 実引数情報 報告書、画面、および通信記述名 デッドコード情報 制御フロー情報
KEDL1261A-I	'(n/m) *** ソースファイルパス ***'を解析中...	COBOL2002 Net Developer を開始するコマンド (ccbl2002 コマンド) でコンパイルする直前に出力される情報メッセージです。
KEDL1262A-W	COBOL ソースファイル'*** ソースファイルパス ***'の解析時にエラーが発生したため、ドキュメント生成を警告で終了しました。	COBOL2002 Net Developer でのコンパイルがエラーになった場合に、一部の解析情報から一部のドキュメントを生成できたときに警告として出力されます。 正しい解析結果が出力されるようにコンパイルエラーを修正してください。
KEDL1263A-W	COBOL ソースファイル'*** ソースファイルパス ***'の解析時に警告エラーが発生しました。	COBOL2002 Net Developer でのコンパイルが警告エラーになった場合に、解析の警告メッセージとして出力されます。 正しい解析結果が出力されるようにコンパイルエラーを修正して、解析/生成を実行してください。
KEDL1264A-E	COBOL ソースファイル'*** ソースファイルパス ***'の解析時に重大/回復不能エラーが発生しました。	COBOL2002 Net Developer でのコンパイルが重大/回復不能エラーになった場合に、解析のエラーメッセージとして出力されます。 正しい解析結果が出力されるようにコンパイルエラーを修正して、解析/生成を実行してください。
KEDL1265A-E	重複するプログラム名の個数が 999 を超えています。、プログラム名=*** 1 ***	COBOL ソースファイルの入力時に、重複するプログラム名の個数が 999 を超えました。 重複するプログラム名を変更してください。
KEDL1266A-E	重複するプログラム名の個数が 999 を超えています。、プログラム名=*** 1 ***	COBOL ソースファイルの解析時に、重複するプログラム名の個数が 999 を超えました。 重複するプログラム名を変更してください。
KEDL1267A-W	プログラム名の長さが 160 バイトを超えています。先頭の 160 バイトだけを採用します。、COBOL ソースファイル名 (** 1 **)	プログラム名の長さが 160 バイトを超えています。プログラム名の長さを 160 バイト以内に変更してください。
KEDL1268A-W	*** 1 ***の数が 32767 を超えています。32768 以上の *** 1 ***は解析しません。	1 つの COBOL ソースファイル内の、次に示す項目の個数が 32,767 個を超えています。メッセージ

メッセージ ID	メッセージテキスト	対処
		<p>に表示されている項目の数を 32,767 個以内に變更してください。</p> <ul style="list-style-type: none"> • プログラム • 部見出し • 節名（手続き部の先頭が節の見出しで始まっていない場合に節名として仮定される「{PROCEDURE_START}」を含む） • 段落名 • ファイル管理記述項 • ファイルの入出力文（READ, WRITE, REWRITE, DELETE） • 登録集原文 • COPY 文 • PERFORM 文の呼出先情報 • CALL 文 • CRUD 操作情報
KEDL1270A-E	ソースファイル'*** ソースファイルパス ***'中に不当なコードがあります。コード値=*** 不当コード ***	COBOL ソースファイル中の不当なコードを削除してください。
KEDL1271A-E	プログラム呼出関係_CSV 専用プログラム呼び出しの最大階層(*** 1 ***)を超えたため、プログラム呼出関係の CSV ドキュメントを出力できません。	環境定義ファイル（INI ファイル）の PROGRAMCALLCSV セクションにある、CALLDEPTHMAX の値を拡張してください。
KEDL1273A-E	重複する登録集原文名の個数が 99 を超えています。 登録集原文名=*** 1 ***	登録集原文名が重複しないように、環境変数 CBLLIB に設定した登録集原文のパス名を見直してください。
KEDL1274A-E	重複する COBOL ソースファイル名の個数が 99 を超えています。COBOL ソースファイル名=*** 1 ***	COBOL ソースファイル名が重複しないように、COBOL ソースファイル名を見直してください。
KEDL1275A-W	ソース原文がシフト JIS で定義されていない文字を含んでいます。1 バイト文字は疑問符(?), 2 バイト文字は中点(・)に置き換えます。	—
KEDL1276A-E	埋め込み SQL 文の解析中に予期しないエラーが発生しました。	メッセージ中のエラー詳細を基に原因を調査してください。
KEDL1277A-W	埋め込み SQL 文に構文エラーがあるため表名を抽出できませんでした。SQL 文=(*** 1 ***)	SQL 文を見直してください。
KEDL1278A-W	SQL 識別子(*** 1 ***)が不正な文字を含んでいます。@に置き換えます。	表名やカーソル名を見直してください。
KEDL1279A-W	*** 1 ***文(SQL)に指定されたカーソル名(*** 2 ***)の定義が見つからないため表名を抽出できませんでした。	カーソル名を見直してください。

メッセージ ID	メッセージテキスト	対処
KEDL1280A-W	動的 SQL のための構文が指定されている*** 1 *** 文は CRUD 操作情報抽出の対象になりません。	—
KEDL1281A-W	動的 SQL または SQL 手続きを実行する EXECUTE 文は CRUD 操作情報抽出の対象になりません。	—
KEDL1282A-W	リスト名を使った SELECT 文や問合せは CRUD 操作情報抽出の対象になりません。	—
KEDL1283A-E	埋め込み SQL 文の解析中に予期しないエラーが発生しました。詳細メッセージ= (*** 1 ***)	詳細メッセージを基にエラーの原因を調査してください。
KEDL1284A-W	SQL 識別子(*** 1 ***)の長さが 31 文字を超えています。先頭の 31 文字だけを採用します。	—
KEDL1285A-W	SQL 文中の問合せ式(集合演算)のネストの深さが 上限を超えています。	集合演算子 (UNION, EXCEPT, INTERSECT, MINUS) を使った問合せ式のネストの深さが 255 以内となるように SQL 文を修正して再実行してください。
KEDL1286A-W	SQL 文中の副問合せのネストの深さが上限を超えています。	副問合せのネストの深さが 255 以内となるように SQL 文を修正して再実行してください。
KEDL1287A-W	SQL 文中の表結合のネストの深さが上限を超えています。	表結合の括弧のネストの深さが 255 以内となるように SQL 文を修正して再実行してください。
KEDL1288A-W	プログラム(*** 1 ***)の節(*** 2 ***)の実行順序表示の行数が上限を超えています。60,001 以上は解析しません。	—
KEDL1289A-W	プログラム(*** 1 ***)の実行順序リストの行数が上限を超えています。1,000,000 以上は解析しません。	—

(凡例)

—：対処が不要であることを示します。

(3) COBOL ソース解析のメッセージ (KEDL2000～KEDL2999) と対処

表 9-3 COBOL ソース解析のメッセージ (KEDL2000～KEDL2999) と対処

メッセージ ID	メッセージテキスト	対処
KEDL2001A-E	組み込みデータベースとの切断に失敗しました。 エラー番号=*** 1 ***,エラー詳細=*** 2 ***	メッセージ中のエラー番号およびエラー詳細を基に原因を調査してください。
KEDL2007A-E	ファイル(*** 1 ***)のオープンで予期しないエラーが発生しました。 エラー番号=*** 2 ***,エラー詳細=*** 3 ***	メッセージ中のエラー番号およびエラー詳細を基に原因を調査してください。
KEDL2021A-E	ODBC データソースに指定されたドライバが HiRDB 用ではありません。	ODBC データソース名に HiRDB のドライバを指定して再実行してください。

メッセージID	メッセージテキスト	対処
KEDL2022A-E	他の COBOL2002 Professional Tool Kit のツールが組み込みデータベースを使用中です。	データベースを使用している DB ユティリティ、または開発マネージャを終了したあとで、COBOL ソース解析を再実行してください。cblcsa コマンドがデータベースを使用している場合は、コマンドの実行が終わったあとで、COBOL ソース解析を再実行してください。
KEDL2025A-E	データベースが COBOL2002 Professional Tool Kit の組み込みデータベースではありません。	組み込みデータベースをセットアップし直して再実行してください。 組み込みデータベースのセットアップについては、マニュアル「COBOL2002 Professional 製品 導入ガイド」を参照してください。
KEDL2026A-E	他のユーザが組み込みデータベースを使用中です。	ターミナルサービス接続で、ほかのユーザが COBOL2002 Professional Tool Kit を使い終わるまで待ってから、COBOL ソース解析を起動してください。
KEDL2028A-E	本製品の前提である COBOL2002 製品がインストールされていません。 COBOL2002 製品をインストールしてから、再起動してください。	前提の COBOL2002 Net Developer がインストールされていません。COBOL2002 Net Developer をインストールしてから、COBOL2002 Professional Tool Kit を使用してください。
KEDL2029A-E	本製品の前提と異なるバージョンの COBOL2002 製品がインストールされています。 前提バージョンの COBOL2002 製品をインストールし直してから、再起動してください。	COBOL2002 Professional Tool Kit と異なるバージョン/リビジョンの COBOL2002 Net Developer がインストールされています。同じバージョン/リビジョンの COBOL2002 Net Developer をインストールしてから、COBOL2002 Professional Tool Kit を使用してください。
KEDL2030A-E	ccbl2002 コマンドが見つかりません。 COBOL2002 を再インストールしてください。	COBOL2002 Net Developer を開始するコマンド (ccbl2002 コマンド) がありません。COBOL2002 Net Developer を再インストールしてください。
KEDL2033A-E	プロジェクトにソースファイルが登録されていません。	プロジェクトにソースファイルを登録してから再実行してください。
KEDL2035A-E	プロジェクト数の上限を超えました。	不要なプロジェクトを削除してから、再度プロジェクトを新規作成してください。
KEDL2036A-E	同じ名前のプロジェクトがすでに存在します。	別の名前を指定するか、既存のプロジェクト名を削除または変更してから、再度新しいプロジェクト名を指定してください。
KEDL2037A-E	組み込みデータベースは COBOL2002 Professional Tool Kit *** 4 ***で作成されたデータベースではありません。DB ユティリティコマンドで組み込みデータベースを初期化して、再度、環境を設定してください。	DB ユティリティコマンドでデータベースを初期化してから再起動してください。

メッセージID	メッセージテキスト	対処
KEDL2038A-E	プロジェクト名に @'¥/:*?"<> ; および空白文字は使えません。	@'¥/:*?"<> ; 空白文字を使わないプロジェクト名を指定してください。
KEDL2039A-E	プロジェクト名は 30 文字以内で指定してください。	プロジェクト名は 30 文字以内で指定してください。
KEDL2041A-E	Windows の環境変数 TEMP が定義されていません。 Windows の環境変数 TEMP の定義を確認してください。	Windows の環境変数 TEMP を設定してください。
KEDL2042A-E	環境定義ファイル (CBLANALYZER.INI) に指定された解析ログ出力先フォルダのパス名が見つかりません。環境定義ファイルを修正して再起動してください。	ANALYZERLOGFILE セクションの LOGFOLDER パラメータに正しいフォルダのパス名を指定して再起動してください。
KEDL2043A-E	環境定義ファイル (CBLANALYZER.INI) に指定された解析ログ出力先フォルダのパス名が長すぎます。環境定義ファイルを修正して再起動してください。	ANALYZERLOGFILE セクションの LOGFOLDER パラメータに、長さが上限を超えていないフォルダのパス名を指定して再起動してください。上限は開発マネージャのソース解析プロジェクト使用の場合で 238 文字、cblcsa コマンドの場合で 190 文字です。cblcsa コマンドの場合は絶対パス名が 190 文字以下のフォルダのパス名を -LogFolder オプションに指定することで回避できます。
KEDL2044A-E	解析ログファイル (** 1 **) に読み書きの権限がありません。解析ログファイルまたは環境定義ファイルに指定した解析ログ出力先フォルダのアクセス権限を確認してください。	ANALYZERLOGFILE セクションの LOGFOLDER パラメータに指定したフォルダに、ログ出力に必要なアクセス権がない場合は、アクセス権を持ったフォルダのパス名を指定して再起動してください。メッセージの解析ログファイルにアクセス権がないときはアクセス権を付与して再起動してください。
KEDL2045A-E	環境定義ファイル (CBLANALYZER.INI) に指定された操作ログ出力先フォルダのパス名が見つかりません。環境定義ファイルを修正して再起動してください。	OPERATIONLOGFILE セクションの LOGFOLDER パラメータに正しいフォルダのパス名を指定して再起動してください。
KEDL2046A-E	環境定義ファイル (CBLANALYZER.INI) に指定された操作ログ出力先フォルダのパス名が長すぎます。環境定義ファイルを修正して再起動してください。	OPERATIONLOGFILE セクションの LOGFOLDER パラメータに、長さが上限を超えていないフォルダのパス名を指定して再起動してください。上限は開発マネージャのソース解析プロジェクト使用の場合で 238 文字、cblcsa コマンドの場合で 190 文字です。cblcsa コマンドの場合は絶対パス名が 190 文字以下のフォルダのパス名を -LogFolder オプションに指定することで回避できます。
KEDL2047A-E	操作ログファイル (** 1 **) に読み書きの権限がありません。操作ログファイルまたは環境定義ファイルに指定した操作ログ出力先フォルダのアクセス権限を確認してください。	OPERATIONLOGFILE セクションの LOGFOLDER パラメータに指定したフォルダに、ログ出力に必要なアクセス権がない場合は、アクセス権を持ったフォルダのパス名を指定して再起動してください。

メッセージ ID	メッセージテキスト	対処
		動してください。メッセージの操作ログファイルにアクセス権がない場合はアクセス権を付与して再起動してください。
KEDL2048A-E	環境定義ファイル (CBLANALYZER.INI) の分布情報一覧の分布範囲指定が不正です。環境定義ファイルを修正して再起動してください。	分布情報一覧関連のセクションの RANGEnn パラメータに指定された範囲指定が次のどれかに該当しないかを確認し、該当する場合は修正して再起動してください。 <ul style="list-style-type: none"> 最小値または最大値が整数値でない。 最小値または最大値が負値である。 最大値より最小値の方が大きい。
KEDL2049A-E	-LogFolder オプションで指定されたフォルダに解析ログまたは操作ログの出力に必要なアクセス権がありません。	-LogFolder オプションにログ出力に必要なアクセス権のあるフォルダのパス名を指定して再起動してください。
KEDL2050A-E	環境変数 TEMP に指定されたフォルダ名が不正です。	正しいフォルダ名を指定してください。
KEDL2051A-E	環境変数 TEMP に指定されたフォルダが見つかりません。	存在するフォルダの名称を指定してください。
KEDL2052A-E	環境変数 TEMP に指定されたフォルダ名が長すぎます。	指定されたフォルダに対応する絶対パス名が 180 文字以下となるようにフォルダのパス名を指定してください。
KEDL2053A-E	99999 個を超えるソースファイルは追加できません。	すでに登録されている COBOL ソースファイルで削除できるものを検討して、削除してください。削除できるファイルがなければ、プロジェクトを複数に分割してください。
KEDL2054A-W	プログラム名を変換しました。変換前 (** 1 **), 変換後 (** 2 **)	—
KEDL2055A-E	組み込みデータベースとの接続に失敗しました。詳細メッセージ= (** 1 **)	詳細メッセージを基にエラーの原因を調査してください。
KEDL2056A-E	組み込みデータベースの処理中にエラーが発生しました。詳細メッセージ= (** 1 **)	詳細メッセージを基にエラーの原因を調査してください。
KEDL2057A-E	ソースファイル " ** 1 ** " の読み込みに失敗しました。詳細メッセージ= (** 2 **)	アクセス権がない場合はアクセス権を付与してください。アクセス権の問題ではない場合は、詳細メッセージを基に原因を調査してください。
KEDL2058A-E	ソースフォルダ " ** 1 ** " またはその下位フォルダにアクセス権がありません。詳細メッセージ= (** 2 **)	指定したパスおよびフォルダ名を修正して再実行してください。または、アクセス権を修正して再実行してください。
KEDL2059A-E	ソース解析オプションファイルの読み込みに失敗しました。詳細メッセージ= (** 1 **)	アクセス権がない場合はアクセス権を付与してください。アクセス権の問題ではない場合は、詳細メッセージを基に原因を調査してください。
KEDL2060A-E	ソース解析オプションファイルの XML の解析に失敗しました。詳細メッセージ= (** 1 **)	ソース解析オプションファイルの XML が正しいかどうかを確認してください。

メッセージID	メッセージテキスト	対処
KEDL2061A-E	ソース解析オプションファイルの書き込みに失敗しました。詳細メッセージ= (** 1 **)	書き込み権限がない場合は書き込み権限を付与してください。書き込み権限の問題ではない場合は、詳細メッセージを基に原因を調査してください。
KEDL2062A-E	ソース解析ソースリストファイルの読み込みに失敗しました。詳細メッセージ= (** 1 **)	アクセス権がない場合はアクセス権を付与してください。アクセス権の問題ではない場合は、詳細メッセージを基に原因を調査してください。
KEDL2063A-W	ソースファイル " ** 1 **" はすでにプロジェクトに登録されています。	—
KEDL2064A-W	ソースファイル " ** 1 **" はプロジェクトに登録されていません。	—
KEDL2065A-E	ソース解析オプションファイルのオプション (** 1 **) の設定値 (** 2 **) が不正です。	正しいソース解析オプションファイルを指定して再実行してください。
KEDL2066A-E	環境変数 ** 1 ** に指定されたフォルダ名が不正です。	環境変数に指定したフォルダ名を修正して再実行してください。
KEDL2067A-E	環境変数 ** 1 ** に指定されたフォルダ名が見つかりません。	環境変数に指定したフォルダ名を修正して再実行してください。
KEDL2068A-E	ソース解析オプションファイルの拡張子は「.xml」でなければなりません。	ファイルの拡張子を xml に変更してください。
KEDL2069A-W	ファイル " ** 1 **" は COBOL ソースファイルの拡張子ではないため登録されませんでした。	タイプミスの場合は正しい COBOL ソースファイル名に変更してください。タイプミスでない場合は、環境変数 CBLFIX や CBLFREE に拡張子を登録して再実行してください。
KEDL2070A-E	ソース解析オプションファイルのフォルダ " ** 1 **" が存在しません。	存在するフォルダのファイルパス名を指定して再実行してください。
KEDL2071A-W	操作ログファイル (** 1 **) の書き込みに失敗しました。出力先を " ** 2 **" に切り替えます。詳細メッセージ= (** 3 **)	再実行してください。解析生成中は操作ログファイルをエディタで開かないでください。
KEDL2072A-E	操作ログファイル (** 1 **) の書き込みに失敗しました。処理を中止します。詳細メッセージ= (** 2 **)	ハードディスクに空き容量があるかどうかを確認してください。
KEDL2073A-E	登録可能なソースファイルが指定されていません。	正しいソースファイルを指定して再実行してください。
KEDL2074A-E	環境定義ファイル(CBLANALYZER.INI)に分布範囲指定(RANGE01～RANGE10)がすべて指定されていない分布情報セクションがあります。環境定義ファイルを修正して再起動してください。	環境定義ファイルを修正して再起動してください。
KEDL2075A-E	ソース解析ソースリストファイル中に絶対パスでないソースファイル名が指定されています。	ソース解析ソースリストファイル中のソースファイル名をすべて絶対パス名に変えて再実行してください。

メッセージID	メッセージテキスト	対処
KEDL2076A-E	環境定義ファイル(*** 1 ***)が見つかりません。	環境定義ファイルを作成してメッセージの示すパスに格納してください。 環境定義ファイルについては、「 10.10 COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) 」を参照してください。
KEDL2077A-W	*** 1 *** に合致する COBOL ソースファイルは見つかりません。	正しいソースファイルのパターンを指定して再実行してください。
KEDL2078A-W	ソースファイル *** 1 *** は、パス名がセミコロン(;), アポストロフィ(')またはカンマ(,)を含んでいるため登録されませんでした。	正しいソースファイルを指定して再実行してください。
KEDL2079A-E	ドキュメントの出力先フォルダ *** 1 *** が作成できません。詳細メッセージ=(*** 2 ***)	書き込み権限がない場合は書き込み権限を付与してください。書き込み権限の問題ではない場合は、メッセージ中の詳細情報を基に原因を調査してください。
KEDL2080A-E	スタイルシートの生成に失敗しました。詳細メッセージ=(*** 1 ***)	書き込み権限がない場合は書き込み権限を付与してください。書き込み権限の問題ではない場合は、メッセージ中の詳細情報を基に原因を調査してください。
KEDL2082A-E	解析生成オプション (*** 1 ***)の指定値の長さが上限を超えています。	解析生成オプションの値を 32,000 バイト以内に變更して再実行してください。
KEDL2083A-E	組み込みデータベースの処理中にエラーが発生しました。解析／生成を中止します。詳細メッセージ=(*** 1 ***)	詳細メッセージを基にエラーの原因を排除した上で、-Analyzer ALL オプションを使用して、すべての COBOL ソースファイルを再解析してください。
KEDL2084A-E	埋め込み SQL 文の CRUD 操作情報の書き込み中にエラーが発生しました。詳細メッセージ=(*** 1 ***)	詳細メッセージを基にエラーの原因を調査してください。
KEDL2100A-E	プログラムの実行がキャンセルされました。	バッチコマンドを再実行してください。
KEDL2308A-E	プロジェクト名 *** 1 *** がデータベース内に見つかりません。	プロジェクト名を確認してください。

(凡例)

－：対処が不要であることを示します。

(4) COBOL ソース解析のメッセージ (KEDL3000～KEDL3999) と対処

表 9-4 COBOL ソース解析のメッセージ (KEDL3000～KEDL3999) と対処

メッセージID	メッセージテキスト	対処
KEDL3001A-E	ファイル(*** 1 ***)のオープンに失敗しました。エラー番号=*** 2 ***	メッセージ中のエラー番号を基に原因を調査してください。

メッセージ ID	メッセージテキスト	対処
KEDL3002A-E	ファイル(*** 1 ***)の読み込みに失敗しました。エラー番号=*** 2 ***	メッセージ中のエラー番号を基に原因を調査してください。
KEDL3003A-E	ファイル(*** 1 ***)の書き込みに失敗しました。エラー番号=*** 2 ***	メッセージ中のエラー番号を基に原因を調査してください。
KEDL3004A-E	ファイル(*** 1 ***)の情報取得に失敗しました。エラー番号=*** 2 ***	メッセージ中のエラー番号を基に原因を調査してください。
KEDL3005A-E	ファイル(*** 1 ***)のクローズに失敗しました。エラー番号=*** 2 ***	メッセージ中のエラー番号を基に原因を調査してください。
KEDL3006A-E	メモリ不足が発生しました。エラー番号=*** 1 ***	メッセージ中のエラー番号を基に原因を調査してください。
KEDL3007A-W	プログラム名の長さが 160 バイトを超えています。先頭の 160 バイトだけを採用します。、行番号 (*** 1 ***) ,COBOL ソースファイル名 (*** 2 ***)	プログラム名の長さが 160 バイトを超えています。プログラム名の長さを 160 バイト以内に変更してください。
KEDL3008A-E	ソース解析情報ファイルのファイル形式バージョン番号が不一致のため、解析を続行できません。COBOL ソース解析の番号=*** 1 ***,ソース解析情報ファイルの番号=*** 2 ***	COBOL ソース解析が前提としている COBOL2002 をインストールしてください。
KEDL3009A-E	論理エラーが発生しました。エラー情報=*** 1 ***	当社保守員に連絡してください。

注
 メッセージ中のエラー番号の内容については、「[9.3.4 システムエラー番号](#)」を参照してください。

(5) COBOL ソース解析のメッセージ (KEDL9000～KEDL9999) と対処

表 9-5 COBOL ソース解析のメッセージ (KEDL9000～KEDL9999) と対処

メッセージ ID	メッセージテキスト	対処
KEDL9001A-E	COBOL ソース解析でエラーが発生しました。処理を中止します。 *** 1 *** ** 2 *** ** 3 *** ** 4 ***	メッセージ中の詳細情報を基に原因を調査してください。

9.3.4 システムエラー番号

KEDL3000～KEDL3999 のメッセージテキストの後ろに出力されるエラー番号について説明します。

表 9-6 システムエラー番号とエラー内容の対応

システムエラー番号	エラーの内容
2	該当するファイルまたはディレクトリがありません。
5	入出力エラーが発生しました。

システムエラー 番号	エラーの内容
12	メモリが不足しています。
13	アクセス許可が拒否されました。
17	ファイルが存在します。
28	デバイスに空き領域がありません。

9.4 そのほかの方法から探す

事例およびメッセージの内容を確認してもトラブルが解決しない場合、次の方法で対処してください。

- COBOL2002 でコンパイルして、構文的なエラーや警告がないことを確認する。
- COBOL ソース解析の対象か、および制限事項に該当しないかを確認する。
対象については、「[1.3 解析できる COBOL ソースファイル](#)」を、制限事項については、「[10. COBOL ソース解析の仕様範囲のリファレンス](#)」を参照してください。
- エラーの個所をコメントにする。
エラーの個所をコメントにして、COBOL ソース解析を再実行してください。
- ウィルスチェックを実行していないか確認する。
ウィルスチェックを実行しているときに、COBOL ソース解析を実行すると、起動できない、出力結果がエラーになるなどの影響が出ることがあります。ウィルスチェックを実行していない状態で、COBOL ソース解析を再実行してください。

そのほか、動作がふだんよりも遅いと感じたとき、COBOL ソース解析の組み込みデータベースを再編成する必要があります。組み込みデータベースの再編成については、マニュアル「COBOL2002 Professional 製品 導入ガイド」のエラー時の対処を参照してください。

これらの対処を実施してもトラブルが解決しない場合、保守員に連絡してください。保守員に連絡するときには、次のフォルダおよびファイルを採取して、現象の詳細と発生日時をあわせてご連絡ください。

- *COBOL2002 Professional Tool Kit* のインストール先フォルダ¥hirdb_e_CT0¥SPOOL
- *COBOL2002 Professional Tool Kit* のインストール先フォルダ¥hirdb_e_CT0¥UXPLDIR¥spool
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥COBOL Analyzer
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥COBOLAnalyzeLog.tag
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥COBOLAnalyzeLog2.tag
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥COBOLControlLog.txt
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥COBOLControlLog2.txt
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥pderr1.trc
- *Windows* の環境変数 *TEMP* で定義されているフォルダ¥pderr2.trc

10

COBOL ソース解析の仕様範囲のリファレンス

COBOL ソース解析では、COBOL2002 の規格に沿ってコーディングされた COBOL ソースファイルを解析の対象としていますが、解析できない COBOL ソースファイルがあります。COBOL ソース解析の仕様範囲について説明します。

10.1 解析情報の上限値

COBOL ソース解析に登録できる COBOL ソースファイルの最大数、プログラム名の長さなど、COBOL ソース解析を使用するときの上限値について説明します。

表 10-1 COBOL ソース解析を使用するときの上限値

項番	対象	上限値
1	登録できる COBOL ソースファイルの数	99,999 個
2	ソースファイル名の長さ	Windows で作成できるパス名長 (259 バイト)
3	登録集原文フォルダのパス名長	Windows で作成できるパス名長 (259 バイト)
4	プロジェクトの解析オプションのコンパイラ環境 変数 CBLLIB に指定できる文字列の長さ	32,000 バイト
5	プログラム名の長さ	31 文字 (先頭の 31 文字だけ採用) 定数指定の場合は 160 バイト
6	プログラムコメント行の文字列の長さ	65 バイト
7	節名の長さ	31 文字
8	節コメント行の文字列の長さ	65 バイト
9	COPY 文または INCLUDE 文の入れ子	20 階層の入れ子
10	プログラム階層図 (HTML ファイル) の階層	10 階層
11	セクション階層図 (HTML ファイル) の階層	10 階層
12	「解析生成オプション設定」画面の「生成ドキュメント」タブで指定できる出力先フォルダのパス名の長さ	167 文字
13	1 つの COBOL ソースファイル内の、次に示す項目の個数 • プログラム • 部見出し • 節名 (「{PROCEDURE_START}」も含む) • ファイル管理記述項 • ファイル入出力文 (READ, WRITE, REWRITE, DELETE) • 登録集原文 (未登録の登録集原文を含む) • COPY 文 (入れ子になっている COPY 文を含む) • PERFORM 文の呼出先情報 • CALL 文 • CRUD 操作情報	32,767 個 1 つの COBOL ソースファイル内での、それぞれの項目の定義個数
14	重複プログラムの数	999 個

項番	対象	上限値
15	プロジェクトの数	1,000 個
16	重複する登録集原文名の数	99 個
17	重複する COBOL ソースファイル名の数	99 個
18	1 つのプログラム内の、01, 77 レベルのデータ項目の個数	32,767 個
19	1 つの COBOL ソースファイル内の、次に示す項目の定義個数の合計（各種解析情報数の合計） <ul style="list-style-type: none"> • ソース行情報（約 256 行で 1 個） • プログラム • 部見出し • 節名 • ファイル管理記述項 • ファイル入出力文 • 登録集原文 • COPY 文 • 01, 77 レベルデータ項目（記述項情報） • 01 レベル集団項目（構成情報）※ • 仮引数 • PERFORM 文の呼出先情報 • CALL 文 • 実引数 • 報告書、画面、および通信記述名 • デッドコード情報 • 制御フロー情報 	400,000 個
20	[実行順序表示] 画面の実行順序ツリー（左ペイン）の行数	60,000 行
21	実行順序リストの行数	999,999 行

注※

解析オプション[レベル番号 02～49,66 のデータ項目の情報を抽出する]をチェックしていない場合、「01 レベル集団項目（構成情報）」は合計値に含まれません。

10.2 制限がある構文

COBOL ソース解析で制限がある COBOL2002 の構文および機能について説明します。

制限の対象になる構文を含む COBOL ソースファイルを解析したときに、ソース解析情報が生成されるとエラーメッセージは出力されます。ソース解析情報が生成されないときは、エラーメッセージは出力されません。

COBOL ソース解析で制限になる構文を次の表に示します。

表 10-2 COBOL ソース解析で制限になる構文

項番	制限の対象	COBOL ソース解析の動作
1	次に示す文のファイルアクセス情報 <ul style="list-style-type: none">• CLOSE• MERGE• OPEN• RELEASE• RETURN• START• SORT• USE• UNLOCK	解析を継続します。 ソース解析情報に生成されません。 ファイルアクセス情報の対象となる文を次に示します。 <ul style="list-style-type: none">• READ• WRITE• REWRITE• DELETE
2	「データベース操作シミュレーション機能」の INCLUDE	解析を継続します。 表名情報を出力する「SQL データ操作機能」として認識します。
3	アドレス名および指標名	解析を継続します。 データ項目として認識しません。
4	節がコメントで終了している COBOL ソースファイル	解析を継続します。 節コメント行の文字列を取得できません。
5	ASSIGN 句の定数に、Windows のフォルダ名に使用できない文字列（¥/:*?<> など）を指定している	解析を継続します。 生成結果は不正になることがあります。
6	利用者定義語以外の COBOL の語に全角文字を使用	全角文字を使用するとソースファイルが登録されないことがあります。また、ソースが登録できても予約語が認識されないで、ソース解析情報が正しく生成されないことがあります。
7	PROGRAM-ID を行の先頭から始めていないとき	プログラムの登録時にプログラム名を取得できないため、エラーになります。
8	ENTRY 文で定義した入口名	ENTRY 文で定義した入口名は、解析/生成の対象にしません。
9	数字編集項目の小数部に浮動挿入文字が存在	数字編集項目の小数部に、次に示す浮動挿入文字があると、[プログラムデータ] 画面の「ファイル節、作業場所節、連絡節のデータ定義」属性欄の浮動挿入文字は「9」と表示されます。 <ul style="list-style-type: none">• ¥

項番	制限の対象	COBOL ソース解析の動作
		<ul style="list-style-type: none"> • + • - <p>正しい属性は、ソース表示のデータ定義を参照してください。 (例) 「PIC ¥.¥¥」は「¥.99」と表示 「PIC +.++」は「+.99」と表示 「PIC -.--」は「-.99」と表示</p>
10	報告書節のデータ項目数と記述項数	報告書節がある場合、報告書節のデータ項目は、[プログラム詳細] 画面の「プログラム概要」のデータ項目数と記述項数に加算されません。
11	節見出しで始まらない手続き部先頭の節	<p>次に示す節を仮定して解析を継続します。</p> <ul style="list-style-type: none"> • 節名：{PROCEDURE_START} • 節の開始行：手続き部の最初の実行文または段落の開始行 <p>COBOL ソース解析時の注意事項を次に示します。</p> <ul style="list-style-type: none"> • 節{PROCEDURE_START}の節コメント行は抽出できません。 • -SQL,XDM コンパイラオプションが指定されている場合、埋め込み SQL 文 (EXEC SQL~END-EXEC) は、一律、覚え書きの実行文として扱われます。このため、実行文でない DECLARE CURSOR 文や埋め込み例外宣言も、節見出しのない手続き部先頭に記述されていれば、節 {PROCEDURE_START}の先頭として扱われます。
12	同一プログラム内で 01 レベルデータ項目の名前 (ファイルレコードの場合はファイル名で修飾した名前) が重複 (参照されていない場合)	最初のデータ項目の名前はそのまま解析し、2 番目以降のデータ項目の名前には"(n)"の重複番号 (n=2~) を付加して解析を継続します。
13	特殊名段落で CURRENCY SIGN 句を指定した場合の PICTURE 文字列中の通貨編集用文字	<p>PICTURE 文字列中の通貨編集用文字は常に通貨記号(¥)で表示されます。</p> <p>(例) 特殊名段落で CURRENCY SIGN IS '\$'が指定されている場合 「PIC \$\$\$,\$\$\$,\$\$\$」は「¥¥¥,¥¥¥,¥¥¥」と表示されます。</p>
14	定数の連結式	<p>定数の連結式は、順番に評価して 1 つの定数に変換します。</p> <p>(例 1)</p> <p>'A' & 'B' & 'C' → 'ABC' 'A' & X'42' & 'C' → 'ABC' X'41' & 'B' & X'43' → 'ABC' '41' & X'42' & X'43' → X'414243' N'あい' & N'う' → N'あいう' NX'82a082a2' & N'う' → NX'82a082a282a4' N'あい' & NX'4142' → N'あい AB'</p> <p>ただし、連結後の英数字定数や日本語定数が非印字可能文字 (制御文字や非シフト SJIS 文字) を含む場合は、それぞれ 16 進英数字定数や 16 進日本語定数に変換します。また、-JPN コンパイ</p>

項番	制限の対象	COBOL ソース解析の動作
		<p>ラオプション指定時に日本語の定数（日本語定数および 16 進日本語定数）と、英数字の定数（英数字定数および 16 進英数字定数）を連結した場合、連結後の日本語定数や 16 進日本語定数のバイトサイズが奇数になる場合は、それぞれ英数字定数、16 進英数字定数に変換します。</p> <p>（例 2）連結後の英数字定数や日本語定数が非印字可能文字を含む場合 'ABC' & X'00' → X'41424300' N'あいう' & NX'0020' → NX'82a082a282a40020'</p> <p>（例 3）日本語の定数と英数字の定数を連結した場合 'あいう' & 'AB' → N'あいう AB' N'あいう' & 'ABC' → 'あいう ABC' NX'82a082a282a4' & 'AB' → NX'82a082a24142' NX'82a082a282a4' & 'ABC' → X'82a082a2414243'</p>
15	連絡節に指定された LIMIT 指定のない動的長基本項目のサイズ	連絡節に指定された LIMIT 指定のない動的長基本項目は実行時までサイズが決定しません。このため、文字列長を 0 として、動的長英数字項目の場合は 1、動的長日本語項目の場合は 2 がサイズとして表示されます。

10.3 COBOL2002 の言語仕様の解析結果

COBOL2002 の言語仕様のうち、制限がある項目と COBOL ソース解析の動作の関係を次の表に示します。

表 10-3 COBOL2002 の言語仕様で制限がある項目と COBOL ソース解析の動作の関係

項番	制限の対象	COBOL ソース解析の動作
1	TYPE 句	解析を継続します。 TYPE 句の型名で定義された属性とサイズで表示します。また、エラーメッセージは出力しません。
2	SAME AS 句	解析を継続します。 SAME AS 句で定義された属性とサイズで表示します。また、エラーメッセージは出力しません。
3	局所場所節	解析を継続します。 エラーメッセージは出力しません。 LOCAL-STORAGE SECTION.のデータ名は、仕様書のプログラムデータの作業場所節のデータ定義名一覧と、プログラムデータの連絡節のデータ定義名一覧には出力しません。
4	利用者定義関数の定義 (FUNCTION-ID)	無視して、解析を継続します。 エラーメッセージは出力しません。
5	利用者定義関数の呼び出し	無視して、解析を継続します。 エラーメッセージは出力しません。
6	TYPEDEF 句	無視して、解析を継続します。 エラーメッセージは出力しません。
7	オブジェクト指向	オブジェクト指向機能の文法を使用している COBOL ソースファイルは登録または解析できません。また、オブジェクト指向機能の文法の有効行数やコメント数はカウントしません。COBOL ソースファイルの解析時にコンパイルエラーになった場合は、解析処理を中止します。 COBOL ソース解析でサポートしないオブジェクト指向機能の定義を次に示します。 <ul style="list-style-type: none">クラス定義ファクトリ定義 (そのメソッド定義群も含む)インスタンス定義 (そのメソッド定義群も含む)メソッド定義インタフェース定義 (そのメソッド原型も含む) また、次に示す項目は、ソース解析情報に出力できますが、情報が欠落することがあります。 <ul style="list-style-type: none">リポジトリ段落：ソース表示だけ表示されます。オブジェクトプロパティ：ソース表示だけ表示されます。INVOKE 文：ソース表示だけ表示されます。
8	動的長基本項目	作業場所節データ定義一覧と連絡節データ定義一覧には、LIMIT の指定値 + 1 のサイズを持つ、英数字項目または日本語項目として表示します。エラーメッセージは出力しないで、解析/生成処理を継続します。

項番	制限の対象	COBOL ソース解析の動作
9	日本語集団項目	作業場所節データ定義一覧と連絡節データ定義一覧には、英数字の集団項目として表示します。 エラーメッセージは出力しないで、解析/生成処理を継続します。
10	句や文の指定での覚え書き	句や文の指定で、COBOL2002 の言語仕様で覚え書きと見なしているものは、ドキュメントに反映されません。 (例) ASSIGN TO SYS001-DA-DK-S SYS002-DA-DK-S 上記の「SYS002-DA-DK-S」は覚え書きと見なすため、ソース解析情報には反映されません。
11	END PROGRAM	END PROGRAM と以降のコメント行はステップ数にカウントしません。
12	プログラム名段落 (PROGRAM-ID)	先頭に書かれたプログラムが次のどちらかの条件に該当する COBOL ソースファイルをプロジェクトに登録すると、プログラム名が取得できないため登録できなかったことを示す警告メッセージが出力されて登録できません。 <ul style="list-style-type: none"> プログラム名段落の予約語 PROGRAM-ID を登録集原文ファイルに記述して COPY 文で取り込んでいる。 プログラム名段落のプログラム名が英数字定数で記述されている場合に、行継続を使用して複数行にわたって記述されているか、または連結式を使用して記述されている。

「解析を中止する」とは、対象の COBOL ソースファイルだけの解析を中止することを意味します。該当する COBOL ソースファイルの解析の中止後、ほかの COBOL ソースファイルの解析に移行します。

10.4 COBOL ソース解析で利用できるコンパイラオプション

COBOL2002 のコンパイラオプションと COBOL ソース解析での使用可否について次の表に示します。コンパイラオプションの指定方法については、マニュアル「COBOL2002 ユーザーズガイド」を参照してください。

カテゴリの意味は次のとおりです。

最終生成物の種類（プロジェクトの種類）

開発マネージャで最終生成物の種類（プロジェクトの種類）を設定するコンパイラオプションです。最終生成物とは、コンパイラが最終的に生成する実行可能ファイル、DLL、または標準ライブラリのことです。

製品連携

他製品との連携を設定するコンパイラオプションです。

実行

実行時の動作を設定するコンパイラオプションです。

最適化

プログラムの最適化を設定するコンパイラオプションです。

デバッグ

デバッグを設定するコンパイラオプションです。

リンク

リンクを設定するコンパイラオプションです。

規格

規格仕様のチェックを設定するコンパイラオプションです。

移行

他システムとの移行を設定するコンパイラオプションです。

リスト出力

リスト出力を設定するコンパイラオプションです。

その他

その他の項目を設定するコンパイラオプションです。

表 10-4 COBOL2002 のコンパイラオプションと COBOL ソース解析での使用可否

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース解析での使用
1	-Main,{System V3} ファイル名	最終生成物の種類	先頭の外部プログラムを主プログラムと仮定し、コマンドの引数を受け取ります。	×
2	-Dll,{Stdcall Cdecl} -Dll	最終生成物の種類	DLL 形式を指定します。	×

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース 解析での使用
	-noDll			
3	-SQL,{XDM ODBC [,NoCont]} -noSQL	製品連携	埋め込み SQL 文を XDM/RD または ODBC インタフェース機能で使えるようにします。	×
4	-SQLDisp -noSQLDisp	製品連携	埋め込み SQL 文に用途 (USAGE 句) が表示用 (DISPLAY) のデータ項目を指定できるようにします。	×
5	-RDBTran -noRDBTran	製品連携	COMMIT 文/ROLLBACK 文を HiRDB による索引編成ファイルに対して適用します。	○
6	-IsamExtend [,Zone] -noIsamExtend	製品連携	Btrieve (Pervasive.SQL) による索引編成ファイルを使用します。	○
7	-XMAP,LinePrint -noXMAP	製品連携	書式印刷機能を使用して、順編成ファイルをプリンタに出力します。	○
8	-OpenTP1 -noOpenTP1	製品連携	OpenTP1 を使用したデータコミュニケーション機能を使用できるようにします。	○
9	-SQLAutoCopy -noSQLAutoCopy	製品連携	埋め込み SQL 文を含む COBOL ソースファイルを解析できるように CBLAUTOSQLCOPY.CBL ファイルを自動的に展開します。	○※
10	-NumAccept -noNumAccept	実行	ACCEPT 文に数字項目を指定できるようにします。	○
11	-NumCsv -noNumCsv	実行	CSV 編成ファイルで、セルデータを数値として入出力できるようにします。	○
12	-MultiThread -noMultiThread	実行	マルチスレッド対応 COBOL プログラムを作成します。	○
13	-MainNotCBL -noMainNotCBL	実行	すべて副プログラムとして扱います。	×
14	-DllInit -noDllInit	実行	DLL を初期化します。	×
15	-Optimize,{0 1 2 3}	最適化	コンパイル時の最適化のレベルを指定します。	×
16	-ScreenSpeed -noScreenSpeed	最適化	画面の表示速度を重視します。	×
17	-DebugLine -noDebugLine	デバッグ	デバッグ行を有効にします。	○
18	-DebugInf [,Trace] -noDebugInf	デバッグ	異常終了時、エラー要約情報を出力します。	×

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース解析での使用
19	-DebugCompati -noDebugCompati	デバッグ	実行時に次のチェックをします。 • 添字、指標名の繰り返し回数の範囲外チェック • プログラム間整合性チェック	×
20	-DebugData [,ValueHex] -noDebugData	デバッグ	データ例外を検出します。	×
21	-TDInf -noTDInf	デバッグ	テストデバッグ情報を出力します。	×
22	-CVInf -noCVInf	デバッグ	カバレッジ情報を出力します。	×
23	-DebugRange -noDebugRange	デバッグ	添字、指標名の繰り返し回数について、次元ごとの範囲外チェックをします。	×
24	-TestCmd{,Full ,Break ,Sim}+ -noTestCmd	デバッグ	TD コマンド格納ファイルに出力する情報の種類を指定します。	×
25	-SimMain プログラム名 -noSimMain	デバッグ	主プログラムをシミュレーションします。	×
26	-SimSub プログラム名 [,…] -noSimSub	デバッグ	副プログラムをシミュレーションします。	×
27	-SimIdent -noSimIdent	デバッグ	副プログラム（一意名 CALL 文）をシミュレーションします。	×
28	-StdCall -noStdCall	リンク	stdcall 呼び出し指示ファイルを有効にします。	×
29	-StdCallFile ファイル名 -noStdCallFile	リンク	stdcall 呼び出し指示ファイル名を指定します。	×
30	-Lib,{GUI CUI}	リンク	GUI モード/CUI モードのどちらの実行時ライブラリをリンクするかを指定します。	×
31	-Compile,{CheckOnly NoLink} -noCompile	リンク	コンパイルの処理範囲を指定します。	×
32	-DefFile ファイル名 -noDefFile	リンク	生成する.def ファイル名を指定します。	×
33	-IconFile ファイル名 -noIconFile	リンク	アイコンファイル名を指定します。	×
34	-ResrcFile ファイル名 -noResrcFile	リンク	生成するリソース定義ファイル名を指定します。	×
35	-OutputFile ファイル名 -noOutputFile	リンク	生成する実行可能ファイル名、または DLL ファイル名を指定します。	×

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース解析での使用
36	-Link オプションの並び -noLink	リンク	リンクに渡すオプションを指定します。	×
37	-DynamicLink,Call -noDynamicLink	リンク	定数指定の CALL 文実行に動的リンク機能を使用するときに指定します。	×
38	-ManifestFileExt -noManifestFileExt	リンク	マニフェストファイルを実行可能ファイルや DLL に埋め込まず、外部マニフェストファイルとするときに指定します。	×
39	-StdMIA{,13 ,14}+ -noStdMIA	規格	MIA 仕様の範囲外チェックをします。	×
40	-Std85{,{High Middle Low} ,Obso ,Report}+ -noStd85	規格	JIS 仕様をチェックします。	×
41	-Std2002{,OutOfRange ,Obso ,Archaic}+ -noStd2002	規格	COBOL2002 規格仕様をチェックします。	×
42	-StdVersion,{1 2} -noStdVersion	規格	第 1 次規格/第 2 次規格の解釈でコンパイルします。	○
43	-CompatiM7 -noCompatiM7	移行	MIOS7 との互換機能を有効にします。	○
44	-CompatiV3 -noCompatiV3	移行	VOS3 COBOL85 との互換機能を有効にします。	○
45	-Compati85{,IoStatus ,Linage ,Call ,Power ,Syntax ,IDParag ,RsvWord ,NoPropagate ,All}+ -noCompati85	移行	COBOL85 互換機能を有効にします。指定されたオプションに対して、COBOL85 と同等の動作を保証します。	○
46	-H8Switch -noH8Switch	移行	HITAC8000 シリーズの仕様でコンパイルします。	○
47	-Cblctr -noCblctr	移行	CBL-CTR 特殊レジスタを使用できるようにします。	○
48	-DigitsTrunc -noDigitsTrunc	移行	転記文で上位桁を切り捨てます。	○
49	-IgnoreLCC -noIgnoreLCC	移行	行送り制御文字を無視します。	○
50	-CmAster -noCmAster	移行	1 カラム目が「*」の行を注記行とします。	○
51	-CmDol	移行	7 カラム目が「\$」の行を注記行とします。	○

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース 解析での使用
	-noCmDol			
52	-Comp5 -noComp5	移行	COMP-5 を指定できるようにします。	○
53	-V3Spec [,CopyEased] -noV3Spec	移行	VOS3 COBOL85 に対する COBOL2002 固有の構文をチェックします。	○
54	-V3ConvName -noV3ConvName	移行	VOS3 COBOL85 からのソースファイル互 換のため、COPY 文の原文名定数中の「¥」 と「@」を変換します。	○
55	-Switch,{EBCDIC EBCDIK} [,Unprintable] [,noApplyJpnItem] -noSwitch	移行	字類条件を指定します。	○
56	-V3Rec,{Fixed Variable} -noV3Rec	移行	メインフレーム (VOS3) の固定長または可 変長レコード形式のプログラムを、VOS3 の日本語文字の扱いに合わせてコンパイルし ます。	○
57	-V3RecFCSpace -noV3RecFCSpace	移行	空白に関する機能キャラクタの扱いをメイン フレーム (VOS3) と同等にします。	○
58	-V3RecEased{,QuoteCheck ,WordCheck}+ -noV3RecEased	移行	-V3Rec オプション指定時の仕様チェックを 緩和します。	○
59	-DoubleQuote -noDoubleQuote	移行	引用符 (") を分離符と見なしてコンパイル します。	○
60	-BigEndian{,Bin ,Float}+ -noBigEndian	移行	2 進データ項目または浮動小数点データ項目 をビッグエンディアン形式で処理します。	○
61	-VOSCBL{,OccursKey ,ReportControl}+ -noVOSCBL	移行	メインフレーム互換機能を有効にします。	○
62	-PortabilityCheck{,Literal ,Numeric}+ -noPortabilityCheck	移行	移行向けチェック機能を有効にします。	○
63	-IgnoreAPPLY,FILESHARE -noIgnoreAPPLY	移行	APPLY FILE-SHARE 句を覚え書きと見な します。	○
64	-SrcList,{OutputAll CopyAll CopySup NoCopy} [,NoFalsePath] [,DataLoc] -noSrcList	リスト出力	コンパイルリストを出力します。	×

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース解析での使用
65	-ErrSup{I ,W}+ -noErrSup	リスト出力	I レベルまたは W レベルエラーの出力を抑止します。	×
66	-Bin1Byte -noBin1Byte	その他	1 バイトの 2 進項目を有効にします (PICTURE 句の指定で 2 桁までは 1 バイトとして扱います)。	○
67	-JPN,{Alnum V3JPN V3JPNSpace} -noJPN	その他	日本語項目の扱いを指定します。	○
68	-EquivRule,{NotExtend NotAny StdCode} -noEquivRule	その他	文字の等価規則をどう変更するか指定します。	○
69	-UscoreStart -noUscoreStart	その他	先頭がアングスコアの CALL 定数を指定できるようにします。	○
70	-BinExtend -noBinExtend	その他	2 進データ項目に指定できる初期値を拡張します。	○
71	-MinusZero -noMinusZero	その他	10 進項目で負の符号を持つゼロを正の符号を持つゼロに変換します。	○
72	-TruncCheck -noTruncCheck	その他	転記でのデータ切り捨てをチェックします。	○
73	-LowerAsUpper -noLowerAsUpper	その他	定数指定の CALL に指定された英小文字を英大文字に変換してプログラムを呼び出します。	○
74	-CBLVALUE -noCBLVALUE	その他	環境変数 CBLVALUE を有効にします。	○
75	-Repository,{Gen Sup} -noRepository	その他	リポジトリファイルの生成時、強制的に出力するか、更新しないかを指定します。	×
76	-RepositoryCheck -noRepositoryCheck	その他	同じソースファイル中の翻訳単位の定義と外部リポジトリ中の情報に相違があるかどうかをチェックします。	○
77	-Define 翻訳変数名 [=値] [,翻訳変数名 [=値] ...] -noDefine	その他	コンパイル時に有効となる、翻訳変数名とその値を定義します。	○
78	-Details -noDetails	その他	コンパイラオプションの詳細情報を出力します。	×
79	-OldForm "旧オプションの並び"	その他	旧形式 (COBOL85) のオプションを指定します。	○
80	-Help -?	その他	ccbl2002 コマンドのヘルプを出力します。	×

項番	コンパイラオプション	カテゴリ	意味	COBOL ソース 解析での使用
81	-UniObjGen -noUniObjGen	その他	シフト JIS の COBOL ソースから Unicode のオブジェクトを生成します。	×
82	-UniEndian,{Little Big} -noUniEndian	その他	シフト JIS で記述された日本語文字定数を UTF-16LE, または UTF-16BE に変換しま す。	×
83	-MaxDigits38	その他	数字項目, 数字編集項目, および数字定数に 指定できる最大桁数を 18 桁から 38 桁に拡 張します。 64bit 版 COBOL2002 で使用できます。	○
84	-IntResult,DecFloat40	その他	算術演算の中間結果の表現形式を 40 桁の 10 進浮動小数点形式とします。 64bit 版 COBOL2002 で使用できます。	○
85	-LiteralExtend,Alnum -noLiteralExtend	その他	英数字定数と定数指定のプログラム名の長さ を拡張します。	×
86	-SpaceAsZero -noSpaceAsZero	その他	外部 10 進項目中に空白文字があるとき, ゼ ロとみなして比較, 演算, 転記を実行する。	○
87	-CheckUninitData -noCheckUninitData	その他	データ項目の初期化漏れをチェックする。	○

(凡例)

- ：使用できます。
- ×
- ×

注※

COBOL2002 Professional 製品固有のコンパイラオプションです。

10.5 COBOL ソース解析で利用できるコンパイラ環境変数

COBOL2002 のコンパイラ環境変数と COBOL ソース解析での使用可否を次の表に示します。

表 10-5 COBOL2002 のコンパイラ環境変数と COBOL ソース解析での使用可否

項番	環境変数名	説明	COBOL ソース解析での使用
1	CBL_RDBSYS	HiRDB による索引編成ファイルで操作対象となるデータベースシステムの種別	○
2	CBLCOPT	ccbl コマンドに指定する旧形式のオプション列	×
3	CBLCOPT2002	ccbl2002 コマンドに指定する新形式のオプション列	○
4	CBLERRMAX	コンパイルを打ち切る S レベルのエラーの数	○
5	CBLFIX	固定形式正書法の COBOL ソースファイルの拡張子	○
6	CBLFREE	自由形式正書法の COBOL ソースファイルの拡張子	○
7	CBLLIB	登録集原文の検索フォルダ	○
8	CBLLINKER	コンパイラが使用するリンカの絶対パス名	×
9	CBLLINKINTERVAL	対象ファイルへのアクセス処理の前に空ける間隔（ミリ秒単位）	×
10	CBLMANIFESTTOOL	コンパイラが使用するマニフェストツールの絶対パス名	×
11	CBLPIDIR	プログラム情報ファイル (.cbp) の生成先フォルダ	×
12	CBLREP	リポジトリファイルの出力先のフォルダ、およびリポジトリ段落で指定した翻訳単位名を含むリポジトリファイルを検索するフォルダ	×
13	CBLRESOURCECOMPILER	コンパイラが使用するリソースコンパイラの絶対パス名	×
14	CBLSYSREP	リポジトリファイルの参照時に検索するフォルダ	×
15	CBLTAB	タブコードを空白に変換するときの空白の数	○
16	CBLVALUE	VALUE 句の指定のないデータ項目の初期値	○
17	CBLV3UNICODE	-UniObjGen オプション指定時に、-CompatV3 オプションおよび-V3Rec オプションの指定を有効としないかどうか	×
18	登録集環境変数	登録集原文の検索フォルダ	○
19	CBLFIXEDFORMLINE	固定形式正書法の 1 行の長さの制限値	×

(凡例)

- ：使用できます。
- ×

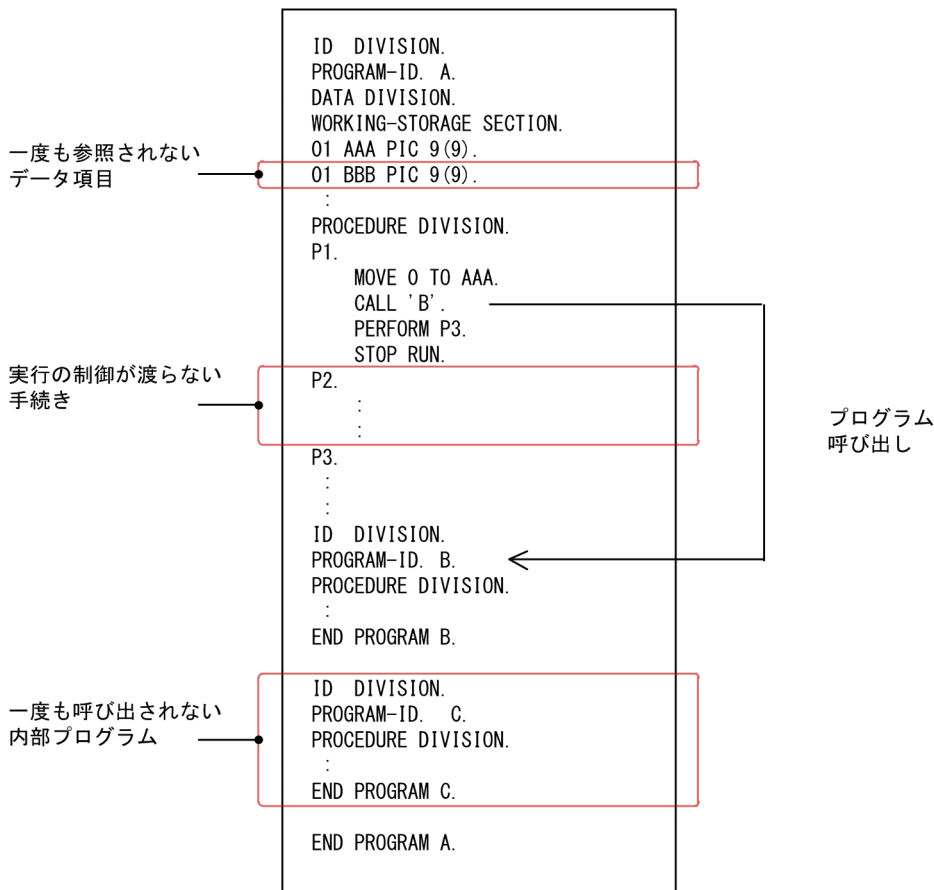
10.6 デッドコード候補の検知方法

デッドコード候補の検知方法について説明します。

デッドコードとは、プログラム実行時に実行される可能性がない文、プログラムなど、プログラムの実行結果に影響を与えないソースコードのことです。デッドコード候補から、テストが不要なプログラムの選定や、プログラムの処理の誤りを発見できます。

デッドコードの例を次の図に示します。

図 10-1 デッドコードの例



デッドコード検出の対象

COBOL ソース解析では、登録集原文を取り込んだあとの COBOL プログラム中のプログラム定義で、コンパイル対象になる行を検出対象とします。

デッドコード検出の対象でない項目

次に示すコードは検出対象外です。

- ・ 利用者定義関数、クラス、インタフェース定義
- ・ コンパイル対象にならないデバッグ行、条件翻訳結果の無効行
- ・ S レベル、U レベルエラーが発生する COBOL プログラム

- 覚え書きと見なされた文

デッドコード候補の種類

検出されるデッドコード候補の種類を次の表に示します。

表 10-6 検出されるデッドコード候補の種類

項番	デッドコード候補の種類	内容	ソース表示(COPY 展開後)時に背景をグレーにする行
1	未使用の外部プログラム	一度も呼び出されることがない外部プログラム	PROGRAM-ID の行
2	未使用の内部プログラム	一度も呼び出されることがない内部プログラム	PROGRAM-ID の行
3	未使用のデータ項目	一度も参照されていないデータ項目	データ項目の定義位置
4	未使用のファイル名、報告書名、画面名、通信記述名	一度も参照されていないファイル名、報告書名、画面名、通信記述名	報告書名、画面名、通信記述名の定義位置
5	制御が渡らない文	実行時に制御が渡ることがない文	制御が渡ることがない文の先頭行から終了行
6	制御が渡らない手続き	実行時に制御が渡ることがない手続き	制御が渡ることがない手続きの定義位置

デッドコード候補の詳細

デッドコード候補として検出される内容を次に示します。

未使用の外部プログラム

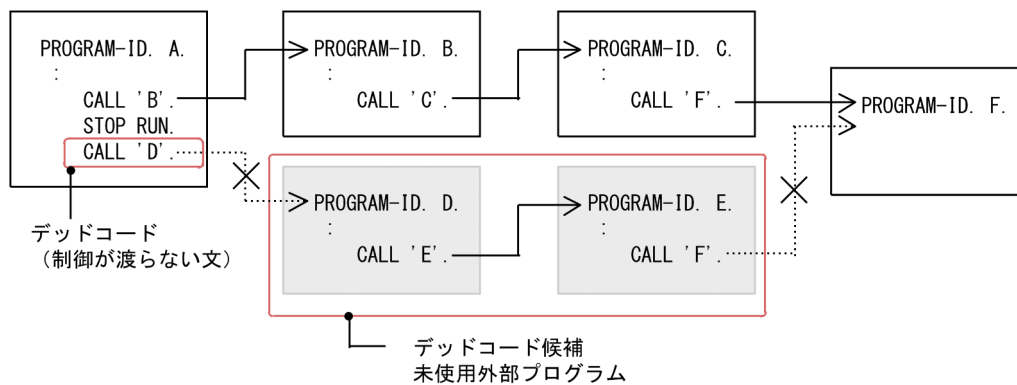
解析対象の翻訳単位間で、どのプログラムからも呼び出されていない COBOL 主プログラム以外の外部プログラムが検出されます。COBOL ソース解析では、次に示すプログラムを未使用の外部プログラムとします。

- 解析対象の翻訳単位間で、「CALL 定数」によって呼ばれることがない外部プログラム
- 「CALL 一意名」からだけ呼ばれている外部プログラムのうち、CALL 一意名に対応する呼び出し先プログラムとして解決できなかった外部プログラム
- 解析対象外の翻訳単位（他言語プログラムを含む）からだけ呼び出される外部プログラム
- 主プログラムでない、呼び出し元の CALL 文がすべてデッドコード候補である外部プログラム

(例) 未使用の外部プログラムの例

次の例では、プログラム A にプログラム D を呼ぶ CALL 文が存在するが、CALL 文がデッドコードであるためプログラム D もデッドコード候補として検出されます。また、プログラム D が呼ばれていないため、プログラム D からしか呼ばれていないプログラム E もデッドコード候補となります。プログラム F はプログラム C からの呼び出しがあるため、デッドコード候補にはなりません。

図 10-2 未使用の外部プログラムの例



注意

- 重複するプログラム名/ENTRY 文で定義した入口点の名称が存在する場合で、かつそのプログラムを呼び出す CALL 文があるときは、それらのプログラムはすべて呼ばれるものと仮定して処理されます。
- 未使用外部プログラムの場合、プログラム中のほかのデッドコードも検出します。

未使用の内部プログラム

最外プログラム中で、一度も呼び出されていない内部プログラムが検出されます。COBOL ソース解析では、次に示すプログラムを未使用の内部プログラムとします。

- プログラム内で、CALL/CANCEL 文によって呼ばれることがない内部プログラム
- 呼び出し元の CALL/CANCEL 文がすべてデッドコードである内部プログラム

プログラム中に CALL/CANCEL 一意名が存在する場合、COBOL ソース解析では内部プログラムを次のように扱います。

- CALL/CANCEL 一意名から呼ぶことができる内部プログラムは、使用される可能性がある内部プログラムとして扱われ、デッドコードではなくデッドコード候補として検出されます。
ただし、次の方法で、CALL 一意名に対応する呼び出し先プログラムとして解決された内部プログラムは、呼ばれるプログラムとして扱い、デッドコード候補から除きます。
- [解析生成オプション設定] 画面の [呼出先プログラム解決] タブに一意名呼び出し対応付けファイルを指定

なお、デッドコードの検出は、CALL 一意名に対応する呼び出し先プログラムが解決されたあとに実施します。ある内部プログラムが CALL 一意名の呼び出し先プログラムとして解決されていても、その CALL 一意名がデッドコード（制御が渡らない文）で、呼び出し元の CALL/CANCEL 文がすべてデッドコードの場合、その内部プログラムは呼ばれるプログラムとは扱われないで、未使用の内部プログラムとして検出されます。

一意名呼び出し対応付けファイル (idc ファイル) の作成方法については、「[付録 B.1 一意名呼び出し対応付けファイル](#)」を参照してください。

未使用のデータ項目

データ部中の作業場所節，局所場所節，連絡節，画面節で定義している 01，77，レベルのデータ項目定義で，プログラム中で 1 回も参照されていない項目が検出されます。COBOL ソース解析では，定義したデータ項目が集団項目である場合，集団項目に属するすべてのデータ項目が未使用である場合だけ未使用のデータ項目として検出されます。また，デッドコード候補の処理からだけ参照されているデータ項目も，未使用のデータ項目として検出されます。

注意事項

- COBOL ソース解析では，ファイル節で定義しているレコード記述項および報告書節で定義している報告集団記述項が未使用かどうかは検出されません。これらが未使用かどうかは，未使用のファイル名，報告書名の検出結果で判断してください。
- 埋め込み SQL 宣言節内で定義しているデータ項目が SQL 文中で使用されていても，COBOL の構文中で使用されていない場合は，未使用のデータ項目として検出します。このため，埋め込み SQL 宣言節内のデータ項目が未使用のデータ項目として検出された場合は，SQL 文中でも使用されていないかどうかを確認してください。
- データ項目が参照された場合，そのデータ項目の定義に必要なデータ項目は未使用のデータ項目として検出されません。

(例)

未使用のデータ項目として検出されない例

```
01 A PIC X.  
01 B REDEFINES A PIC 9.
```

上記の定義に対し，データ項目「B」が参照されている場合，データ項目「A」は未使用のデータ項目として検出されません。

未使用のファイル名，報告書名，画面名，通信記述名

ファイル名，報告書名，画面名，通信記述名の中で，手続き部から 1 回も参照されていないものが検出されます。

ファイル名，報告書名，画面名が参照された場合，各項目の定義に必要なデータ項目は未使用のデータ項目として検出されません。

(例)

未使用のデータ項目として検出されない例

```
FILE-CONTROL.  
  SELECT FILE1  
    ASSIGN TO 'FNAME'  
    FILE STATUS FS1.  
  :  
WORKING-STORAGE SECTION.  
  01 FS1 PIC X(2).
```

上記の定義に対し、ファイル「FILE1」が参照されている場合、データ項目「FS1」は未使用のデータ項目として検出されません。

制御が渡らない文、手続き

実行時に制御が渡る可能性がない文、手続きが検出されます。COBOL ソース解析では、次の文および手続きを制御が渡らない文と判断します。

- GO TO 文、GOBACK 文など、次の文に実行の制御を移さない文があるとき、その文の後ろに続く文から、その文が属する節、段落、無条件文の終わり、または ENTRY 文の前まで。
- GO TO 文や PERFORM 文による参照がなく、先行する節、段落からも制御が渡ることがない手続き。

ただし、次に示す文および手続きは、実行時に制御が渡る可能性がある文および手続きと判断します。

- 宣言部分の USE 手続き
- 非宣言部分の最初の文
- ENTRY 文

(例)

実行時に制御が渡る可能性がある文および手続きと判断する例

```
PROCEDURE DIVISION.  
DECLARATIVES.  
USE-PROC SECTION.                *> 宣言部分のUSE手続きは制御が  
    USE AFTER EXCEPTION CONDITION *> 渡る可能性があると判断する  
        EC-SIZE-ZERO-DIVIDE.  
        DISPLAY 'ゼロ除算発生'.  
        RESUME 復帰手続き.  
        DISPLAY 'DEAD CODE'.      *> デッドコード候補  
END DECLARATIVES.  
PROC-SEC SECTION.  
    ACCEPT X.                     *> 非宣言部分の最初の文は制御が  
                                *> 渡る可能性があると判断する  
>>TURN EC-SIZE CHECKING ON  
    DIVIDE 10 BY X GIVING RESULT.  
    DISPLAY RESULT.  
    STOP RUN.  
    DISPLAY 'DEAD CODE'.          *> デッドコード候補  
  
    ENTRY 'ENTRY1'.               *> ENTRY文は制御が渡る可能性が  
    DISPLAY '入口点 : ENTRY1'.    *> あると判断する  
    :
```

次の文に実行の制御を移さない文

次の文に実行の制御を移さない文を次に示します。

- 画面機能の ACCEPT 文/ REPLY 文 (USE FOR WINDOW 節内)
- EXIT PERFORM 文
- EXIT USE 文

- GO TO 文
- GOBACK 文
- IF 文/SEARCH 文の NEXT SENTENCE
- RESUME 文
- STOP 文
- 手続き先から制御が戻って来ないそと PERFORM 文、PROCEDURE 指定のある SORT/MERGE 文（USE 手続き中に「RESUME 手続き名」が存在する場合を除く）

(例)

PERFORM 文の例

```
A.
  PERFORM B.
  DISPLAY 'A'.  *> デッドコード候補
B.
  STOP RUN.
```

USE 手続き中に「RESUME 手続き名」が存在する場合

「RESUME 手続き名」が存在する場合、そと PERFORM 文と PROCEDURE 指定のある SORT/MERGE 文は、指定された手続き先から必ず制御が戻ってくるものとして扱います。

次に示すプログラムでは、1.の PERFORM 文を実行すると手続き A 中で 4. STOP RUN 文を実行するため、PERFORM 文に制御が戻ってきません。このため、2.の DISPLAY 文は実行されません（デッドコード候補）。しかし、手続き A 中の 3.で例外が発生すると、USE 手続きを通して手続き B へ制御が渡ることがあります。この場合、1.の PERFORM 文に制御が戻り、2.の DISPLAY 文も実行されます。

COBOL ソース解析、PERFORM/SORT/MERGE 文に指定した手続き先で、実行時に例外が発生するかどうかは判断しません。このため、手続き先での例外の発生を考慮し、「RESUME 手続き名」が存在するプログラムの場合、PERFORM/SORT/MERGE 文に指定された手続き先からは、必ず制御が戻ると仮定します。

このプログラムの例では、1.の PERFORM 文は必ず制御が戻ると仮定するため、2.の DISPLAY 文はデッドコード候補として検出されません。

(例)

PERFORM 文の例

```
USE-PROC SECTION.
  USE AFTER EXCEPTION CONDITION EC-ALL.
  RESUME B.
END DECLARATIVES.
ST.
  ACCEPT      X.
  MOVE 123 TO Y.
  PERFORM A THRU B.      *> 1.
  DISPLAY 'AFTER PERFORM'. *> 2. デッドコード候補ではないと仮定
  STOP RUN.
A.
  DISPLAY 'A'.
```



```

COMPUTE X = Y / X.      *> 3.
STOP RUN.               *> 4.
B.
DISPLAY 'B'.

```

- 条件文 (IF, EVALUATE) や条件指定 ([NOT] ON SIZE ERROR など) のある文で、すべての分岐先の無条件文に制御を移す文が存在

(例)

すべての分岐先の無条件文に制御を移す文が存在する例

```

A.
  IF X = 1 THEN
    GO TO B
  ELSE
    GO TO C
  END-IF.
DISPLAY 'A'. *> デッドコード候補
B.
C.

```

注意

- 条件文、または条件指定付きの文がある場合、すべての分岐先が実行される可能性があるとは仮定されません。

ALTER 文で飛び先が変更されている GO TO 文がある場合、その GO TO 文を参照しているすべての ALTER 文に指定された飛び先の手続きは、実行されると仮定されます。

(例) 次に示す例では、手続き A 中の GO TO 文は、ALTER 文によって飛び先を手続き X または Y に変更しています。このプログラムで手続き A 中の GO TO 文に制御が渡る場合、手続き X、手続き Y の両方とも実行される手続きと仮定されます。

```

ALTER A TO X.
:
ALTER A TO Y.
:
A.
GO TO.
B.
STOP RUN.
:
X.      *> 実行される可能性があるとは仮定する
:
Y.      *> 実行される可能性があるとは仮定する
:

```

(例 1) 制御が渡らない文の例 1

次のプログラムでは、GOBACK 文の後ろにある文は実行されることがないため、デッドコード候補として検出されます。

```

A.
GOBACK.
DISPLAY 'this is dead code'. *> デッドコード候補

```

(例 2) 制御が渡らない文の例 2

次のプログラムでは、手続き B の最初の文である DISPLAY 文に制御が渡ることはないが、ENTRY 文は実行される可能性があると判断されます。このため、手続き B の最初の DISPLAY 文だけデッドコード候補として検出されます。

```
PROCEDURE DIVISION.  
A.  
    DISPLAY 'XXX'.  
    STOP RUN.  
B.  
    DISPLAY 'this is dead code'. *> デッドコード候補  
    ENTRY 'YYY'.  
    DISPLAY 'YYY'.
```

(例 3) 制御が渡らない手続きの例

次のプログラムでは、IF 文の THEN 節、ELSE 節は両方実行される可能性があると判断されます。その結果、手続き B をデッドコード候補として検出されます。

```
      :  
PROCEDURE DIVISION.  
A.  
    ACCEPT X.  
    IF X = 1 THEN  
        PERFORM C  
    ELSE  
        GO TO D  
    END-IF.  
    STOP RUN.  
B.          *> デッドコード候補の手続き  
    GO TO D.  
C.  
D.
```

デッドコード候補の検知の注意

- COBOL ソース解析では、登録集原文取り込み後の COBOL プログラムをデッドコード候補の検出対象とするため、登録集原文中の行に対してデッドコード候補が検出されることがあります。しかし、そのデッドコード候補となった登録集原文中の行は、ほかの COBOL プログラムではデッドコード候補にならないことがあります。デッドコード候補の個所を修正するときは、呼び出し関連を確認してください。
- デッドコード候補の検知以外のコンパイルエラーが出力されるのを防ぐため、解析する COBOL プログラムに必要なすべてのコンパイラオプションとコンパイラ環境変数を指定してください。
- デッドコード（制御が渡らない手続き）として検出された節や段落は、[ソース表示(COPY 展開後)]画面で節見出しや段落見出しの部分だけが網掛けされます。節がデッドコード（制御が渡らない手続き）の場合、その節に属するすべての段落もデッドコード（制御が渡らない手続き）ですが、段落見出しは網掛けされません。節見出しが網掛けされている場合は、節全体（節に属する段落すべてを含む）がデッドコードであると解釈してください。
- ENTRY 文は COBOL ソース解析では解析対象外です。このため、ENTRY 文の入口名で呼び出されていても、プログラム名で呼び出されていないプログラムは「未使用」として表示されます。

10.7 埋め込み SQL 文を含む COBOL ソースファイルの解析

埋め込み SQL 文を含む COBOL ソースファイルの解析方法について説明します。

COBOL ソース解析では、埋め込み SQL 文を含む COBOL ソースファイルを解析して、埋め込み SQL 文から表に対するデータ操作（CRUD 操作）の情報を抽出します。抽出対象となる CRUD 操作の情報（CRUD 操作情報）を次に示します。

- 埋め込み SQL 文で操作する表名
- 埋め込み SQL 文を含むプログラム名
- 表に対する操作種別（C：生成（Create）、R：読み取り（Read）、U：更新（Update）、D：削除（Delete））

COBOL ソース解析では、すべての COBOL ソースファイルを解析したあとに、システム全体のすべての CRUD 操作情報を分析し、各表(DB)に対して、どのプログラムが表データの生成（Create）、読み取り（Read）、更新（Update）、削除（Delete）を実行しているかを示す情報（CRUD 情報）を生成します。

[表(DB)詳細] 画面での CRUD 情報の表示例を次に示します。

図 10-3 [表(DB)詳細] 画面での CRUD 情報の表示例



CRUD 情報を使用すると、表データのライフサイクルに関するプログラムを確認できるため、次の作業を効率的かつ効果的に実施できます。

- 表操作の矛盾（表に対する読み取りがあるが生成がないなど）の洗い出し
- 表形式やインデクスの変更時の影響個所の特定
- 障害発生時の問題個所の絞り込み
- 表の棚卸し対象（表データが使用されていない表）の洗い出し

解析対象の埋め込み SQL 文を含む COBOL ソースファイル

COBOL ソース解析では、次に示す SQL 埋め込み COBOL ソースファイル（埋め込み SQL 文を含む COBOL ソースファイル）を解析対象とします。

- COBOL2002 の ODBC インタフェース機能で利用できる埋め込み SQL 文を含む COBOL ソースファイル
- XDM/RD の埋め込み SQL 文を含む COBOL ソースファイル
- COBOL 言語用 SQL プリプロセサの UAP ソースファイル

注意事項

COBOL 言語用 SQL プリプロセサの UAP ソースファイルの場合、次のどちらかに該当する COBOL ソースファイルはコンパイルエラーになるため、COBOL ソース解析では解析できません。

- SQLCODE 変数, SQLSTATE 変数, SQL 連絡領域 (SQLCA) または SQL 記述領域 (SQLDA) を COBOL プログラム内で参照しているが、明示的な埋め込み SQL 宣言がない。
- SQL TYPE 句を使用している。

この場合、COBOL2002 コンパイラの環境変数 CBLCOPT2002 に -SQLAutoCopy コンパイラオプションを指定する必要があります。-SQLAutoCopy コンパイラオプションについては、[\[付録 B.3 -SQLAutoCopy コンパイラオプション\]](#) を参照してください。

解析対象の埋め込み SQL 文

COBOL ソース解析で解析対象になる埋め込み SQL 文は、次に示す形式です。

```
EXEC SQL
  [AT {SQL識別子 | :埋め込み変数名} ]   [FOR {整数 | :埋め込み変数名} ]
  文種別キーワード [句] ...
END-EXEC
```

COBOL ソース解析で解析対象になる埋め込み SQL 文の構文規則を次に示します。

解析対象の埋め込み SQL 文の構文規則

- EXEC, SQL, および END-EXEC の語は、固定継続標識を使って複数行にわたって記述できません。
- 埋め込み SQL 文中の全角・半角および英大文字・英小文字の扱いを次に示します。埋め込み SQL 文中では、-EquivRule コンパイラオプションの規則は埋め込み変数名だけに適用され、それ以外の部分には適用されません。

SQL の予約語の場合

半角英字だけ使用できます（全角英字は使用できません）。英大文字・英小文字を区別しません。

SQL の予約語以外の語の場合

全角・半角および英大文字・英小文字を区別します。ただし、引用符 (") で囲まれていないカーソル名や表名中の半角英小文字は半角英大文字に変換します。

- 埋め込み SQL 文中の注記行は COBOL の規則に準じます。
- 埋め込み SQL 文中の行の継続は COBOL の規則に準じます。
- 埋め込み SQL 文中のデバッグ行は COBOL の規則に準じます。-EquivRule,NotAny コンパイラオプションが指定されている場合、小文字の d はデバッグ標識と見なされないため、デバッグ行の扱いにはなりません。-DebugLine コンパイラオプションが指定されている場合、デバッグ行は有効行として扱います。
- 埋め込み SQL 文中は SQL のコメント指定 (--~および/*~*/) と、COBOL の行内注記 (*>~) の両方が使用できます。
- 埋め込み SQL 文中のタブ文字は COBOL の規則に準じます。タブを空白に置き換えるときの空白の個数は、環境変数 CBLTAB を参照してください。
- 埋め込み例外宣言 (WHENEVER 文) およびカーソル宣言 (DECLARE CURSOR 文) は、手続き部にだけ記述できます。

COBOL ソース解析の SQL 文の解析では、表データに対して CRUD 操作を行う埋め込み SQL 文 (CRUD 操作文) を検出して、CRUD 操作種別と表名を抽出します。

COBOL ソース解析で解析対象になる埋め込み SQL 文と抽出する CRUD 操作種別を次の表に示します。

表 10-7 COBOL ソース解析で解析対象になる埋め込み SQL 文と抽出する CRUD 操作種別

種類	分類	SQL 文	説明	抽出する CRUD 操作 種別
定義系	表操作	CREATE TABLE	表の作成	—
		ALTER TABLE	表の変更	—
		DROP TABLE	表の削除	D
	その他	その他の定義系 SQL 文	上記以外の定義	—
宣言系	埋め込み SQL 宣言節	BEGIN DECLARE SECTION	埋込み SQL 開始宣言	—
		END DECLARE SECTION	埋込み SQL 終了宣言	—
	埋め込み例外宣言	WHENEVER	埋込み例外宣言	—
	カーソル宣言	DECLARE CURSOR	カーソル宣言	—
制御系	接続	CONNECT	データベースとの接続	—
		DISCONNECT	データベースからの切断	—
	トランザクション	COMMIT	コミットでのトランザクションの終了	—

種類	分類	SQL 文	説明	抽出する CRUD 操作 種別
		ROLLBACK	ロールバックでのトランザクションの終了	—
操作系	静的 SQL	SELECT	表から条件が一致する 1 行の取り出し	R
		INSERT	表への行の挿入	C, R※1
		UPDATE (探索)	表の条件が一致する行の更新	U, R※1
		DELETE (探索)	表から条件が一致する行の削除	D, R※1
		TRUNCATE	表の全行削除	D
		PURGE	表の全行削除	D
		ASSIGN LIST	リストの作成	—
	カーソル操作	OPEN	カーソルのオープン	R※2
		FETCH	カーソルが指す行の取り出し	R※2
		UPDATE (位置付け)	カーソルが指す行の更新	U, R※1
		DELETE (位置付け)	カーソルが指す行の削除	D
		CLOSE	カーソルのクローズ	—
	動的 SQL	EXECUTE IMMEDIATE	動的 SQL の前処理と実行	—
		PREPARE	動的 SQL の前処理	—
		EXECUTE	動的 SQL の実行	—
		DEALLOCATE PREPARE	動的 SQL の前処理無効化	—
	その他	その他の操作系 SQL 文	上記以外の操作	—

(凡例)

C：CRUD 操作種別として作成 (Create) を抽出します。

R：CRUD 操作種別として読み取り (Read) を抽出します。

U：CRUD 操作種別として更新 (Update) を抽出します。

D：CRUD 操作種別として削除 (Delete) を抽出します。

—：CRUD 操作文でない埋め込み SQL 文です。この SQL 文は無視されます。

注※1

問合せ式や副問合せに指定された表名に対する CRUD 操作情報です。

注※2

カーソルを定義する DECLARE CURSOR 文の問合せに指定された表名の CRUD 操作情報です。

解析対象 SQL 文の表参照

COBOL ソース解析で解析対象になる SQL 文の表参照の形式を次に示します。

{ [認可識別子.] 表識別子 [句] ... |
 導出表副問合せ [句] ... |
 [() 表参照 {INNER | {LEFT | RIGHT} [OUTER] | {CROSS | FULL} } JOIN 表参照 ON 結合条件 [()] }

注意事項

認可識別子はスキーマ名またはスキーマ認可識別子です。表識別子は、実表名、ビュー表名または問合せ名です。相関名は CRUD 情報の抽出対象になりません。

構文的なコンパイルエラーのある SQL 文の扱い

COBOL ソース解析では、構文的なコンパイルエラーがないものを解析対象としています。このため、SQL 文に構文エラーがあってもエラーメッセージは出力しません。ただし、SQL 文から正しい表名が抽出できない場合は、次に示す警告メッセージを出力します。

表 10-8 正しい表名が抽出できない場合に出力する警告メッセージと SQL 文の扱い

構文エラーで出力する警告メッセージ	メッセージの出力条件	SQL 文の扱い
KEDL1277A-W 埋め込み SQL 文に構文エラーがあるため表名を抽出できませんでした。 SQL 文=(** 1 **)	<ul style="list-style-type: none"> 表名の文脈を検出するための予約語が見つからない。 表名やカーソル名の文脈に予約語や分離符の記号（引用符を除く）が現れた。 	「== SQL 構文エラー ==」を表名として CRUD 操作情報を抽出します。 [表(DB)詳細] 画面では、表名「== SQL 構文エラー ==」に、SQL 文で構文エラーがあるプログラムの一覧が表示されます。
KEDL1278A-W SQL 識別子(** 1 **)が不正な文字を含んでいます。@に置き換えます。	<ul style="list-style-type: none"> SQL 識別子の通常識別子が英数字文字、下線、\$, #, 半角カナ文字以外の半角文字を含んでいる。 	不正な文字を@に置き換えます。
KEDL1279A-W ** 1 **文(SQL)に指定されたカーソル名(** 2 **)の定義が見つからないため表名を抽出できませんでした。	<ul style="list-style-type: none"> OPEN 文や FETCH 文に指定されたカーソル名を定義する DELCLARE CURSOR 文が見つからない。 	「== カーソル名未定義 ==」を表名として CRUD 操作情報を抽出します。 [表(DB)詳細] 画面では、表名「== カーソル名未定義 ==」に、SQL 文でカーソル名未定義エラーがあるプログラムの一覧が表示されます。

CRUD 情報を出力するソース解析情報

CRUD 情報を出力するソース解析情報（HTML ファイル形式、CSV ファイル形式）を次の表に示します。

表 10-9 CRUD 情報を出力するソース解析情報

項番	ソース解析情報	CRUD 情報の出力項目
1	表(DB)一覧	<ul style="list-style-type: none"> 表名※1 入出力種別（CRUD 操作種別※2） 使用プログラム数
2	表(DB)詳細	<ul style="list-style-type: none"> 表名※1

項番	ソース解析情報	CRUD 情報の出力項目
		<ul style="list-style-type: none"> • 使用プログラム数 • CRUD の分布 • 使用プログラム名 • 入出力種別 (CRUD 操作種別※2)
3	プログラム詳細の表(DB)一覧	<ul style="list-style-type: none"> • 表名※1 • 入出力種別 (CRUD 操作種別※2)
4	セクション階層図	<ul style="list-style-type: none"> • 表名※1 • 入出力種別 (CRUD 操作種別※2)
5	プログラム階層図	<ul style="list-style-type: none"> • 表名※1 • 入出力種別 (CRUD 操作種別※2)
6	表(DB)参照関係 CSV	<ul style="list-style-type: none"> • 表名※1 • 使用プログラム名 • 入出力種別 (CRUD 操作種別※2)

注※1

表名は、次の形式で出力します。

〔認可識別子.〕 表識別子

注※2

CRUD 操作種別は、C, R, U, D の順序で出力されます。

例えば、表データに対して生成 (Create)、更新 (Update)、読み取り (Read) を操作した CRUD 操作文の場合、CRUD 操作種別には「CRU」が出力されます。

注意事項

- COBOL ソースファイルで使用している表名が、通常識別子 (引用符 (")) で囲まれていない SQL 識別子) の場合、半角英小文字は半角英大文字に変換されてソース解析情報 (HTML ファイル形式、CSV ファイル形式) に出力されます。
- COBOL ソースファイルで使用している表名が、区切り識別子 (引用符 (")) で囲まれた SQL 識別子) の場合、引用符を除いた名前がソース解析情報 (HTML ファイル形式、CSV ファイル形式) に出力されます。また、引用符を除いた名前が次の半角記号文字を含んでいる場合、これらの半角記号文字は全角に変換されます。
" % & ' () * , / : ; < > ? ¥ ^ !
- 構文エラーで表名を抽出できなかった CRUD 操作文がプログラムにある場合、表名として「== SQL 構文エラー ==」が、ソース解析情報 (HTML ファイル形式) に出力されます。
- 未定義のカーソル名を指定した OPEN 文や FETCH 文がプログラムにある場合、表名として「== カーソル名未定義 ==」が、ソース解析情報 (HTML ファイル形式) に出力されます。

CRUD 情報の抽出例

COBOL ソースファイルから CRUD 情報を抽出する例を次に示します。

COBOL ソースファイルの例

```

IDENTIFICATION DIVISION.
PROGRAM-ID. データ追加.
:
EXEC SQL
INSERT INTO TBL1(CLM1,CLM2) VALUES( (SELECT MIN(CLM3) FROM TBL2), NULL )
END-EXEC.
:
END PROGRAM データ追加.

IDENTIFICATION DIVISION.
PROGRAM-ID. データ検索.
:
EXEC SQL DECLARE CSR1 CURSOR FOR SELECT * FROM TBL1 WHERE CLM1 = :PARAM END-EXEC.
EXEC SQL OPEN CSR1 END-EXEC.
:
EXEC SQL FETCH CSR1 INTO :CLM1-VALUE, :CLM2-VALUE END-EXEC.
:
EXEC SQL CLOSE CSR1 END-EXEC.
:
END PROGRAM データ検索.

IDENTIFICATION DIVISION.
PROGRAM-ID. データ更新.
:
EXEC SQL
UPDATE TBL1 SET CLM2=(SELECT MAX(CLM3) FROM TBL2) WHERE CLM1 = :PARAM
END-EXEC.
:
END PROGRAM データ更新.

IDENTIFICATION DIVISION.
PROGRAM-ID. データ削除.
:
EXEC SQL DELETE FROM TBL1 WHERE CLM1 = :PARAM END-EXEC.
:
END PROGRAM データ削除.

```

CRUD 情報の抽出結果

表名	CRUD 操作を行うプログラム名	CRUD 操作種別	備考
TBL1	データ追加	C	INSERT 文の CRUD 操作
	データ検索	R	OPEN 文/FETCH 文の CRUD 操作
	データ更新	U	UPDATE 文の CRUD 操作
	データ削除	D	DELETE 文の CRUD 操作
TBL2	データ追加	R	INSERT 文の副問合せの CRUD 操作
	データ更新	R	UPDATE 文の副問合せの CRUD 操作

注意事項

- ソースファイルの種別に応じた-SQL コンパイラオプションを指定して解析してください。
ソースファイルの種別と-SQL コンパイラオプションの対応を次に示します。

ソースファイルの種別	-SQL コンパイラオプションの指定有無		
	指定なし	-SQL,XDM コンパイラ オプションの指定あり	-SQL,ODBC コンパイラ オプションの指定あり
COBOL2002 の ODBC インタフェース用 COBOL ソースファイル	◎	○※1	○※2
XDM/RD の埋め込み SQL を含む COBOL ソースファイル	△※3	○※1	×
COBOL 言語用 SQL プリプロセサの UAP ソースファイル	◎	○※1	×

(凡例)

- ◎：解析できます。
- ：解析できますが、制限があります。
- △：一部解析できません。
- ×：解析できません。

注※1

埋め込み変数がデッドコード情報※4 で不当に「未使用」として報告されることがあります。また、-SQL,XDM コンパイラオプションは、-SQLAutoCopy コンパイラオプションと併用できません。

注※2

埋め込み SQL 宣言内で定義したデータ項目は、デッドコード情報※4 で「未使用」として報告されません。

注※3

RDB-STATUS, RDB-COUNT, または RDB-RDNODE 特殊レジスタが使用されている場合や、未定義のデータ項目や手続き名を埋め込み変数に指定している場合は、コンパイラで構文エラーとなります。

注※4

【プログラム詳細】画面の「プログラム概要」や【デッドコード情報一覧】画面に表示される未使用のデータ項目数、および【ソース表示(COPY 展開後)】画面内のデータ記述項の網掛け表示（背景をグレーで表示）が該当します。

- データベース管理システムのエンハンスで追加または変更された SQL の構文では、CRUD 操作情報が抽出できないことがあります。
- COBOL 言語用 SQL プリプロセサの UAP ソースファイルを解析する場合、SQL プリプロセサでエラーがないことが前提となります。SQL 文が正しくないと、次のような解析結果となることがあります。
 - 構文エラーを示す警告メッセージが出力され、その SQL 文の CRUD 操作情報が抽出されない
 - 表名ではない SQL 識別子や定数が表名として表示される
 - データベース管理システムで定義する SQL の予約語が表名として表示される
- 動的 SQL のための構文が指定されている SQL 文は、CRUD 操作情報の抽出対象にはなりません。
COBOL ソースファイルに次の SQL 文が記述されている場合は、CRUD 操作情報の抽出対象外であることを示す警告メッセージが表示されます。

- FOR 句に SQL 識別子が指定されている DECLARE CURSOR 文
- カーソル指定の文脈に埋め込み変数が指定されている OPEN 文または FETCH 文
- 動的 SQL または SQL 手続きを実行する EXECUTE 文は、CRUD 操作情報の抽出対象にはなりません。COBOL ソースファイルに次の SQL 文が記述されている場合は、CRUD 操作情報の抽出対象外であることを示す警告メッセージが表示されます。
 - EXECUTE 文
 - EXECUTE IMMEDIATE 文
- リスト名を使った SELECT 文や問合せは、CRUD 操作情報の抽出対象にはなりません。COBOL ソースファイルに次の SQL 文が記述されている場合は、CRUD 操作情報の抽出対象外であることを示す警告メッセージが表示されます。
 - リスト名を使った SELECT 文
 - リスト名を使った問合せ
- WITH 句（WITH 問合せ名～）を用いた問合せ式は、DECLARE CURSOR 文には指定できません。WITH 句（WITH 問合せ名～）を用いた問合せ式が DECLARE CURSOR 文に指定されている場合、SQL 文が解析できないことを示すコンパイラのメッセージが表示されます。また、カーソル名未定義のコンパイルエラーが、その DECLARE CURSOR 文のカーソル名を参照する OPEN 文で表示されます。
- -SQL コンパイラオプションの指定なしで COBOL 言語用 SQL プリプロセサの UAP ソースファイルを解析した場合、次のどれかの条件を満たす SQL 文に対してコンパイラで解析できないことを示すメッセージが表示されます。ただし、S レベル以上のコンパイルエラーが出力されていないかぎり、COBOL ソース解析の CRUD 情報の解析には影響ありません。
 - カーソル名を引用符 (") で囲んでいる
 - カーソル名に記号 (#, \$) を使用している
 - SQL のコメント指定 (/~*/) を使用している
- 開発マネージャに COBOL 用 SQL プリプロセサの UAP ソースファイルを登録する場合に、SQL プリプロセサを実行してプリプロセス後の COBOL ソースファイルを生成する設定で登録したとき、ソース解析プロジェクトの解析の実行で解析されるのはプリプロセス後の COBOL ソースファイルとなります。プリプロセス後の COBOL ソースファイルは埋め込み SQL 文を含まないため、表(DB)一覧や表(DB)詳細などの CRUD 情報のドキュメントは作成されません。CRUD 情報のドキュメントが必要な場合は、cblcsa コマンドを使って、プリプロセス前の SQL プリプロセサの UAP ソースファイルを解析／生成してください。

10.8 実行文として数える文

COBOL ソース解析で、実行文として数える文の一覧です。

COBOL ソース解析では、ACCEPT や ADD 文を実行文として数えます。

表 10-10 COBOL ソース解析で実行文として数える文

項番	文
1	ACCEPT 文
2	ADD 文
3	ALTER 文
4	CALL 文
5	CANCEL 文
6	CLOSE 文
7	COMMIT 文
8	COMPUTE 文
9	CONNECT 文※
10	CONTINUE 文※
11	DELETE 文
12	DISABLE 文
13	DISCONNECT 文※
14	DISPLAY 文
15	DIVIDE 文
16	ENABLE 文
17	ENTRY 文
18	ERASE 文
19	EVALUATE 文
20	EXAMINE 文
21	EXEC SQL SQL 文 END-EXEC
22	EXIT 文
23	FETCH 文※
24	FIND 文
25	GENERATE 文

項番	文
26	GET 文
27	GO TO MORE-LABELS 文
28	GO TO 文
29	GOBACK 文
30	IF 文
31	INITIALIZE 文
32	INITIATE 文
33	INSPECT 文
34	INVOKE 文
35	MERGE 文
36	MODIFY 文
37	MOVE 文
38	MULTIPLY 文
39	NULLIFY 文
40	OPEN 文
41	PERFORM 文
42	RAISE 文
43	READ 文
44	RECEIVE 文
45	RECONNECT 文
46	RELEASE 文
47	REPLY 文
48	RESUME 文
49	RETURN 文
50	REWRITE 文
51	ROLLBACK 文※
52	SEARCH 文
53	SEND 文
54	SET 文
55	SORT 文
56	START 文

項番	文
57	STOP 文
58	STORE 文
59	STRING 文
60	SUBTRACT 文
61	SUPPRESS 文
62	TERMINATE 文
63	TRANSCIVE 文
64	TRANSFORM 文
65	UNLOCK 文
66	UNSTRING 文
67	USE 文
68	WRITE 文

注※

SQL 文のときは、「EXEC SQL SQL 文 END-EXEC」でまとめて 1 文として数えます。

10.9 解析結果に出力される COBOL ソースファイル中の名称

COBOL ソース解析の解析結果に出力される COBOL ソースファイル中の名称について説明します。-EquivRule コンパイラオプションの指定有無によって、名称の変換規則が異なります。

-EquivRule コンパイラオプションの指定がない場合の名称の変換規則

-EquivRule コンパイラオプションの指定がない場合、COBOL ソースファイルで使用している名称は変換されて、ソース解析情報に出力されます。-EquivRule コンパイラオプションの指定がない場合の各名称の変換規則を次の表に示します。

表 10-11 COBOL ソースファイルで使用している名称の変換規則 (-EquivRule コンパイラオプションの指定がない場合)

名称	変換規則
プログラム名	プログラム名指定の場合 全角文字は対応する半角文字に変換されます。 半角英小文字は半角英大文字に変換されます。 半角の-（ハイフン）は、半角の_（アンダスコア）に変換されます。 定数指定の場合 半角の-（ハイフン）は、半角の_（アンダスコア）に変換されます。 半角の#は、先頭文字のときは「J」、2文字目以降のときは「0」に変換されます。 半角の¥は、先頭文字のときは「A」、2文字目以降のときは「1」に変換されます。 半角の@は、先頭文字のときは「B」、2文字目以降のときは「2」に変換されます。 半角の1～9と0は、先頭文字のときはA～IとJに変換されます。2文字目以降のときは変換されません。
登録集原文名	登録集原文名指定の場合 半角英小文字は半角英大文字に変換されます。 定数指定の場合 半角、全角に関係なく、英小文字は英大文字に変換されます。
節名	全角文字は対応する半角文字に変換されます。
段落名	半角英小文字は半角英大文字に変換されます。
データ名	
ファイル名	
報告書名	
画面名	
通信記述名	
表名	引用符（"）で囲まれていない場合 半角英小文字は半角英大文字に変換されます。

名称	変換規則
	引用符 (") で囲まれている場合 引用符を除いた名前に変換されます。また、引用符を除いた名前が次の半角記号文字を含んでいる場合、これらの半角記号文字は全角に変換されます。 " % & ' () * , / : ; < > ? ¥ ^ ;

-EquivRule コンパイラオプションの指定がある場合の名称の変換規則

-EquivRule コンパイラオプションの指定がある場合、COBOL ソースファイルで使用している名称は、全角文字および半角英小文字は変換されないでそのままソース解析情報に出力されます。

プログラム名、節名、段落名、データ名、ファイル名、報告書名、画面名、および通信記述名は、-EquivRule コンパイラオプションのサブオプションの指定によって、変換規則が異なります。

-EquivRule コンパイラオプションの指定がある場合の各名称の変換規則を次の表に示します。

表 10-12 COBOL ソースファイルで使用している名称の変換規則 (-EquivRule コンパイラオプションの指定がある場合)

名称	NotExtend サブオプション、StdCode サブオプション指定時の変換規則	NotAny サブオプション指定時の変換規則
プログラム名	<p>プログラム名指定の場合</p> <p>全角文字は対応する半角文字に変換されません。 半角英小文字は半角英大文字に変換されます。 半角の- (ハイフン) は、半角の_ (アンダスコア) に変換されます。</p> <p>定数指定の場合</p> <p>-EquivRule コンパイラオプションの指定がない場合の名称の変換規則と同じです。</p>	<p>プログラム名指定の場合</p> <p>全角文字は対応する半角文字に変換されません。 半角英小文字は半角英大文字に変換されません。 半角の- (ハイフン) は、半角の_ (アンダスコア) に変換されます。</p> <p>定数指定の場合</p> <p>-EquivRule コンパイラオプションの指定がない場合の名称の変換規則と同じです。</p>
登録集原文名	-EquivRule コンパイラオプションの指定がない場合の名称の変換規則と同じです。	-EquivRule コンパイラオプションの指定がない場合の名称の変換規則と同じです。
節名	<p>全角文字は対応する半角文字に変換されません。 半角英小文字は半角英大文字に変換されます。</p>	<p>全角文字は対応する半角文字に変換されません。 半角英小文字は半角英大文字に変換されません。</p>
段落名		
データ名		
ファイル名		
報告書名		
画面名		
通信記述名		
表名	-EquivRule コンパイラオプションの指定がない場合の名称の変換規則と同じです。	-EquivRule コンパイラオプションの指定がない場合の名称の変換規則と同じです。

10.10 COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI)

COBOL ソース解析の動作環境をカスタマイズする環境定義ファイル (CBLANALYZER.INI) に指定できるパラメータ、および環境定義ファイルのパラメータに不正な値が指定された場合の動作について説明します。

10.10.1 COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) の説明

COBOL ソース解析の動作環境をカスタマイズする環境定義ファイル (CBLANALYZER.INI) の説明です。

以降に示すそれぞれのセクションを変更すると、COBOL ソース解析の動作環境をカスタマイズできます。無効な値を指定すると、デフォルト値が仮定されます。

注意事項

環境定義ファイル (CBLANALYZER.INI) は、次のフォルダに格納されています。

*COBOL2002 Professional Tool Kit*のインストール先フォルダ¥Bin

環境定義ファイル (CBLANALYZER.INI) を変更するときは、元の状態に戻せるように、バックアップを取得しておくことをお勧めします。

接続データベース情報：DataBase セクション

接続先のデータベースに関する設定です。

接続先 ODBC データソース名：ODBCDSN パラメータ

接続先の ODBC データソース名です。この項目は変更しないでください。

デフォルト値：COBOL2002ToolKitDB

解析ログ情報：ANALYZERLOGFILE セクション

解析ログの設定です。

解析ログの出力有無：LOGOUTPUT パラメータ

解析ログを出力するかどうかを「YES」もしくは「NO」、または「Y」もしくは「N」で示します。

デフォルト値：YES

解析ログの出力先フォルダ：LOGFOLDER パラメータ

解析ログの出力先フォルダです。絶対パス名で示します。

デフォルト値：%TEMP%

解析ログの最大サイズ (MB)：LOGMAXSIZE パラメータ

解析ログの最大サイズです。1～512 の範囲の数値 (単位は MB) で示します。

デフォルト値：30

操作ログ情報：OPERATIONLOGFILE セクション

操作ログの設定です。

操作ログの出力有無：LOGOUTPUT パラメータ

操作ログを出力するかどうかを「YES」もしくは「NO」、または「Y」もしくは「N」で示します。

デフォルト値：YES

操作ログの出力先フォルダ：LOGFOLDER パラメータ

操作ログの出力先フォルダです。絶対パス名で示します。

デフォルト値：%TEMP%

操作ログの最大サイズ（MB）：LOGMAXSIZE パラメータ

操作ログの最大サイズです。1～512 の範囲の数値（単位は MB）で示します。

デフォルト値：30

分布情報一覧：DISTRIBUTE セクション名セクション

分布情報一覧の範囲です。この値を変更すると、分布一覧の棒グラフの表示をカスタマイズできます。

セクション名に指定するセクション名と意味を次に示します。

セクション名	意味
DISTRIBUTELINESTEP	行数の分布
DISTRIBUTEDATA	記述項数の分布
DISTRIBUTEEXECUTESTATEMENT	実行文の数の分布
DISTRIBUTEBRANCHSTATEMENT	分岐の数の分布
DISTRIBUTESECTION	節数の分布
DISTRIBUTEPARAGRAPH	段落数の分布
DISTRIBUTECALLERSTATEMENT	呼び出し元プログラム数の分布
DISTRIBUTECALLSTATEMENT	呼び出し先プログラム数の分布

分布範囲指定：RANGE 分布情報の範囲パラメータ

分布情報の範囲を、RANGE01～RANGE10 の間でそれぞれ、最小値と最大値をコンマ「,」で区切って指定します。0 以上の半角数字を指定してください。RANGE10 の最大値は省略でき、指定の有無に関係なく「上限なし（無限大）」の扱いとなります。

分布情報セクション全体または RANGE01～RANGE10 パラメータをすべて省略した場合は、デフォルト値を仮定します。デフォルト値は、各分布情報によって異なります。環境定義ファイル（CBLANALYZER.INI）の初期値を参照してください。なお、RANGE01～RANGE10 パラメータを部分的に省略することはできません。

プログラム呼出関係 CSV：PROGRAMCALLCSV セクション

プログラム呼出関係 CSV の指定です。

プログラム呼び出し最大階層：CALLDEPTHMAX パラメータ

プログラム呼び出しの最大階層を 0～2000 の範囲で指定します。

デフォルト値：255

範囲外および数値以外を指定した場合には、デフォルト値となります。

注意

プログラム呼び出し最大階層を超える COBOL ソースファイルで、プログラム呼出関係 CSV を出力した場合、操作ログにエラーが出力されます。

インストール直後の CBLANALYZER.INI ファイルの内容

```

;
; 接続データベース情報
[DATABASE]
; 接続先ODBCデータソース名
ODBCDSN=COBOL2002ToolKitDB

; 解析ログ情報 (COBOLAnalyzeLog.tag)
[ANALYZERLOGFILE]
; 解析ログの出力有無 (YES/NO)
LOGOUTPUT=YES
; 解析ログの出力先フォルダ
LOGFOLDER=%TEMP%
; 解析ログの最大サイズ (MB)
LOGMAXSIZE=30

; 操作ログ情報 (COBOLControlLog.txt)
[OPERATIONLOGFILE]
; 操作ログの出力有無 (YES/NO)
LOGOUTPUT=YES
; 操作ログの出力先フォルダ
LOGFOLDER=%TEMP%
; 操作ログの最大サイズ (MB)
LOGMAXSIZE=30

; 分布情報一覧 - 行数の分布
[DISTRIBUTELINESTEP]
RANGE01=0,200
RANGE02=201,500
RANGE03=501,1000
RANGE04=1001,1500
RANGE05=1501,2000
RANGE06=2001,3000
RANGE07=3001,5000
RANGE08=5001,7000
RANGE09=7001,10000
RANGE10=10001,

; 分布情報一覧 - 記述項数の分布
[DISTRIBUTEDATA]
RANGE01=0,200
RANGE02=201,500
RANGE03=501,1000
RANGE04=1001,2000
RANGE05=2001,3000
```

```
RANGE06=3001, 5000  
RANGE07=5001, 7000  
RANGE08=7001, 10000  
RANGE09=10001, 20000  
RANGE10=20001,
```

; 分布情報一覧 - 実行文の数の分布
[DISTRIBUTEEXECUTESTATEMENT]

```
RANGE01=0, 20  
RANGE02=21, 50  
RANGE03=51, 100  
RANGE04=101, 200  
RANGE05=201, 500  
RANGE06=501, 1000  
RANGE07=1001, 1500  
RANGE08=1501, 2000  
RANGE09=2001, 3000  
RANGE10=3001,
```

; 分布情報一覧 - 分岐の数の分布
[DISTRIBUTEBRANCHSTATEMENT]

```
RANGE01=0, 2  
RANGE02=3, 5  
RANGE03=6, 10  
RANGE04=11, 20  
RANGE05=21, 50  
RANGE06=51, 100  
RANGE07=101, 200  
RANGE08=201, 500  
RANGE09=501, 1000  
RANGE10=1001,
```

; 分布情報一覧 - 節数の分布
[DISTRIBUTESECTION]

```
RANGE01=0, 1  
RANGE02=2, 3  
RANGE03=4, 5  
RANGE04=6, 10  
RANGE05=11, 20  
RANGE06=21, 30  
RANGE07=31, 50  
RANGE08=51, 70  
RANGE09=71, 100  
RANGE10=101,
```

; 分布情報一覧 - 段落数の分布
[DISTRIBUTEPARAGRAPH]

```
RANGE01=0, 5  
RANGE02=6, 10  
RANGE03=11, 20  
RANGE04=21, 30  
RANGE05=31, 50  
RANGE06=51, 70  
RANGE07=71, 100  
RANGE08=101, 140  
RANGE09=141, 200  
RANGE10=201,
```

```
; 分布情報一覧 - 呼出元プログラム数の分布
[DISTRIBUTE CALLER STATEMENT]
RANGE01=0, 0
RANGE02=1, 1
RANGE03=2, 3
RANGE04=4, 5
RANGE05=6, 7
RANGE06=8, 10
RANGE07=11, 15
RANGE08=16, 30
RANGE09=31, 50
RANGE10=51,

; 分布情報一覧 - 呼出先プログラム数の分布
[DISTRIBUTE CALL STATEMENT]
RANGE01=0, 0
RANGE02=1, 1
RANGE03=2, 3
RANGE04=4, 5
RANGE05=6, 7
RANGE06=8, 10
RANGE07=11, 15
RANGE08=16, 30
RANGE09=31, 50
RANGE10=51,

; プログラム呼出関係CSV
[PROGRAM CALL CSV]
; プログラム呼び出し最大階層(0 - 2000)
CALLDEPTHMAX=255
```

10.10.2 COBOL ソース解析の環境定義ファイル (CBLANALYZER.INI) の不正なパラメータ値の扱い

環境定義ファイルのパラメータに不正な値が指定された場合の動作を説明します。

パラメータに不正な値が指定された場合、デフォルト値の仮定による実行の結果が致命的な問題を引き起こすおそれがあるときは、エラーメッセージを出力して終了します。そうでないときはデフォルト値を仮定して処理を続行します。

ログの出力有無：LOGOUTPUT パラメータ

「YES」、「NO」、「Y」または「N」のどれかが指定されていない場合、デフォルト値「YES」を仮定します。エラーメッセージは出力しません。

ログの出力先フォルダ：LOGFOLDER パラメータ

パス名が次のどちらかに該当する場合、メッセージを出力して終了します。メッセージの出力先は、開発マネージャの場合はメッセージウィンドウ、cblcsa コマンドの場合は標準エラー出力です。

- 存在しないフォルダ名である。またはファイル名が指定されている。

解析ログ用メッセージ：KEDL2042A-E

操作ログ用メッセージ：KEDL2045A-E

- 長さが上限（開発マネージャは 238 文字，cblcsa コマンドは 190 文字）を超えている。

解析ログ用メッセージ：KEDL2043A-E

操作ログ用メッセージ：KEDL2046A-E

ログの最大サイズ (MB)：LOGMAXSIZE パラメータ

0 以下の値，または数値以外が指定された場合はデフォルト値を仮定します。512 を超える値が指定された場合は「512」を仮定します。エラーメッセージは出力しません。

分布範囲指定：RANGE 分布情報の範囲パラメータ

次のどちらかに該当する場合は，メッセージ (KEDL2048A-E) を出力して終了します。メッセージの出力先は，開発マネージャの場合はメッセージウィンドウ，cblcsa コマンドの場合は標準エラー出力です。

- 最小値と最大値に正の整数値以外が指定されている
- 最大値に指定した値より最小値に指定した値の方が大きい

該当するものが複数あっても最初に検出した時点でエラーメッセージを出力して中断します。

プログラム呼出関係 CSV：PROGRAMCALLCSV セクション

範囲外の値，または数値以外が指定された場合はデフォルト値を仮定します。エラーメッセージは出力しません。

10.11 引数整合性チェック

プロジェクト内の CALL 文の実引数と呼出先プログラムの仮引数に不整合（矛盾）がないかをチェックする機能（引数整合性チェック）について説明します。

10.11.1 引数整合性チェックとは

引数整合性チェックとは、プロジェクトのすべての COBOL ソースファイル中の CALL 文に対して、CALL 文の実引数と呼出先プログラムの仮引数に不整合（矛盾）がないかをチェックする機能のことです。

引数整合性チェックで実施する検査項目を次の表に示します。引数整合性チェックでは、この表に示す項番の順番に、実引数と仮引数の整合性を検査します。

表 10-13 引数整合性チェックで実施する検査項目

項番	検査項目		検査内容
1	引数の個数のチェック		実引数と仮引数の個数が一致しているかを検査します。
2	引数ごとの個別 チェック	受渡方法のチェック	実引数と仮引数の受け渡し方法（BY REFERENCE, BY CONTENT, BY VALUE, BY ATTRIBUTE）が一致しているかを検査します。
3		データ型の適合 チェック	実引数と仮引数のデータ型が適合しているかを検査します。
4		サイズのチェック	実引数と仮引数のサイズが一致しているかを検査します。
5		集団項目の構造 チェック	実引数と仮引数が集団項目の場合に、集団項目に属するデータ項目の構造が一致しているかを検査します。

引数整合性チェックを実施するには、[解析生成オプション設定] 画面の [生成ドキュメント] タブで不整合引数一覧の出力を有効（チェックボックスをオン）にします。また、引数整合性チェックの検査項目や検査条件は、[解析生成オプション設定] 画面の [引数整合性チェック] タブで設定します。

引数整合性チェックで引数の不整合が検出されると、次のファイルに引数の不整合情報が出力されます。それらのファイルを参照し、引数ごとの不整合情報を確認して、COBOL プログラムを修正してください。

- 不整合引数一覧 CSV ファイル（不整合引数一覧.csv）

引数不整合のある実引数と仮引数の組み合わせの情報を出力します。引数ごとの個別チェックでは、最初に検出した不整合だけを出力します。

- 不整合引数一覧タグファイル（不整合引数一覧.tag）

引数不整合のある CALL 文、実引数および仮引数に対する警告メッセージを出力します。CSV ファイル形式のソース解析情報が格納されたフォルダでこのタグファイルを選択すると、COBOL エディタが起動します。警告メッセージをダブルクリックすると、COBOL ソースファイルが表示され、該当する CALL 文、実引数または仮引数のソースコードにカーソルが移動します。不整合引数一覧タグファイルに出力される警告メッセージは、解析ログや操作ログには出力されません。

各ファイルに出力される内容については、「[10.11.2 引数整合性チェックの検査内容](#)」を参照してください。

関連項目

- ・ [「5.2.6 \[引数整合性チェック\] タブ](#)」
- ・ [「5.2.8 \[生成ドキュメント\] タブ](#)」

10.11.2 引数整合性チェックの検査内容

引数整合性チェックの検査項目ごとに実施する検査の内容と、不整合引数一覧 CSV ファイル、および不整合引数一覧タグファイルに出力される報告内容について説明します。

(1) 引数の個数のチェック

(a) 検査内容

実引数と仮引数の個数が一致しているかどうかを検査します。ただし、[解析生成オプション設定] 画面の [引数整合性チェック] タブで [実引数個数>仮引数個数の場合は検出しない] を有効にしている場合は、実引数個数>仮引数個数を検出しても不整合引数として報告されません。

(b) 報告内容

実引数と仮引数の個数が一致していない場合、不整合引数一覧 CSV ファイルに不整合種別が「個数不一致」の不整合引数情報を出力します。また、不整合引数一覧タグファイルに警告メッセージを出力します。

各ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

```
項番,<<呼出元情報>>,実引数個数,"-", "-", ..., "-",呼出先プログラム名,仮引数個数,"-", ..., "-",  
"個数不一致"
```

引数番号, 実引数情報, 仮引数情報のすべての列には「-」が出力されます。不整合引数一覧 CSV ファイルに出力される情報の内容については、「[8.2.6 不整合引数一覧](#)」を参照してください。

不整合引数一覧タグファイル

```
# [項番] ソースファイル名, ソース行番号 CALL '呼出先プログラム名' 個数不一致  
"パス名1", line 行番号1: KEDL4000A-W CALL文の実引数の個数が仮引数の個数と一致していま  
せん。呼出元プログラム=*** 1 ***, 呼出先プログラム=*** 2 ***  
"パス名2", line 行番号2: KEDL4001A-I 実引数 個数=*** 1 ***  
"パス名3", line 行番号3: KEDL4002A-I 仮引数 個数=*** 1 ***
```

不整合引数一覧タグファイルに出力される情報の内容を次に示します。

[項番]

不整合引数一覧.csv 内の不整合引数情報の通番です。

[ソースファイル名]

CALL 文のソースファイルのフォルダパス名なしのファイル名です。

[ソース行番号]

CALL 文の [ソース表示(COPY 展開後)] 画面上の行番号です。

[パス名 1], [パス名 2]

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 1], [行番号 2]

CALL 文のファイル (パス名 1) 内の相対行番号です。

[パス名 3]

呼出先プログラムの手続き部見出しの COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 3]

呼出先プログラムの手続き部見出しのファイル (パス名 2) 内の相対行番号です。

(2) 受渡方法のチェック

(a) 検査内容

実引数と仮引数の受け渡し方法が指定できる組み合わせかどうかを検査します。実引数と仮引数の受け渡し方法の組み合わせを次の表に示します。

表 10-14 実引数と仮引数の受け渡し方法の組み合わせ

実引数	仮引数	
	BY REFERENCE	BY VALUE
BY REFERENCE	○	×
BY CONTENT	○	×
BY VALUE	×	○
BY ATTRIBUTE	○	×

(凡例)

- ：組み合わせは指定できます。
- ×：組み合わせは指定できません。

(b) 報告内容

実引数と仮引数の受け渡し方法が指定できない組み合わせの場合、不整合引数一覧 CSV ファイルに不整合種別が「受渡方法不一致」の不整合引数情報を出力します。また、不整合引数一覧タグファイルに警告メッセージを出力します。

各ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

項番,<<呼出元情報>>,"-",引数番号,<<実引数情報>>,呼出先プログラム名,"-",<<仮引数情報>>,"受渡方法不一致"

不整合引数一覧 CSV ファイルに出力される情報の内容については、「[8.2.6 不整合引数一覧](#)」を参照してください。

不整合引数一覧タグファイル

```
# [項番] ソースファイル名, ソース行番号 CALL '呼出先プログラム名' (第<引数番号>引数) 受渡方法不一致
"パス名1", line 行番号1: KEDL4010A-W CALL文の*** 1 ***番目の実引数の受渡方法が仮引数と一致していません。呼出元プログラム=*** 2 ***, 呼出先プログラム=*** 3 ***
"パス名2", line 行番号2: KEDL4011A-I 実引数=*** 1 *** 受渡方法=*** 2 ***
"パス名3", line 行番号3: KEDL4012A-I 仮引数=*** 1 *** 受渡方法=*** 2 ***
```

不整合引数一覧タグファイルに出力される情報の内容を次に示します。

[項番]

不整合引数一覧.csv 内の不整合引数情報の通番です。

[ソースファイル名]

CALL 文のソースファイルのフォルダパス名なしのファイル名です。

[ソース行番号]

CALL 文の [ソース表示(COPY 展開後)] 画面上の行番号です。

[パス名 1], [パス名 2]

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 1], [行番号 2]

CALL 文のファイル (パス名 1) 内の相対行番号です。

[パス名 3]

呼出先プログラムの手続き部見出しの COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 3]

呼出先プログラムの手続き部見出しのファイル (パス名 2) 内の相対行番号です。

(3) データ型の適合チェック

(a) 検査内容

適合規則を使用して、実引数と仮引数のデータ型が適合しているかどうかを検査します。

引数のデータ型は、次の表に示す規則に従って分類されます。

表 10-15 引数のデータ型の分類

引数のデータ型	該当するデータ項目または定数
集団項目	固定長の英数字集団項目
集団項目(可変長)	可変長の英数字集団項目
日本語集団項目	日本語集団項目 (GROUP-USAGE IS NATIONAL)
英字	英字項目
英数字	英数字項目, 英数字定数 -JPN,Alnum または-JPN,V3JPN コンパイラオプションを指定すると, CALL 文の引数に日本語定数が指定できますが, 英数字定数として扱われます。
動的長英数字	動的長英数字項目
英数字編集	英数字編集項目
外部 10 進	外部 10 進項目
内部 10 進	内部 10 進項目 (COMP-3 または PACKED-DECIMAL)
2 進	2 進項目 (COMP, COMP-4, または BINARY), 数字定数, LENGTH OF
2 進 (COMP-X)	COMP-X 項目
2 進 (COMP-5)	COMP-5 項目
数字編集	数字編集項目
外部浮動	外部浮動小数点項目
内部浮動	内部浮動小数点項目 (COMP-1 または COMP-2), 浮動小数点定数
日本語	日本語項目
動的長日本語	動的長日本語項目
日本語編集	日本語編集項目
外部ブール	外部ブール項目
内部ブール	内部ブール項目
アドレス	アドレス項目 (ポインタ項目), ADDR 関数, ADDRESS OF, NULL 表意定数
指標	指標項目
オブジェクト参照	オブジェクト参照項目
OLE 参照	OLE オブジェクト参照項目
OLE バリエント	OLE バリエント項目

データ型の適合規則を次に示します。

1. 次に示すデータ項目は, 実引数と仮引数の両方のデータ型が一致している場合にだけ適合していると判定されます。

- ・ 内部ブール項目
- ・ アドレス項目
- ・ 指標項目
- ・ オブジェクト参照項目
- ・ OLE オブジェクト参照項目
- ・ OLE バリエント項目

2. 項番 1 以外のデータ項目は、次の表に示す規則に従って、適合しているかどうか判定されます。

図 10-4 引数のデータ型の適合規則

仮引数 実引数	集団項目	可変長集団項目	日本語集団項目	英字	英数字	動的長英数字	英数字編集	外部10進	内部10進	2進	COM P-X	COM P-5	数字編集	外部浮動	内部浮動	日本語	動的長日本語	日本語編集	外部ブール
集団項目	○	○	△	△	○	△	△	△	△	△	△	△	△	△	△	△	△	△	×
可変長集団項目	○	○	△	△	○	△	△	△	△	△	△	△	△	△	△	△	△	△	×
日本語集団項目	△	△	○	×	×	×	×	×	×	×	×	×	×	×	×	(1)	×	(1)	×
英字	△	△	×	(1)	(1)	×	×	×	×	×	×	×	×	×	×	×	×	×	×
英数字	○	○	×	(1)	(1)	×	(1)	○	×	×	×	×	○	○	×	×	×	×	○
動的長英数字	△	△	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×	×	×
英数字編集	△	△	×	×	(1)	×	(2)	×	×	×	×	×	×	×	×	×	×	×	×
外部10進	△	△	×	×	○	×	×	(3)	×	×	×	×	×	×	×	×	×	×	×
内部10進	△	△	×	×	×	×	×	(4)	×	×	×	×	×	×	×	×	×	×	×
2進	△	△	×	×	×	×	×	×	(5)	×	(5)	×	×	×	×	×	×	×	×
2進 (COMP-X)	△	△	×	×	×	×	×	×	×	○	×	×	×	×	×	×	×	×	×
2進 (COMP-5)	△	△	×	×	×	×	×	×	(5)	×	(5)	×	×	×	×	×	×	×	×
数字編集	△	△	×	×	○	×	×	×	×	×	×	×	(6)	×	×	×	×	×	×
外部浮動	△	△	×	×	○	×	×	×	×	×	×	×	×	(7)	×	×	×	×	×
内部浮動	△	△	×	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×	×
日本語	△	△	(1)	×	×	×	×	×	×	×	×	×	×	×	×	(1)	×	(1)	×
動的長日本語	△	△	×	×	×	×	×	×	×	×	×	×	×	×	×	×	○	×	×
日本語編集	△	△	(1)	×	×	×	×	×	×	×	×	×	×	×	×	(1)	×	(2)	×
外部ブール	×	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×	×	×	○

(凡例)

○：適合していると判定されます。

△：引数の受け渡し内容渡し (BY CONTENT) または値渡し (BY VALUE) の場合は適合している、参照渡し (BY REFERENCE) の場合は適合していないと判定されます。

×

(n)：番号 n (1～7) の条件を満たしている場合は、適合していると判定されます。

1：JUSTIFIED 句が一致している

2：JUSTIFIED 句と PICTURE 文字列が一致している

3：PICTURE 文字列の符号有無と小数部桁数および SIGN 句が一致している

4：PICTURE 文字列の符号有無と小数部桁数が一致している

5：PICTURE 文字列の符号有無と小数部桁数が一致している (ただし、BY CONTENT 指定または BY VALUE 指定のとき、実引数が符号なしなら符号有無は一致とみなす)。

6：PICTURE 文字列と BLANK WHEN ZERO が一致している (DECIMAL-POINT IS COMMA 句は考慮されない)。

7：PICTURE 文字列が一致している (DECIMAL-POINT IS COMMA 句は考慮される)。

注意事項

- 日本語集団項目は、日本語項目として扱われます。
- 英字項目、外部 10 進項目、英数字編集項目および数字編集項目の部分参照は、英数字項目として扱われます。また、日本語編集項目の部分参照は、日本語項目として扱われます。そのほかのデータ項目の部分参照は、データ項目のデータ型を持つものとして扱われます。
- 2 進項目（COMP-5 項目を含む）で、かつ BY CONTENT 指定または BY VALUE 指定の場合は、実引数が符号なしで仮引数が符号ありの組み合わせでも適合として扱われます。
- 実引数と仮引数の両方がオブジェクト参照項目の場合、USAGE 句のクラス名やインタフェース名の指定に違いがあっても適合として扱われます。
- 数字編集項目の PICTURE 句の等価判断では、DECIMAL-POINT IS COMMA 句の差異は考慮されません。

(b) 報告内容

実引数と仮引数のデータ型が適合していない場合、不整合引数一覧 CSV ファイルに不整合種別が「データ型不適合」の不整合引数情報を出力します。また、不整合引数一覧のタグファイル警告メッセージを出力します。

各ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

項番,<<呼出元情報>>,"-",引数番号,<<実引数情報>>,呼出先プログラム名,"-",<<仮引数情報>>,"データ型不適合"

不整合引数一覧 CSV ファイルに出力される情報の内容については、「[8.2.6 不整合引数一覧](#)」を参照してください。

不整合引数一覧タグファイル

```
# [項番] ソースファイル名, ソース行番号 CALL '呼出先プログラム名' (第<引数番号>引数)
データ型不適合
"パス名1", line 行番号1: KEDL4020A-W CALL文の*** 1 ***番目の実引数のデータ型が仮引数の
データ型と適合していません。呼出元プログラム=*** 2 ***, 呼出先プログラム=*** 3 ***
"パス名2", line 行番号2: KEDL4021A-I 実引数=*** 1 *** データ型=*** 2 ***
"パス名3", line 行番号3: KEDL4022A-I 実引数=*** 1 *** データ型=*** 2 ***
```

不整合引数一覧タグファイルに出力される情報の内容を次に示します。

[項番]

不整合引数一覧.csv 内の不整合引数情報の通番です。

[ソースファイル名]

CALL 文のソースファイルのフォルダパス名なしのファイル名です。

[ソース行番号]

CALL 文の [ソース表示(COPY 展開後)] 画面上の行番号です。

[パス名 1]

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 1]

CALL 文のファイル (パス名 1) 内の相対行番号です。

[パス名 2]

実引数がデータ項目の場合

データ記述項の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

実引数がデータ項目以外の場合

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 2]

実引数がデータ項目の場合

実引数のデータ記述項のファイル (パス名 2) 内の相対行番号です。

実引数がデータ項目以外の場合

CALL 文のファイル (パス名 2) 内の相対行番号です。

[パス名 3]

仮引数のデータ記述項の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 3]

仮引数のデータ記述項のファイル (パス名 3) 内の相対行番号です。

(4) サイズのチェック

(a) 検査内容

実引数と仮引数のサイズが一致しているかどうかを検査します。ただし、次の場合、サイズチェックの検査対象から除外されます。

- [解析生成オプション設定] 画面の [引数整合性チェック] タブで [次の条件をすべて満たすケースを検出しません] を有効 (チェックボックスがオン) にして、かつ実引数と仮引数が次の条件をすべて満たす場合
 - 参照渡し、または内容渡しである
 - 実引数と仮引数の一方または両方が集団項目である
 - 実引数サイズ > 仮引数サイズである
- [解析生成オプション設定] 画面の [引数整合性チェック] タブで [一方が可変長集団項目の場合はサイズ検査対象外とする] を有効 (チェックボックスがオン) にして、かつ実引数または仮引数の一方が可変長集団項目の場合

なお、無効（チェックボックスがオフ）にしている場合は、可変長集団項目は最大長を使って検査されます。

- 実引数が部分参照で長さが省略、変数または式の場合
- 実引数または仮引数のどちらかが LIMIT 指定のない動的長基本項目の場合

(b) 報告内容

実引数と仮引数のサイズが一致していない場合、不整合引数一覧 CSV ファイルに不整合種別が「サイズ不一致」の不整合引数情報を出力します。また、不整合引数一覧タグファイルに警告メッセージを出力します。

各ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

項番,<<呼出元情報>>,"-",引数番号,<<実引数情報>>,呼出先プログラム名,"-",<<仮引数情報>>,"サイズ不一致"

不整合引数一覧 CSV ファイルに出力される情報の内容については、「[8.2.6 不整合引数一覧](#)」を参照してください。

不整合引数一覧タグファイル

```
# [項番] ソースファイル名, ソース行番号 CALL '呼出先プログラム名' (第<引数番号>引数) サイズ不一致
"パス名1", line 行番号1: KEDL4030A-W CALL文の*** 1 ***番目の実引数のサイズが仮引数のサイズと一致していません。呼出元プログラム=*** 2 ***, 呼出先プログラム=*** 3 ***
"パス名2", line 行番号2: KEDL4031A-W 実引数=*** 1 *** サイズ=*** 2 ***
"パス名3", line 行番号3: KEDL4032A-W 仮引数=*** 1 *** サイズ=*** 2 ***
```

不整合引数一覧タグファイルに出力される情報の内容を次に示します。

[項番]

不整合引数一覧.csv 内の不整合引数情報の通番です。

[ソースファイル名]

CALL 文のソースファイルのフォルダパス名なしのファイル名です。

[ソース行番号]

CALL 文の [ソース表示(COPY 展開後)] 画面上の行番号です。

[パス名 1]

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 1]

CALL 文のファイル (パス名 1) 内の相対行番号です。

[パス名 2]

実引数がデータ項目の場合

データ記述項の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

実引数がデータ項目以外の場合

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 2]

実引数がデータ項目の場合

データ記述項のファイル（パス名 2）内の相対行番号です。

実引数がデータ項目以外の場合

CALL 文のファイル（パス名 2）内の相対行番号です。

[パス名 3]

仮引数のデータ記述項の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 3]

仮引数のデータ記述項のファイル（パス名 3）内の相対行番号です。

(5) 集団項目の構造チェック

(a) 検査内容

実引数と仮引数が集団項目の場合に、集団項目に属するデータ項目の構造が一致しているかを検査します。

実引数と仮引数の集団項目に属するすべての従属項目（再定義項目と再命名項目を除く）が、次に示す条件を満たしている場合、集団項目の構造が一致しているとみなされます。

1. 従属項目の集団項目先頭からのオフセットが等しい
2. 従属項目の OCCURS 句の有無と反復回数（DEPENDING ON 指定の場合は最大反復回数）が等しい
3. 従属項目のデータ構造が等しい

従属項目のデータ構造の例を次に示します。

01 集団項目A.		01 集団項目B.
02 従属項目A-1 PIC X(5).	←→	02 従属項目B-1 PIC X(5).
02 従属項目A-2.	←→	02 従属項目B-2.
03 従属項目A-2-1 PIC 9(4).		03 従属項目B-2-1 PIC.
:		:
02 従属項目A-N PIC S9(9) COMP.	←→	02 従属項目B-N PIC S9(9) COMP.

(凡例)

←→ : 従属項目の対応を示します。

次の条件を満たす場合、データ構造は等しいとして扱われます。

- 集団項目同士の場合、集団項目の構造が等しい（構造を再帰的に検査する）
- 基本項目同士の場合、サイズが等しい（データ型は検査しない）
- 基本項目と集団項目の場合は、サイズが等しい

従属項目が被再定義項目の場合は、被再定義項目のデータ構造が非等価でも、再定義項目の中にデータ構造が一致するものがあれば、データ構造は等しいとして扱われます。

注意事項

- 次の場合、集団項目の構造チェックは実施されません。
 - ・ 実引数と仮引数の受渡方法が不一致の場合
 - ・ 実引数または仮引数の一方が集団項目ではない場合
 - ・ 実引数がファイル節、作業場所節、局所場所節、連絡節の集団項目ではない場合
 - ・ 実引数が再命名項目である場合
 - ・ 実引数が部分参照である場合
- 実引数と仮引数の集団項目のサイズが異なる場合、2つの集団項目を重ね合わせたときに領域が重なる部分だけが検査されます。

(b) 報告内容

実引数と仮引数の集団項目の構造が一致していない場合、不整合引数一覧 CSV ファイルに不整合種別が「構造不一致」の不整合引数情報を出力します。また、不整合引数一覧タグファイルに警告メッセージを出力します。

各ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

項番,<<呼出元情報>>,"-",引数番号,<<実引数情報>>,呼出先プログラム名,"-",<<仮引数情報>>,"構造不一致"

不整合引数一覧 CSV ファイルに出力される情報の内容については、「8.2.6 不整合引数一覧」を参照してください。

不整合引数一覧タグファイル

[項番] ソースファイル名, ソース行番号 CALL '呼出先プログラム名' (第<引数番号>引数) 構造不一致
"パス名1", line 行番号1: KEDL4040A-W CALL文の*** 1 ***番目の実引数の集団項目の構造が仮引数の集団項目の構造と一致していません。呼出元プログラム=*** 2 ***, 呼出先プログラム=*** 3 ***

集団項目の構造チェックでは、不一致の内容によって出力される警告メッセージが異なります。

集団項目先頭からのオフセットが不一致の場合

"パス名2", line 行番号2: KEDL4041A-W 実引数 集団項目=*** 1 *** 従属項目=*** 2 **
* オフセット=*** 3 ***
"パス名3", line 行番号3: KEDL4042A-W 仮引数 集団項目=*** 1 *** 従属項目=*** 2 **
* オフセット=*** 3 ***

OCCURS 句の有無または反復回数が不一致の場合

"パス名2", line 行番号2: KEDL4043A-W 実引数 集団項目=*** 1 *** 従属項目=*** 2 **
* OCCURS=*** 3 ***
"パス名3", line 行番号3: KEDL4044A-W 仮引数 集団項目=*** 1 *** 従属項目=*** 2 **
* OCCURS=*** 3 ***

サイズが不一致の場合

```
"パス名2", line 行番号2: KEDL4045A-W 実引数 集団項目=*** 1 *** 従属項目=*** 2 **
* サイズ=*** 3 ***
"パス名3", line 行番号3: KEDL4046A-W 仮引数 集団項目=*** 1 *** 従属項目=*** 2 **
* サイズ=*** 3 ***
```

不整合引数一覧タグファイルに出力される情報の内容を次に示します。

[項番]

不整合引数一覧.csv 内の不整合引数情報の通番です。

[ソースファイル名]

CALL 文のソースファイルのフォルダパス名なしのファイル名です。

[ソース行番号]

CALL 文の [ソース表示(COPY 展開後)] 画面上の行番号です。

[パス名 1]

CALL 文の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 1]

CALL 文のファイル (パス名 1) 内の相対行番号です。

[パス名 2]

実引数のデータ記述項の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 2]

実引数のデータ記述項のファイル (パス名 2) 内の相対行番号です。

[パス名 3]

仮引数のデータ記述項の COBOL ソースファイルまたは登録集原文ファイルの絶対パス名です。

[行番号 3]

仮引数のデータ記述項のファイル (パス名 3) 内の相対行番号です。

(6) 対応する仮引数のない実引数の情報の検出

実引数個数>仮引数個数の場合に、対応する仮引数のない実引数（実引数番号>仮引数個数）の情報を検出します。不整合引数一覧 CSV ファイルには、不整合種別が「対応する仮引数なし」の不整合引数情報を出力します。なお、不整合引数一覧タグファイルには警告メッセージは出力されません。

ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

```
項番,<<呼出元情報>>,"-",引数番号,<<実引数情報>>,呼出先プログラム名,"-", "-", "...", "-", "対応する仮引数なし"
```

仮引数情報のすべての列には「-」が出力されます。不整合引数一覧 CSV ファイルに出力される情報の内容については、「[8.2.6 不整合引数一覧](#)」を参照してください。

(7) 対応する実引数のない仮引数の情報の検出

実引数個数<仮引数個数の場合に、対応する実引数のない仮引数（仮引数番号>実引数個数）の情報を検出します。不整合引数一覧 CSV ファイルには、不整合種別が「対応する実引数なし」の不整合引数情報を出力します。なお、不整合引数一覧タグファイルには警告メッセージは出力されません。

ファイルのフォーマットを次に示します。

不整合引数一覧 CSV ファイル

項番,<<呼出元情報>>,"-",引数番号,"-",...,"-",呼出先プログラム名,"-",<<仮引数情報>>,"対応する実引数なし"

実引数情報のすべての列には「-」が出力されます。不整合引数一覧 CSV ファイルに出力される情報の内容については、「[8.2.6 不整合引数一覧](#)」を参照してください。

10.11.3 引数整合性チェック使用時の注意事項

引数整合性チェック使用時の注意事項を説明します。

1. 呼出先のプログラムがプロジェクト内に存在する CALL 文だけが引数整合性チェックの対象になります。
2. 一意名指定の CALL 文の場合、呼出先のプログラムが自動解決や一意名呼び出し対応付けファイルで解決したものだけが引数整合性チェックの対象になります。
3. 仮引数として引数整合性チェックの対象になるのは、手続き部（PROCEDURE DIVISION）の USING 指定に指定したデータ項目（仮引数）だけです。手続き部（PROCEDURE DIVISION）の USING 指定に指定していない仮引数の被再定義項目や再定義項目は、引数整合性チェックでは無視されます。

（例 1）被再定義項目が仮引数の例

```
LINKAGE SECTION.  
  01 PARAM1 PIC X(10).  
  01 PARAM2 REDEFINES PARAM1 PIC N(5).  *> 無視  
PROCEDURE DIVISION USING PARAM1.  
  :
```

（例 2）再定義項目が仮引数の例

```
LINKAGE SECTION.  
  01 PARAM1 PIC X(10).  *> 無視  
  01 PARAM2 REDEFINES PARAM1 PIC N(5).  
PROCEDURE DIVISION USING PARAM2.  
  :
```

4. 返却項目と返却値受取項目は、引数整合性チェックの対象外です。
5. ENTRY 文で指定している入口点の呼び出しの引数は、引数整合性チェックの対象外です。
6. 利用者定義関数とメソッドの呼び出しの引数は、引数整合性チェックの対象外です。

7. 実引数がオブジェクトプロパティの場合、データ型のチェック、サイズのチェック、および集団項目の構造チェックは実施されません。
8. 実引数の受渡方法が BY ATTRIBUTE 指定の場合、データ型のチェック、サイズのチェック、および集団項目の構造チェックは実施されません。
9. 集団項目の構造チェックは、集団項目のすべての従属項目を検査するため、従属項目数が多い集団項目を引数に多数指定している場合は、時間が掛かることがあります。
10. 不整合引数一覧 CSV ファイルや不整合引数一覧タグファイルに実引数の情報として出力されるデータ名には、添字や部分参照子の情報は付加されません。
11. 引数整合性チェックを開始する前に、不整合引数一覧タグファイルは削除されます。引数整合性チェックで不整合が検出されない場合、不整合引数一覧タグファイルは作成されません。
12. 引数不整合が検出されてタグファイルに警告メッセージが出力されても、COBOL ソースファイルの解析状態や生成状態は「警告」にはなりません。

10.12 主プログラムの指定方法

COBOL ソース解析では、システム資産の棚卸しを支援するため、次の「使用有無」の情報をドキュメントに出力しています。

- COBOL ソースファイルの使用／未使用の情報
- プログラムの使用／未使用の情報
- 節や段落の使用／未使用の情報
- ファイルの使用／未使用の情報
- データの使用／未使用の情報
- デッドコード情報（実行される可能性がない文、プログラムなどの情報）

これらの情報の基となる「プログラムの使用／未使用の情報」は、次のプログラムの使用／未使用の定義を使って抽出します。

プログラムの使用／未使用の定義

- 主プログラムは、常に「使用プログラム」です。
- 主プログラムから直接的または間接的に呼び出されるプログラムは、「使用プログラム」です。
- 主プログラムから直接的にも間接的にも呼び出されないプログラムは、「未使用プログラム」です。

ここで、主プログラムとは、プロジェクトの中で最初に制御が渡る COBOL プログラムであり、次のプログラムを指します。

- オペレーティングシステムから呼び出される外部プログラム（-Main コンパイラオプションで主プログラム指定するプログラム）
- COBOL 言語以外のプログラムから呼び出される外部プログラム
- 部品ライブラリの外部プログラム

正しい「プログラムの使用／未使用の情報」を抽出するには、解析時に、プロジェクト内の主プログラムがすべて正しく認識できている必要があります。

プロジェクト内の主プログラムは、[解析生成オプション設定] 画面の [主プログラム指定] タブで指定します。指定方法には、プロジェクト内のプログラムの中から自動検出する方法と、主プログラム名を指定する方法の 2 つがあります。

デフォルトの設定では、次のプログラムが主プログラムとして自動検出されるようになっています。検出条件の初期値については、[10.12.1 主プログラム自動検出](#) を参照してください。

- 引数のない COBOL ソースファイル先頭のプログラム
- メインフレームの主プログラムの引数の特徴（可変長文字列型の集団項目）を持つ COBOL ソースファイル先頭のプログラム

- Windows および UNIX の主プログラムの引数の特徴（argc と argv）を持つ COBOL ソースファイル先頭のプログラム

関連項目

- [\[5.2.2 \[主プログラム指定\] タブ\]](#)

10.12.1 主プログラム自動検出

資産分析の棚卸しでは、次に示すプログラムを取り扱うことが少なくありません。これらのプログラムでは、どれが主プログラムかがわからなくなっていることが多く、主プログラムが多数存在することもあります。そのため、主プログラムの洗い出しに多大な時間が掛かることがあります。

- 開発マネージャのプロジェクトファイルや、メイクファイルのない COBOL ソースファイル
- 保守ドキュメントのない COBOL ソースファイル
- 複数のプロジェクトが入り交じった状態の COBOL ソースファイル

このような場合は、主プログラム自動検出を使用します。主プログラム自動検出を有効にすると、プロジェクト内の、オペレーティングシステムから呼び出される外部プログラム※を、引数と返却項目の特徴から自動的に検出して主プログラムとして扱います。

注※

-Main コンパイラオプションで主プログラム指定するプログラムのことです。

主プログラム自動検出を使用する場合は、[解析生成オプション設定] 画面の [主プログラム指定] タブで [主プログラムを COBOL ソースファイル先頭の最外側プログラムの中から検出する] チェックボックスをチェックします。そして、主プログラムとして検出する COBOL ソースファイル先頭の最外側プログラムの特徴を [USING 検出条件] と [RETURNING 検出条件] で指定します。USING 検出条件の各項目は OR 条件です。USING 検出条件と RETURNING 検出条件の両方を満たすものが主プログラムとして検出されます。

USING 検出条件をチェックした場合の動作を次の表に示します。なお、USING 検出条件のすべての項目のチェックを外した場合は、COBOL ソースファイル先頭のすべての最外側プログラムが USING 検出条件を満たすものとして扱われます。

表 10-16 USING 検出条件をチェックした場合の動作

USING 検出条件	USING 検出条件をチェックした場合の動作	初期値
引数なし	COBOL ソースファイルの先頭の最外側プログラムが引数を持たない場合に、USING 指定の条件を満たすプログラムとして扱います。 コマンドラインに引数を持たないアプリケーションの主プログラムの特徴です。	チェックあり

USING 検出条件	USING 検出条件をチェックした場合の動作	初期値
引数 1 個で英数字である	<p>COBOL ソースファイルの先頭の最外側プログラムが引数 1 個で参照渡し の英数字項目の場合に、USING 指定の条件を満たすプログラムとして扱 います。</p> <p>コマンドラインに引数を持つメインフレームアプリケーションの主プログ ラムで使用されている可能性のあるインタフェースです。</p>	チェックなし
引数 1 個で可変長文字列 型の集団項目である	<p>COBOL ソースファイルの先頭の最外側プログラムが引数 1 個で参照渡し の可変長文字列型の集団項目の場合に、USING 指定の条件を満たすプログ ラムとして扱います。可変長文字列型の集団項目とは、長さ領域の 2 バイ ト 2 進整数と文字列領域の英数字項目（集団項目も可）の従属項目だけで 構成される集団項目です。</p> <p>コマンドラインに引数を持つメインフレームアプリケーションの主プログ ラムの特徴です。</p> <p>(例 1)</p> <pre> 01 ARG. 02 ARG-LENGTH PIC 9(4) USAGE COMP. 02 ARG-STRING PIC X(100). </pre> <p>(例 2)</p> <pre> 01 ARG. 02 ARG-LENGTH PIC 9(4) USAGE COMP. 02 ARG-STRING OCCURS 100 TIMES. 03 ARG-STRING-C PIC X. </pre>	チェックあり
引数 2 個で 2 進整数とポ インタである	<p>COBOL ソースファイルの先頭の最外側プログラムが引数 2 個で、第 1 引 数が値渡しの 2 進整数項目、第 2 引数が値渡しのポインタ項目または参照 渡しでポインタ項目だけから成る集団項目（従属基本項目がすべてポイン タ項目）の場合に、USING 指定の条件を満たすプログラムとして扱いま す。ポインタ項目はアドレス項目と同じ意味です。</p> <p>コマンドラインに引数 (argc, argv) を持つ UNIX や Windows のアプリ ケーションの主プログラムの特徴です。</p> <p>(例 1)</p> <pre> LINKAGE SECTION. 01 ARGC PIC 9(9) COMP. 01 ARGV USAGE ADDRESS. PROCEDURE DIVISION USING BY VALUE ARGC ARGV. </pre> <p>(例 2)</p> <pre> LINKAGE SECTION. 01 ARGC PIC 9(9) COMP. 01 ARGV. 02 ARGV-LIST USAGE POINTER OCCURS 10 TIMES. PROCEDURE DIVISION USING BY VALUE ARGC BY REFERENCE ARGV. </pre>	チェックあり

RETURNING 検出条件をチェックした場合の動作を次の表に示します。なお、RETURNING 検出条件の
チェックを外した場合、COBOL ソースファイルの先頭の最外側プログラムは、常に RETURNING 検出
条件を満たすものとして扱われ、USING 検出条件だけで主プログラムとするかどうかは決定されます。

表 10-17 RETURNING 検出条件をチェックした場合の動作

RETURNING 検出条件	RETURNING 検出条件をチェックした場合の動作	初期値
RETURNING 指定なし	COBOL ソースファイルの先頭の最外側プログラムが RETURNING 指定を持たない場合に、RETURNING 指定の条件を満たすプログラムとして扱います。	チェックあり

注意事項

- 次の主プログラムは、主プログラム自動検出では検出できません。
 - ・ COBOL 言語以外のプログラムから呼び出される外部プログラム
 - ・ 部品ライブラリの外部プログラム

これらの主プログラムを「主プログラム」扱いにする場合は、主プログラム名を明示的に指定する方法を使用します。主プログラム名を明示的に指定する方法については、「[10.12.2 主プログラム名による主プログラム指定](#)」を参照してください。

- 主プログラム自動検出では、引数の条件が一致すれば、主プログラムでないプログラムでも主プログラムとして扱われます。このため、システムのプログラム構成によっては、主プログラムでないプログラムが主プログラムとして検出されてしまうことがあります。検出条件を調整しても主プログラムでないプログラムが主プログラムとして検出されてしまう場合は、主プログラム名を明示的に指定する方法に切り替えてください。手順を次に示します。

手順

1. 主プログラム自動検出による解析結果から主プログラム扱いのプログラムの一覧を作成する
2. 主プログラム扱いのプログラムの COBOL ソースファイルの一つ一つ目視で確認して、主プログラムかどうかを正確に判断し、正確な主プログラムの一覧を作成する
3. 手順 2 で作成した一覧を基に、主プログラム名を明示的に指定する

主プログラム名を明示的に指定する方法については、「[10.12.2 主プログラム名による主プログラム指定](#)」を参照してください。

10.12.2 主プログラム名による主プログラム指定

主プログラムの名称がわかっているプロジェクトの場合は、主プログラム名で主プログラムを指定します。また、次のようなプロジェクトの場合も、COBOL ソースファイルからプロジェクトの主プログラムの一覧を洗い出した上で、主プログラム名で主プログラムを指定します。

- ・ アプリケーションの主プログラムが COBOL 言語以外のプログラムのプロジェクトの場合
- ・ 部品ライブラリのプロジェクトの場合（エクスポートする外部プログラムがすべて主プログラム）

主プログラム名で主プログラムを指定するには、[解析生成オプション設定] 画面の [主プログラム指定] タブで [主プログラムを COBOL ソースファイル先頭の最外側プログラムの中から検出する] チェック

ボックスのチェックを外し、[主プログラム名の指定] に主プログラムとするプログラム名の一覧を指定します。

主プログラム名の一覧は、次のどちらかの方法で指定します。

- 1 行ずつ改行して主プログラム名を指定する

(例)

主プログラム名の指定

主プログラム名 1 ↵ 主プログラム名 2 ↵ 主プログラム名 3 ↵ 主プログラム名 4 ↵

- コンマで区切って、1 行に複数の主プログラム名を指定する

(例)

主プログラム名の指定

主プログラム名 1, 主プログラム名 2 ↵ 主プログラム名 3, 主プログラム名 4, 主プログラム名 5 ↵ 主プログラム名 6 ↵

注意事項

- 主プログラムの名称は、COBOL ソースファイルで使用している名称はなく、変換規則に従って変換される名称を指定します。プログラム名の変換規則については、「[10.9 解析結果に出力される COBOL ソースファイル中の名称](#)」を参照してください。

(例)

AB-CD → AB_CD

'#PROGRAM@01' → 'JPROGRAM201'

'0#¥@-1' → 'JJAB_1'

- 主プログラム自動検出と、主プログラム名による主プログラム指定は併用できます。併用する場合は「主プログラムを COBOL ソースファイル先頭の最外側プログラムの中から検出する」チェックボックスをチェックしたままにします。主プログラム自動検出で主プログラムと認識できないプログラムを補足で指定する場合に使用します。

10.13 プログラムコメントの抽出

プログラムコメントの抽出方法について説明します。

10.13.1 プログラムコメント抽出とは

プログラムコメント抽出とは、プログラムに対して書かれたコメントの中から、プログラムの機能を要約する文字列（プログラムコメントと呼びます）を抽出して、プログラム一覧や、プログラム概要のドキュメントに出力する機能です。COBOLでは、プログラム名を英数字の略号で記述することが多く、そのようなプログラムではプログラム名から機能を推測することが困難です。プログラムコメント抽出を使用することで、プログラムの機能要約が記載された保守ドキュメントが生成できます。

プログラムコメントの抽出例を次の図に示します。

図 10-5 プログラムコメントの抽出例



デフォルトの設定では、プログラムコメントは抽出されません。プログラムコメントを抽出するには、[解析生成オプション設定] 画面の [プログラムコメント抽出] タブで [プログラムコメントを抽出する] チェックボックスをチェックします。

10.13.2 プログラムコメントの抽出位置の指定

プログラムコメントが記述されているコメント行をプログラムコメント行と呼びます。このプログラムコメント行の検索場所と検索条件を [プログラムコメント抽出] タブの [抽出位置] で指定します。システム開発では、プロジェクトのコーディング規則によって、プログラムコメントの記述方法を定めるのが通例です。その規則が示すプログラムコメント行の場所を、検索場所と検索条件で指定します。

(1) プログラムコメント行の検索場所の指定

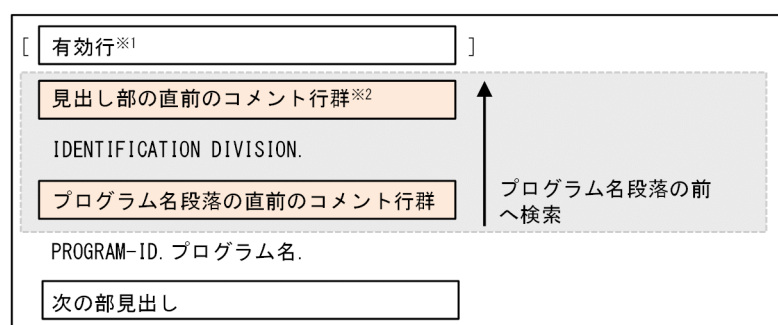
[抽出位置] の [プログラムコメント行の検索場所] で、プログラムコメント行がプログラム名段落の前にあるか、プログラム名段落のあとにあるかを指定します。デフォルトの設定では、[プログラム名段落の前] オプションボタンが選択されています。

プログラムコメント行の検索場所

☒ プログラム名段落の前

☐ プログラム名段落の後

[プログラム名段落の前] オプションボタンを選択すると、プログラム名段落 (PROGRAM-ID) の 1 行前の行からソースファイルの先頭に向けて、検索条件を満たすコメント行が検索されます。ファイルの先頭、または見出し部の部見出し (IDENTIFICATION DIVISION) の前にある有効行に到達するまで検索されます。



(凡例)

: 検索範囲を示します。

注※1 前のプログラムのEND PROGRAM. または親プログラムの最後の文のことです。

注※2 見出し部の部見出し (IDENTIFICATION DIVISION) の直前の行からソースファイルの先頭に向けて連続するコメント行です。ファイル先頭または有効行に到達するまでのコメント行が対象です。

[プログラム名段落の後] オプションボタンを選択すると、プログラム名段落 (PROGRAM-ID) の次の行から見出し部の末尾まで、検索条件を満たすコメント行が検索されます。

IDENTIFICATION DIVISION.

PROGRAM-ID. プログラム名.

プログラム名段落の直後のコメント行群 ↓ プログラム名段落の後へ検索

次の部見出し

(凡例)

検索範囲を示します。

注意事項

- コメント行は、注記行、空白行、デバッグ行、または翻訳指令行です。
- 空白行、デバッグ行および翻訳指令行から抽出するコメント行の文字列は、空文字列（""）になります。

(2) プログラムコメント行の検索条件の指定

[抽出位置] の [プログラムコメント行の検索条件] で、プログラムコメント行の検索条件を指定します。

プログラムコメント行の検索条件

☐ 抽出キー文字列で始まるコメント行（プログラムコメントは抽出キーの後の文字列）

☒ 抽出キー文字列で始まるコメント行の次行
抽出キー文字列

☐ プログラム名段落から指定行目の位置にあるコメント行
 行目(1~99)

(a) 空文字列でない最初のコメント行をプログラムコメント行として検出する

[抽出キー文字列で始まるコメント行] オプションボタンを選択して、[抽出キー文字列] を空にします。

この設定で解析／生成すると、検索場所で、最初の空文字列でないコメント行がプログラムコメント行として検出され、行の文字列がプログラムコメントとして抽出されます。

IDENTIFICATION DIVISION の前にプログラムコメントがある場合

COBOL ソースファイルの例

```
000100
000200* コメント1
000300
000400 IDENTIFICATION DIVISION.
000500 PROGRAM-ID. プログラム名. ... プログラム名段落
```

- 「コメント 1」をプログラムコメントとして抽出するための指定

[プログラムコメント行の検索場所]：

[プログラム名段落の前] オプションボタンを選択する。

[プログラムコメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

何も指定しない。

PROGRAM-ID の前後にプログラムコメントがある場合

COBOL ソースファイルの例

```
000100 IDENTIFICATION DIVISION.  
000200* コメント1  
000300  
000400 PROGRAM-ID. プログラム名.      ... プログラム名段落  
000500  
000600* コメント2
```

- 「コメント 1」をプログラムコメントとして抽出するための指定

[プログラムコメント行の検索場所]：

[プログラム名段落の前] オプションボタンを選択する。

[プログラムコメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

何も指定しない。

- 「コメント 2」をプログラムコメントとして抽出するための指定

[プログラムコメント行の検索場所]：

[プログラム名段落の後] オプションボタンを選択する。

[プログラムコメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

何も指定しない。

注意事項

- [抽出キー文字列で始まるコメント行] オプションボタンの代わりに、[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択しても、結果は同じになります。
- 検索範囲内で空文字列でないコメント行が見つからない場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。

(b) 抽出キー文字列で始まる最初のコメント行をプログラムコメント行として検出する

[抽出キー文字列で始まるコメント行] オプションボタンを選択して、[抽出キー文字列] に抽出キーを指定します。

この設定で解析／生成すると、検索場所で、最初の抽出キー文字列で始まるコメント行がプログラムコメント行として検出され、抽出キー文字列の直後から行末尾までの文字列がプログラムコメントとして抽出されます。

COBOLソースファイルの例

```
000100* コメント
000200* 抽出キー コメント1
000300* コメント
000400 IDENTIFICATION DIVISION.
000500* コメント
000600 PROGRAM-ID. プログラム名.      ... プログラム名段落
000700* コメント
000800* 抽出キー コメント2
000900* コメント
001000* コメント
```

- 「コメント 1」をプログラムコメントとして抽出するための指定
[プログラムコメント行の検索場所]：
[プログラム名段落の前] オプションボタンを選択する。
[プログラムコメント行の検索条件]：
[抽出キー文字列で始まるコメント行] オプションボタンを選択する。
[抽出キー文字列]：
「抽出キー」を指定する。
- 「コメント 2」をプログラムコメントとして抽出するための指定
[プログラムコメント行の検索場所]：
[プログラム名段落の後] オプションボタンを選択する。
[プログラムコメント行の検索条件]：
[抽出キー文字列で始まるコメント行] オプションボタンを選択する。
[抽出キー文字列]：
「抽出キー」を指定する。

注意事項

- 検索範囲内に抽出キー文字列で始まるコメント行が見つからない場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。
- 抽出キー文字列のあとに空白以外の文字がないコメント行の場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。

(c) 抽出キー文字列で始まる最初のコメント行の次行をプログラムコメント行として検出する

[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択して、[抽出キー文字列] に抽出キーを指定します。

この設定で解析／生成すると、検索場所で、最初の抽出キー文字列で始まるコメント行の次の行がプログラムコメント行として検出され、行の文字列がプログラムコメントとして抽出されます。

COBOL ソースファイルの例

```
000100* コメント  
000200* 抽出キー コメント  
000300* コメント1  
000400 IDENTIFICATION DIVISION.  
000500* コメント  
000600 PROGRAM-ID. プログラム名.      ... プログラム名段落  
000700* コメント  
000800* 抽出キー コメント  
000900* コメント2  
001000* コメント
```

- 「コメント 1」をプログラムコメントとして抽出するための指定
[プログラムコメント行の検索場所]：
[プログラム名段落の前] オプションボタンを選択する。
[プログラムコメント行の検索条件]：
[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択する。
[抽出キー文字列]：
「抽出キー」を指定する。
- 「コメント 2」をプログラムコメントとして抽出するための指定
[プログラムコメント行の検索場所]：
[プログラム名段落の後] オプションボタンを選択する。
[プログラムコメント行の検索条件]：
[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択する。
[抽出キー文字列]：
「抽出キー」を指定する。

注意事項

- 検索範囲内に抽出キー文字列で始まるコメント行が見つからない場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。
- 抽出キー文字列で始まる行はコメント行でも、その次の行がコメント行でない、またはコメント文字列が空文字列のコメント行である場合があります。その場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。

(d) プログラム名段落から特定の行目にあるコメント行をプログラムコメント行として検出する

[プログラム名段落から指定行目の位置にあるコメント行] オプションボタンを選択して、[行目] にプログラム名段落から何行目のコメント行かを指定します。

この設定で解析／生成すると、検索場所で、プログラム名段落から指定行目にあるコメント行がプログラムコメント行として検出され、行の文字列がプログラムコメントとして抽出されます。

COBOL ソースファイルの例

000100*	-----	+8
000200*	コメント	+7
000300*	コメント1	+6
000400*	コメント	+5
000500*	-----	+4
000600	IDENTIFICATION DIVISION.	
000700		+3
000800*	コメント	+2
000900		+1
001000	PROGRAM-ID. プログラム名.	... プログラム名段落
001100		+1
001200*	コメント	+2
001300*	コメント2	+3
001400*	コメント	+4
001500	ENVIRONMENT DIVISION.	

(凡例)

+n: プログラム名段落からカウントした行数 (n行目) を示します。

- 「コメント 1」をプログラムコメントとして抽出するための指定

[プログラムコメント行の検索場所]:

[プログラム名段落の前] オプションボタンを選択する。

[プログラムコメント行の検索条件]:

[プログラム名段落から指定行目の位置にあるコメント行] オプションボタンを選択する。

[行目]:

「6」を指定する。

- 「コメント 2」をプログラムコメントとして抽出するための指定

[プログラムコメント行の検索場所]:

[プログラム名段落の後] オプションボタンを選択する。

[プログラムコメント行の検索条件]:

[プログラム名段落から指定行目の位置にあるコメント行] オプションボタンを選択する。

[行目]:

「3」を指定する。

注意事項

- コメント行（注記行、空白行、デバッグ行、または翻訳指令行）でない行は、行数にはカウントされません。
- 指定行目のコメント行が検索範囲外のコメント行の場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。
- 指定行目のコメント行が空白行、デバッグ行、または翻訳指令行の場合、プログラムコメントはなしとなり、プログラムコメント行の文字列は空文字列となります。

関連項目

- [「5.2.3 \[プログラムコメント抽出\] タブ」](#)

10.13.3 プログラムコメント抽出時の注意事項

- プログラムコメントの抽出を有効にした場合、[抽出位置] で指定した位置からプログラムコメントが抽出できなかったプログラム（プログラムコメント行の文字列が空文字列となったプログラム）に対して、警告メッセージが出力されます。

10.14 節コメントの抽出

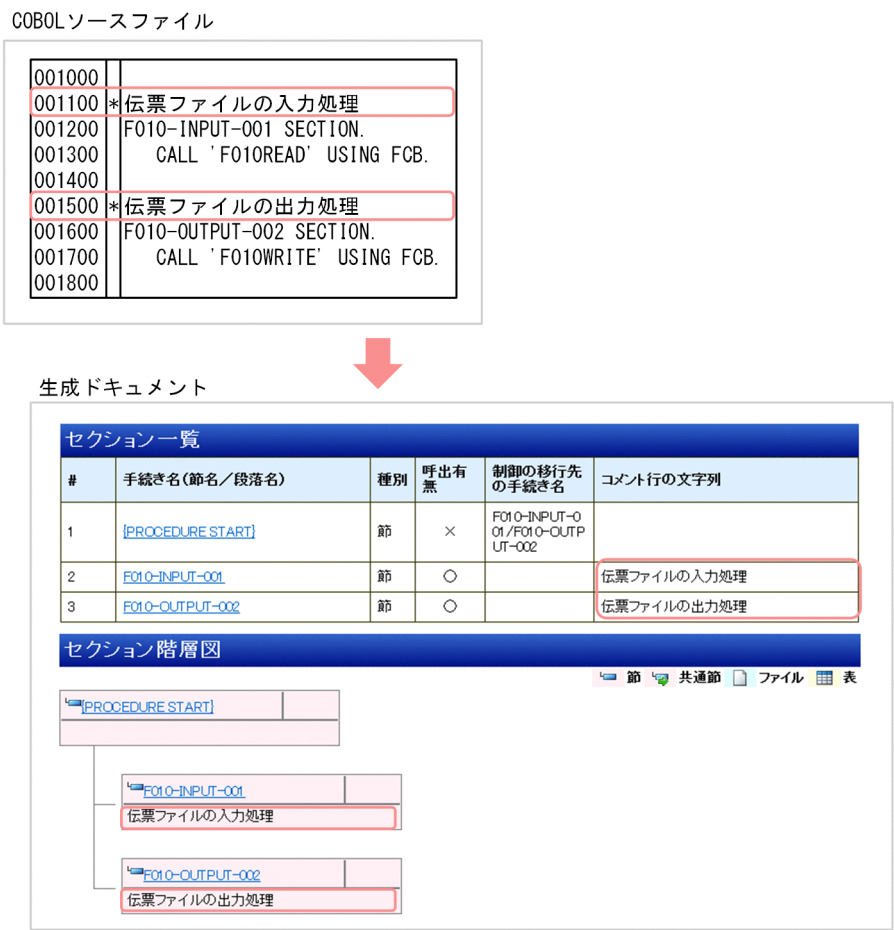
節コメントの抽出方法について説明します。

10.14.1 節コメント抽出とは

節コメント抽出とは、手続き部の節に対して書かれたコメントの中から、節の機能を要約する文字列（節コメントと呼びます）を抽出して、セクション一覧や、セクション階層図のドキュメントに出力する機能です。COBOL では、節名を英数字の略号で記述することが多く、そのようなプログラムでは節名から機能を推測することが困難です。節コメント抽出を使用することで、節の機能要約が記載された保守ドキュメントが生成できます。

節コメントの抽出例を次の図に示します。

図 10-6 節コメントの抽出例



デフォルトの設定では、節コメントは抽出されません。節コメントを抽出するには、[解析生成オプション設定] 画面の [節コメント抽出] タブで [節コメントを抽出する] チェックボックスをチェックします。

10.14.2 節コメントの抽出位置の指定

節コメントが記述されているコメント行を節コメント行と呼びます。この節コメント行の検索場所と検索条件を「節コメント抽出」タブの「抽出位置」で指定します。システム開発では、プロジェクトのコーディング規則によって、節コメントの記述方法を定めるのが通例です。その規則が示す節コメント行の場所を、検索場所と検索条件で指定します。

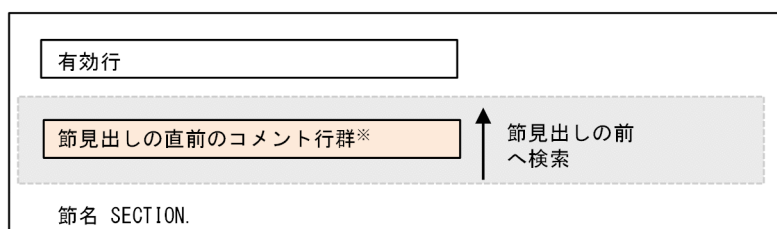
(1) 節コメント行の検索場所の指定

「抽出位置」の「節コメント行の検索場所」で、節コメント行が節見出しの前にあるか、節見出しのあとにあるかを指定します。デフォルトの設定では、「節見出しの前」オプションボタンが選択されています。

節コメント行の検索場所

- ☒ 節見出しの前
☐ 節見出しの後

「節見出しの前」オプションボタンを選択すると、節見出しの1行前の行からソースファイルの先頭に向けて、検索条件を満たすコメント行が検索されます。有効行が出現するまで検索されます。

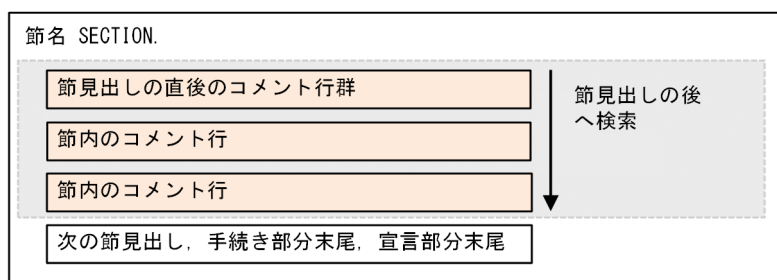


(凡例)

■ : 検索範囲を示します。

注※ 節見出しの直前の行からソースファイル先頭に向けて連続するコメント行です。
有効行が出現するまでのコメント行が対象です。

「節見出しの後」オプションボタンを選択すると、節見出しの次の行から節の末尾まで、検索条件を満たすコメント行が検索されます。



(凡例)

■ : 検索範囲を示します。

注意事項

- コメント行は、注記行、空白行、デバッグ行、または翻訳指令行です。
- 空白行、デバッグ行、および翻訳指令行から抽出するコメント行の文字列は、空文字列（""）になります。

(2) 節コメント行の検索条件の指定

[抽出位置] の [節コメント行の検索条件] で、節コメント行の検索条件を指定します。

節コメント行の検索条件

☐ 抽出キー文字列で始まるコメント行（節コメントは抽出キーの後の文字列）

☒ 抽出キー文字列で始まるコメント行の次行

抽出キー文字列

☐ 節見出しから指定行目の位置にあるコメント行

行目(1～99)

(a) 空文字列でない最初のコメント行を節コメント行として検出する

[抽出キー文字列で始まるコメント行] オプションボタンを選択して、[抽出キー文字列] を空にします。

この設定で解析／生成すると、検索場所で、最初の空文字列でないコメント行が節コメント行として検出され、行の文字列が節コメントとして抽出されます。

COBOLソースファイルの例

```
000100
000200* コメント1
000300
000400 節名 SECTION.      ... 節見出し
000500
000600* コメント2
000700
```

- 「コメント 1」を節コメントとして抽出するための指定

[節コメント行の検索場所]：

[節見出しの前] オプションボタンを選択する。

[節コメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

何も指定しない。

- 「コメント 2」を節コメントとして抽出するための指定

[節コメント行の検索場所]：

[節見出しの後] オプションボタンを選択する。

[節コメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

何も指定しない。

注意事項

- [抽出キー文字列で始まるコメント行] オプションボタンの代わりに、[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択しても、結果は同じになります。
- 検索範囲内で空文字列でないコメント行が見つからない場合、節コメントはなしとなり、節コメント行の文字列は空文字列となります。

(b) 抽出キー文字列で始まる最初のコメント行を節コメント行として検出する

[抽出キー文字列で始まるコメント行] オプションボタンを選択して、[抽出キー文字列] に抽出キーを指定します。

この設定で解析／生成すると、検索場所で、最初の抽出キー文字列で始まるコメント行が節コメント行として検出され、抽出キー文字列の直後から行末尾までの文字列が節コメントとして抽出されます。

COBOLソースファイルの例

```
000100* コメント  
000200* 抽出キー コメント1  
000300* コメント  
000400 節名 SECTION.          ... 節見出し  
000500* コメント  
000600* 抽出キー コメント2  
000700* コメント
```

- 「コメント 1」を節コメントとして抽出するための指定

[節コメント行の検索場所]：

[節見出しの前] オプションボタンを選択する。

[節コメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

「抽出キー」を指定する。

- 「コメント 2」を節コメントとして抽出するための指定

[節コメント行の検索場所]：

[節見出しの後] オプションボタンを選択する。

[節コメント行の検索条件]：

[抽出キー文字列で始まるコメント行] オプションボタンを選択する。

[抽出キー文字列]：

「抽出キー」を指定する。

注意事項

- 検索範囲内に抽出キー文字列で始まるコメント行が見つからない場合、節コメントはなしとなり、節コメント行の文字列は空文字列となります。
- 抽出キー文字列のあとに空白以外の文字がないコメント行の場合、節コメントはなしとなり、節コメント行の文字列は空文字列となります。

(c) 抽出キー文字列で始まる最初のコメント行の次行を節コメント行として検出する

[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択して、[抽出キー文字列] に抽出キーを指定します。

この設定で解析／生成すると、検索場所で、最初の抽出キー文字列で始まるコメント行の次の行が節コメント行として検出され、行の文字列が節コメントとして抽出されます。

COBOLソースファイルの例

```
000100* コメント  
000200* 抽出キー コメント  
000300* コメント1  
000400* コメント  
000500 節名 SECTION.          ... 節見出し  
000600* コメント  
000700* 抽出キー コメント  
000800* コメント2  
000900* コメント
```

- 「コメント 1」を節コメントとして抽出するための指定

[節コメント行の検索場所]：

[節見出しの前] オプションボタンを選択する。

[節コメント行の検索条件]：

[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択する。

[抽出キー文字列]：

「抽出キー」を指定する。

- 「コメント 2」を節コメントとして抽出するための指定

[節コメント行の検索場所]：

[節見出しの後] オプションボタンを選択する。

[節コメント行の検索条件]：

[抽出キー文字列で始まるコメント行の次行] オプションボタンを選択する。

[抽出キー文字列]：

「抽出キー」を指定する。

注意事項

- 検索範囲内に抽出キー文字列で始まるコメント行が見つからない場合、節コメントはなしとなり、節コメント行の文字列は空文字列となります。
- 抽出キー文字列で始まる行はコメント行でも、その次の行がコメント行でない、またはコメント文字列が空文字列のコメント行である場合があります。その場合、節コメントはなしとなり、節コメント行の文字列は空文字列となります。

(d) 節見出しから特定の行目にあるコメント行を節コメント行として検出する

[節見出しから指定行目の位置にあるコメント行] オプションボタンを選択して、[行目] に節見出しから何行目のコメント行かを指定します。

この設定で解析／生成すると、検索場所で、節見出しから指定行目にあるコメント行が節コメント行として検出され、行の文字列が節コメントとして抽出されます。

COBOL ソースファイルの例

000100*	-----	+6
000200*	コメント	+5
000300*	コメント1	+4
000400*	コメント	+3
000500*	-----	+2
000600		+1
000700	節名	
000800	SECTION.	} 節見出し
000900		+1
001000*	コメント	+2
001100*	コメント2	+3
001200*	コメント	+4
001300*	ENVIRONMENT DIVISION.	

(凡例)

+n : 節見出しからカウントした行数 (n行目) を示します。

- 「コメント 1」を節コメントとして抽出するための指定

[節コメント行の検索場所] :

[節見出しの前] オプションボタンを選択する。

[節コメント行の検索条件] :

[節見出しから指定行目の位置にあるコメント行] オプションボタンを選択する。

[行目] :

[4] を指定する。

- 「コメント 2」を節コメントとして抽出するための指定

[節コメント行の検索場所] :

[節見出しの後] オプションボタンを選択する。

[節コメント行の検索条件] :

[節見出しから指定行目の位置にあるコメント行] オプションボタンを選択する。

[行目]：

「3」を指定する。

■ 注意事項

- コメント行（注記行，空白行，デバッグ行，または翻訳指令行）でない行は，行数にはカウントされません。
- 指定行目のコメント行が検索範囲外のコメント行の場合，節コメントはなしとなり，節コメント行の文字列は空文字列となります。
- 指定行目のコメント行が空白行，デバッグ行，または翻訳指令行の場合，節コメントはなしとなり，節コメント行の文字列は空文字列となります。

関連項目

- [「5.2.4 \[節コメント抽出\] タブ」](#)

10.14.3 節コメント抽出時の注意事項

- 節コメントの抽出を有効にした場合，[抽出位置] で指定した位置から節コメントが抽出できなかった節（節コメント行の文字列が空文字列となった節）に対して，警告メッセージが出力されます。

10.15 そと PERFORM 文からの呼出先情報の抽出規則

そと PERFORM 文から呼出先（制御の移行先）の情報を抽出する規則について説明します。

そと PERFORM 文から抽出する呼出先情報

呼出先情報は、そと PERFORM 文から抽出する呼出先に関する情報です。この情報は、[プログラム詳細] 画面の「セクション一覧」の [呼出有無] と [制御の移行先の手続き名]、[セクション関連図] 画面のセクション間のリンクを構築するために使用されます。

呼出先情報（単一指定）

呼出先が単一指定の場合、そと PERFORM 文から抽出する呼出先情報は、次の情報を持ちます。

- 呼出先手続きの情報（手続き種別、節名、段落名）

[プログラム詳細] 画面の「セクション一覧」の [制御の移行先の手続き名] では、この情報を使って次のように表示します。

PERFORM 文	制御の移行先の手続き名
PERFORM 節名 1	節名 1
PERFORM 段落名 1 IN 節名 2	段落名 1 [節名 2]

呼出先情報（範囲指定）

呼出先が範囲指定（手続き名 1 THRU 手続き名 2）の場合、そと PERFORM 文から抽出する呼出先情報は、次の 2 つの情報を持ちます。

- 呼出先手続きの情報（手続き種別、節名、段落名）
- 出口手続きの情報（手続き種別、節名、段落名）

[プログラム詳細] 画面の「セクション一覧」の [制御の移行先の手続き名] では、この情報を使って次のように表示します。

PERFORM 文	制御の移行先の手続き名
PERFORM 節名 1 THRU 節名 2	節名 1：節名 2
PERFORM 節名 1 THRU 段落名 2 IN 節名 2	節名 1：段落名 2 [節名 2]
PERFORM 段落名 1 IN 節名 1 THRU 節名 2	段落名 1 [節名 1]：節名 2
PERFORM 段落名 1 IN 節名 1 THRU 段落名 2 IN 節名 2	段落名 1 [節名 1]：段落名 2 [節名 2]

[プログラム詳細] 画面の「セクション一覧」の [呼出有無] や、[セクション関連図] 画面のセクション間のリンクの構築では、呼出先手続きの情報だけが使用されます。

注意事項

〔プログラム詳細〕画面の「セクション一覧」の〔呼出有無〕では、抽出された呼出先情報（単一指定）と、呼出先情報（範囲指定）の呼出先手続きに格納されている節や段落だけが、「参照」として表示されます。

呼出先情報の抽出規則

呼出先情報の抽出規則を次に示します。

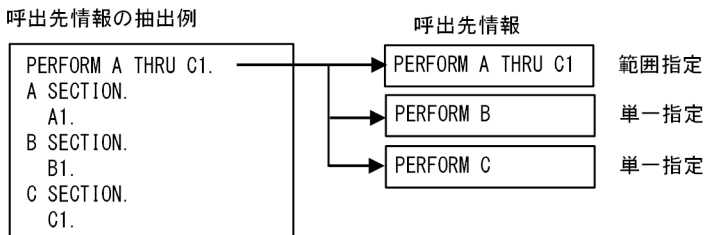
PERFORM 手続き名 1 の抽出規則

手続き名 1（節名／段落名）を呼出先手続きとして、呼出先情報（単一指定）を抽出します。

PERFORM 手続き名 1 THRU 手続き名 2（手続き名 1 は手続き名 2 の前）の抽出規則

- 手続き名 1 を呼出先手続き、手続き名 2 を出口手続きとして、呼出先情報（範囲指定）を抽出します。
- 手続き名 1 の末尾から手続き名 2 の末尾までの範囲に存在する、すべての節見出しの節名を使って呼出先情報（単一指定）を抽出します。

（例）呼出先情報の抽出例



注意事項

2 つ目の規則で抽出された呼出先情報（単一指定）の情報は、〔プログラム詳細〕画面の「セクション一覧」の〔制御の移行先の手続き名〕には出力されません。

PERFORM 手続き名 1 THRU 手続き名 2（手続き名 2 は手続き名 1 の前）の抽出規則

手続き名 1 を呼出先手続き、手続き名 2 を出口手続きとして、呼出先情報（範囲指定）を抽出します。

10.16 制御フロー解析による実行順序表示機能

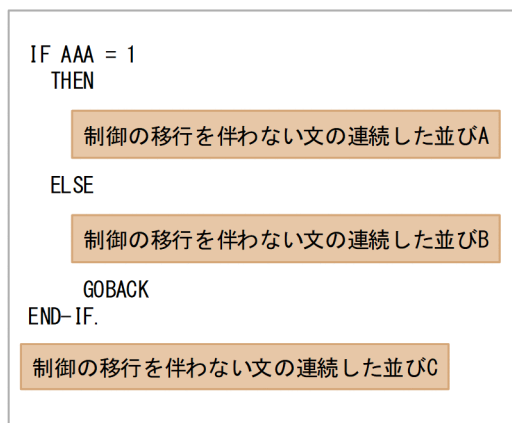
制御フロー解析と、制御フロー情報を使用する実行順序表示機能について説明します。

10.16.1 制御フロー解析とは

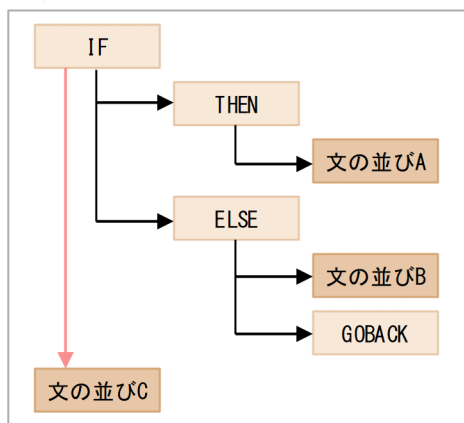
制御フロー解析とは、ソースプログラムを解析して作成した制御フローグラフを使って、プログラムの実行をシミュレートし、プログラムの実行状態を調べる手法です。制御フローグラフとは、プログラムが実行する全経路をグラフで表現したものです。このグラフのノードを制御ブロックといい、必ずセットで実行される文のまとまりを表します。また、それぞれの制御ブロックをつなぐエッジは、制御遷移（「制御の上から下への流れ」と「分岐や呼び出しによる制御の移行（ジャンプ）」）を表します。

図 10-7 制御フロー解析の概要

COBOLソースプログラム



制御フローグラフ



(凡例)

□ : 制御ブロック (■ は基本ブロック)
→ : エッジ

COBOL では次の単位で制御ブロックを作成します。

- ・ 節および段落
- ・ 制御の移行を表現する文（条件文、GO TO 文、PERFORM 文、GOBACK 文、STOP RUN 文など）
- ・ 文の条件指定（THEN/ELSE/WHEN/ [NOT] ON~/ [NOT] AT~など）
- ・ 制御の移行を伴わない文の連続した並び（基本ブロックといいます）

制御フロー解析には、制御ブロック情報のほかに次の 2 つの情報も必要です。

- ・ 文情報
- ・ 制御遷移情報（分岐や呼び出しによる制御の移行の情報）

これらの制御フロー解析に必要な 3 つの情報を総称して制御フロー情報といいます。

〔解析生成オプション設定〕画面の〔その他の解析オプション〕タブで〔プログラムの制御フロー情報（実行順序表示機能で使用する情報）を抽出する〕をチェックして制御フロー情報を抽出すると、実行順序表示機能が利用できるようになります。

10.16.2 実行順序表示機能とは

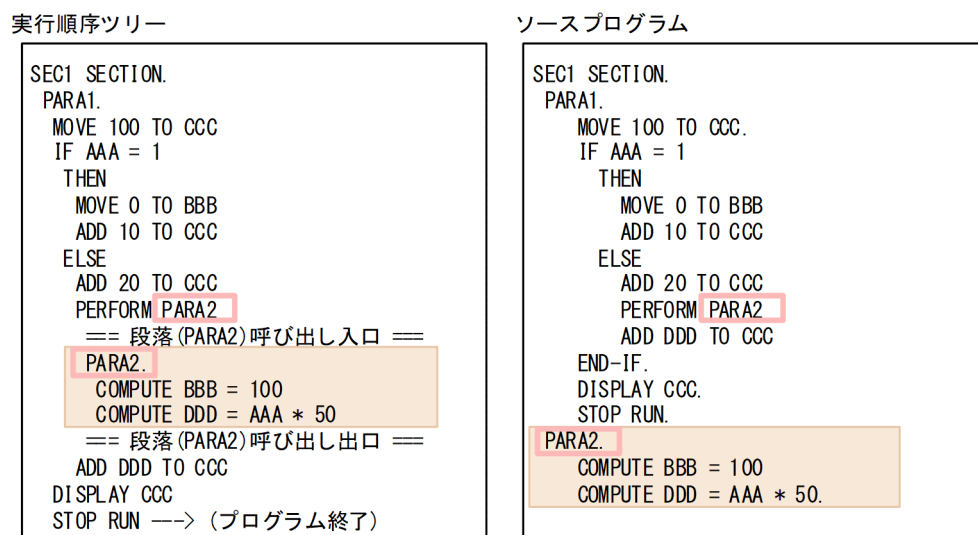
ソースコードの処理を理解するには、通常、ソースファイル（ソースプログラム）のソースコードを読んで行います。このソースコードの読解方法をソースコードリーディングといいます。ソースコードリーディングには、次のメリットがあります。

- ・ プログラム、節、段落に付けられたコメントで処理全体を把握できる
- ・ 処理コメントが処理を理解する手助けになる
- ・ 過去に読み書きした処理コードパターンの記憶が処理を理解する手助けになる
- ・ 処理コードパターンを記憶することでソースコードの読解力を身に着けることができる

しかし、実行パスを頭の中で記憶しながら目視でトレースするので、条件文や分岐が複雑に絡み合い実行パスが多数になると、頭の中でのトレースが困難になります。

実行順序表示機能は、ソースコードリーディングのこの欠点部分を支援するための機能です。この機能では、制御フロー解析の制御フロー情報を使って、制御ブロック（節／段落／文／条件指定）のソースコードを実行順にツリー形式で表示します。このツリーのことを実行順序ツリーといいます。

図 10-8 実行順序ツリーとソースプログラムの関係



この図では、ソースプログラムのソースコードは「PERFORM PARA2」で段落 PARA2 に制御が移行しています。実行順序ツリーでは、ソースコードの実行順に従って、「PERFORM PARA2」のあとに呼出先（制御の移行先）の段落 PARA2 を表示しています。

実行順序ツリーを使うと、次のようなメリットがあり、効率良くかつ効果的にトレースできます。

- ソースコードリーディングのように分岐や手続き呼び出しのたびに分岐先や呼出先を探す手間で見視トレースが中断されることはない
- 実行順序ツリーはすべての実行経路を表示するため、頭の中に実行経路をスタック（記憶）する必要がなく、記憶の混乱で発生する実行経路のトレース漏れを防止できる

この実行順序ツリーは、次に示す COBOL ソース解析の生成ドキュメントに表示または出力されます。

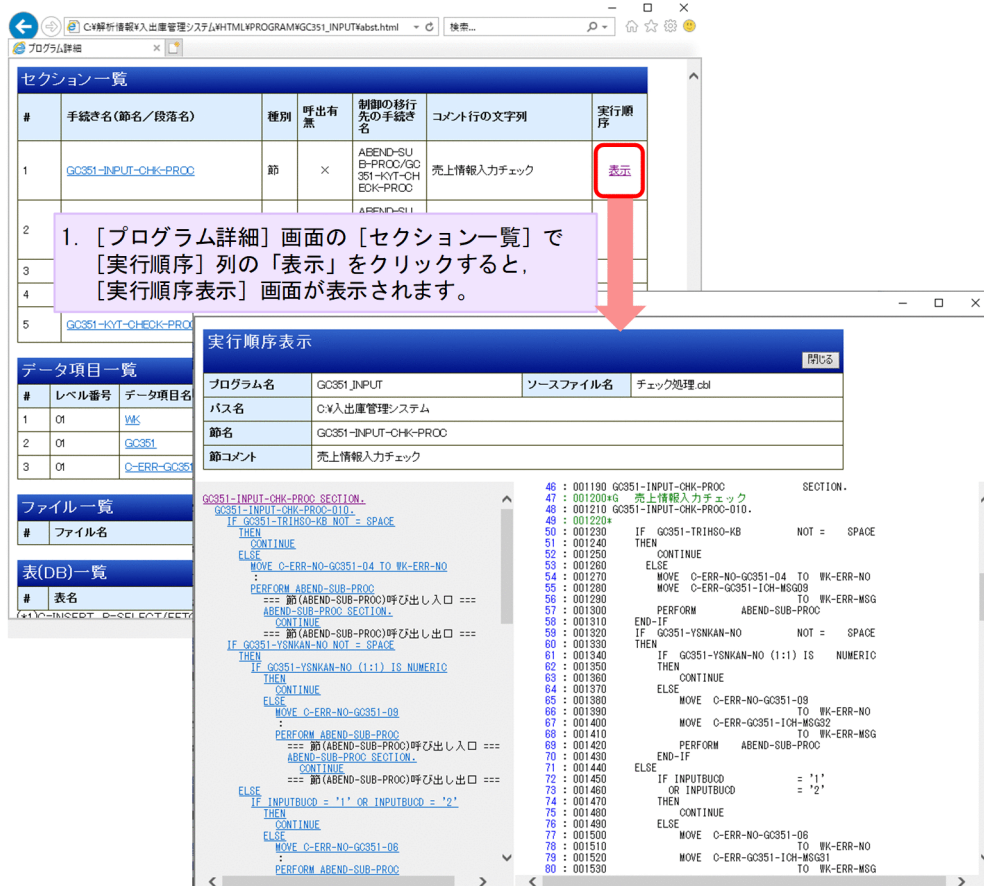
- [実行順序表示] 画面
画面の左ペインのしおり（目次）に実行順序ツリー、右ペインにソースプログラムを表示します。
- 実行順序リスト
実行順序ツリーをプログラムごとに CSV ファイルに出力します。

10.16.3 実行順序ツリーでトレースできる実行順序情報

実行順序情報は、[実行順序表示] 画面または実行順序リストの実行順序ツリーを使用してトレースできます。それぞれの実行順序ツリーの内容を次に示します。

(1) [実行順序表示] 画面での実行順序ツリーの表示内容

[解析生成オプション設定] 画面の [その他の解析オプション] タブで [プログラムの制御フロー情報（実行順序表示機能で使用する情報）を抽出する] をチェックして、解析／生成を実行すると、[プログラム詳細] 画面の [セクション一覧] の [実行順序] 列に「表示」が表示されます。「表示」をクリックすると、その節に対する [実行順序表示] 画面が表示されます。



プログラム全体の「実行順序表示」画面を表示させたい場合は、手続き部分先頭の節の「表示」をクリックします。手続き部先頭が節見出しではない場合は、「{PROCEDURE_START}」が手続き部分先頭の節です。

左ペインでは、実行順序ツリーをしおり（目次）として表示します。右ペインでは、[ソース表示(COPY 展開後)] 画面の右ペインと同じソースプログラムのソースコードを表示します。左ペインの下線のリンクをクリックすると、該当する行を右ペインの最上行に位置付けます。[実行順序表示] 画面については、[7.4.5 \[実行順序表示\] 画面](#)を参照してください。

実行順序ツリーの表示規則

- 各行は次の規則でソース行 1 行分を表示します。ただし、条件式を含む IF 文、EVALUATE 文、WHEN 条件指定、および PERFORM 文は、条件式の終わりまで表示します。
 - 節と段落の見出しの末尾には終止符を表示します。
 - 文中の不要な空白（連続した空白の先頭以外の部分）は削除します。
 - コメント（空白行、注記行、行内注記、翻訳指令行、および条件翻訳の無効行）は削除します。
 - 継続行は継続対象行に含めます。
- 実行文の終止符や明示範囲終了子（END-PERFORM と END-EXEC を除く）は表示しません。また、END-PERFORM は、コメントとして表示し、リンク（下線）は付加しません。
- 文が複数行にわたる場合は、先頭の 1 行（埋め込み SQL 文の場合は 2 行）を表示します。

- 条件指定は条件記述部分だけ表示し、無条件文は次の行に表示します。
- 条件指定の予約語部分は常に完全形（補助語の省略なし）を半角英数字文字で表示します。また、「EOP」と書いても「END-OF-PAGE」と表示します。
- インデントーションで制御ブロックの親子関係を表現します。
- 手続き呼び出しの場合、開始と終了を次のコメントで囲みます。

節のとき

```
=== 節(節名)呼び出し入口 ===
```

```
=== 節(節名)呼び出し出口 ===
```

段落のとき

```
=== 段落(段落名)呼び出し入口 ===
```

```
=== 段落(段落名)呼び出し出口 ===
```

ただし、再帰呼び出しとなっている場合は、呼出先の実行順序ツリーは表示しないで「"*** 再帰呼び出しのため省略 ***"」を表示します。

- GO TO 文の分岐先が通過済みの場合は、分岐先の実行順序ツリーは表示しないで「"*** この先省略 ***"」を表示します。ただし、分岐先の節・段落が単純でかつ制御フローの終端となる場合を除きます。
- プログラム全体ではなく、節個別の実行順序表示の場合で GO TO 文の分岐先が節外のときは、分岐先の実行順序ツリーは出力しないで「"*** 節外への分岐 ***"」を表示します。
- 手続き部末尾で仮定される暗黙の GOBACK 文に対しては「"END PROGRAM"」を表示します（END PROGRAM で終わっていない手続き部末尾の暗黙の GOBACK 文に対しても「"END PROGRAM"」を表示します）。
- プログラム終了となる次の文は半角英数字文字で表示し、後ろには「"---> (プログラム終了)"」を付加します。
 - STOP RUN 文（埋め込み例外宣言の STOP 指定は STOP RUN 文と同じ表示となる）
 - GOBACK 文
 - EXIT PROGRAM 文
 - END PROGRAM 文（暗黙の GOBACK 文）
 - CBLABN サービスルーチン呼び出しの CALL 文（区切り記号はアポストロフィ（'）固定で引数は表示しない）
- プログラム全体ではなく、節個別の実行順序表示の場合で、プログラム終了となる文や USE 節終了となる文（USE 節の場合）と遭遇しないで節の終わりに達したときは、「"=== 節(節名)の終わり ==="」を表示します。ただし、節が空の場合は何も表示しません。
- 基本ブロック（制御の移行を伴わない文の連続した並び）の場合は、先頭の文だけ表示し、2つ目以降の文は「"："」で代替します。
- 右ペインに対応する行がある場合はリンク（下線）を付加します。内部的に仮定された文や条件指定、コメント（手続き呼び出しの入口と出口など）にはリンクは付加しません。

注意事項

しおり（目次）の実行順序ツリーの行数が上限（60,000）を超えた場合は、「"===== 上限 (60,000)を超えたため打ち切り =====」が表示され、以降の行は表示されません。

(2) 実行順序リストでの実行順序ツリーの出力内容

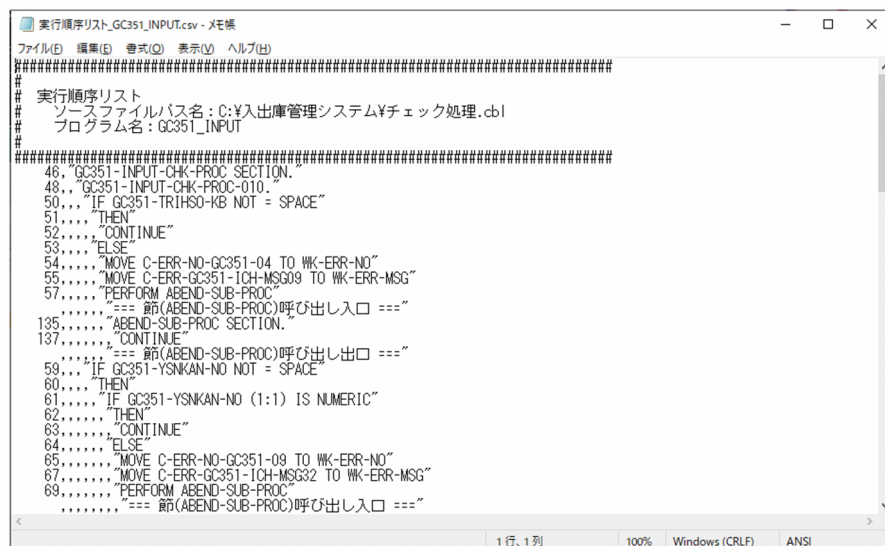
[解析生成オプション設定] 画面の [その他の解析オプション] タブの [プログラムの制御フロー情報（実行順序表示機能で使用する情報）] を抽出する, および [生成ドキュメント] タブの [実行順序リスト] をチェックして、解析／生成を実行すると、プログラムごとに実行順序リスト CSV ファイルが出力されます。

実行順序リスト CSV ファイルの出力先

生成ドキュメントの出力先フォルダ¥プロジェクト名¥CSV¥実行順序リスト¥実行順序リスト_プログラム名.csv

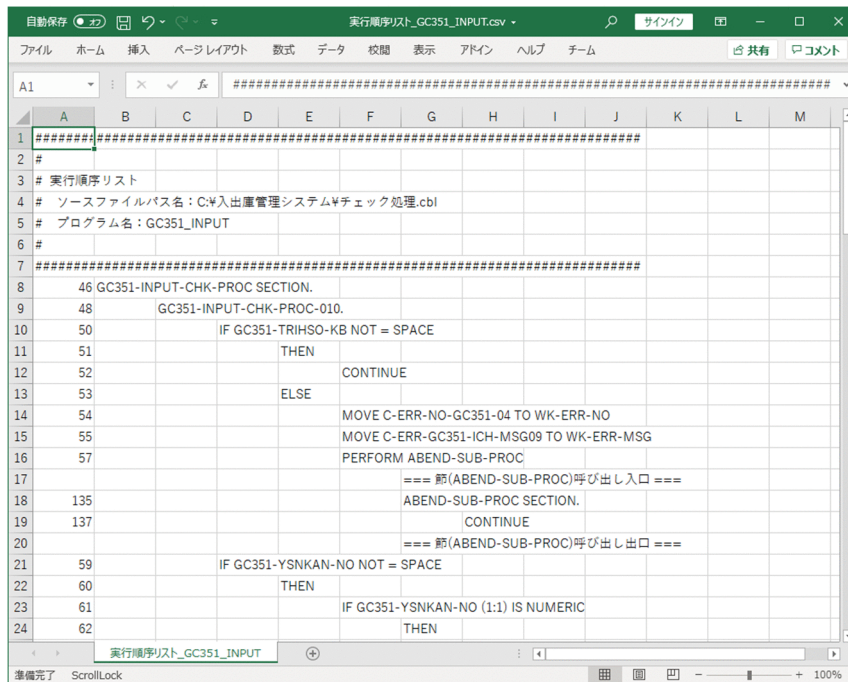
実行順序リスト CSV ファイルの出力例

- テキストエディタの場合



```
#####
##
## 実行順序リスト
## ソースファイルパス名: C:\¥入出庫管理システム¥チェック処理.cb1
## プログラム名: GC351_INPUT
##
#####
48., GC351-INPUT-CHK-PROC SECTION.
48., GC351-INPUT-CHK-PROC-010.
50., IF GC351-TRIHSO-KB NOT = SPACE"
51., THEN
52., "CONTINUE"
53., ELSE
54., "MOVE C-ERR-NO-GC351-04 TO WK-ERR-NO"
55., "MOVE C-ERR-GC351-1CH-MSG09 TO WK-ERR-MSG"
57., "PERFORM ABEND-SUB-PROC"
57., "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
135., "ABEND-SUB-PROC SECTION."
137., "CONTINUE"
137., "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
59., IF GC351-YSNKAN-NO NOT = SPACE"
60., THEN
61., "IF GC351-YSNKAN-NO (1:1) IS NUMERIC"
62., THEN
63., "CONTINUE"
64., ELSE
65., "MOVE C-ERR-NO-GC351-09 TO WK-ERR-NO"
67., "MOVE C-ERR-GC351-1CH-MSG32 TO WK-ERR-MSG"
69., "PERFORM ABEND-SUB-PROC"
69., "=== 節(ABEND-SUB-PROC)呼び出し出口 ==="
```

- 表計算ソフトの場合



実行順序リスト CSV ファイルについては、「[8.2.7 実行順序リスト](#)」を参照してください。

実行順序ツリーの出力規則

- 各行の先頭にはソース行番号を出力します。6 桁に満たない場合は前に半角空白を補います。内部的に仮定された文や条件指定、コメント（手続き呼び出しの入口と出口など）の場合は空文字列を出力します。
- 各行は次の規則で文や条件指定の先頭から末尾までのすべてを 1 行で出力します。
 - 節と段落の見出しの末尾には終止符を出力します。
 - 文中の不要な空白（連続した空白の先頭以外の部分）は削除します。
 - コメント（空白行、注記行、行内注記、翻訳指令行、および条件翻訳の無効行）は削除します。
 - 継続行は継続対象行に含めます。
 - 実行文の終止符や明示範囲終了子（END-PERFORM と END-EXEC を除く）は出力しません。また、END-PERFORM は、コメントとして出力し、行の先頭はソース行番号ではなく、空文字列となります。
 - 条件指定は条件記述部分だけ出力し、無条件文は次の行に出力します。
- コンマ (,) によるインデントーションで制御ブロックの親子関係を表現します。
- 手続き呼び出しの場合、開始と終了を次のコメントで囲みます。

節のとき

=== 節(節名)呼び出し入口 ===

=== 節(節名)呼び出し出口 ===

段落のとき

=== 段落(段落名)呼び出し入口 ===

=== 段落(段落名)呼び出し出口 ===

ただし、再帰呼び出しとなっている場合は、呼出先の実行順序ツリーは出力しないで「"*** 再帰呼び出しのため省略 ***"」を出力します。

- GO TO 文の分岐先が通過済みの場合は、分岐先の実行順序ツリーは出力しないで「"*** この先省略 ***"」を出力します。ただし、分岐先の節・段落が単純でかつ制御フローの終端となる場合を除きます。
- 手続き部末尾で仮定される暗黙の GOBACK 文に対しては「"END PROGRAM"」を出力します (END PROGRAM で終わっていない手続き部末尾の暗黙の GOBACK 文に対しても「"END PROGRAM"」を出力します)。
- プログラム終了となる次の文はの後ろには「"---> (プログラム終了)"」を付加します。
 - STOP RUN 文
 - GOBACK 文
 - EXIT PROGRAM 文
 - END PROGRAM 文 (暗黙の GOBACK 文)
 - CBLABN サービスルーチン呼び出しの CALL 文
- 基本ブロック (制御の移行を伴わない文の連続した並び) の場合は、制御ブロック内のすべての文を出力します。

注意事項

- 実行順序リストでは、手続き部分先頭の節に対する [実行順序表示] 画面の実行順序表示と同じ実行経路を表示します。手続き部分先頭から到達できない手続き (節・段落) や文は表示されません。
- プログラムの手続き部分が空の場合、実行順序リスト CSV ファイルには実行順序リストは出力されないでヘッダだけが出力されます。
- 実行順序リストの実行順序ツリーの行数が上限 (999,999) を超えた場合は、「"===== 上限(999,999)を超えたため打ち切り ====="」が出力され、以降の行は出力されません。
- プログラム名が英小文字を含む場合、出力ファイル名は、実行順序リスト_プログラム名.英小文字識別子コード.csv となります。英小文字識別コードは 16 進数で、英大文字と英小文字だけが異なるファイル名が同一とみなされて上書きされることを回避するために付加されます。

10.16.4 実行順序表示機能使用時の注意事項

- プログラムのデータ項目には常に正しい値が入っていて、文の実行で例外が発生することはないものとして解析します。

- PERFORM 文以外の文から USE 手続きへの制御の移行はないものとして解析します。
- RESUME AT 手続き名は、RESUME AT NEXT STATEMENT として扱います。USE 節終了となる実行順序ツリーになり、手続き名に分岐する実行順序ツリーにはなりません。
- RAISE 文は制御の移行を伴わない文として扱います。基本ブロックに含まれます。
- EXIT PROGRAM 文は、主プログラムであっても CONTINUE 文としては扱いません。常にプログラム終了となる実行順序ツリーになり、後続の文に続く実行順序ツリーにはなりません。
- END-PERFORM と END-EXEC を除き、明示範囲終了子は表示／出力されません。END-PERFORM は出力されますが、コメントとして表示／出力されます。
- 次の覚え書きの文は表示／出力されません。
 - ENTER 文
 - CANCEL 文(WINDOW 節)
 - WAIT 文
 - うち PERFORM 文の中にない EXIT PERFORM 文
- 次の予約語は常に半角英数字文字で表示／出力されます。
 - 節見出しの SECTION
 - 埋め込み SQL 文に適用された WHENEVER SQLERROR／WHENEVER SQLWARNING／WHENEVER NOT FOUND
 - 暗黙的に仮定された THEN
- 1 行に表示する文や指定の文字列の長さが 1,024 文字を超えている場合は 1,024 文字で切り捨てます。
- そと PERFORM 文が次に示す PERFORM 文の一般規則に該当する場合、[実行順序表示] 画面や実行順序リストの実行順序が、実際のプログラムの実行順序と一致しないことがあります。

注意事項

マニュアル「COBOL2002 言語 標準仕様編」の PERFORM 文の一般規則

次の条件が重なる場合の PERFORM 文の並びを実行した結果は規定されません。また、この並びを実行したとき、例外の成立は設定されません。

- ある PERFORM 文が実行されて、まだ終了していない
 - その PERFORM 文の範囲の中でもう一つの PERFORM 文が実行される
 - 第 2 の PERFORM 文の実行が、第 1 の PERFORM 文の出口を通過して行われる
- 解析ログに「けたあふれは発生しないので ON SIZE ERROR 指定を無視する」ことを示す COBOL2002 コンパイラのメッセージが出力されることがありますが、その ON SIZE ERROR 指定は実行順序表示機能では無視されないで表示／出力されます。
 - 節や段落のメインパス（条件指定の中でない部分）に STOP RUN 文などのプログラムを終了させる文がある場合、その節や段落の手続き呼び出しでは「=== 節(節名)呼び出し入口 ===」や「=== 段落

(段落名)呼び出し入口 ===」に対応する「=== 節(筋名)呼び出し出口 ===」や「=== 段落(段落名)呼び出し出口 ===」は表示されません。

- 次の埋め込み SQL 文は解析対象外です。CONTINUE 文として解析します。
 - COMMENT 文
 - VAR 文
 - カーソル宣言以外の DECLARE 文
 - 解析できないことを示すコンパイルエラー（警告）が出力される SQL 文（WITH 句のある DECLARE CURSOR など）
- -SQL,XDM コンパイラオプションが必要なプログラムの制御フロー情報は抽出できません。[解析生成オプション設定] 画面の [その他の解析オプション] タブで [プログラムの制御フロー情報（実行順序表示機能で使用する情報）を抽出する] をチェックした場合、-SQL,XDM コンパイラオプションまたは-SQL,ODBC コンパイラオプションが指定されていても、指定はないものとして解析します。埋め込み SQL 文は解析できますが、VOS3 XDM/RD 固有の特殊レジスタ（RDB-STATUS, RDB-COUNT, および RDB-RDNODE）が使用されているときはコンパイルエラーになります。

11

使用上の注意事項

COBOL ソース解析の使用上の注意事項について説明します。

11.1 解析／生成に関する注意事項

COBOL ソースファイルの解析／生成を実行する前に確認しておく注意事項を説明します。

1. 正しい解析結果および生成結果を出力するために、解析／生成を実行する前に次の作業を実施してください。
 - 解析／生成オプションで主プログラムを正しく設定しているか確認する
解析／生成オプションの主プログラム指定が正しく設定されていないと、プログラムの使用／未使用の情報が正しく抽出できないため、ドキュメント中の「使用有無」の情報が正しく表示されません。[解析生成オプション設定] 画面の [主プログラム指定] タブで、主プログラムを過不足なく指定して、COBOL ソース解析が解析時に主プログラムを正しく認識できるようにしてください。解析／生成オプションの主プログラム指定については、「[10.12 主プログラムの指定方法](#)」を参照してください。
 - 一意名によるプログラム呼び出しの呼び出し先をコンパイラで完全に解決しているか確認する
COBOL ソースファイルのコンパイルで一意名呼び出し（一意名指定の CALL 文）に対する警告メッセージが出力されている場合、一意名呼び出しの呼出先が確定していません。この場合、プログラムおよびプログラムの構成要素の使用／未使用が判断できないため、ドキュメント中の「使用有無」の情報が正しく表示されないことがあります。一意名呼び出し対応付けファイルを作成して、必ず一意名呼び出し（一意名指定の CALL 文）の呼出先を完全に解決してください。
 - COBOL ソースファイルに対するコンパイルエラーや解析エラーの原因がすべて排除されているか確認する
一部の COBOL ソースファイルに対して、S または U レベルのコンパイルエラーや E レベルの解析エラーが出力されている場合、「システム全体の情報」のドキュメントの生成結果は保証できません。エラーメッセージを基に、すべての COBOL ソースファイルに対するエラーの原因を排除してください。
2. COBOL ソース解析の実行中は、ファイルを排他制御するエディタで解析ログや操作ログを開かないでください。COBOL ソース解析を実行する前に、開いているログファイルがあれば閉じてください。
3. COBOL ソース解析がメッセージを出力するたびに解析ログおよび操作ログがファイルに追加書きされます。現行ログの最大容量を超えた場合、退避ログにコピーしたあとで、現行ログが初期化されてログが出力されます。退避ログにすでにログがある場合は上書きされ、古いログは消去されます。このため、解析ログおよび操作ログを保持する場合は、適時、退避ログにある解析ログファイルおよび操作ログファイルを任意の場所にバックアップしてください。
4. ソース解析プロジェクト名、または COBOL ソースファイル中のプログラム名、ファイル名、表名、登録集原文名で次の Windows の予約デバイス名が使用されている場合、ドキュメント生成でエラーとなります。
 - CON
 - PRN
 - AUX
 - NUL

- LPT1～LPT9（LPT0 は予約デバイス名ではありません。ただし、エクスプローラの名称変更では指定できません）
- COM1～COM9（COM0 は予約デバイス名ではありません。ただし、エクスプローラの名称変更では指定できません）

付録

付録 A ソース解析情報のフォルダ構成

COBOL ソース解析のソース解析情報（HTML ファイル形式、CSV ファイル形式）のフォルダ構成です。ソース解析情報は、[解析生成オプション設定] 画面の [生成ドキュメント] タブで設定したフォルダパスの下に、解析対象プロジェクト名のフォルダが作成され、その下に「HTML」「CSV」フォルダが作成されます。

フォルダパスが円記号（¥）で終わっている個所は、フォルダを示します。

表 A-1 COBOL ソース解析のソース解析情報のフォルダ構成

フォルダパスとファイル名		ソース解析情報
任意のフォルダ¥CSV		ソース解析情報（CSV）の出力先フォルダ COBOL ソース解析の [解析生成オプション設定] 画面の [生成ドキュメント] タブで設定します。
	COBOL ソースファイル一覧.csv	COBOL ソースファイル一覧
	ファイル参照関係.csv	ファイル参照関係
	プログラム一覧.csv	プログラム一覧
	セクション一覧.csv	セクション一覧
	ファイル一覧.csv	ファイル一覧
	プログラム呼出関係.csv	プログラム呼び出し関係
	登録集原文参照関係.csv	登録集原文参照関係
	表 DB 参照関係.csv	表 DB 参照関係
	データ項目一覧¥	プログラムごとのデータ項目一覧の格納フォルダ
	データ項目一覧_プログラム名 n.csv	プログラム n のデータ項目一覧
	不整合引数一覧.csv	不整合引数一覧
	不整合引数一覧.tag	不整合引数の警告メッセージ一覧
	実行順序リスト¥	プログラムごとの実行順序リストの格納フォルダ
	実行順序リスト_プログラム名 n.csv	プログラム n の実行順序リスト
任意のフォルダ¥HTML		ソース解析情報（HTML）の出力先フォルダ COBOL ソース解析の [解析生成オプション設定] 画面の [生成ドキュメント] タブで設定します。
	cblsrcfilelist.html	[COBOL ソースファイル一覧]
	copyfilelist.html	[登録集原文ファイル一覧]
	copylist.html	[登録集原文一覧]
	deadinflist.html	[デッドコード情報一覧]

フォルダパスとファイル名		ソース解析情報
	filelist.html	[ファイル一覧]
	index.html	プロジェクトサマリ
	moddupliidlist.html	[重複するプログラム一覧]
	modlist.html	[プログラム一覧]
	modunureg.html	[未登録のプログラム一覧]
	tablelist.html	[表(DB)一覧]
	unregcopylist.html	[未登録の登録集原文一覧]
	COPY¥	登録集原文に関するソース解析情報の格納フォルダ
	登録集原文名¥	登録集原文の COBOL プログラムに関するソース解析情報の格納フォルダ
	COPY 番号※1¥	1～99 の番号
	copydetail.html	[登録集原文詳細] の「登録集原文概要」
	copyedmodlist.html	[登録集原文詳細] の「COPY 元プログラム」
	FILE¥	ファイルに関するソース解析情報の格納フォルダ
	ファイル名¥	該当するファイルに関するソース解析情報の格納フォルダ
	filedetail.html	[ファイル詳細]
	filemodlist.html	[ファイル詳細] の「ファイル使用プログラム」
	Files¥	画像ファイルの格納フォルダ
	Figure¥	
	commod.png	
	comsec.png	
	file.png	
	graph.png	
	kaisou.png	
	program.png	
	section.png	
	table.png	
	ToolKitlogo.png	
	MODHIER¥	プログラム階層図に関するソース解析情報の格納フォルダ
	modhier.html	[プログラム階層図]
	modhier_共通プログラム名.html	[プログラム階層図(共通プログラム)]
	PROGRAM¥	COBOL プログラムのソース解析情報の格納フォルダ

フォルダパスとファイル名		ソース解析情報
	プログラム名¥	COBOL プログラムに関するソース解析情報の格納フォルダ
	abst.html	[プログラム概要]
	dataitemlist.html	[データ項目一覧]
	deadodelist.html	[制御が渡らない文、手続き一覧]
	filesecdatadef.html	[ファイル節のデータ定義]
	intermodlist.html	[内部プログラム一覧]
	linksecdatadef.html	[連絡節のデータ定義]
	modcall.html	[呼出先プログラム一覧]
	modcalled.html	[呼出元プログラム一覧]
	moddata.html	[プログラムデータ]
	modfilelist.html	[ファイル一覧]
	modsrcview.html	[ソース表示(COPY 展開後)]
	modsrcviewindex.html	[ソース表示(COPY 展開後)] 画面の左ペインの左ペインの目次（しおり）
	modsrcviewlist.html	[ソース表示(COPY 展開後)] 画面のソースコード表示
	modsrcviewtitle.html	[ソース表示(COPY 展開後)] 画面のタイトル部分
	modtablelist.html	[表(DB)一覧]
	refcopylist.html	[参照する登録集原文一覧]
	rep_src_commlist.html	[報告書、画面、通信記述名一覧]
	seclist.html	[セクション一覧]
	worksecdatadef.html	[作業場所節のデータ定義]
	SECHIER¥	セクション階層図に関するソース解析情報の格納フォルダ
	sechier.html	[セクション階層図]
	sechier_共通節名.html	[セクション階層図（共通節）]
	SECTION¥	節に関するソース解析情報の格納フォルダ
	"節名"の並び※2¥	各節に関するソース解析情報の格納フォルダ
	execseqview.html	[実行順序表示]
	execseqviewindex.html	[実行順序表示] 画面の左ペインの目次（しおり）
	execseqviewtitle.html	[実行順序表示] 画面のタイトル部分
	SOURCE¥	COPY 展開前の [ソース表示]
	COBOL ソースファイル名または登録集原文ファイル名¥	[ソース表示] 画面对象の COBOL ソースファイル名または登録集原文ファイル名

フォルダパスとファイル名			ソース解析情報
		<i>FILE</i> 番号※ ³ または <i>COPY</i> 番号※ ¹ ¥	1～99 の番号
		srcview.html	[ソース表示]
		srcviewindex.html	[ソース表示] 画面の左ペインの目次（しおり）
		srcviewlist.html	[ソース表示] 画面のソースコード表示
		srcviewtitle.html	[ソース表示] 画面のタイトル部分
	STAT¥		分析情報に関するソース解析情報の格納フォルダ
		branchcountdist.html	[分岐の数の分布]
		descriptioncountdist.html	[記述項数の分布]
		modcallcountdist.html	[呼出先プログラム数の分布]
		modcalledcountdist.html	[呼出元プログラム数の分布]
		paracountdist.html	[段落数の分布]
		seccountdist.html	[節数の分布]
		statementcountdist.html	[実行文の数の分布]
		stepcountdist.html	[行数の分布]
	Stylesheet¥		ソース解析情報の HTML ファイル用スタイルシート格納フォルダ
		graph.css	
		kaiso.css	
		srcview.css	
		stylesheet.css	
	TABLE¥		表に関するソース解析情報の格納フォルダ
		テーブル名¥	各表名に関するソース解析情報の格納フォルダ
		tabledetail.html	[表(DB)詳細]
		tablemodlist.html	[表(DB)使用プログラム一覧]

注※1

同名の登録集原文が存在する場合は、次の規則でフォルダが作成されます。

"登録集原文名" + "¥" + *COPY* 番号

注※2

手続き部の先頭が節の見出しで始まっていない場合は、「{PROCEDURE_START}」という名前の節が仮定されます。

節名が重複する場合は、次の規則でフォルダが作成されます。

節名+(重複番号)

節名が英小文字を含んでいる場合は、次の規則でフォルダが作成されます。

節名+英小文字識別コード

注※3

同名の COBOL ソースファイルが存在する場合は、次の規則でフォルダが作成されます。

"*COBOL* ソースファイル名" + "." + "拡張子" + "¥" + *FILE* 番号

付録 B 解析を補助する機能

解析を補助する機能について説明します。

付録 B.1 一意名呼び出し対応付けファイル

一意名でプログラムを呼び出している COBOL プログラムの解析で、VALUE 句または MOVE 文による英数字定数の転記以外の方法で一意名が更新される場合、COBOL ソース解析のコンパイラでは一意名によるプログラム呼び出しを解析できません。一意名呼び出し対応付けファイルは、COBOL ソース解析で自動的に一意名の呼び出しを解析できない場合、一意名の呼び出し先を手動で解決するための機能です。

COBOL ソース解析のコンパイラで呼び出し先を解析できない場合は、一意名の呼び出し先が完全に解決していないことを示す警告メッセージが表示されます。この警告メッセージで COBOL プログラム名と該当行番号を確認し、一意名呼び出し対応付けファイルを作成してください。

拡張子は、.idc で作成してください。

作成した一意名呼び出し対応付けファイルは、[解析生成オプション設定] 画面の [呼出し先プログラム解決] タブに指定します。

形式

"COBOLソースファイル名"△"プログラム名"△"一意名"△"呼び出し先プログラム名"

(凡例)

△：1 個以上の半角空白を示します。

説明

一意名呼び出し対応付けファイルに指定する項目です。

COBOL ソースファイル名

一意名指定の CALL 文が記述されている COBOL ソースファイル名です。

プログラム名

一意名指定の CALL 文が記述されているプログラム名です。

一意名

CALL 文の一意名です。

呼び出し先プログラム名

呼び出し先のプログラム名です。

注意

- 一意名呼び出し対応付けファイルの 1 行の長さは、4,095 バイト以内で指定してください。4,095 バイトを超えていると、警告メッセージが出力され、その行は無視されます。

- COBOL ソースファイル名、プログラム名、一意名、呼び出し先プログラム名はそれぞれ引用符 (") で囲み、1 個以上の半角空白またはタブ文字で区切って、1 行に 1 組だけ指定してください。
- COBOL ソースファイル名の長さは、255 バイト以内で指定してください。
- COBOL ソースファイル名には、引用符 (") を含むファイル名は指定できません。
- COBOL ソースファイル名には、絶対パス名を指定してください。ファイル名だけ、または相対パス名で指定した場合は、その行は無視されます。
- プログラム名と呼び出し先プログラム名は、次に示すプログラム名の構成規則に従って指定してください。構成規則に従っていない場合は、警告メッセージが出力され、その行は無視されます。
 - 次に示す文字で構成されている。
 - 英文字 (A~Z, a~z)
 - 数字 (0~9)
 - 日本語文字 (ただし、拡張仕様)
 - (ハイフン)
 - _ (アンダスコア)
- 文字列の長さは、先頭は英文字または日本語文字で、かつ 160 バイト (-Compati85,Syntax オプションの指定が有効な場合は 30 バイト) 以内です。
- 一意名には、CALL 文の一意名として参照されるデータ名を指定します。一意名が部分参照、修飾、または添字が付けられている場合は、警告メッセージが出力され、その行は無視されます。
- 一意名の長さは 31 文字 (-Compati85,Syntax オプションありの場合は 30 文字) 以内で指定してください。一意名の長さが上限を超えた場合は、警告メッセージが出力され、その行は無視されます。
- 1 カラム目が「#」で始まる行はコメント行として扱い、その行は無視されます。
- 一意名呼び出し対応付けファイルは、コメント行を含め 1,000 行以内で記述してください。1,000 行を超えている場合は、警告メッセージが出力され、以降の行は無視されます。
- 一意名呼び出し対応付けファイルでは、次の文字は区別されます。
 - 各英小文字と対応する英大文字
 - 拡張コードの各文字と対応する標準コードの文字
- COBOL ソースファイル中の利用者定義語には文字の等価規則が適用されるため、プログラム名と一意名には、それぞれ等価規則の適用後のプログラム名やデータ名を指定してください。ただし、COBOL ソースファイル名の英小文字と対応する英大文字は同等と見なされます。
- COBOL ソースファイル名で指定されたソースファイル名とプログラム情報収集対象 (コンパイル対象) のソースファイル名とが一致しない場合、その行は無視されます。
- 指定した COBOL ソースファイルに、プログラム名と同じ名称のプログラム名が定義されていない場合、警告メッセージが出力され、その行は無視されます。
- プログラム名に指定したプログラム定義中で、一意名と同じ名称のデータ項目が CALL 文の一意名として参照されていない場合、または一意名が一意に参照できない場合、警告メッセージが出力され、その行は無視されます。

なお、CALL 文の一意名に添字が付けられている場合は、CALL 文の一意名から添字を除いたデータ名と一意名が一致していれば、一致したものと見なされます。

- 一意名と同じ名称のデータ項目が CALL 文の一意名として参照されているとき、呼び出し先プログラム名を一意名指定の CALL 文の呼び出し先として解決し、次の定数指定の CALL 文が書かれていたと見なされ、プログラム情報が収集されます。

CALL '呼び出し先プログラム名' USING 一意名指定の CALL 文と同じ引数の並び

- 一意名呼び出し対応付けファイルがある場合は、次の順に評価され、呼び出し先が解決されます。
 - COBOL ソースファイルの解析による呼び出し先の解決
 - 一意名呼び出し対応付けファイルによる呼び出し先の解決
- 1 つの一意名指定の CALL 文に対して、複数の呼び出し先プログラムを複数の行を使って指定できます。その場合、指定された呼び出し先プログラムの数だけ定数指定の CALL 文が並んでいると見なされ、プログラム情報が収集されます。ただし、COBOL ソースファイルの解析による呼び出し先の解決で仮定された定数指定の CALL 文も含めて呼び出し先プログラムが同じものは 1 つとして扱われます。
- プログラム名、呼び出し先プログラム名の中にハイフンがある場合、それぞれアンダスコアに変換してプログラム名、呼び出し先プログラム名と見なされます。
- 呼び出し先プログラム名の長さ（バイト数）が一意名の長さを超えている場合、呼び出し先プログラム名は切り捨てないで、そのままを呼び出し先プログラム名と見なされます。
- 一意名呼び出し対応付けファイルに対する警告メッセージは、コンパイルリストには出力されません。
- 一意名呼び出し対応付けファイルに対する警告メッセージは、次に示す形式で出力されます。

”一意名呼び出し対応付けファイル”， line 行番号：警告メッセージ

- 一意名呼び出し対応付けファイルは、シフト JIS コードで記述してください。
- 呼び出し先として解決されたプログラム名が、CALL 文から呼び出せる内部プログラム名と一致している場合、その CALL 文は内部プログラム呼び出しの CALL 文と見なされ、一致しない場合は、外部プログラム呼び出しの CALL 文と見なされます。このとき、プログラム名が一致するかどうかは、プログラム名の文字列が一致するかどうかだけをチェックされ、引数と返却項目の適合規則はチェックされません。
- 一意名呼び出し対応付けファイルは、一意名指定の CANCEL 文には適用しません。

付録 B.2 サービスルーチンファイル

COBOL プログラムからライブラリ製品のプログラムや他言語で開発したプログラムを呼び出している場合は、COBOL ソースファイルがないため、「未登録のプログラム」としてソース解析情報のドキュメントに表示されます。サービスルーチンファイルは、このようなプログラムを「未登録のプログラム」として表示しないための機能です。

拡張子は、.svw（テキストファイル）で作成してください。また、フォルダパス名を除いたファイル名と拡張子が 100 バイト以内となる名前で作成してください。

形式

```
*コメント
S△サービスルーチン名またはプログラム名
:
**△END
```

(凡例)

△：1 個以上の半角空白を示します。

説明

サービスルーチンファイルに指定する項目です。

*コメント
コメント行です。

S△サービスルーチン名またはプログラム名
サービスルーチン名またはライブラリのプログラム名です。

**△END
サービスルーチン名またはプログラム名の並びの終わりです。

注意

- サービスルーチン名またはプログラム名の「-」（ハイフン）と「_」（アンダスコア）は、等価として扱われます。
- CALL 文に記述されているプログラム名が、#，¥，@の半角文字を含む場合や、先頭文字が半角数字の場合は、次の変換規則で変換されたプログラム名（目的プログラムの外部シンボル名）を指定してください。

変更前のプログラム名（CALL 文に記述されているプログラム名）中の半角文字	変換後のプログラム名（目的プログラムの外部シンボル名）	
	先頭の文字	2 番目以降の文字
#	J	0
¥	A	1
@	B	2
1～9, 0	A～I, J	変更なし

付録 B.3 -SQLAutoCopy コンパイラオプション

-SQLAutoCopy コンパイラオプションについて説明します。

COBOL 言語用 SQL プリプロセサの UAP ソースファイルの場合、次のどちらかに該当する COBOL ソースファイルはコンパイルエラーになるため、解析できません。

- SQLCODE 変数, SQLSTATE 変数, SQL 連絡領域 (SQLCA) または SQL 記述領域 (SQLDA) を COBOL プログラム内で参照しているが、明示的な埋め込み SQL 宣言がない
- SQL TYPE 句を使用している

コンパイルエラーを回避するためには、ソース解析プロジェクトの解析生成オプションを次のように変更します。

- SQL 連絡領域 (SQLCA) や SQL 記述領域 (SQLDA) の登録集原文ファイルを用意して、登録集原文ファイルのパス名を環境変数 CBLLIB に設定します。
- COBOL2002 のコンパイラオプションに -SQLAutoCopy コンパイラオプションを追加します。

(1) 形式

```
-SQLAutoCopy  
-noSQLAutoCopy
```

(2) 機能

-SQLAutoCopy

埋め込み SQL 文を含む COBOL ソースファイルの場合、「WORKING-STORAGE SECTION.」のあとに、「INCLUDE CBLAUTOSQLCOPY.」を仮定します。

-noSQLAutoCopy

-SQLAutoCopy コンパイラオプションの指定を打ち消します。

(3) 注意事項

- 次のどれかのコンパイラオプションを指定すると、-SQLAutoCopy コンパイラオプションは無効になります。
-StdMIA -Std85 -Std2002

付録 C 各バージョンの変更内容

各バージョンの変更内容を示します。

変更内容 (3021-3-616-30) COBOL2002 Developer Professional 04-30, COBOL2002 Developer Professional(64) 04-30

追加・変更内容
<p>[分布情報一覧] 画面に分布情報（段落数の分布）を追加した。</p> <p>また、次の分布情報の名称を変更した。</p> <ul style="list-style-type: none">・「実行文の行数の分布」から「実行文の数の分布」に変更・「セクション数の分布」から「節数の分布」に変更
<p>次の画面に段落情報を表示するようにした。</p> <ul style="list-style-type: none">・プロジェクトサマリ画面（段落数の平均／最小／最大）・[プログラム詳細] 画面のプログラム概要（段落数）・[プログラム詳細] 画面のセクション一覧（手続き名（節名／段落名）／種別／呼出有無／制御の移行先の手続き名／コメント行の文字列）・[ソース表示] 画面の左ペインの表示（段落名）
<p>[解析生成オプション設定] 画面の [主プログラム指定] タブの画面項目を改善し、主プログラムの指定方法の説明を追加した。</p>
<p>[解析生成オプション設定] 画面の [プログラムコメント抽出] タブの画面項目を改善し、プログラムコメントの抽出方法の説明を追加した。</p>
<p>[解析生成オプション設定] 画面の [主プログラム指定] タブの USING 検出条件に「引数 1 個で可変長文字列型の集団項目である」を追加した。</p>
<p>[解析生成オプション設定] 画面の [節コメント抽出] タブの画面項目を改善し、節コメントの抽出方法の説明を追加した。</p>
<p>次の一覧情報 CSV に段落数の情報を追加した。</p> <ul style="list-style-type: none">・プログラム一覧

COBOL2002 Developer Professional 04-21, COBOL2002 Developer Professional(64) 04-21

追加・変更内容
<p>COBOL ソース解析で、一覧情報 CSV のセクション一覧に出力する情報に段落を追加した。</p>
<p>COBOL ソース解析で、[ソース表示] 画面に 80 カラムを超えるソースコードを表示できるようにした。</p>
<p>次に示すメッセージの対処中の項目に、「段落名」を追加した。</p> <p>KEDL1260A-E</p> <p>KEDL1268A-E</p>
<p>COBOL2002 のコンパイラ環境変数 CBLFIXEDFORMLINE を、COBOL ソース解析では使用不可として追加した。</p>
<p>デッドコード候補の検知の注意に、[ソース表示(COPY 展開後)] 画面での節見出しや段落見出しの表示方法を追加した。</p>

変更内容 (3021-3-616-20) COBOL2002 Developer Professional 04-20, COBOL2002 Developer Professional(64) 04-20

追加・変更内容
プログラム呼び出しの実引数と仮引数の整合性をチェックして、不整合引数一覧を出力できるようにした。 これに伴い、開発マネージャで次に示す内容を変更した。 <ul style="list-style-type: none">• CSV 形式の解析結果の格納フォルダを表示するメニューの名称を「解析結果(CSV/TAG)の表示」に変更した。• 「解析生成オプション設定」画面に「引数整合性チェック」タブを追加した。• 「解析生成オプション設定」画面の「生成ドキュメント」タブで一覧情報 CSV に不整合引数一覧を追加した。
COBOL ソース解析で 02～49, 66 レベルのデータ項目を抽出して、データ項目一覧を出力できるようにした。 これに伴い、開発マネージャで次に示す内容を変更した。 <ul style="list-style-type: none">• 「解析生成オプション設定」画面に「その他の解析オプション」タブを追加した。• 「解析生成オプション設定」画面の「生成ドキュメント」タブで一覧情報 CSV にデータ項目一覧を追加した。
次に示すメッセージを追加した。 KEDL1260A-E
COBOL ソース解析で利用できる COBOL2002 のコンパイラオプションに、-CheckUninitData オプションを追加した。

変更内容 (3021-3-616-10) COBOL2002 Developer Professional 04-10, COBOL2002 Developer Professional(64) 04-10

追加・変更内容
適用 OS に「Windows Server 2019」を追加した。
COBOL ソース解析で解析対象とする SQL 埋め込み COBOL ソースファイル（埋め込み SQL 文を含む COBOL ソースファイル）として、次のソースファイルをサポートした。 <ul style="list-style-type: none">• XDM/RD の埋め込み SQL 文を含む COBOL ソースファイル• COBOL 言語用 SQL プリプロセサの UAP ソースファイル
ソース解析情報に未稼働のソフトウェア資産の情報（未使用資産情報）を出力できるようにした。 これに伴い、COBOL ソース解析で生成するドキュメントに、次の CSV ファイルを追加した。 <ul style="list-style-type: none">• セクション一覧.csv• ファイル一覧.csv
cblcsa コマンド使用時に、解析または生成でデータベース管理システムや ODBC のエラーが発生した場合の対処について説明を追加した。
次の COBOL ソース解析の出力結果が意図したとおりではない事例の対処の説明を変更した。 <ul style="list-style-type: none">• 「ページが表示されません」と表示されるページがある
次に示すメッセージを追加した。 KEDL1015A-E, KEDL1017A-E, KEDL1018A-E, KEDL1072A-E, KEDL1134A-E, KEDL1275A-W, KEDL1276A-E, KEDL1277A-W, KEDL1278A-W, KEDL1279A-W, KEDL1280A-W, KEDL1281A-W, KEDL1282A-W, KEDL1283A-E, KEDL1284A-W, KEDL1285A-W, KEDL1286A-W, KEDL1287A-W, KEDL2082A-E, KEDL2083A-E, KEDL2084A-E
次に示すメッセージを変更した。 KEDL1020A-E, KEDL1021A-E, KEDL1057A-E, KEDL1135A-E, KEDL1143A-E, KEDL1268A-W, KEDL2055A-E, KEDL2056A-E

追加・変更内容
次に示すメッセージを削除した。 KEDL1058A-E, KEDL1140A-E, KEDL1269A-W, KEDL1272A-W
COBOL ソース解析で次の構文の制限を解除した。 <ul style="list-style-type: none"> • -EquivRule コンパイラオプション • ほかのプログラムと名前が重複する内部プログラム • 全角と半角の違いだけの同じ字面のプログラム名
COBOL ソース解析が仮定する、節見出しで始まらない手続き先頭の節の名称を、DUMMY_SEC_1 から {PROCEDURE_START}に変更した。
COBOL ソースファイルの解析／生成を実行する前に確認しておく注意事項の説明を追加した。

変更内容 (3021-3-616) COBOL2002 Developer Professional 04-00, COBOL2002 Developer Professional(64) 04-00

追加・変更内容
開発マネージャで COBOL ソース解析の実行をサポートした。これに伴い、次に示す内容を変更した。 <ul style="list-style-type: none"> • 開発マネージャのプロジェクトを対象に COBOL ソース解析を実行するように変更した。 • 開発マネージャで、ソース解析プロジェクトの指定、解析生成オプションの設定、COBOL ソース解析の実行をできるように変更した。 • 解析オプションと生成オプションを同じダイアログで設定するように変更した。 • 解析ログと操作ログの生成内容を変更した。
バッチモードで次に示す操作をサポートした。 <ul style="list-style-type: none"> • ソース解析プロジェクトの作成、名称の変更、削除 • ソース解析プロジェクトの COBOL ソースファイルの登録と削除 • ソース解析プロジェクトの解析生成オプションのインポートとエクスポート • ソース解析用データベースの初期化 • ソース解析オプションファイルの編集 これに伴い、cblcsaSetOptions コマンドを追加した。
次に示すメッセージを追加した。 KEDL0003A-E, KEDL0004A-E, KEDL0005A-E, KEDL0105A-E, KEDL0107A-E, KEDL0108A-W, KEDL0109A-W, KEDL0110A-E, KEDL0111A-E, KEDL0112A-E, KEDL0113A-E, KEDL0114A-E, KEDL0115A-E, KEDL0116A-E, KEDL0117A-E, KEDL0118A-E, KEDL0119A-E, KEDL0124A-E, KEDL1005A-E, KEDL1133A-W, KEDL1229A-E, KEDL1230A-E, KEDL2021A-E, KEDL2025A-E, KEDL2033A-E, KEDL2050A-E, KEDL2051A-E, KEDL2052A-E, KEDL2053A-E, KEDL2054A-W, KEDL2055A-E, KEDL2056A-E, KEDL2057A-E, KEDL2058A-E, KEDL2059A-E, KEDL2060A-E, KEDL2061A-E, KEDL2062A-E, KEDL2063A-W, KEDL2064A-W, KEDL2065A-E, KEDL2066A-E, KEDL2067A-E, KEDL2068A-E, KEDL2069A-W, KEDL2070A-E, KEDL2071A-W, KEDL2072A-E, KEDL2073A-E, KEDL2074A-E, KEDL2075A-E, KEDL2076A-E, KEDL2077A-W, KEDL2078A-W, KEDL2079A-E, KEDL2080A-E, KEDL3009A-E, KEDL9001A-E
次に示すメッセージを変更した。 KEDL0104A-E, KEDL1141A-E, KEDL1142A-E, KEDL1146A-E, KEDL1267A-W, KEDL1271A-E, KEDL2038A-E, KEDL2039A-E, KEDL2044A-E, KEDL2046A-E, KEDL2047A-E, KEDL2055A-E, KEDL2056A-E, KEDL2057A-E, KEDL2058A-E
次に示すメッセージを削除した。

追加・変更内容

KEDL0001A-E, KEDL0002A-E, KEDL1002A-E, KEDL1014A-E, KEDL1015A-E, KEDL1018A-E, KEDL1019A-E, KEDL1022A-W, KEDL1023A-W, KEDL1026A-E, KEDL1027A-E, KEDL1028A-E, KEDL1030A-E, KEDL1032A-W, KEDL1033A-W, KEDL1037A-E, KEDL1038A-E, KEDL1039A-E, KEDL1041A-W, KEDL1043A-E, KEDL1045A-W, KEDL1046A-E, KEDL1047A-E, KEDL1048A-E, KEDL1049A-E, KEDL1050A-E, KEDL1051A-E, KEDL1054A-E, KEDL1055A-E, KEDL1069A-E, KEDL1070A-E, KEDL1072A-E, KEDL1073A-E, KEDL1076A-E, KEDL1078A-E, KEDL1080A-E, KEDL1085A-E, KEDL1088A-E, KEDL1095A-E, KEDL1096A-E, KEDL1097A-E, KEDL1100A-E, KEDL1101A-E, KEDL1102A-E, KEDL1103A-E, KEDL1104A-E, KEDL1105A-E, KEDL1106A-E, KEDL1107A-E, KEDL1109A-E, KEDL1110A-E, KEDL1111A-E, KEDL1112A-E, KEDL1117A-E, KEDL1118A-E, KEDL1119A-E, KEDL1122A-E, KEDL1123A-E, KEDL1125A-E, KEDL1128A-E, KEDL1133A-E, KEDL1134A-E, KEDL1137A-E, KEDL1138A-E, KEDL1139A-E, KEDL1144A-E, KEDL1145A-E, KEDL1147A-E, KEDL1153A-E, KEDL1154A-E, KEDL1155A-E, KEDL1156A-E, KEDL1157A-E, KEDL1158A-E, KEDL1159A-E, KEDL1160A-E, KEDL1161A-E, KEDL1162A-E, KEDL1163A-E, KEDL1164A-E, KEDL1165A-E, KEDL1166A-E, KEDL1167A-E, KEDL1168A-E, KEDL1169A-E, KEDL1170A-E, KEDL1171A-E, KEDL1172A-E, KEDL1173A-E, KEDL1174A-E, KEDL1175A-E, KEDL1176A-E, KEDL1177A-E, KEDL1178A-E, KEDL1179A-E, KEDL1180A-E, KEDL1181A-E, KEDL1182A-E, KEDL1183A-E, KEDL1184A-E, KEDL1185A-E, KEDL1186A-E, KEDL1187A-E, KEDL1188A-E, KEDL1189A-E, KEDL1190A-E, KEDL1191A-E, KEDL1192A-E, KEDL1193A-E, KEDL1194A-E, KEDL1195A-E, KEDL1196A-E, KEDL1197A-E, KEDL1198A-E, KEDL1199A-E, KEDL1201A-E, KEDL1202A-E, KEDL1203A-E, KEDL1204A-E, KEDL1205A-E, KEDL1206A-E, KEDL1207A-E, KEDL1208A-E, KEDL1209A-E, KEDL1210A-E, KEDL1211A-E, KEDL1212A-E, KEDL1213A-E, KEDL1214A-E, KEDL1215A-E, KEDL1216A-E, KEDL1217A-E, KEDL1224A-E, KEDL1225A-E, KEDL1228A-E, KEDL1230A-E, KEDL1237A-W, KEDL1238A-W, KEDL1239A-E, KEDL1243A-I, KEDL1244A-E, KEDL1245A-E, KEDL1246A-E, KEDL1247A-E, KEDL1248A-E, KEDL1249A-E, KEDL1250A-E, KEDL1251A-W, KEDL1275A-E, KEDL2005A-E, KEDL2006A-E, KEDL2008A-E, KEDL2009A-E, KEDL2010A-W, KEDL2011A-W, KEDL2012A-E, KEDL2013A-Q, KEDL2014A-E, KEDL2015A-Q, KEDL2017A-E, KEDL2018A-I, KEDL2019A-E, KEDL2020A-E, KEDL2023A-Q, KEDL2024A-E, KEDL2025A-E, KEDL2027A-I, KEDL2031A-E, KEDL2032A-W, KEDL2034A-Q, KEDL2401A-E

PROCEDURE USING または CALL 文の引数の個数の上限値を、各引数の（引数の名前のバイト数 + 1）を第 1 引数から順に合計した長さが 320 バイトに納まる個数に変更した。

COBOL ソース解析で利用できるコンパイラオプションに、-SpaceAsZero オプションを追加した。

付録 D このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 D.1 関連マニュアル

関連マニュアルを次に示します。必要に応じてお読みください。

COBOL2002 Professional 製品

COBOL2002 Professional 製品 導入ガイド (3021-3-615)

COBOL2002 Professional Tool Kit データ影響波及分析ガイド (3021-3-617)

COBOL2002 Professional Tool Kit 単体テスト支援ガイド (3021-3-618)

COBOL2002

COBOL2002 ユーザーズガイド (3021-3-600)

COBOL2002 操作ガイド (3021-3-601)

COBOL2002 言語 標準仕様編 (3021-3-604)

COBOL2002 言語 拡張仕様編 (3021-3-605)

COBOL2002 メッセージ (3021-3-609)

付録 D.2 このマニュアルでの表記

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

マニュアルでの表記		製品名
Excel		Microsoft Excel
		Microsoft Office Excel
Internet Explorer		Microsoft Internet Explorer
		Windows Internet Explorer
Visual Basic		Microsoft Visual Basic
Windows 7	Windows 7(x86)	Microsoft Windows 7 Professional 日本語版(32 ビット版)
		Microsoft Windows 7 Enterprise 日本語版(32 ビット版)
		Microsoft Windows 7 Ultimate 日本語版(32 ビット版)
	Windows 7(x64)	Microsoft Windows 7 Professional 日本語版(64 ビット版)
		Microsoft Windows 7 Enterprise 日本語版(64 ビット版)
		Microsoft Windows 7 Ultimate 日本語版(64 ビット版)

マニュアルでの表記		製品名
Windows 8.1	Windows 8.1(x86)	Windows 8.1 Pro 日本語版(32 ビット版)
		Windows 8.1 Enterprise 日本語版(32 ビット版)
	Windows 8.1(x64)	Windows 8.1 Pro 日本語版(64 ビット版)
		Windows 8.1 Enterprise 日本語版(64 ビット版)
Windows 10	Windows 10(x86)	Windows 10 Pro 日本語版(32 ビット版)
		Windows 10 Enterprise 日本語版(32 ビット版)
	Windows 10(x64)	Windows 10 Enterprise 日本語版(64 ビット版)
		Windows 10 Pro 日本語版(64 ビット版)
Windows Server 2008 R2		Microsoft Windows Server 2008 R2 Standard 日本語版
		Microsoft Windows Server 2008 R2 Enterprise 日本語版
		Microsoft Windows Server 2008 R2 Datacenter 日本語版
Windows Server 2012		Microsoft Windows Server 2012 Standard 日本語版
		Microsoft Windows Server 2012 Datacenter 日本語版
Windows Server 2012 R2		Microsoft Windows Server 2012 R2 Standard 日本語版
		Microsoft Windows Server 2012 R2 Datacenter 日本語版
Windows Server 2016		Microsoft Windows Server 2016 Standard 日本語版
		Microsoft Windows Server 2016 Datacenter 日本語版
Windows Server 2019		Windows Server 2019 Standard 日本語版
		Windows Server 2019 Datacenter 日本語版

- Windows 7, Windows 8.1, Windows 10, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, および Windows Server 2019 とで機能差がない場合、Windows と表記しています。

また、このマニュアルでは、製品名を次のように表記しています。

表記	製品名
COBOL2002	COBOL2002 Net Developer
	COBOL2002 Net Developer(64)

付録 D.3 KB（キロバイト）などの単位表記について

1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ 1,024 バイト、 $1,024^2$ バイト、 $1,024^3$ バイト、 $1,024^4$ バイトです。

索引

記号

-SQLAutoCopy コンパイラオプション 345

C

cblcsaSetOptions コマンド - ソース解析オプションファイルを編集する 113

cblcsa コマンド - バッチモードで COBOL ソース解析を実行する 100

COBOL2002 の言語仕様の解析結果 249

COBOL2002 のコンパイラオプション 251

COBOL2002 のコンパイラ環境変数 258

COBOL 資産を棚卸する 54

COBOL ソース解析で利用できるコンパイラオプション 251

COBOL ソース解析で利用できるコンパイラ環境変数 258

COBOL ソース解析の基本的な使い方 21

COBOL ソース解析の仕様範囲のリファレンス 243

COBOL ソース解析のメッセージ (KEDL0000～KEDL0999) と対処 226

COBOL ソース解析のメッセージ (KEDL1000～KEDL1999) と対処 228

COBOL ソース解析のメッセージ (KEDL2000～KEDL2999) と対処 234

COBOL ソース解析のメッセージ (KEDL3000～KEDL3999) と対処 239

COBOL ソース解析のメッセージ (KEDL9000～KEDL9999) と対処 240

COBOL ソース解析のログを参照する 52

COBOL ソース解析を実行できない [トラブルシュート] 209

COBOL ソースファイル一覧 169

[COBOL ソースファイル一覧] 画面 121

COBOL ソースファイルの内容と差異があるソース解析情報の項目 160

COBOL ソースファイルをすべて解析する 22

CSV ファイル形式のソース解析情報を加工する 49

CSV ファイルの種類 168

H

HTML ファイル形式のソース解析情報を参照する 38

い

一意名呼び出し対応付けファイル 342

一時ファイルが削除されない [トラブルシュート] 213

一覧情報 CSV 169

う

埋め込み SQL 文を含む COBOL ソースファイルの解析 267

え

エラーがある COBOL プログラムを修正する 47

エラー時の対処 208

か

解析エラー発生時のエラー位置の表示 226

解析結果の出力先の変更が開発マネージャに反映されない 213

解析結果のリファレンス (CSV ファイル形式) 167

解析結果のリファレンス (HTML ファイル形式) 114

解析した COBOL ソースファイルの結果を参照する 35

解析したプログラムの情報とソースを参照する 40

解析情報の上限値 244

[解析生成オプション設定] 画面 83

解析前に前回のソース解析情報を残す 44

解析ログ 47

解析ログを参照する 215

解析を補助する機能 342

開発マネージャでの COBOL ソース解析の操作メニュー項目と設定画面 80

各種の分布情報を参照する 41

関係情報 CSV 200

き

[記述項数の分布] 155

基本ブロック 323

行数、行番号に関する項目 161

[行数の分布] 155

こ

効率良くかつ効果的にソースコードリーディングを実施する 19

コマンドリファレンス 99

コメント行の文字列を解析結果に表示する 18

コンパイラオプション 251

[コンパイラオプション] タブ 83

コンパイラ環境変数 258

さ

サービスルーチンファイル 344

し

システムエラー番号 240

システム固有画面 121

実行順序ツリー 324

[実行順序表示] 画面 151

実行順序表示機能 324

実行順序リスト 196

実行文として数える文 276

[実行文の数の分布] 156

出力結果が意図したとおりではない〔トラブルシュー
ト〕 210

[主プログラム指定] タブ 85

主プログラム自動検出 302

主プログラムの指定方法 301

主プログラム名による主プログラム指定 304

使用上の注意事項 333

処理ロジックが複雑なプログラムを調査する 71

事例から探す〔トラブルシュート〕 209

せ

制御フロー解析 323

制御フロー解析による実行順序表示機能 323

制御フローグラフ 323

制御フロー情報 323

制御ブロック 323

制限がある構文 246

[生成ドキュメント] タブ 96

セクション一覧 174

[セクション階層図] 画面 149

節コメント 314

節コメント行 315

節コメント抽出 314

[節コメント抽出] タブ 88

節コメントの抽出 314

[節数の分布] 157

設定した解析環境を保存/読み出しをして使う 18

そ

操作ログ 52

操作ログを参照する 217

ソース解析情報 13

ソース解析情報の画面遷移 115

ソース解析情報の画面遷移（各種の分布情報） 116

ソース解析情報の画面遷移（システム全体の情報）
115

ソース解析情報の画面遷移（デッドコード情報） 116

ソース解析情報の画面遷移（プログラム固有の情報）
115

ソース解析情報の出力先を変更する 45

ソース解析情報の内容 15

ソース解析情報の表示規則 160

ソース解析情報のフォルダを退避する 44

ソース解析プロジェクトの解析生成の実行で想定以上
に時間が掛かる 213

[ソース表示] 画面 147

ソース表示で表示されるソース行の最大長 166

ソース表示の色と行数カウント 161

そと PERFORM 文からの呼出先情報の抽出規則 321

[その他の解析オプション] タブ 94

た

[段落数の分布] 158

ち

[重複するプログラム一覧] 画面 125

つ

使われていないソースコードの候補を確認する 42

て

データ項目一覧 180

デッドコード候補の検知方法 259

[デッドコード情報一覧] 画面 136

と

[登録集原文一覧] 画面 126

登録集原文参照関係 202

[登録集原文詳細] 画面 128

[登録集原文ファイル一覧] 画面 122

は

バッチモードで COBOL ソース解析を実行する 19, 77

バッチモードで実行する 76

ひ

引数整合性チェック 287

引数整合性チェック使用時の注意事項 299

[引数整合性チェック] タブ 92

引数整合性チェックの検査内容 288

[表(DB)一覧] 画面 132

表(DB)参照関係 205

[表(DB)詳細] 画面 132

ふ

ファイル一覧 177

[ファイル一覧] 画面 130

ファイル参照関係 204

[ファイル詳細] 画面 131

不整合引数一覧 190

プログラム一覧 172

[プログラム一覧] 画面 123

[プログラム階層図] 画面 133

プログラムコメント 306

プログラムコメント行 307

プログラムコメント抽出 306

[プログラムコメント抽出] タブ 86

プログラムコメントの抽出 306

プログラム固有の情報 138

[プログラム詳細] 画面 138

[プログラムデータ] 画面 146

プログラムの処理を調査する 56

プログラムの入出力情報を調査する 59

プログラム呼出関係 200

プログラム呼び出しの実引数と仮引数の整合性をチェックする 68

プログラム呼び出しの引数の整合性をチェックする 19

プロジェクトサマリ画面 117

[分岐の数の分布] 156

[分布情報一覧] 画面 154

へ

変換して表示されるソース解析情報の項目 160

変更した COBOL ソースファイルだけを解析する 18, 31

み

[未登録の登録集原文一覧] 画面 127

[未登録のプログラム一覧] 画面 124

め

メインフレームの COBOL ソースファイルを解析する 19

メインフレームの固定長形式がある COBOL ソースファイルを解析する 66

メッセージから探す [トラブルシュート] 225

メッセージの一覧 226

メッセージの形式 225

よ

呼出先情報 321

[呼出先プログラム解決] タブ 90

[呼出先プログラム数の分布] 159

ろ

ログが蓄積されない〔トラブルシュート〕 212

ログを参照する 215