

SAP Variant Configuration 解説

技術設定・依存関係・AVC・統合設計の詳細ガイド

2026年6月

はじめに： VC が必要な理由と全体アーキテクチャ

製造業において「顧客ごとに仕様が異なる製品」を効率的に管理することは、ERP 設計の最難関テーマの一つです。SAP Variant Configuration (VC) はこの課題に対する解答として、「一つのコンフィギュラブル品目 (KMAT) と製品モデル (特性・クラス・依存関係) から、受注時の顧客選択に応じて BOM・ルーティング・価格を動的に生成する」仕組みを提供します。

VC を正しく設計・実装するには「特性 (CT04) ・クラス (CL02) ・コンフィギュラブル品目 (KMAT) ・スーパー BOM ・スーパールーティング・依存関係 (CU01~CU04) ・コンフィギュレーションプロファイル (CU41) 」という七つのオブジェクトの関係と、各オブジェクトの詳細な技術設定を理解することが必要です。本稿ではそれらを設定手順・構文・制約事項まで含めて徹底的に解説します。

VC の全体オブジェクト関係図

- KMAT 品目 (MM01) : VC の製品テンプレート。コンフィギュレーションクラス (タイプ 300) を持つ。
- 特性 (CT04) : 仕様軸 (COLOR・ENGINE_TYPE・VOLTAGE 等) を定義するマスタ。
- クラス (CL02, タイプ 300) : 特性を束ねるコンテナ。KMAT に割り当て。
- コンフィギュレーションプロファイル (CU41) : KMAT 上で VC の動作方式を制御する設定レコード。
- スーパー BOM (CS01, 用途 1 または 5) : 全バリエーションの部品を網羅した BOM。各明細に選択条件を付与。
- スーパールーティング (CA01) : 全バリエーションの工程を網羅したルーティング。各工程に選択条件を付与。
- 依存関係 (CU01~CU04) : 特性値間のビジネスルールを記述するロジック。
- バリエーション条件 (Variant Conditions) : VC と価格設定 (SD Condition Technique) を接続する価格ルール。

1. 特性 (Characteristic) の詳細設定 — CT04

特性の基本構造

特性 (Characteristic) は VC における「選択肢の軸」です。CT04 で作成・管理します。特性は DMS・Classification 等の他 SAP モジュールとも共用されますが、VC 専用の設定項目が存在します。

データ型と技術パラメータ

文字型 (CHAR)

文字型は「外装色・素材・製品グレード」等のカテゴリ値に使用します。

- フィールド長 (Field Length) : 最大 30 文字。VC のコンフィギュレーション画面での表示幅に影響します。
- 大文字/小文字区別: 「大文字のみ (Uppercase Only) 」フラグを有効にすると、入力値が自動で大文字に変換されます。依存関係の比較式では大文字小文字が厳密に区別されるため、このフラグの設定は依存関係の動作に影響しません。
- 固定値リスト (Allowed Values) : CT04 の「Values」タブで選択可能な値を定義します。値コード (Value) と説明 (Description) の両方を各言語で設定します。VC では基本的に全値を固定値として定義し「追加値の許可 (Additional Values) 」は無効にすることが推奨されます。

数値型 (NUM)

数値型は「電圧・出力・重量・サイズ」等の量的な値に使用します。

- 桁数・小数桁 (Number of Chars / Decimal Places) : 整数部と小数部の桁数を定義します。例: 電圧特性は「整数 5 桁・小数 0 桁」。
- 単位 (Unit of Measurement) : 数値型特性には単位 (kW・V・kg・mm 等) を設定できます。単位付き特性値は依存関係の数値比較でも単位変換を考慮したロジックが必要です。
- 許容範囲 (Interval Values) : 数値型では固定値の代わりに「最小値・最大値・増分 (Increment) 」で許容範囲を定義できます。例: 電圧 100～

240V・増分 10V。増分設定により 100V・110V・120V...240V が選択可能値になります。

日付型 (DATE) ・時刻型 (TIME)

VC 用途での使用は限定的ですが、「有効開始日・生産開始予定日」等の時間的な仕様軸が必要な場合に使用します。依存関係での日付比較には専用の関数 (DATE_TO_DAYS 等) を使用します。

VC 固有の特性設定タブ

「Additional Values」の制御

- 「Additional Values (追加値の許可)」: ON にすると固定値リスト以外の任意入力が可能になります。VC 設計では通常 OFF にして選択肢を固定値に制限します。フリー入力が必要な特性 (顧客固有の刻印文字列等) のみ ON にします。

「Chars」タブ (特性の追加属性)

- 「Entry Required (必須入力)」: このフラグを ON にすると、コンフィギュレーション完了の判定時にこの特性への値入力が必要とみなされます。VC の「Complete Configuration」チェックに連動します。
- 「No Display (非表示)」: コンフィギュレーション画面で表示しない特性 (依存関係のアクションで自動設定される内部特性) にはこのフラグを ON にします。
- 「Not Ready for Input (入力禁止)」: アクションによって自動設定される特性に ON にします。ユーザーが手動変更できないことを強制します。
- 「Reference Characteristic (参照特性)」: 品目マスタ・受注・製造指図の特定フィールドを特性として参照する「参照特性」設定です。例: 品目マスタのウェイト (MARA-NTGEW) を参照する参照特性を定義し、依存関係でウェイトに基づく選択条件を設定できます。

特性の多言語設定

特性の「説明 (Description)」と「固定値のテキスト」は多言語で設定できます。CT04 の言語切替ボタンで日本語・英語等の説明を個別に入力します。コンフィギュレーション画面の表示言語はログインユーザーの言語設定に依存するため、使用

するすべての言語でテキストを設定することが必須です。設定漏れがあると特定言語のユーザーに文字化けまたは空白が表示されます。

2. クラス (Class) の詳細設定 — CL02

クラスタイプ 300 (Configuration Class) の特性

VCで使用するクラスはクラスタイプ「300 (Configuration Class)」です。クラスタイプ 300 は他のクラスタイプ (001: 汎用・002: 材質等) と異なり、VC 専用の属性を持ちます。

- クラスの作成: CL02 を起動→クラス名入力→クラスタイプ「300」を選択→基本データタブで説明入力→「Chars」タブで特性を追加。
- クラスへの特性割り当て順序: 「Chars」タブで特性を追加する際、順番がコンフィギュレーション画面での表示順に直接影響します。ユーザーが選択しやすい順序 (重要度の高い特性を上部に配置) で設計します。
- 特性の必須設定 (Required): クラス内で特定の特性を「必須 (Required)」に設定できます。この設定は CT04 の「Entry Required」フラグと連動します。

クラス階層とクラスノード

VC の高度な機能として「クラスノード (Class Node)」があります。スーパー BOM の明細に品目ではなく「クラス」を配置し、コンフィギュレーション時にそのクラスに属する品目の中から動的に品目を選択する仕組みです。

- クラスノードの利用場面: 例として「ギヤボックス」という明細にギヤボックス品目を直接配置するのではなく、クラス「GEARBOX_CLASS」をクラスノードとして配置します。GEARBOX_CLASS に属する品目 (5 速 MT・6 速 MT・8 速 AT 等) の中から、特性値に応じた品目が自動的に選択されます。
- クラスノードの設定: CS01 の BOM 明細で品目タイプを「クラスノード (Class Node: D)」に選択し、クラス番号 (CL02 で作成したクラス) を入力します。クラスノードにも通常の明細と同様に選択条件が付与できます。
- クラスノードの制約: クラスノードから選択される品目は、そのクラスに「配賦 (Assignment)」されている品目でなければなりません。CL20N で品目をクラスに配賦する作業が必要です。品目とクラスの管理が増えるため、クラスノードは本当に必要な場合に限定することを推奨します。

3. コンフィギュラブル品目 (KMAT) の設定 — MM01/MM02

品目タイプ KMAT の特性

コンフィギュラブル品目は品目タイプ「KMAT」で作成します。通常の完成品 (FERT) と異なる重要な設定があります。

主要ビューと設定項目

基本データ 2 ビュー

- 「Configurable Material (コンフィギュラブル品目)」フラグ: このチェックボックスを ON にすることで品目が VC 対象であることを宣言します。このフラグがないとコンフィギュレーションプロファイル (CU41) を作成できません。

MRP 2 ビュー

- 「Configuration Management (コンフィギュレーション管理)」: 「Variant Configuration (バリエーション構成)」を選択します。これにより製造指図・受注でのコンフィギュレーション画面が有効になります。

販売: 一般/プラントデータビュー

- 「Variable-Size Item (可変サイズ品目)」: 寸法指定品 (切断鋼材・電線等) でサイズを VC 特性で指定する場合に使用するフラグです。通常の VC 用途では不要です。

分類ビュー (Classification)

- コンフィギュレーションクラスの割り当て: 分類ビューでクラスタイプ 300 のクラスを品目に割り当てます。この割り当てによって、コンフィギュレーション画面にそのクラスの特徴が表示されます。複数のクラスタイプ 300 クラスを一つの KMAT に割り当てることも可能ですが、プロファイルでの制御が複雑になるため 1 クラスに集約する設計が推奨されます。

コンフィギュレーションプロファイル (CU41) の詳細

コンフィギュレーションプロファイルは KMAT の VC 動作を制御する設定レコードです。CU41 (作成) /CU42 (変更) /CU43 (表示) で管理します。一つの KMAT に対して複数のプロファイルを作成し、用途 (見積・受注・製造指図) ごとにプロファイルを切り替えることも可能です。

プロファイルの主要設定項目

- 「BOM Application (BOM 用途)」: コンフィギュレーション時に使用するスーパー BOM の用途を指定します (例: 用途 1=設計 BOM・用途 5=販売 BOM)。受注コンフィギュレーションと製造コンフィギュレーションで異なる用途の BOM を使用する場合、それぞれに対応するプロファイルを作成します。
- 「Configuration Type (コンフィギュレーションタイプ)」: 「User Interface Configuration (UI コンフィギュレーション)」か「Background Configuration (バックグラウンドコンフィギュレーション)」を選択します。UI コンフィギュレーションは担当者がコンフィギュレーション画面で手動入力するモード。バックグラウンドは依存関係のアクションと参照特性のみで自動的に値が設定されるモード (自動コンフィギュレーション)。
- 「Class (クラス)」: プロファイルで使用するコンフィギュレーションクラスを明示的に指定します。分類ビューで割り当てたクラスと一致している必要があります。
- 「SCE Usage (SCE の使用)」: 制約ネット (Constraint Net) を使用する場合に「Solution and Constraint Engine」を有効化するフラグです。SCE を使用しない場合は OFF にします (性能上のオーバーヘッドを避けるため)。
- 「Variant Matching (バリエーションマッチング)」: コンフィギュレーション値が完全一致する既存の「バリエーション (Variant: KMAT 配下の個別品目)」が存在する場合、そのバリエーションを自動的に選択するかどうかの設定。「Find Variants (バリエーション検索)」ON 時はマッチするバリエーションが自動特定されます。

4. スーパー BOM: 選択条件と数量式の技術詳細

スーパー BOM の設計思想

スーパー BOM (Super BOM) は「あらゆるバリエントで使用される可能性のある全部品を一つの BOM に列挙したもの」です。各明細に「選択条件 (Selection Condition)」を付与し、特定の特性値組み合わせの場合のみその明細が有効となります。

スーパー BOM の設計原則として「常にすべてのバリエントに使用される部品」と「特定バリエントにのみ使用される部品」を混在させます。前者は選択条件なし (無条件有効)、後者に選択条件を付与します。

BOM 明細の選択条件設定

CS01/CS02 で BOM 明細を選択し、「Extra → Conditions (条件)」メニューまたは明細タブ「Conditions」から選択条件を入力します。選択条件は依存関係構文で記述します。

選択条件の基本構文

```
$self.COLOR = 'RED'
```

上記は「COLOR 特性の値が RED の場合、この明細を有効にする」という選択条件です。\$self はコンフィギュレーション対象の品目自身を指す参照語です。

複合条件 (AND / OR)

```
$self.COLOR = 'RED' AND $self.GRADE = 'PREMIUM'  
$self.COLOR = 'RED' OR $self.COLOR = 'WHITE'
```

AND と OR の組み合わせには括弧を使用して優先順位を明示します。

```
($self.COLOR = 'RED' OR $self.COLOR = 'WHITE') AND $self.GRADE = 'PREMIUM'
```

数値比較演算子

```
$self.ENGINE_OUTPUT > 200  
$self.ENGINE_OUTPUT >= 150 AND $self.ENGINE_OUTPUT <= 250
```

数値型特性との比較には「>・<・>=・<=・=・<> (不等号)」が使用できません。

IN 演算子（複数値の一括指定）

```
$self.COLOR IN ('RED', 'WHITE', 'SILVER')
```

IN を使うと複数値の OR 条件を簡潔に記述できます。

数量式（Quantity Formulas）

BOM 明細の数量を固定値でなく、特性値に基づいた計算式で指定できます。数量欄の右側にある「Quantity Formula」フィールドに式を入力します。

数量式の基本構文

```
$self.NUM_FLOORS * 4
```

例：ビル構造物で「フロア数（NUM_FLOORS）×4 本」のボルトが必要な場合の数量式。

```
ROUND($self.WEIGHT / 10, 0)
```

例：重量（WEIGHT）を 10 で割った値を切り上げた数量。

条件付き数量（IF 式）

```
IF $self.OPTION = 'HEAVY' THEN 8 ELSE 4
```

OPTION が HEAVY なら数量 8、それ以外は 4 を返す条件式。

BOM 明細の有効期間と VC の組み合わせ

スーパー BOM の明細には「有効期間（Valid From / Valid To）」を設定できます。選択条件と有効期間を組み合わせることで「2026 年 10 月 1 日以降の受注で、かつ COLOR = RED の場合のみ有効」という精密な制御が可能です。ECM（エンジニアリング変更管理）と連動して、変更番号による有効化も設定できます。

5. スーパールーティング：工程選択条件の設定

スーパールーティングの構造

スーパールーティング（Super Routing）は CA01 で KMAT 品目に対して作成します。通常のルーティングと同様に工程（Operation）を列挙しますが、各工程に選択条件を付与する点が異なります。

工程への選択条件付与

CA01/CA02 でルーティングを開き、工程を選択し「Extras → Conditions」から選択条件を入力します。構文は BOM 明細の選択条件と同一です。

- 例：工程「0020：表面処理（PAINT）」の選択条件 → `$self.SURFACE = 'COATED'`。塗装処理が選択された場合のみこの工程が製造指図に取り込まれます。
- 例：工程「0030：ターボ組付け（TURBO_ASSY）」の選択条件 → `$self.TURBO = 'YES'`。ターボオプションが選択された場合のみ実行されます。

参照工程との組み合わせ

スーパールーティング内で「参照作業手順（Reference Operation Set）」を使用することで、複数の製品で共通する工程（品質検査・梱包・洗浄等）を一元管理できます。スーパールーティングに参照工程をリンクし、参照工程自体に選択条件を付与します。参照工程の変更がすべての参照元スーパールーティングに即時反映されます。

6. 依存関係：完全技術リファレンス

依存関係オブジェクトの種類とトランザクション

依存関係 (Dependencies) は CU01~CU04 のトランザクションで管理します。

- CU01: 前提条件 (Precondition) の作成/変更
- CU02: 選択条件 (Selection Condition) の作成/変更 (ただし通常は BOM/Routing 明細から直接入力する方が多い)
- CU03: アクション/プロシージャ (Action / Procedure) の作成/変更
- CU04: 制約ネット (Constraint Net) の作成/変更

依存関係は KU01 等 (依存関係一覧) や CU05 (依存関係の特性への割り当て) で管理します。依存関係は直接オブジェクト (BOM 明細・特性の固定値) に埋め込むか、独立したオブジェクト (CU01~CU04) として作成して参照する二つの方法があります。再利用性の高い依存関係は独立オブジェクトとして定義します。

依存関係の構文：完全リファレンス

変数と参照語

- \$self: コンフィギュレーション対象品目自身への参照。\$self.COLOR のように特性を参照します。
- \$parent: 親アセンブリへの参照。サブアセンブリ品目の VC で親品目の特性値を参照する際に使用。
- \$root: 製品階層の最上位品目への参照。深い階層構造の VC で最上位の特性を参照する場合。
- TABLES: 特性の「テーブル」形式 (複数の特性値の組み合わせを表形式で定義する表条件) を参照する際の宣言。

基本演算子

```
=      不等号: <>
>      >=   <   <=
AND    OR    NOT
IN ('A', 'B', 'C')
```

数値関数

- ABS(x): 絶対値。
- ROUND(x, n): 小数点 n 桁で四捨五入。
- TRUNC(x, n): 小数点 n 桁で切り捨て。
- SQRT(x): 平方根。
- LOG(x): 自然対数。

文字列関数

- SUBSTR(str, start, length): 部分文字列抽出。
- LENGTH(str): 文字列長を返す。
- CONTAINS(str, 'pattern'): パターン一致チェック（ワイルドカード「*」使用可）。

特殊変数

- \$INOB.MENGE: BOM 明細数量への参照（数量式内で使用）。
- \$INOB.MEINH: BOM 明細単位への参照。

前提条件（Precondition）の詳細

前提条件は「特性の固定値が選択画面に表示される条件」を定義します。前提条件を満たさない場合、その固定値はコンフィギュレーション画面のドロップダウンに表示されません。

前提条件の割り当て方法

CT04 で特性を開き、「Values」タブで固定値を選択→「Extras → Object Dependencies → Preconditions」で依存関係を入力します。

設定例：エンジン出力に応じたターボ選択肢制御

```
/* ターボ特性の値「YES」への前提条件 */  
$self.ENGINE_OUTPUT > 150
```

ENGINE_OUTPUT が 150kW を超えている場合のみ「ターボ=YES」が選択画面に表示されます。150kW 以下のエンジンが選択された状態では「YES」は非表示となり、「NO（または空白）」のみ選択可能です。

アクション (Action / Procedure) の詳細

アクションは「ある特性値が選択されたとき、他の特性値を自動設定する」処理です。CU03で独立オブジェクトとして定義するか、CT04の特性固定値に直接埋め込みます。

アクション構文 (IF-THEN-ELSE)

```
IF $self.COLOR = 'WHITE'  
THEN  
    $self.INTERIOR = 'GREY'  
END_IF
```

```
IF $self.GRADE = 'PREMIUM'  
THEN  
    $self.WARRANTY = '5Y'  
    $self.SUPPORT = 'GOLD'  
END_IF
```

一つのアクションで複数の特性値を設定できます。THEN句に複数の代入文を記述します。

ELSEIF 構文

```
IF $self.ENGINE_OUTPUT > 300  
THEN  
    $self.COOLING = 'WATER_RACE'  
ELSEIF $self.ENGINE_OUTPUT > 150  
THEN  
    $self.COOLING = 'WATER_STD'  
ELSE  
    $self.COOLING = 'AIR'  
END_IF
```

アクションの割り当て方法

アクションはCT04の「Values」タブで固定値を選択→「Extras → Object Dependencies → Actions」で紐付けます。「この値が選択されたときに実行するアクション」という意味付けです。あるいは特性自体にアクションを紐付けることで「この特性に値が入力されたとき（どの値でも）実行する」処理を定義できます。

制約ネット (Constraint Net) の詳細設計

制約ネットのアーキテクチャ

制約ネット (Constraint Net) は SCE (Solution and Constraint Engine) が処理する宣言的ルール群です。通常の依存関係 (IF-THEN 型) と異なり「双方向推論 (Bidirectional Inference)」をサポートします。一組の特性間関係を一度だけ記述すれば、どちらの特性が確定しても他方が自動的に推論されます。

制約の構文

制約ネット内の個々の制約 (Constraint) は以下の構文で記述します。

```
CONSTRAINT constraint_name
CONDITION
  <前提条件式 (この制約が適用される条件) >
RESTRICTIONS
  <制限式 (等式・不等式・集合条件) >
```

制約ネットの例: エンジン出力と燃料タイプの整合性

```
CONSTRAINT ENGINE_FUEL_COMPAT
CONDITION
  $self.ENGINE_OUTPUT IS NOT INITIAL
  AND $self.FUEL_TYPE IS NOT INITIAL
RESTRICTIONS
  ( $self.FUEL_TYPE = 'DIESEL'
    AND $self.ENGINE_OUTPUT >= 100 )
  OR
  ( $self.FUEL_TYPE = 'GASOLINE'
    AND $self.ENGINE_OUTPUT IN [80, 100, 120, 150, 200] )
```

この制約ネットは「ディーゼルなら出力 100kW 以上・ガソリンなら 80/100/120/150/200kW のみ」という制約を双方向で強制します。FUEL_TYPE が先に選択されたときも、ENGINE_OUTPUT が先に選択されたときも、SCE が両方の整合性を検証します。

制約ネットの表条件 (Table Condition)

複雑な組み合わせルールをコードで書かずに「テーブル (Characteristics Table)」として定義することで可読性を大幅に高める機能です。

```
CONSTRAINT VALID_COMBINATIONS
RESTRICTIONS
  TABLE (
    $self.VOLTAGE, $self.FREQ, $self.REGION
    100,           50,           'JAPAN'
    100,           60,           'US'
```

220,	50,	'EUROPE'
240,	50,	'UK'

)

この表条件は「電圧・周波数・地域」の有効な組み合わせを一覧で定義します。SCEはこの表に存在しない組み合わせを自動的に無効とします。

制約ネットの有効化 (CU41でSCE有効化)

制約ネットを機能させるには、コンフィギュレーションプロファイル (CU41) で「SCE Usage」を ON にする必要があります。SCE 有効時は全制約ネットがコンフィギュレーションセッション中に常に評価されます。制約ネットが多い・複雑な場合は応答性能に影響するため、制約ネットの数と複雑さを最小限に抑える設計が重要です。

7. コンフィギュレーションのシミュレーションと検証

CU50: コンフィギュレーションシミュレーター

CU50 は VC モデルの設定テストツールです。実際の受注・製造指図を作成せずにコンフィギュレーションの動作を検証できます。

CU50 の操作手順

- 起動: CU50 を実行→品目番号 (KMAT) とコンフィギュレーションプロファイルを入力→「Execute (実行)」。
- 特性値の入力: コンフィギュレーション画面が表示されます。各特性に値を入力/選択します。依存関係 (前提条件・アクション) の動作をリアルタイムで確認します。
- BOM 爆発確認: 「Extras → Edit BOM (BOM 編集)」でコンフィギュレーション結果の BOM 明細 (有効な明細のみ) を確認します。スーパー BOM から選択条件に合致した明細のみが表示されます。
- ルーティング確認: 「Extras → Edit Routing (ルーティング編集)」でコンフィギュレーション結果のルーティング (有効な工程のみ) を確認します。
- 価格確認: 「Pricing (価格設定)」でコンフィギュレーション価格 (バリエーション条件による価格差額) の計算結果を確認します。

完了チェック (Completeness Check)

コンフィギュレーションが完了しているか (必須特性すべてに値が入力されているか) を判定するのが「完了チェック (Completeness Check)」です。

- CT04 の「Entry Required」フラグを ON にした特性はすべて入力されている必要があります。
- 受注 (VA01) や MTO 製造指図でコンフィギュレーションが「不完全 (Incomplete)」の場合、後続プロセス (スケジュール確認・出荷) がブロックされる場合があります。完了チェックロジックの設計は業務フローとの整合性確認が必要です。

依存関係のデバッグ

依存関係が期待通りに動作しない場合のデバッグ方法を示します。

- CU50の「Log（ログ）」表示：コンフィギュレーション実行中に評価された依存関係・前提条件・アクションの実行ログを確認できます。どの依存関係が評価されてどの値が自動設定されたかを追跡できます。
- CU99（依存関係の一貫性チェック）：VCモデル全体の依存関係の構文エラー・参照エラー（存在しない特性への参照等）を一括チェックするトランザクション。定期的に行ってモデルの整合性を維持します。
- CU70（依存関係一覧表示）：特定の特性・クラス・品目に割り当てられた全依存関係を一覧表示します。依存関係の割り当て漏れや重複の確認に使用します。

8. VC と SD（受注処理）の統合

受注コンフィギュレーション（VA01）のフロー

受注（VA01）で KMAT 品目を入力すると、自動的にコンフィギュレーション画面が開きます。担当者が特性値を選択し、コンフィギュレーションが完了すると以下の処理が自動実行されます。

- 設定済み BOM の生成：スーパー BOM から選択条件に合致した明細のみを含む「設定済み BOM」が受注明細に格納されます。
- コンフィギュレーション価格の計算：バリエーション条件（SD Condition Technique）に基づくオプション価格差額が受注価格に自動加算されます。
- コンフィギュレーション特性値の保存：受注明細のコンフィギュレーション（CUOV）テーブルに選択された全特性値が保存されます。

バリエーション価格（Variant Pricing）の設定

VC の価格設定は SD 条件テクニック（Condition Technique）に基づきます。設定手順を示します。

ステップ 1: バリエーション条件タイプの定義（V/VC）

IMG → 販売管理 → 基本機能 → 価格設定 → 価格設定管理 → 条件タイプの保守。標準の条件タイプ「VA00（バリエーション価格差額）」を使用するか、カスタム条件タイプを作成します。

ステップ 2: 条件テーブルの設定

バリエーション条件の条件テーブルには「コンフィギュラブル品目番号＋特性＋特性値」の組み合わせでキーが構成されます。IMG → 販売管理 → バリエーション構成 → 価格設定 → バリエーション価格設定の条件テーブルを作成します。VC の特性をキー項目として条件テーブルに追加するには「特性参照フィールド」の設定が必要です。

ステップ 3: 条件レコードの作成（VK11/VK12）

VK11 でバリエーション条件レコードを作成します。「KMAT の品目番号・特性名・特性値・価格差額（Surcharge または Discount）」を入力します。

- 例: KMAT=ENGINE_V8・特性 COLOR=RED → 価格差額 +50,000 JPY
- 例: KMAT=ENGINE_V8・特性 TURBO=YES → 価格差額 +350,000 JPY

コンフィギュレーションと ATP (納期回答)

KMATの受注コンフィギュレーション後、MTO (受注生産) または MTS (見込み生産) に応じて ATP (Available to Promise) チェックが実行されます。

- MTO (Make-to-Order) の場合: ATP 量ではなく製造リードタイム (ルーティングの工程時間積み上げ) に基づく予定確認日が計算されます。設定済みルーティングの工程時間がリードタイム計算に直接影響します。
- MTS (計画生産) の場合: 設定済み BOM のコンポーネント在庫・計画入荷を MRP で計画します。KMAT レベルでの在庫管理は行わず (KMAT は在庫を持たない)、展開後の部品レベルで在庫管理します。

9. VC と PP（製造）の統合

製造指図でのコンフィギュレーション

MTO 生産では受注明細からコンフィギュレーションが製造指図に引き継がれます。受注のコンフィギュレーション値（特性値）が製造指図のコンフィギュレーション（CUOV）にコピーされ、製造指図の BOM 爆発・ルーティング取込でスーパー BOM・スーパールーティングの選択条件評価に使用されます。

BOM 爆発（CO01/CO41）での VC 動作

製造指図（CO01）作成時または計画指図（MD16/CO41）の指図変換時に、スーパー BOM が展開（爆発）されます。このとき「コンフィギュレーション値と選択条件の照合」が実行され、有効な明細のみが製造指図コンポーネント（AFKO/AFPO/RESB）に取り込まれます。

- 代替品目の解決：スーパー BOM に代替品目グループが設定されている場合、優先度に従って代替品目が解決されます。在庫量・MRP 調達状況を考慮した代替品目の自動選択は S/4HANA 2020 以降の機能拡張で改善されています。

ルーティング取込での VC 動作

製造指図作成時にスーパールーティングが取り込まれます。各工程の選択条件がコンフィギュレーション値と照合され、有効な工程のみが製造指図工程（AFVC）に取り込まれます。選択条件なしの工程は常に取り込まれます。

製造指図での再コンフィギュレーション

製造指図が作成済みの状態で仕様変更が発生した場合の「再コンフィギュレーション」に注意が必要です。CO02（製造指図変更）でコンフィギュレーション値を変更すると BOM 爆発・ルーティング取込が再実行されます。ただし「すでに払出済みのコンポーネント」や「実績確認済みの工程」の取り扱いは個別の運用判断が必要です。原則として製造着手後のコンフィギュレーション変更は極力避け、変更が必要な場合は製造指図をキャンセルして新規作成することを推奨します。

10. AVC (Advanced Variant Configuration) : 技術詳細

AVC と従来 VC のアーキテクチャ比較

AVC (Advanced Variant Configuration) は S/4HANA 1709 (2017 年) 以降に提供された次世代 VC エンジンです。データモデルは従来 VC と共通ですが、処理エンジンと UI が刷新されています。

- エンジンの違い: 従来 VC は ABAP 上の VC 処理エンジン (LCU*/LVC*ファンクションモジュール) を使用します。AVC は HANA のインメモリ処理を最大限活用する CDS/HANA ネイティブのエンジンで動作します。大規模な制約ネットや多数の特性を持つ複雑な製品モデルでの性能差が顕著です。
- UI の違い: 従来 VC は SAP GUI ベースのコンフィギュレーション画面。AVC は Fiori ベースの「Product Configuration UI (アプリ ID: F3483)」。
- PMEVC: AVC の製品モデル管理は PMEVC (Product Modeling Environment for Variant Configuration) アプリ (Fiori) で行います。特性・クラス・依存関係・制約ネットの管理を GUI エディタで実施できます。

AVC への移行判断

既存の従来 VC システムを AVC に移行するかどうかの判断ポイントを示します。

- 移行推奨ケース: 制約ネットが大規模 (100 件超) ・複雑なコンフィギュレーション計算で性能問題が発生している ・ Fiori UI でのコンフィギュレーション体験を顧客・担当者に提供したい。
- 移行不要または後回しケース: 従来 VC で性能問題がなく業務が安定稼働している ・ VC モデルが比較的シンプル (特性 20 件未満 ・ 依存関係 50 件未満) ・ S/4HANA バージョンが 1709 未満 (AVC サポート外) 。
- 移行時の留意点: データモデル自体 (特性・クラス・スーパー BOM 等) は AVC/従来 VC で共通のため移行工数は比較的少ない。ただし依存関係の構文が一部異なる (AVC では Procedure 構文の一部が非推奨・制約ネット構文の変更) ため、既存依存関係の AVC 互換性確認と修正が必要です。AVC の CU99 相当ツールで互換性チェックを実施します。

LKPM とグローバル特性変更管理

LKPM とは Lokal Klassifizierung/Produktion Modeling (直訳) の略で、コンフィギュラブル製品の特性値を量産全体で一括変更・分析する機能です。

- 利用場面：特定の特性値（廃止になるオプション等）を持つ全受注・製造指図の一括特定と変更対応。マスタデータの VC モデル変更後に既存オーダーへの影響を一括評価・処置する際に使用します。

11. VC の Customizing 設定 (SPRO 詳細)

SCE パラメータの設定

IMG パス: 「SAP NetWeaver → Application Server → Basis Services → Classification System → Settings」

- SCE タイムアウト: 制約ネットの SCE 計算が完了するまでの最大時間 (秒) を設定します。複雑なモデルでタイムアウトが発生する場合はこの値を増やしますが、応答性能とのバランスが必要です。デフォルト値は 30 秒。
- SCE 再帰深度: 制約ネットの推論再帰回数の上限。無限ループ (循環参照制約) を防ぐガード値です。デフォルト値は 100。

バリエーション構成のコア設定

IMG パス: 「Logistics - General → Variant Configuration」

コンフィギュレーション有効フラグの設定

- 「Basic Settings (基本設定)」: VC の基本パラメータ (価格設定方式・完了チェック方法・コンフィギュレーションのコピー動作) を設定します。
- 「Define Configuration Key (コンフィギュレーションキー定義)」: コンフィギュレーションの一意性を判定する「コンフィギュレーションキー (特性の組み合わせから生成されるハッシュ)」の生成方式を設定します。コンフィギュレーションキーが一致する受注・製造指図は同一コンフィギュレーションとみなされ、同一製造指図に集約できます。

バリエーション価格設定の統合

- IMG パス: 「Sales and Distribution → Basic Functions → Pricing → Variant Configuration Pricing」
- 特性を条件テーブルのフィールドとして追加する「Characteristics for Conditions」設定が最重要。この設定なしには VC の特性を価格条件キーとして使用できません。
- 手順: 対象の特性 (CT04) が「KLART (クラスタイプ) 300 に割り当て済み」であることを確認 → 「Assign Characteristic to Condition Table (条

件テーブルへの特性割り当て) 」→条件テーブルのフィールド名として特性名を登録→VK11 での条件レコード入力で選択可能になります。

クラス検索の設定

IMG パス： 「Logistics - General → Classification System → Search Settings」

- クラスタイプ 300 のオブジェクト数が多い場合の検索インデックス (Search Index) の設定。大規模環境ではインデックス再構築スケジュールの管理が必要です。

12. よくある設定ミスとトラブルシューティング

依存関係が動作しない

- チェック 1: 依存関係が特性・固定値に正しく割り当てられているか (CU70 で確認)。依存関係を「作成したが割り当てていない」ミスが最も多い。
- チェック 2: CU99 で依存関係の構文エラーを確認する。構文エラーがあると依存関係は無効として扱われます。
- チェック 3: 前提条件の場合、特性自体がコンフィギュレーションクラスに割り当てられているか (CL02 で確認)。
- チェック 4: コンフィギュレーションプロファイル (CU41) に正しいクラスが指定されているか。

BOM 明細が選択されない / 常に選択される

- チェック 1: BOM 明細の選択条件の構文が正しいか (CU99 で確認)。
- チェック 2: 選択条件で参照している特性名のスペル・大文字小文字が正確か (特性名は完全一致)。
- チェック 3: BOM 用途 (Usage) がコンフィギュレーションプロファイルの「BOM Application」と一致しているか。用途が不一致だとスーパー BOM が参照されません。
- チェック 4: BOM のプラントが製造指図/受注の生産プラントと一致しているか。

コンフィギュレーション価格が計算されない

- チェック 1: VK11 でバリエーション条件レコードが作成されているか。
- チェック 2: 条件タイプ (VA00 等) が受注タイプの価格手続きに組み込まれているか (OVKK/V/08)。
- チェック 3: 「特性参照フィールド (Characteristics for Conditions)」の IMG 設定で対象特性がマッピングされているか。

SCE タイムアウトエラー

- 症状: コンフィギュレーション時に「SCE computation timed out」エラー。
- 対処 1: SCE タイムアウト値を増やす (IMG 設定)。
- 対処 2: 制約ネットの数・複雑さを見直す。不要な制約・循環参照制約を削除。
- 対処 3: 制約ネット内の CONDITION 句を適切に設定し、不要な制約評価をスキップさせる。
- 対処 4: CU50 の「Log」モードで処理時間のかかっている制約を特定して最適化。

クラスノードで品目が選択されない

- チェック 1: クラスノードで指定したクラスに候補品目が配賦 (CL20N) されているか。
- チェック 2: 候補品目の特性値 (分類) が選択条件に合致しているか。クラスノードの品目選択は「候補品目の分類特性値が\$parent の特性値と一致するもの」を自動選択します。

13. 大規模 VC モデルの運用設計

VC モデルのドキュメント管理

VC モデル（特性・クラス・依存関係・スーパー BOM・スーパールーティング）は製品設計の変化とともに継続的に更新されます。モデルのドキュメント管理と変更管理は大規模 VC の安定運用に不可欠です。

- 依存関係カタログ：全依存関係（前提条件・アクション・制約ネット）の一覧・目的・関連特性を Excel または Confluence 等で管理します。SAP システム内の依存関係テキストにも適切なコメント（/* */で記述可能）を残します。
- テストシナリオ一覧：有効なコンフィギュレーションの代表的な組み合わせと期待 BOM・ルーティング・価格を記録した「VC テストシナリオ一覧」を作成します。VC モデル変更後のリグレッションテストに使用します。
- 変更プロセス：VC モデル変更は必ず「開発テナント→品質保証テナント→本番」のトランスポート経路を通過させます。依存関係・スーパー BOM 変更の本番直接入力も絶対禁止です。トランスポートには Customizing トランスポート（S/4HANA のトランスポートマネジメント）を使用します。

大量コンフィギュレーションデータのクリーンアップ

長期稼働のシステムでは受注・製造指図のコンフィギュレーションデータ（CUOV 等のテーブル）が蓄積します。

- アーカイブ：SAP Data Archiving（SARA）でコンフィギュレーション関連テーブルをアーカイブします。受注・製造指図のアーカイブと連動して VC データも削除されます。
- 孤立コンフィギュレーションのクリーンアップ：受注・製造指図が削除されても対応する VC データが残存するケースがあります。定期的なデータ品質確認（カスタム ABAP レポートまたは SAP 標準クリーンアッププログラム）が推奨されます。

以上