

COBOL Enterprise Edition V5.0 / COBOL Standard Edition V5.0 開発環境のご紹介

2022年 4月

日本電気株式会社

目次

1. COBOL開発環境とは
 2. 編集機能
 3. ビルド機能
 4. プロジェクト管理
 5. 資産管理
 6. インポート機能
 7. 連携機能
 8. 動作環境
- 付録A.コンパイル概念（A-VXとの違い）
お問い合わせ先

1. COBOL開発環境とは

EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

1.1. COBOL開発環境

COBOL Enterprise Edition Developerのみ

◆ COBOL開発環境

- COBOLアプリケーションの開発をサポートするWindowsの総合開発環境^(*1)
- プログラムの作成からコンパイルまでの全ての作業をCOBOL開発環境上で行うことができるため、効率的な開発が可能
- Open COBOL Factory 21のWorkbenchやCOBOL85 Remote Workbench for IPFのユーザインタフェースを採用

ビルド機能

コンパイルとリンクを行い、成果物を作成する機能

連携機能

関連製品(Oracle Pro*COBOLなど)やオプション製品およびツールと連携する機能

編集機能

専用のエディタによりCOBOLソースファイルの編集を行う

資産管理^{*2}

外部管理ツールを利用して、複数人数での開発や資産のバージョン管理を行う機能

プロジェクト管理

成果物を作成するファイルの集まりを管理する機能

インポート機能^{*3}

既存COBOL資産^{*4}をCOBOL開発環境で扱うことができる形式にインポートする機能

*1:Windows版はリモートデスクトップ接続でサーバ上の開発環境を操作。Linux版はWindows上の開発環境からLinuxサーバへリモート接続

*2:COBOL開発環境による外部ツールとの自動連携は未サポート。

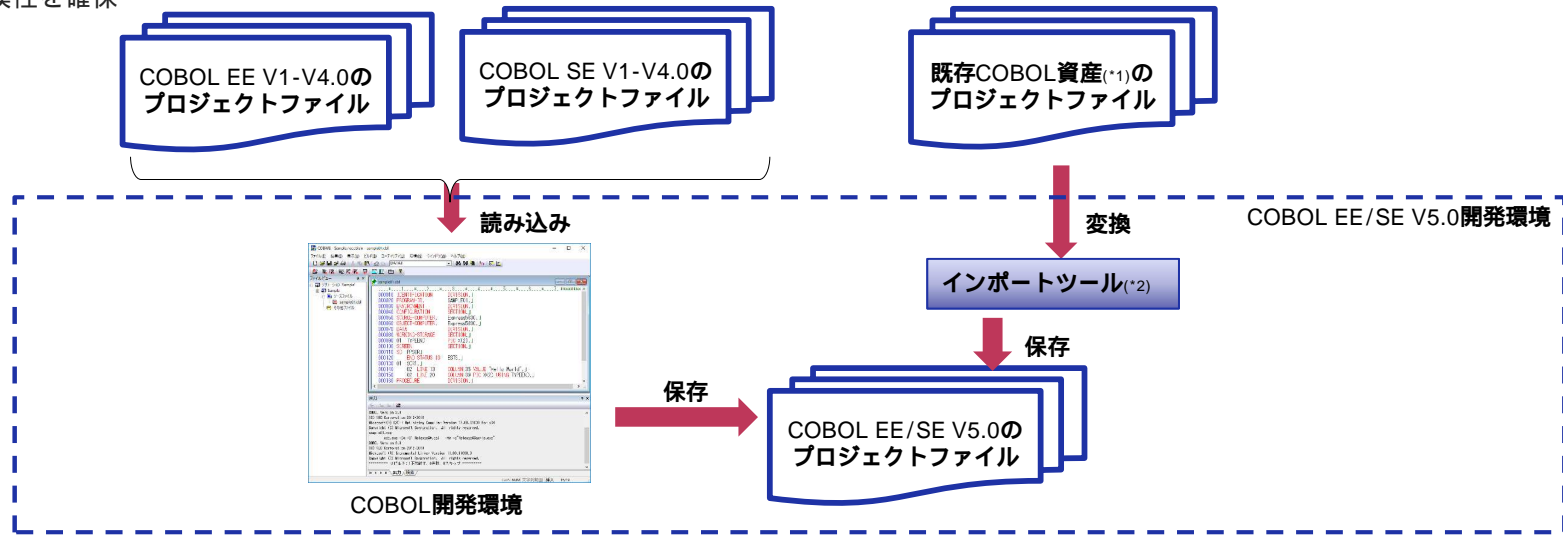
*3:Linux版は非対応

*4:既存COBOL資産とは Open COBOL Factory 21のWorkbenchのプロジェクトファイルやソースファイルを指す

< 参考 > プロジェクトファイルの互換性

◆ プロジェクトファイルの互換性

- COBOL Enterprise Edition V1-V4.0開発環境やCOBOL Standard Edition Ver1.0のVisual Studioアドイン形式のプロジェクトファイルと物理的な互換があるため、そのまま読み込みが可能
- 旧バージョンのプロジェクトファイルも、そのまま読み込み可能
- Open COBOL Factory 21のWorkbenchのプロジェクトファイルは、COBOL開発環境に付属するインポートツールにより形式を変換することで互換性を確保

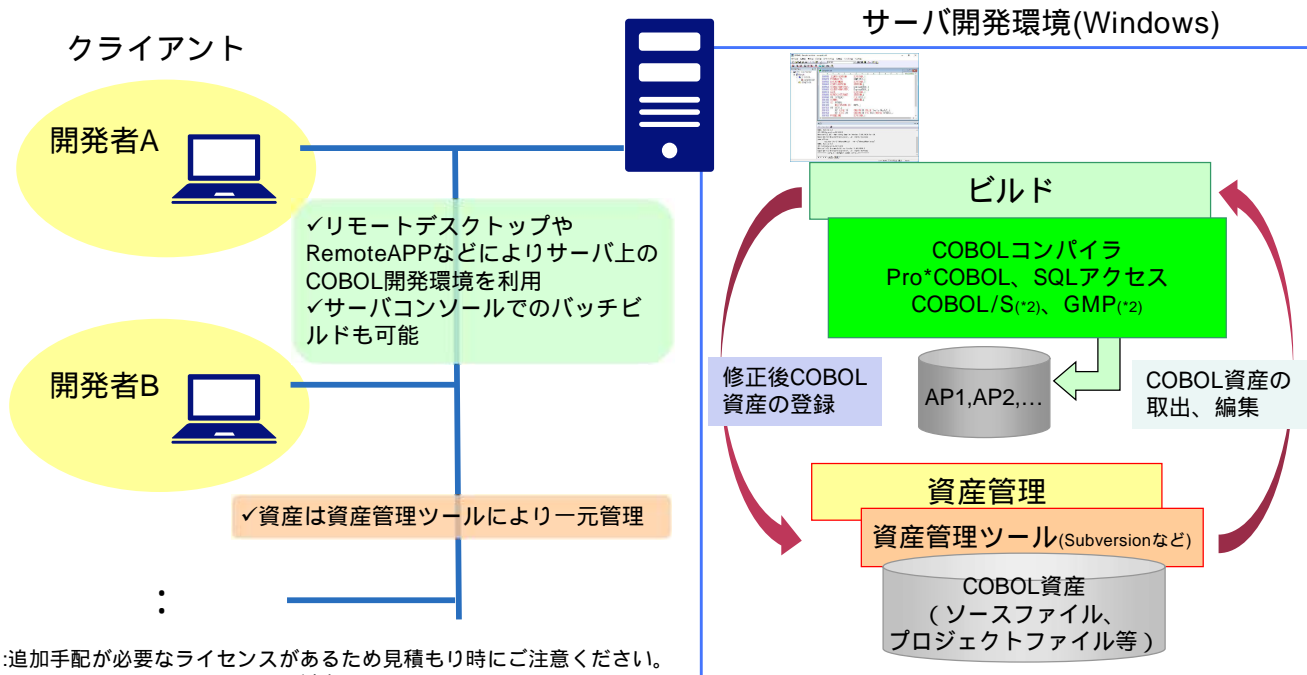


*1:Open COBOL Factory 21/COBOL85 ProおよびOpen COBOL Factory 21/COBOL85のWorkbench、Open COBOL Factory 21/COBOL Remote Workbench、Open COBOL Factory 21/COBOL85 MFA Remote Workbench のプロジェクトファイルやソースファイルを指す

*2:Linux版は非対応

< 参考 > 複数名での開発(Windows)(*1)

- ◆ 資産の一元管理や排他、管理者による一括ビルド、大規模システムでの複数開発者によるサーバOS上での開発は、リモートデスクトップ形態で実現可能

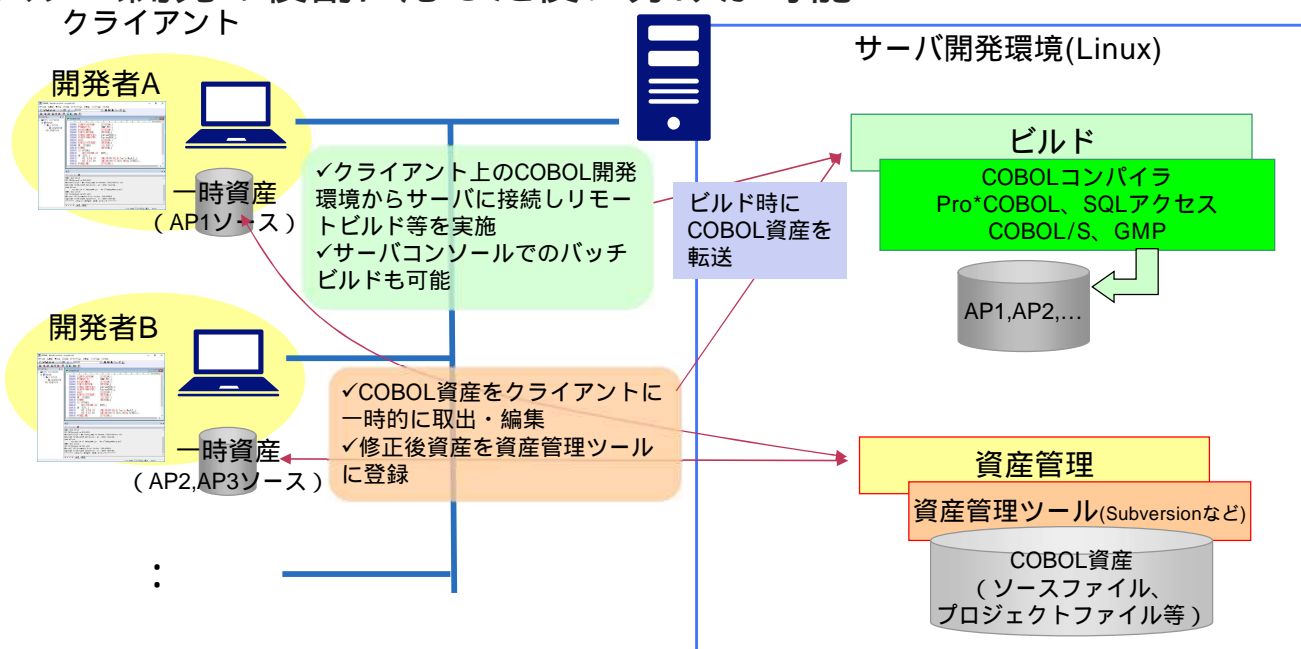


*1:追加手配が必要なライセンスがあるため見積もり時にご注意ください。

*2:COBOL Enterprise Editionのみ対応

< 参考 > 複数名での開発(Linux)(*1)

- ◆ 資産の一元管理や排他により、管理者による一括ビルドやバッチビルド、複数開発者によるクライアントでの編集作業やサーバOS上でのリモートビルドなど、大規模システム開発の役割に応じた使い分けが可能



2. 編集機能

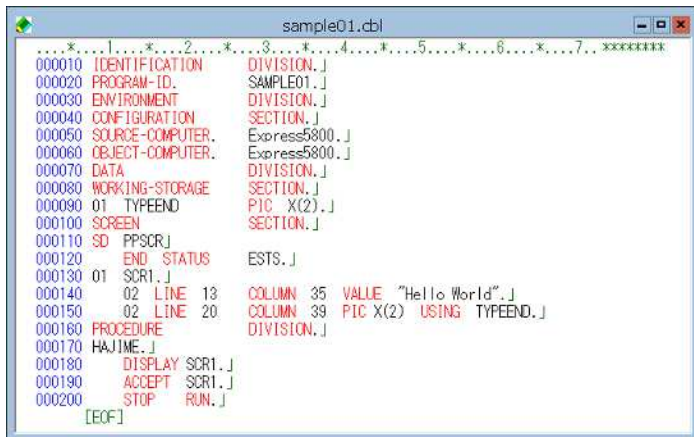
EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

2 . 編集機能

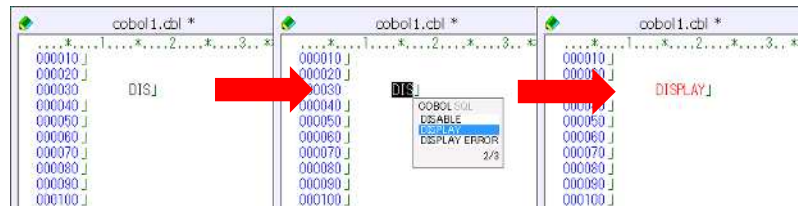
◆ 生産性・効率性を向上させるCOBOL専用エディタをサポート

- 予約語（キーワード）の強調表示と補完機能備えて簡単に編集が可能
- 一連番号領域、識別領域などが分けられ、プログラムの可読性が向上
- COBOL特有の一連番号の振り直し、識別領域の編集が可能
- 検索・置換・ファイル検索・文字変換など一般的なテキストエディタの機能が充実
- 固定形式正書法/自由形式正書法による表示形式の自動変更が可能



◆ キーワード（予約語）補完機能

- 長い予約語も入力が簡単
最初の数文字を入力すると、一覧が表示され、選択すると補完が完了



◆ 領域区分

- スケール行表示で一連番号領域、標識領域、識別領域の区分けが一目瞭然

```

sample01.cbl
.....*.....1.....2.....3.....4.....5.....6.....7.....*.....
00010 IDENTIFICATION DIVISION.
00020 PROGRAM-ID. SAMPLE01.
00030 ENVIRONMENT DIVISION.
00040 CONFIGURATION SECTION.
00050 SOURCE-COMPUTER. Express5800.
00060 OBJECT-COMPUTER. Express5800.
00070 DATA DIVISION.
00080 WORKING-STORAGE SECTION.
00090 01 TYPEND PIC X(2).
00100 SCREEN SECTION.
00110 SD PPSOR.
00120 END STATUS ESTS.
00130 01 SCR1.
00140 02 LINE 13 COLUMN 35 VALUE "Hello World".
00150 02 LINE 20 COLUMN 39 PIC X(2) USING TYPEND.
00160 PROCEDURE DIVISION.
00170 HALTME.
00180 DISPLAY SCR1.
00190 ACCEPT SCR1.
00200 STOP RUN.
[EOF]
  
```

スケール行表示

◆ 自由形式正書法のサポート

- cbl,cob,cbx,cbfそれぞれの形式にあわせて表示する

固定形式正書法

sample.cbl

```

...*...1...*...2...*...3...*...4...*...5...*...6...*...7...*****
000010 IDENTIFICATION DIVISION. J
000020 PROGRAM-ID. SAMPLE01. J
000030 ENVIRONMENT DIVISION. J
000040 CONFIGURATION SECTION. J
  
```

sample.cob *

```

...*...1...*...2...*...3...*...4...*...5...*...6...*...7...*...8...
000010 IDENTIFICATION DIVISION. J
000020 PROGRAM-ID. SAMPLE01. J
000030 ENVIRONMENT DIVISION. J
  
```

sample.cbx *

```

...*...1...*...2...*...3...*...4...*...5...*...6...*...7...*...8...
000010 IDENTIFICATION DIVISION. J
000020 PROGRAM-ID. SAMPLE01. J
000030 ENVIRONMENT DIVISION. J
  
```

.cblの場合、80桁固定長

.cobの場合、識別領域なし、
最大255桁可変長

.cbxの場合、識別領域あり、
最大255桁可変長

自由形式正書法

sample.cbf *

```

...*...3...*...4...*...5...*...6...*...7...*...8...
IDENTIFICATION DIVISION. J
PROGRAM-ID. SAMPLE01. J
ENVIRONMENT DIVISION. J
  
```

.cbfの場合、ソース行のみ、
最大255桁可変長

3. ビルド機能

EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

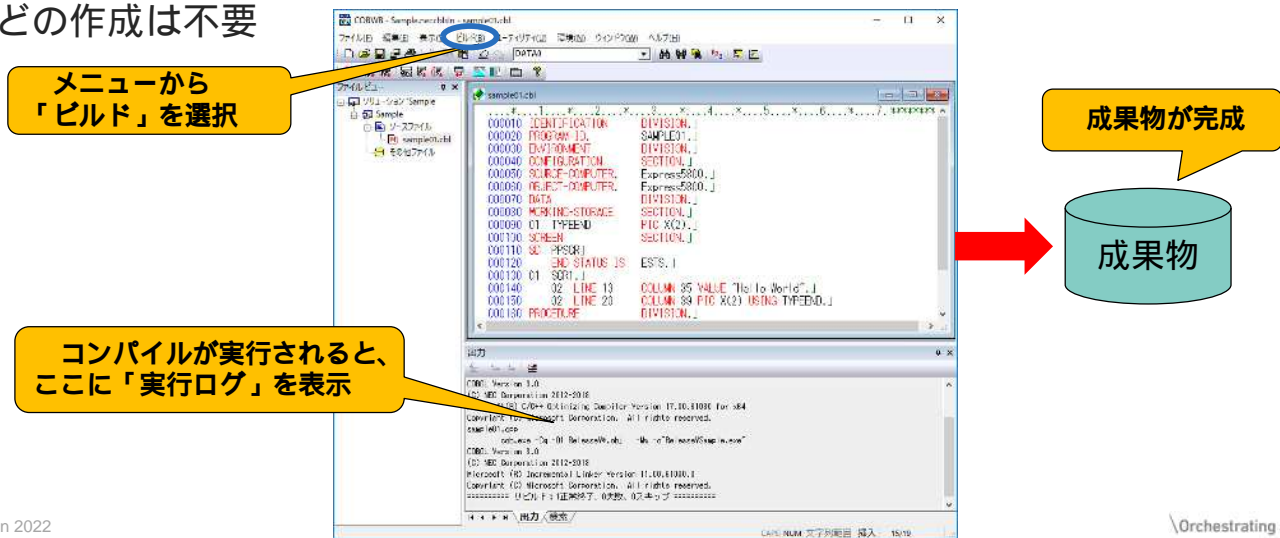
3 . ビルド機能

◆ ビルド

- 対象プログラムのコンパイルやリンクを行い、実行モジュールやライブラリなどの成果物を生成する機能

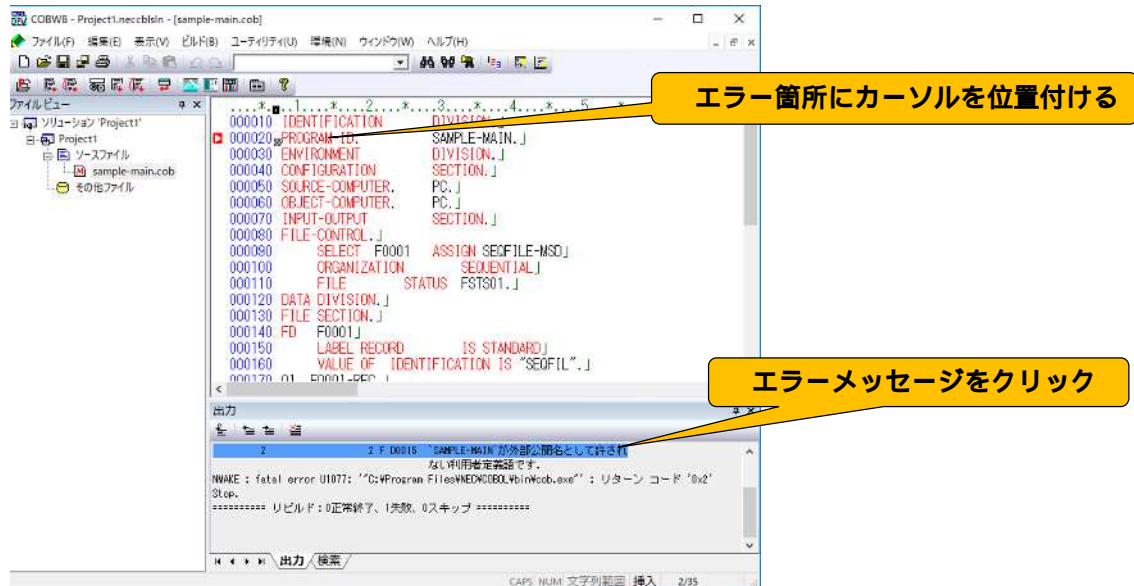
◆ メニューから「ビルド」を選択するだけの簡単操作

- コンパイルの必要性やコンパイルの順序等は開発環境が自動的に判断。makefileやバッチファイルなどの作成は不要



◆ エラー位置付け

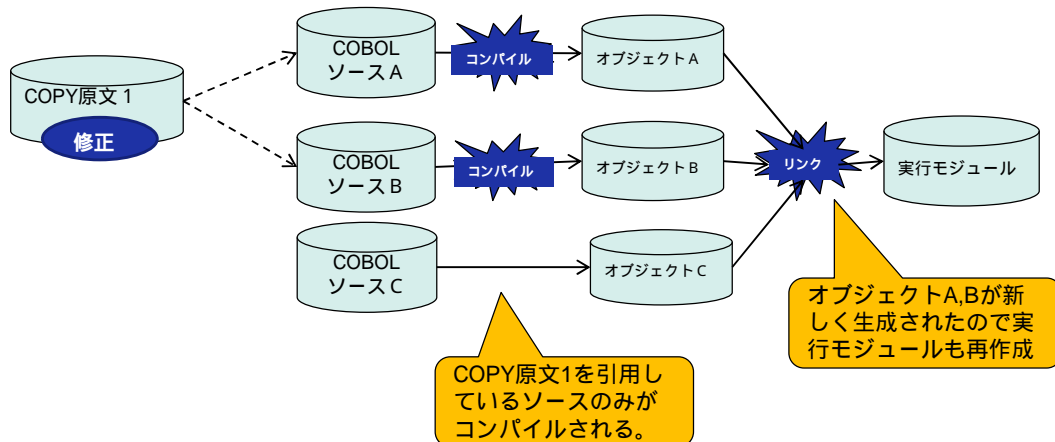
- 出力ウィンドウに表示されるエラーメッセージをマウスでダブルクリックすることにより、エラーが発生したソース位置にカーソルを位置付けることが可能



◆ タイムスタンプを比較し再コンパイルが必要なプログラムを自動抽出

- 各資産の依存関係^(*1)とタイムスタンプを参照し、修正/変更のあったプログラム、および、そのプログラムから影響を受けるプログラムのみを、再コンパイル・リンク可能

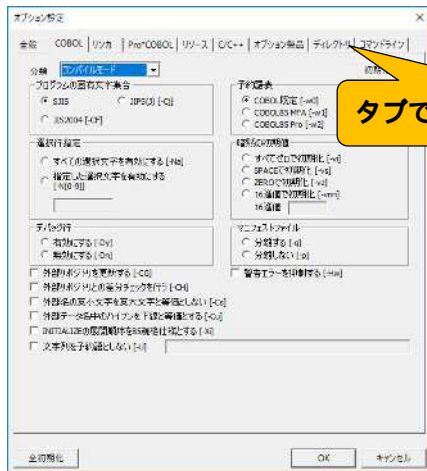
例) 実行モジュールがプログラムA,B,CおよびCOPY原文1から構成されている場合、COPY原文1を変更すると、それに依存したプログラムA、Bのみコンパイルされ、COPY原文1に依存しないプログラムCは再コンパイルされない。



*1:事前に依存関係の登録が必要です

◆ オプション設定

- オプション設定は、カテゴリごとに分類されたタブの指定形式
- 組み合わせ不可な項目は最初から選択できないので、オプション指定ミスによるコンパイルエラーや意図しないオブジェクトの作成を回避することが可能
- 設定したオプション内容をまとめて表示可能



タブで各オプションを分類



設定した内容をまとめて表示

4. プロジェクト管理

EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

4 . プロジェクト管理

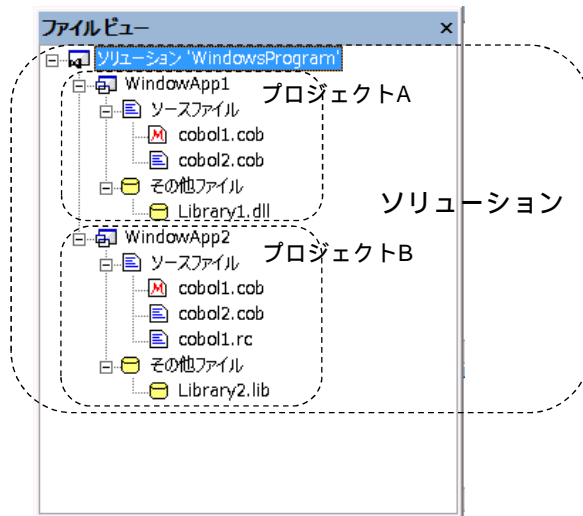
◆ プロジェクトとソリューション

■ プロジェクト

- 一つの成果物（実行モジュールや動的ライブラリ）を構成するファイルの集まり

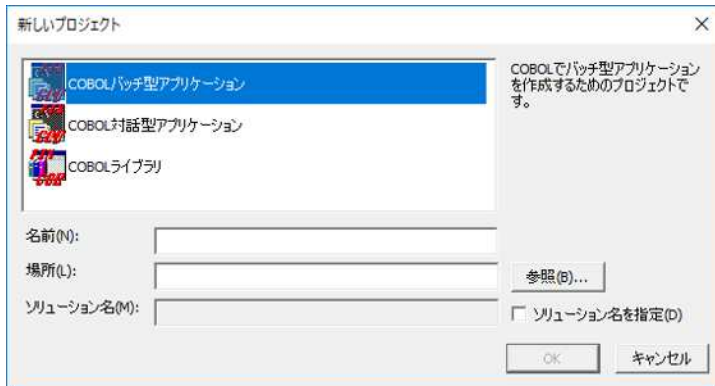
■ ソリューション

- 複数のプロジェクトをまとめて管理



◆ アプリケーションの種別

■ COBOLアプリケーションの種別をプロジェクト作成時に選択可能



プロジェクトの種類	アプリケーションの種別
COBOLバッチ型アプリケーション	COBOLバッチ型アプリケーション
COBOL対話型アプリケーション	COBOL対話型アプリケーション(*1) COBOL GUIアプリケーション(*2) COBOL画面節移行GUIアプリケーション(*2)
COBOLライブラリ	COBOLライブラリ

*1:COBOL Standard EditionおよびCOBOL Enterprise EditionのSE動作モードのみ対応

*2:COBOL Standard Editionのみ対応

5. 資産管理

EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

◆ 資産管理とは

- 複数人での開発や開発資産のバージョン管理で必要となる以下の管理を実施
 - 資産の排他制御
 - 修正履歴の表示^{*1}
 - バージョン間での相違点表示^{*1}
 - 差分のマージ^{*1}

◆ COBOL資産管理の流れ

1. COBOL資産(プログラム, COPY原文, プロジェクトファイル, メイクファイルなど)は資産管理ツールにて管理
2. 資産管理ツールからCOBOL資産を一時的に開発マシンに取り出す
3. COBOL開発環境でプログラム修正やプロジェクトファイルの更新、ビルド実施
4. 修正後のCOBOL資産を資産管理ツールへ登録

^{*1}:別途用意する外部ツールで実現
COBOL開発環境での自動連携は未サポート

6. インポート機能

EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

6. インポート機能

◆ 既存COBOL資産インポート

■ 次の既存製品に付属のWorkbenchのプロジェクトファイルやソースファイルをCOBOL開発環境で扱える形式に変換して取り込むことが可能

- Open COBOL Factory 21/COBOL85 Pro
- Open COBOL Factory 21/COBOL85
- Open COBOL Factory 21/COBOL Remote Workbench
- Open COBOL Factory 21/COBOL85 MFA Remote Workbench

インポート対象の
既存資産を指定

結果出力先を指定

必要に応じて追加の検索ディレクトリお
よびインポート処理のオプションを指定

実行ボタンでインポート

7. 連携機能

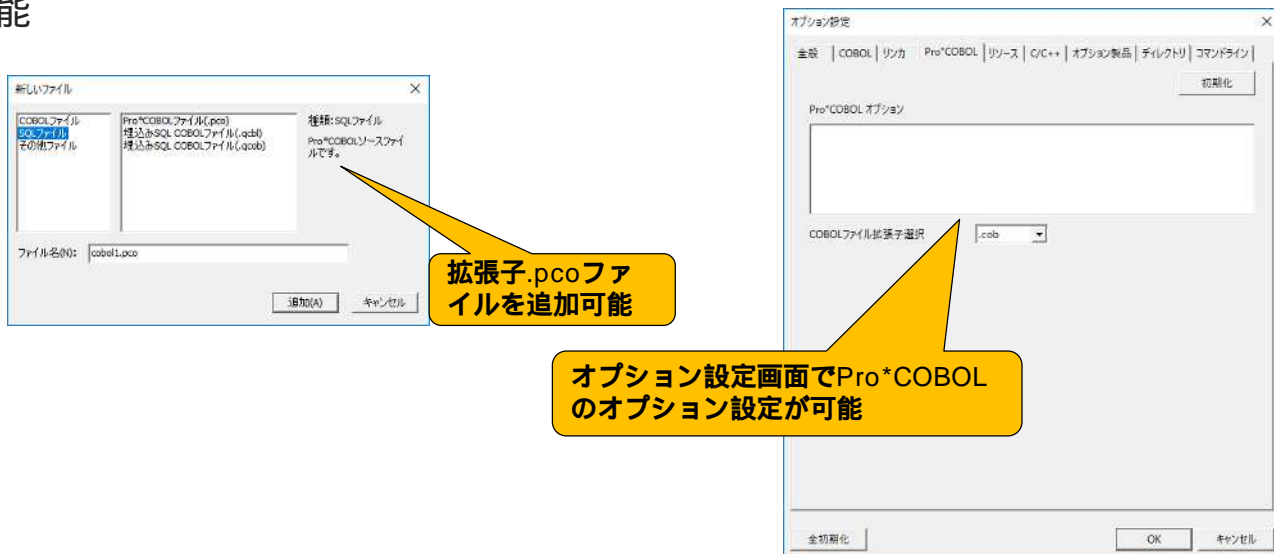
EE : COBOL Enterprise Editionで提供する機能

SE : COBOL Standard Editionで提供する機能

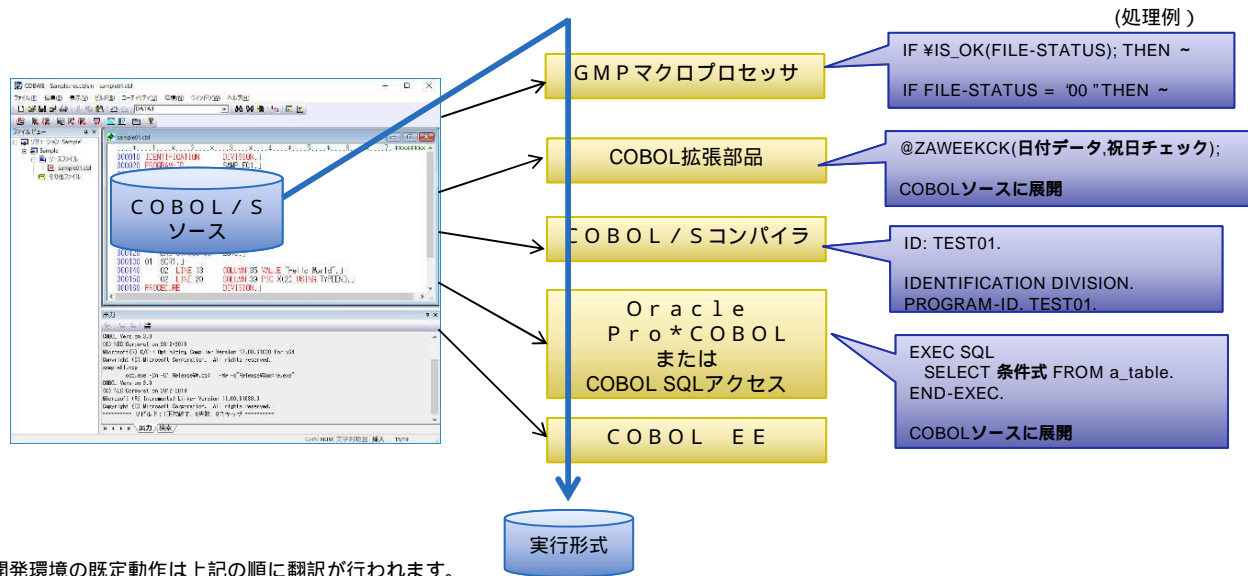
7. 連携機能

◆ 関連製品連携

- プロジェクトにPro*COBOLやC/C++のソースを登録すると、ビルド時にPro*COBOLプリコンパイラの起動やCコンパイラの起動、必要ライブラリのリンクを自動的に実行
- COBOL開発環境の画面で[Pro*COBOL]や[C/C++]のタブを用意しており、オプションなどの指定が可能



- ◆ COBOL/S言語、GMPマクロなどの連携対応 プログラムに、GMPマクロ（¥ ~）記述、SQL文（EXEC SQL）の記述がある場合には、それぞれ、GMP、Oracle Pro*COBOL（もしくはSQLアクセス機能）等呼び出し、翻訳します。



<参考>

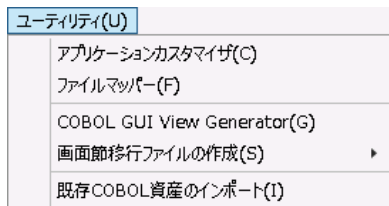
- ・COBOL開発環境の既定動作は上記の順に翻訳が行われます。
- ・「COBOL/Sマクロの中にGMPマクロの記述がある」などで処理順の変更が必要な場合、プロジェクトファイルの「プリコンパイラのビルド設定」の設定情報で変更可能です。

<補足>

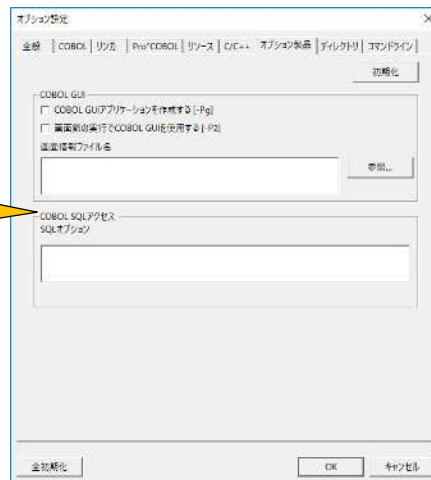
- ・Windows版では標準動作モードのみ連携します。

◆ オプション製品、付属ツール連携

- アプリケーションカスタマイザなどの付属ツールの呼び出し実行が可能
- オプション製品の関連設定を実施することが可能

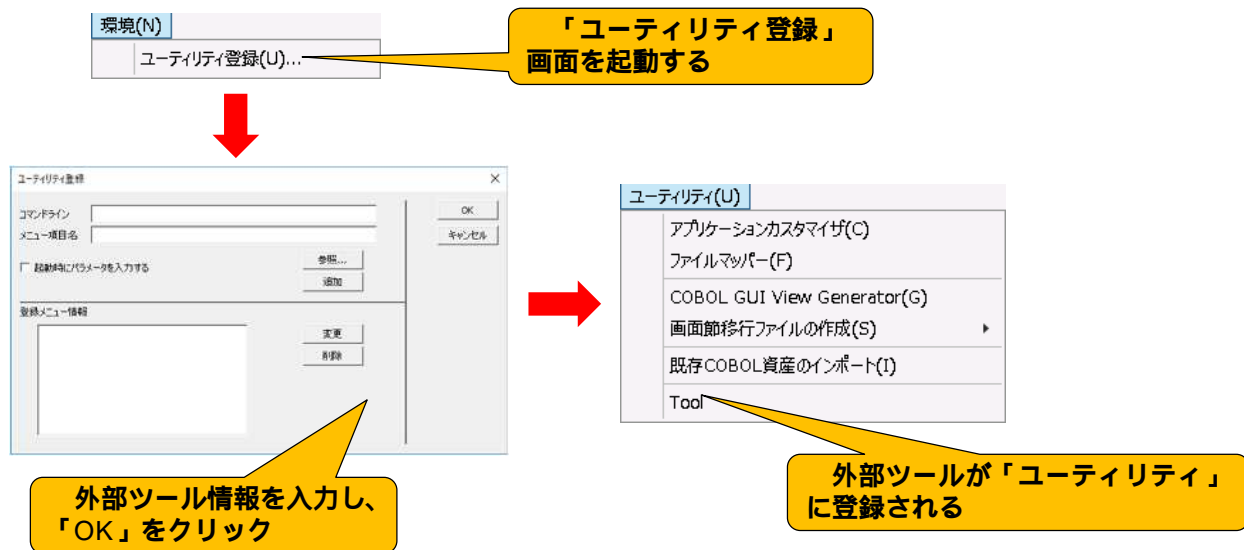


オプション製品
の設定が可能



◆ 外部ツール連携

- 外部ツールを簡単に呼び出せるように登録しておくことが可能



8. 動作環境

8.1. 動作環境(Windows)

製品	OS	Visual Studio
COBOL Standard Edition Developer V5.0	Windows 10 (64ビット版) Windows Server 2016 Windows Server 2019	Visual Studio 2015 Visual Studio 2017 Visual Studio 2019
COBOL Enterprise Edition Developer V5.0	Windows 10 (64ビット版) Windows Server 2016 Windows Server 2019	Visual Studio 2015 Visual Studio 2017 Visual Studio 2019



今後リリースされるOSやVisual Studioの対応は？

ご安心ください。

OSやVisual Studioの新バージョン出荷後、半年～1年目途にCOBOL製品の動作保証を実施予定です。

なお、動作保証は出荷済のCOBOL製品で対応する場合と、新バージョンのCOBOL製品で対応する場合があります。

8.2. 動作環境(Linux)

◆ COBOL Enterprise Edition Developer V5.0

■ サーバ開発環境

- Red Hat Enterprise Linux 7 x86_64
- Red Hat Enterprise Linux 8 x86_64 (サポート対応バージョンの詳細は別途ご確認ください)

■ クライアント開発環境

(COBOL Enterprise Edition Developer クライアントライセンス V5.0も含む)

- Windows 10(64ビット版)



クライアント開発環境にVisual Studio不要！

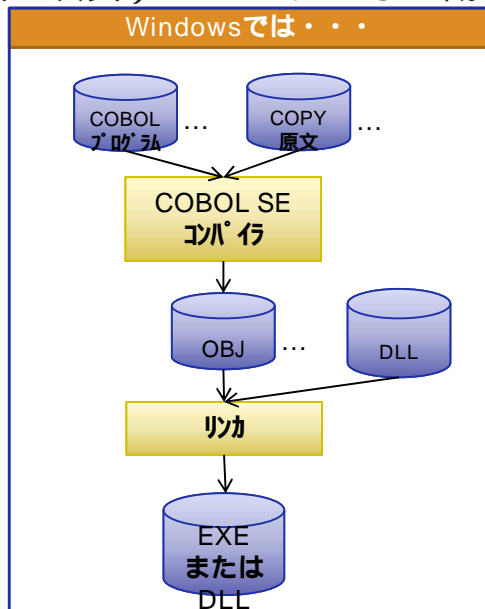
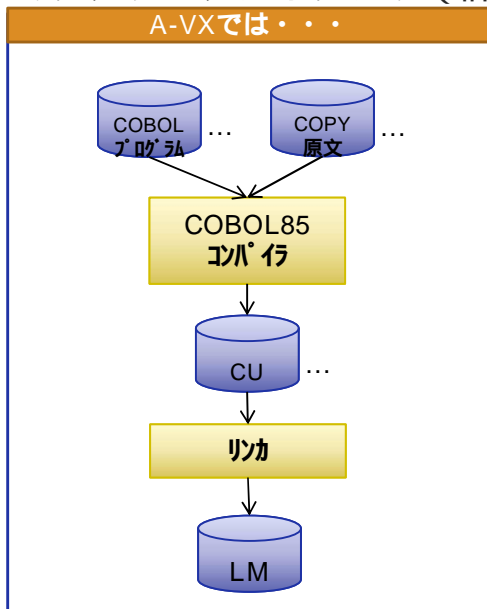
V2.0までは、クライアント開発環境にVisual Studioが別途必要でした。
V3.0からは独自I/Fの開発環境提供により、Visual Studioは不要となりました。

付録A コンパイル概念（A-VXとの違い）

A-VXからCOBOL Standard Editionへ移行されるお客様向け

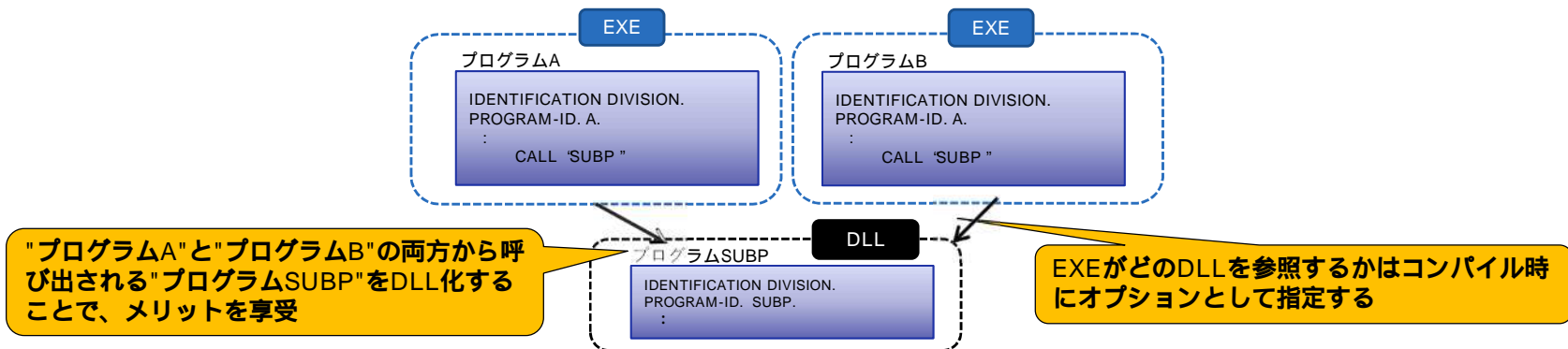
A-1 . コンパイル時の生成物の違い

- ◆ COBOL Standard Editionは、OBJ、EXEおよびDLLの生成が可能
 - OBJ：オブジェクトファイル。A-VXにおけるCUに相当
 - EXE：実行形式。A-VXにおけるLMに相当
 - DLL：ダイナミックリンクライブラリ（詳細は次頁） A-VXにはない概念



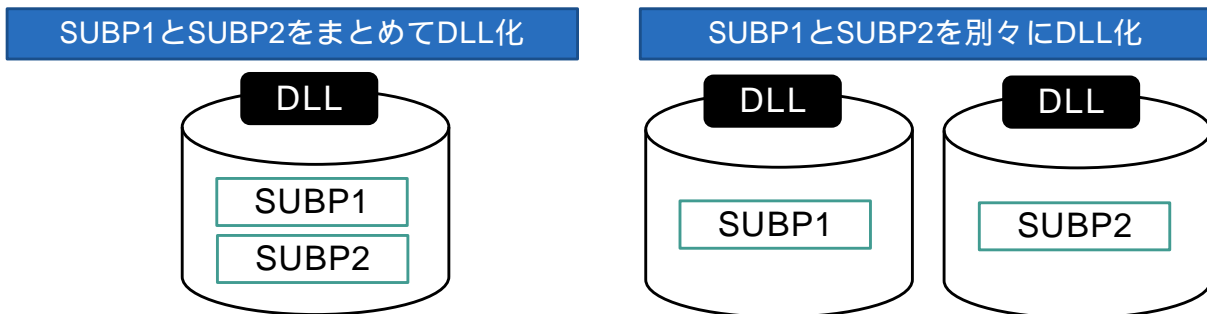
A-2 . DLLとは

- ◆ DLLとは、ダイナミックリンクライブラリの略称であり、実行時に動的にリンクするライブラリのことを指す
 - その用途から共有ライブラリと呼ぶこともある。ファイルの拡張子は「.dll」
- ◆ 複数のプログラムから呼ばれるサブプログラムはDLL化を推奨
 - DLL化のメリット
 - DLLに含まれるプログラムを修正しても、EXEを作りなおす必要はない。（保守性向上）
 - 複数のプログラムから呼ばれた場合、DLL内プログラムのコード部は共有される。（メモリ量削減）



◆ DLLは、1個以上のサブプログラムで構成することが可能

- プログラムSUBP1、SUBP2をまとめて1つのDLLにすることも、SUBP1とSUBP2をそれぞれのDLLとすることも可能

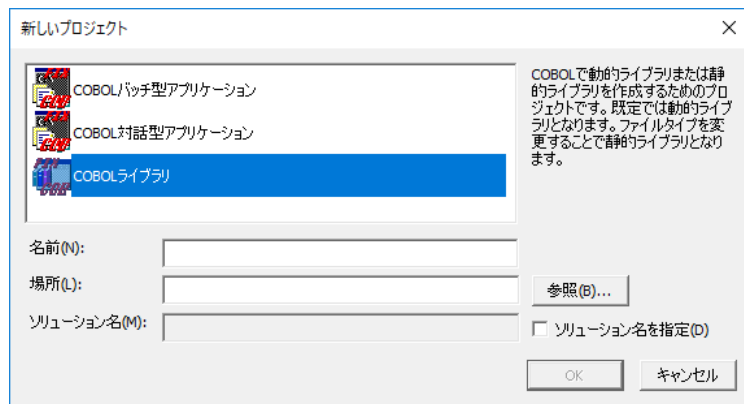


- DLLを構成する場合は、「ファイル操作を行うサブプログラム群」、「コード変換を行うサブプログラム群」のように、同一機能単位のサブプログラム群で構成する

A-3. DLL化の方法

◆ DLLを作るときは新規プロジェクトで「COBOLライブラリ」を選択する

- COBOL開発環境の[新規プロジェクト]を選び、アプリケーションとして「COBOLライブラリ」を選択



- DLL化するプログラムをプロジェクトファイルに追加
- ビルドを実施

A-4. コンパイル時（EXE作成時）のDLLの指定方法

◆ コンパイル時に、EXEが使用するDLLを指定する方法

- オプション設定画面の「リンカ」にある「リンクライブラリ検索ディレクトリ」にDLLが存在するパス、「リンクライブラリ」にDLLのインポートライブラリ名を指定



お問い合わせ

お問い合わせ先



SystemDirectorでは業務システム開発環境に関する様々なご質問やご相談にお応えします
たとえば・・・

再構築を検討中のお客様に何を提案したらよいか
業務アプリケーションの開発や保守を効率化するにはどうしたらよいか
既存資産をどう活用したらよいか

などお気軽にご相談ください

● ご購入前のお問い合わせ

NEC SystemDirector ご相談窓口

● Web <http://jpn.nec.com/SystemDirector/contact.html>

● 本資料の内容についてのお問い合わせ

NEC COBOLご相談窓口

● Web <http://jpn.nec.com/cobol/contact.html>

本資料について

◆ 記載内容に関する注意事項

- 資料作成時点の情報であり、製品出荷時に変更される可能性があります。予めご了承ください。

商標について

- System Director、Open COBOL Factory 21は日本電気株式会社の登録商標です。
- Microsoft®、Windows®、Windows Server®、Visual C++®、Visual Studio®、Excel®、SQL Server®、Access®は、米国あるいはその他の国における米国Microsoft Corporationの商標または登録商標です。
- Windows 10の正式名称は、Microsoft® Windows® 10 Operating Systemです。
- Windows Server 2016の正式名称は、Microsoft® Windows Server® 2016 Operating Systemです。
- Windows Server 2019の正式名称は、Microsoft® Windows Server® 2019 Operating Systemです。
- OracleとJavaは、Oracle Corporation及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
- Subversionは、Apache Software Foundation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、記載されている会社名、製品名は、各社の登録商標または商標です。

\Orchestrating a brighter world

NECは、安全・安心・公平・効率という社会価値を創造し、
誰もが人間性を十分に発揮できる持続可能な社会の実現を目指します。

\Orchestrating a brighter world

NEC