



ESPRIT 2016 の新機能

Copyright © 2016 DP Technology Corp. All rights reserved.

情報は事前通知なしに変更される場合があります。

このマニュアルの全ての内容は、DP Technology Corp.からの文書による事前の許可なく、複写、録音・録画、テープ収録、または任意の情報保存・検索システムを含むグラフィカル、電子的、機械的なあらゆる形態および方法により複製、転送、翻訳することは禁じられています。

この文書で説明されているソフトウェアは、提供されているライセンス契約および秘密保持契約の条件に基づく使用またはコピーのみが許可されます。ライセンスまたは秘密保持契約に指定されている以外の方法でソフトウェアを媒体にコピーすることは違法行為となります。

DP Technology Corporationの全てのソフトウェア製品には、ソフトウェアライセンスが正常に機能するために必要な統合セキュリティプログラムおよびプラグインモジュールが含まれています。これらのセキュリティプログラムやモジュールを無効化や削除しようとして、これらをインストールせずにソフトウェアを操作した場合、DP Technology Corporationの著作権および米国著作権法の違反行為とみなされます。必要なプログラムセキュリティモジュールや統合セキュリティプログラムなしでソフトウェアを操作することを目的とする、DP Technology Corporationから提供されている以外の全てのソフトウェアは、著作権違反となります。

ESPRITはDP Technology Corp.の登録商標です。

この文書で使われている全てのブランドおよび製品名、または専用のファイル形式は、それぞれの所有者の商標または登録商標です。商標および登録についての詳細は、各所有会社にお問い合わせください。

All brand or product names or proprietary file types mentioned in this document are trademarks or registered trademarks of their respective holders. Contact the appropriate companies for more information regarding trademarks and registration.

DP Technology Corp.

1150 Avenida Acaso

Camarillo, CA 93012

USA

電話: +1 805 388 6000

Fax: +1 805 388 3085

www.espritcam.com

このドキュメントはアメリカ合衆国にて印刷されました

目次

システムの強化機能	1
より高速なシミュレーション	1
データ交換	1
加工のためのCAD	1
モデリングの新機能新しい [サーフェスのスムーズ化] コマンドおよびサーフェスの属性	1
ソリッドモデルのフィレットの編集	2
フィーチャーの強化機能	4
フィーチャーの新機能自動チェーン機能でソリッドボディの属性を認識できます。	4
新しい「穴認識」機能	4
工具の強化機能	8
工具の新機能新しくなった工具テクノロジーページ	8
MachiningCloud Connectアドイン	8
新しい工具モデルの設定機能	11
バレル工具のサポートの強化	14
新しいProfitTurning加工法	16
旋盤工具の新機能旋盤荒加工におけるProfitTurning	16
旋盤溝切り加工におけるProfitTurning	19
ミル加工の改善点	21
ミル加工の新機能強化されたProfitMilling機能	21
ポケット加工と変動Z荒加工の新しいオプション [ステップオーバーの最小化]	22
5軸加工の機能強化	23
5軸加工の新機能スワフサイクルでの一定高増分加工条件	23
「ブレードサイクル」が「5軸スパイラル」に名称変更	24
5軸スパイラル仕上げ加工におけるパートサーフェスのスムーズ化	24
5軸スパイラル荒加工での高速送りクリアランス	25
3D工具補正	27
衝突のない外部リンク	28
5軸外部リンクの回転ソリューション	29
ワイヤー放電加工	30
放電加工の新機能コーナーおよび進入確認の新しいリリース機能	30
切り落としの追加クリアランス	30
荒加工の半径を無視	31
最初の切り落としの最適化	32
ポストプロセッサの新機能	33
複数ファイルへの出力転送	33
ヘリカル出力	35
動的な配列	36
実行時における値の変更	37
その他の新しいキーワード	38
更新されたキーワード	40
新しい数学関数	41
ポストプロセッサの定数	42

システムの強化機能

シミュレーションエンジンのアップグレードによりシミュレーションの速度が向上し、Rhino 3dmファイルとCatia V6 3dxmlファイルに対応できるようデータ交換機能が強化されました。

より高速なシミュレーション

シミュレーション機能がアップグレードされ、以前のバージョンに比べて最大14倍の処理速度が可能になりました。また、ストックオートメーションの速度と信頼性もアップグレードされています。

データ交換

ESPRIT 2016では以下がサポートされます。

- ・ Rhino 3dmファイルのインポート
- ・ SolidWorksファイルからのスケッチデータのインポート (最新のSolidWorks FXアドイン使用)
- ・ Catia V6 3dxmlファイルのバージョン2011x-2013のインポート
- ・ ACISファイル、バージョン21まで
- ・ Parasolidファイル、バージョン28まで

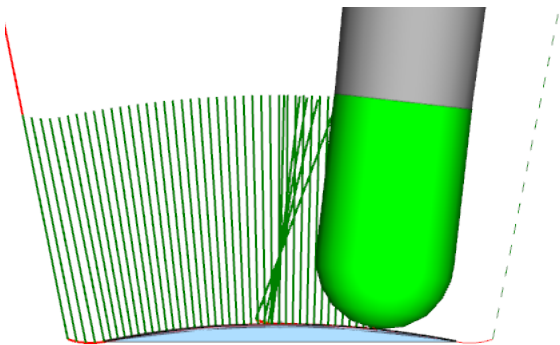
加工のためのCAD

正確なツールパスは、その基盤となるモデルの図形の精度と質に依存します。場合によっては元のCADシステムを使用できないこともあるため、ESPRITにはCADモデルに含まれるマイナーな欠点を補正する機能が用意されています。

モデリングの新機能新しい [サーフェスのスムーズ化] コマンドおよびサーフェスの属性

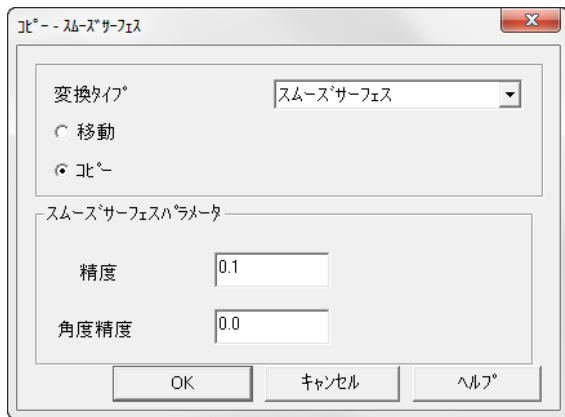
サーフェスには一見ただけでは認識しづらい微小な欠点が含まれていることがあります。図形の微小な欠点によって、そのサーフェスを使用する加工オペレーションに欠点が生じる可能性があります。その代表的な例として、コンポジットサイクルのドライブ曲線としてサーフェスを使用する場合は挙げられます。

図 1. ツールパス分析により、ドライブ曲線の微小な欠点に起因するツールパスの欠点がわかります。



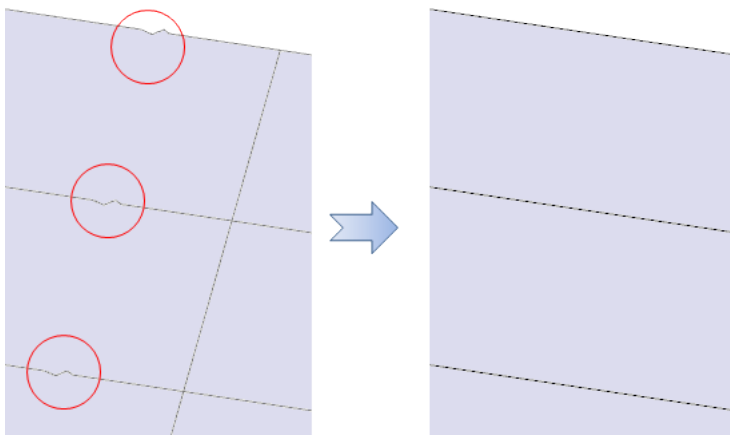
[コピー] コマンドに、[サーフェスのスムーズ化] という新しい変換が追加されました。

図 2. [コピー] ダイアログの [サーフェスのスムーズ化]



[サーフェスのスムーズ化] は、スムージングアルゴリズムを使用して欠点のあるサーフェスをスムーズにし、補正します。

図 3. サーフェスに見られる微小な欠点をスムーズにします。



▶ ESPRITには [最適化] という既存のコマンド (サーフェスを選択し、[コピー] を右クリックして [最適化] を選択) がありますが、U、V度数を設定しなくてもさらに簡単な方法でサーフェスの欠点を補正できるよう、改善されたスムージングアルゴリズムに基づいた [スムーズサーフェス] 機能が追加されました。

サーフェスを選択すると、属性マネージャの新しい属性にコントロールポイントの合計数と、U方向およびV方向の寸法が表示されます。

図 4. スムーズ化されたサーフェスの属性を属性マネージャで比較できます。

キー	値
グループカント	1
一般	
面積	クリックすると計算開始
体積	クリックすると計算開始
最小半径	クリックすると計算開始
内部モーメント	クリックすると計算開始
重心	クリックすると計算開始
制御点 #	24
U方向の次元	12
V方向の次元	2

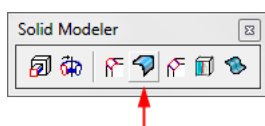
➔

キー	値
グループカント	1
一般	
面積	クリックすると計算開始
体積	クリックすると計算開始
最小半径	クリックすると計算開始
内部モーメント	クリックすると計算開始
重心	クリックすると計算開始
制御点 #	76
U方向の次元	19
V方向の次元	4

ソリッドモデルのフィレットの編集

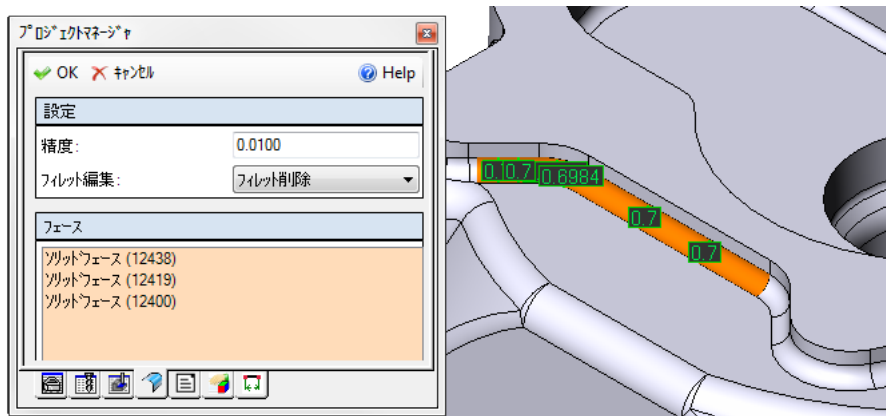
新たに追加された [フィレットの編集] コマンドを使用して、ソリッドモデルのフィレットを簡単に変更したり削除できます。

図 1. [フィレットの編集] コマンドは、[ソリッドモデラー] ツールバーで使用できます。



この新しいコマンドを使用して、フィレットのフェースの削除、半径の変更、フィレット半径の中心に沿ったスパイン曲線の作成などを行い、CADモデルに若干の変更を加えます。

図 2. 選択したフェースには、簡単に確認できるように現在のフィレット半径の値が表示されます。



▶ フェースプロパゲーションを使用して、連続するフィレットフェースを手早くグループ化できます。グループ化されたフェースは、[フィレットの編集]を開くと[フェース]一覧に全て自動的に追加されます。

図 3. モデルからフィレットを削除して、加工時に工具径によりフィレットが形成されるようにします。

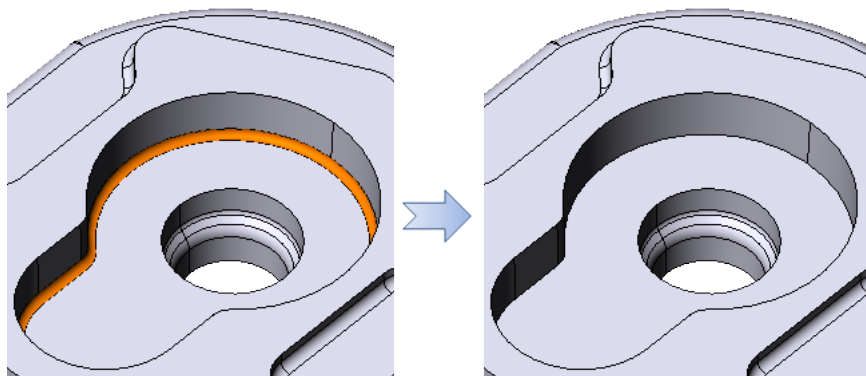


図 4. CADファイルを送信して変更を依頼する代わりに、ESPRITで直接フィレット半径を変更します。

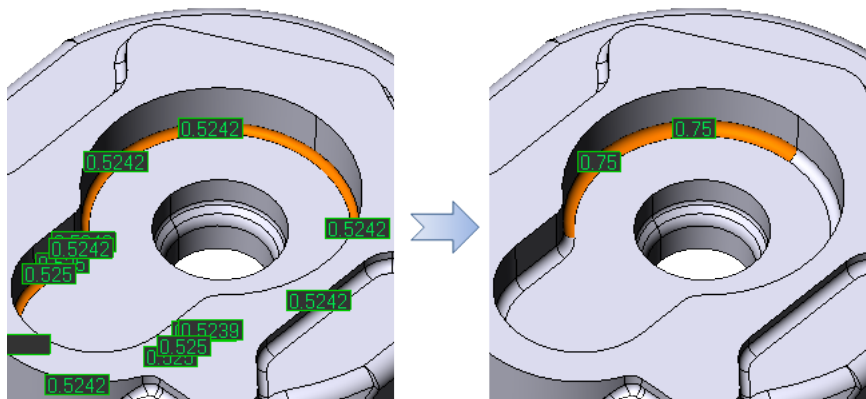
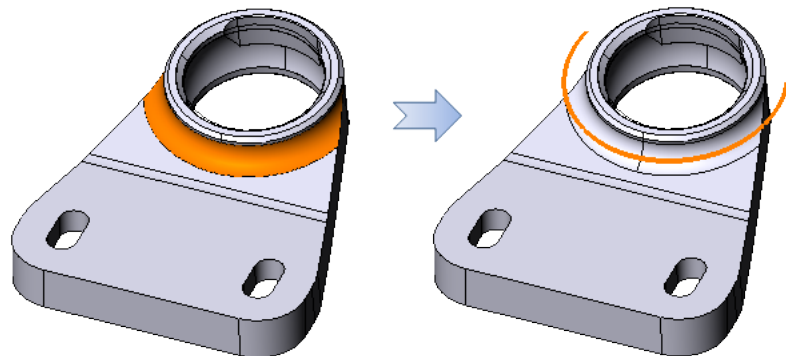


図 5. 加工オペレーションでドライブ曲線として使用されるスパイン曲線を抽出します。



フィーチャーの強化機能

フィーチャーのサブエレメントの端点を認識できるほか、自動チェーン機能によって、ソリッドモデル上に作成する場合の深さ、ドラフト、切削サイドなどの属性を認識できます。また新しい [穴フィーチャー認識] コマンドに、ソリッドモデル上でより多くの複雑なタイプの穴を認識するためのオプションが追加されました。

フィーチャーの新機能自動チェーン機能でソリッドボディの属性を認識できます。

ソリッドエレメント (ループ、エッジなど) を選択すると、平面フィーチャーの深さ、ドラフト角度、および切削サイドが計算されるようになりました。この動作は以前のバージョンの [フェース輪郭] コマンドに似ています。

図 1. 自動チェーンで、ソリッドエレメントから深さ、ドラフト角度、および切削サイドを認識できるようになりました。

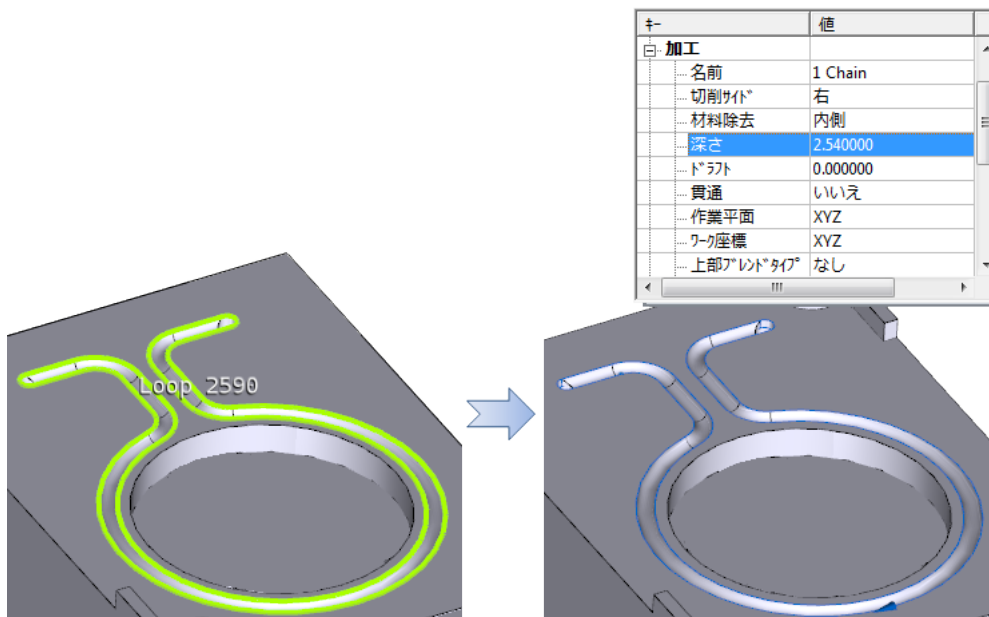


ソリッドから深さが認識されると、通常の場合自動チェーンはループ内で最短のフェースの最大長を使用します。

選択したものがアクティブな作業平面に平行でない場合、その方向が曖昧になることもあります。以前のバージョンの ESPRIT では、[フェース輪郭] コマンドで2つのフィーチャーが作成されました。現バージョンでは、自動チェーン機能が床または底部を見つけてフィーチャーの方向を自動的に判定します。

この強化機能は、隣接している壁が平面ではなく湾曲している場合は特に便利です。その場合、壁の認識コマンドの代わりに自動チェーンを使用できます。ただし、ブレンドタイプは認識されません。

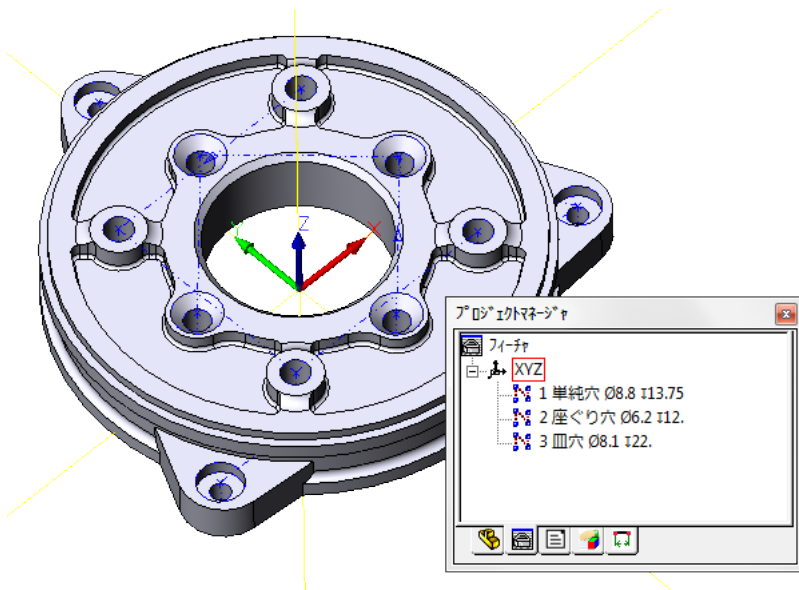
図 2. 自動チェーンでは湾曲した壁の深さも認識されます。



新しい「穴認識」機能

穴認識はソリッドモデルのあらゆる種類の加工穴を簡単に認識できる、まったく新しい機能です。

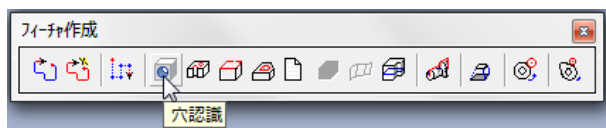
図 1. 穴認識機能によって、あらゆる種類の穴を自動的に認識して適切な名前を付けることができます。



認識される穴の種類:

- ・ 単純な穴
- ・ パーシャル穴
- ・ 複合穴
- ・ 異なる作業平面上にある穴

図 2. 新しい穴認識機能は [フィーチャ作成] ツールバーで使用できます。以前の穴認識コマンドは、旧バージョンの穴認識機能として引き続き使用できます。



認識規則によって、円柱状の開口部が穴として認識されるかどうかが決まります。

図 3. 穴認識のオプション

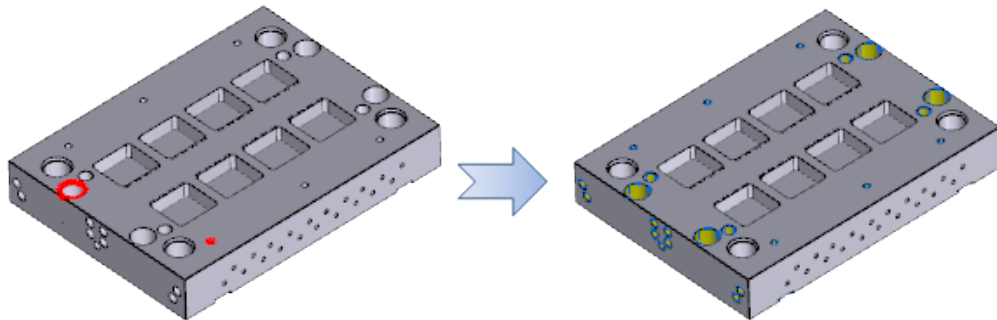


グループ化のオプションは穴認識の後に適用され、これによってフィーチャーマネージャで穴がグループ化される方法が決まります。

- ・ 全穴をPTOPでグループ化: 選択した全ての穴から1つのPTOPフィーチャーを作成します。これは、寸法に関係なく全ての穴を1回でセンタードリル加工する場合に便利です。

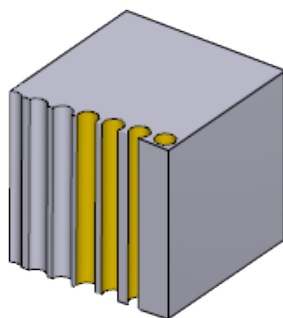
最大直径と最小直径

最大直径と最小直径で指定された範囲内にある直径を持つ穴が認識されます。それ以外の穴は全て無視されます。これは、大きすぎる穴や小さすぎる穴をドリル加工から除外する場合に便利です。



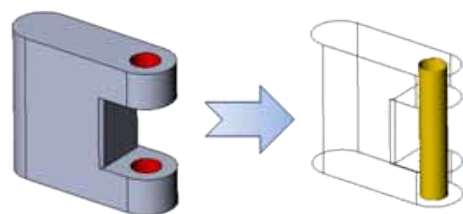
穴開口部の最小角度

この値によって、穴の回転角度に基づいて部分穴が認識されるかどうかが決まります。



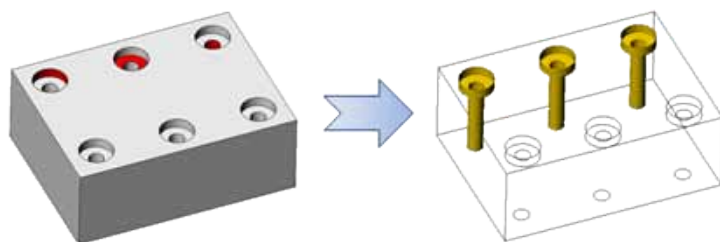
同軸穴の結合

このパラメータは、同じ中心を持つ複数の不完全な穴を1つの穴として認識するか、それとも個別の穴として認識するかを指定します。



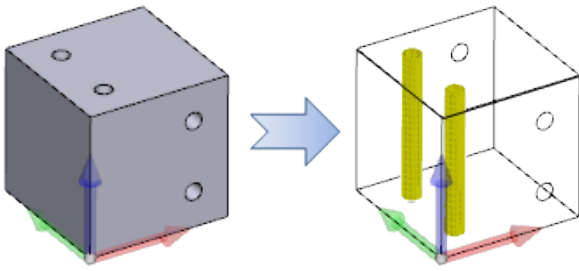
穴フェースの拡張

このパラメータは、選択したフェースの近くにあるソリッドフェースをグループ化するかどうかを指定します。

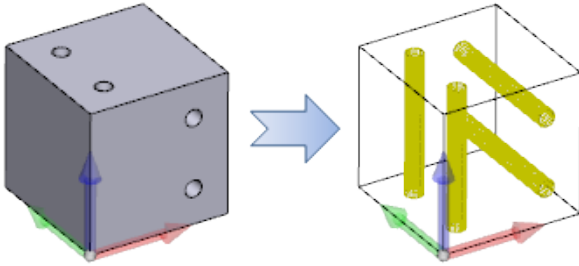


アクティブな作業平面のみ

このパラメータは、穴をアクティブな作業平面のみのZ方向に基づいて認識するかどうかを指定します。



そうでない場合、穴は方向に関係なく認識されます。



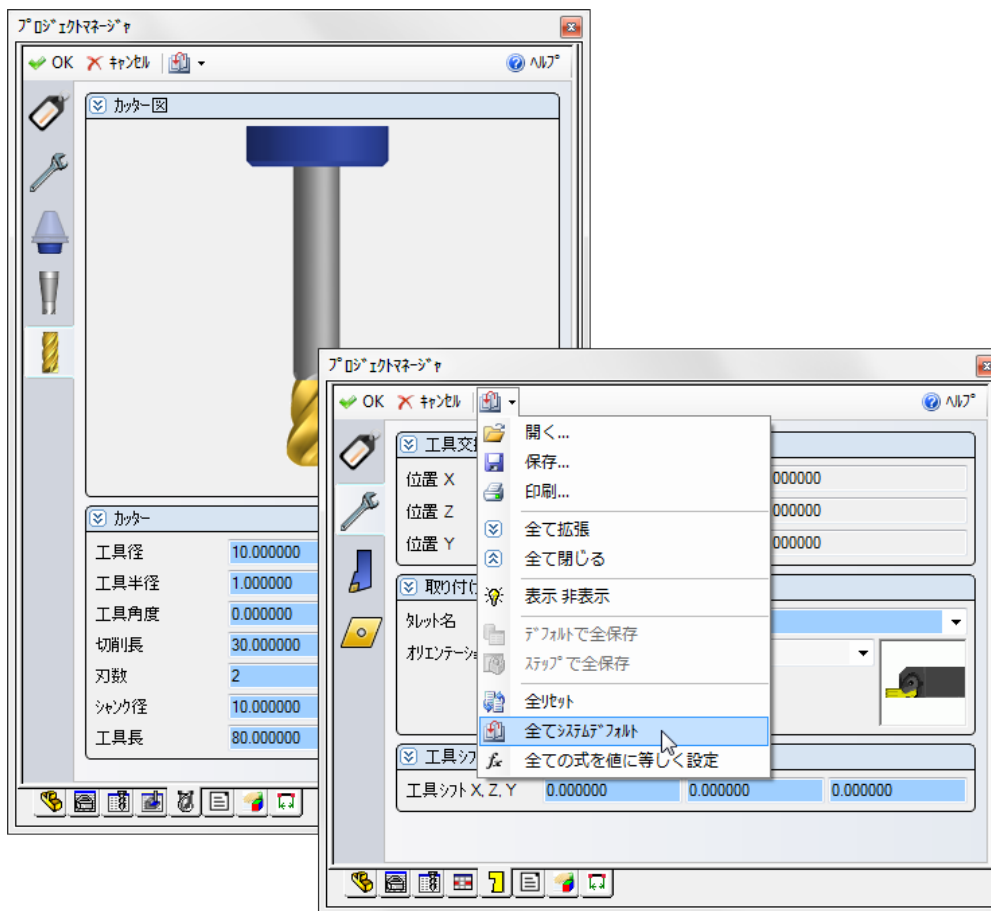
工具の強化機能

工具テクノロジーページが、ESPRITのオペレーションテクノロジーページに合わせて刷新されました。また、新しいMachiningCloudアドインを使用して、ESPRITとMachiningCloudアプリ間で直接通信を行い、大手切削工具メーカーからの工具アセンブリをESPRITドキュメントに簡単にインポートできます。さらにバレル工具のサポートが拡張され、より多くのフリーフォーム加工サイクルを使用できるようになりました。

工具の新機能新しかった工具テクノロジーページ

ミル加工と旋盤加工用工具のテクノロジーページのデザインが更新されました。全ての工具タイプで一貫したインターフェイスが表示されるようになり、各ページがプロジェクトマネージャに組み込まれているので、工具データの入力中でもESPRITの作業を続けることができます。

図 1. 工具ページがプロジェクトマネージャに組み込まれ、オペレーションページと一貫したインターフェイスで作業できます。

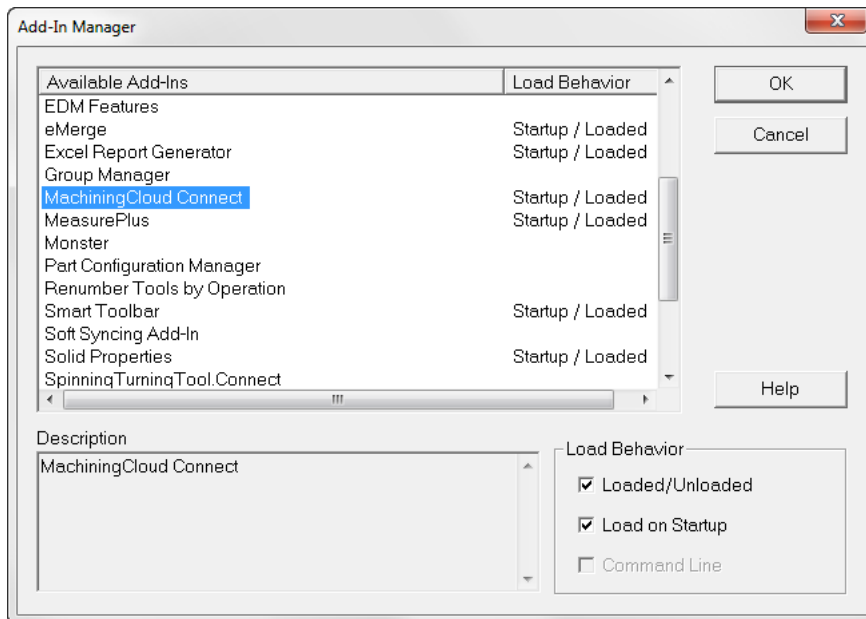


MachiningCloud Connectアドイン

[ファイル] メニューの新しいコマンドを使用して、MachiningCloudアプリを直接操作できます。MachiningCloudアプリを使用すると、大手切削工具メーカーからの工具の検索、選択、組み立てを最も迅速に行えます（アプリは<https://www.machiningcloud.com>から無償でダウンロードできます）。

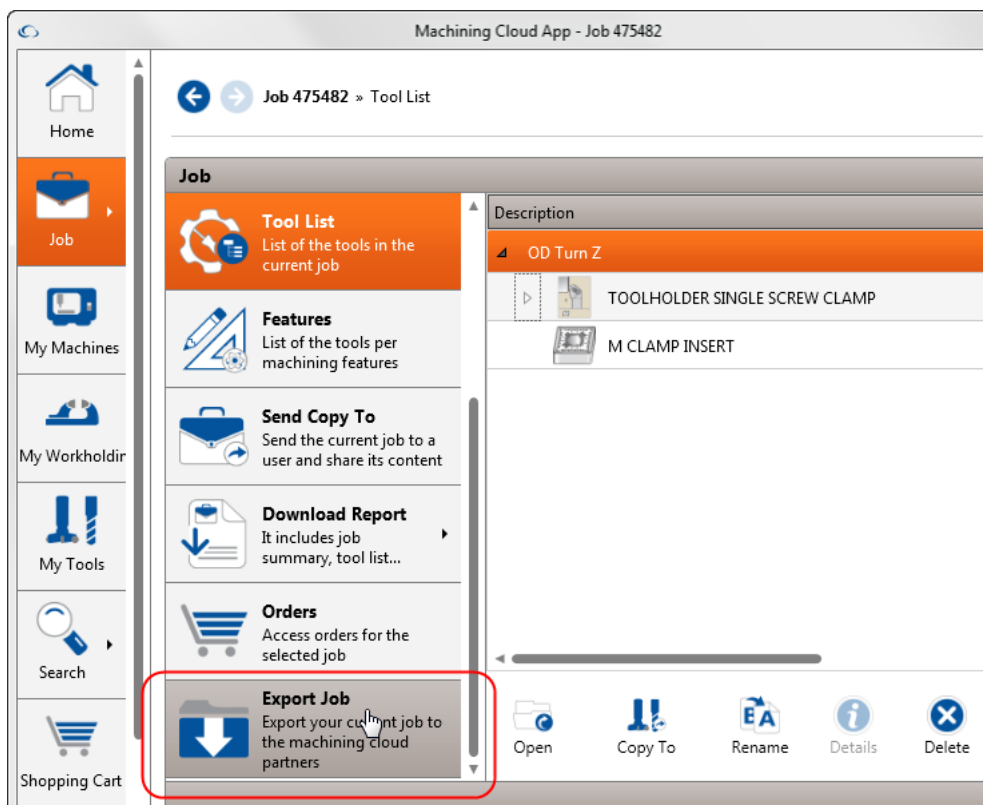
アプリをダウンロードしてインストールした後、新しいMachiningCloud Connectアドインを使用して、ESPRITをMachiningCloudのジョブデータに直接接続します。

図 1. MachiningCloud Connectアドインを使うとESPRITからMachiningCloudアプリに直接接続できます。



MachiningCloudアプリを使って仮想切削工具の選択、定義、組み立てを行い、Machining Cloudジョブをコンピュータに送信することができます。

図 2. MachiningCloudアプリで工具を組み立てて、ジョブをESPRITにエクスポートすることも可能です。

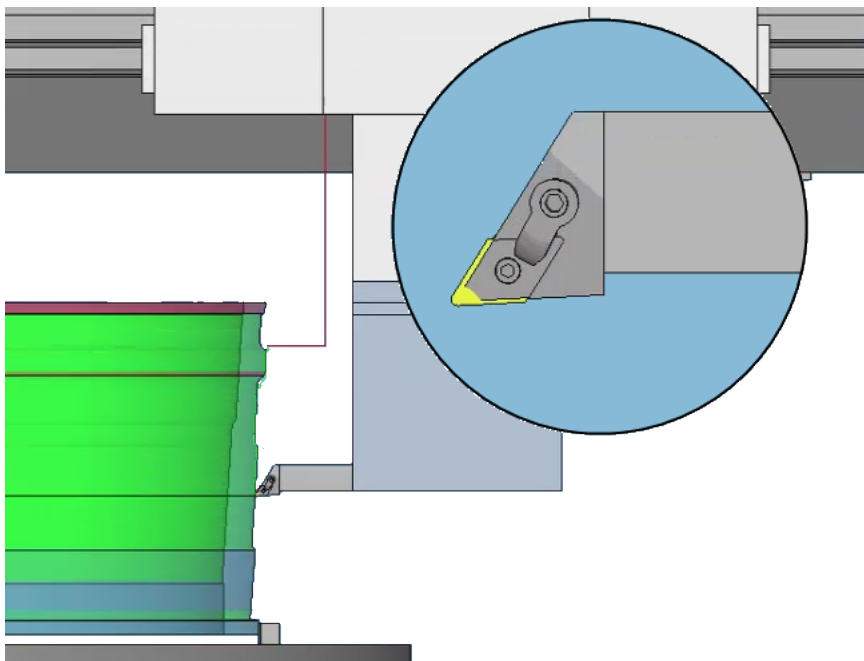


[MachiningCloudジョブを開く] をクリックするだけで、ESPRITで工具をインポートおよび作成し、シミュレーション時に正確な干渉検出を行うことができます。

図 3. MachiningCloud ジョブを直接ESPRITにインポートして、工具を工具マネージャに追加します。

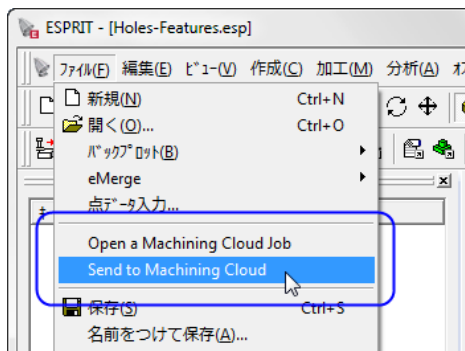


図 4. ESPRITで仮想切削工具をシミュレーションします。



パートに関する情報をMachiningCloudアプリに送信して、推奨される切削工具の一覧を作成することもできます。ESPRITで加工するフィーチャーを選択し、[MachiningCloudに送信] をクリックしてMachiningCloudにエクスポートします。

図 5. フィーチャーに関する情報をESPRITからMachiningCloudアプリに送信します。



MachiningCloudアプリに加工フィーチャーをインポートして、必要な加工方法を指定してから推奨される工具または別の代替工具を選択し、ジョブに追加します。その後、ジョブをESPRITにエクスポートします。

新しい工具モデルの設定機能

[共通加工定義] ツールバーに新しく追加された工具モデルの設定機能を使用して、マシンセットアップのために工具ホルダーのソリッドモデルを追加、修正、削除できます。

図 1. アプリケーションを起動すると、タレットホルダーのツリー表示と、各工具ステーションに現在取り付けられている切削工具を示すダイアログが開きます。

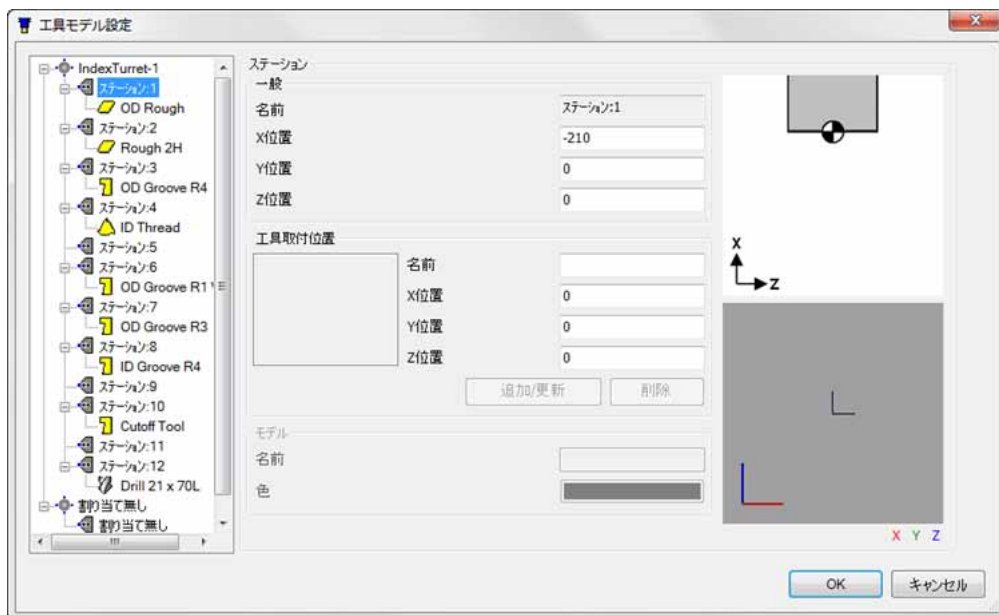


図 2. 目的のステーションを右クリックして、タレットホルダーのソリッドモデルを追加または更新します。

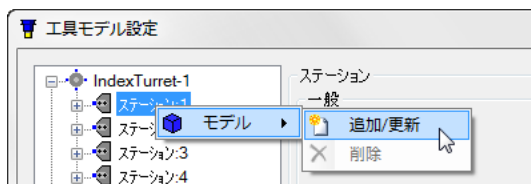


図 3. タレットに取り付けられたホルダー。

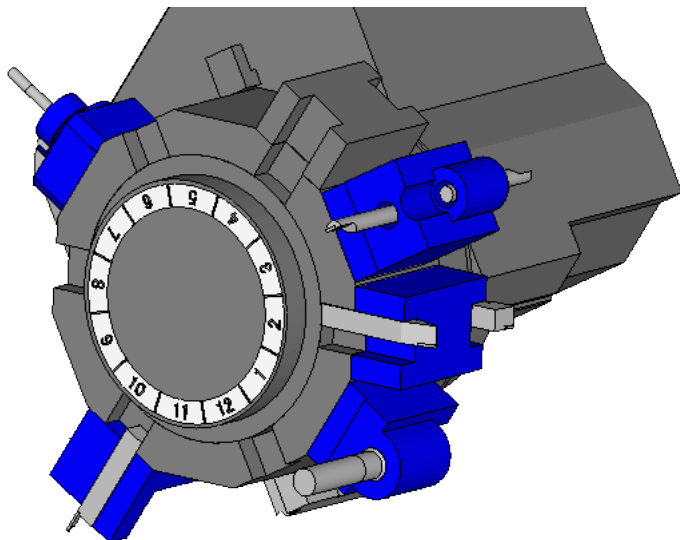


図 4. 目的の工具を右クリックして、工具ホルダーのソリッドモデルを追加または更新します。オフセットを追加して、ホルダーに工具を正しく設定することもできます。

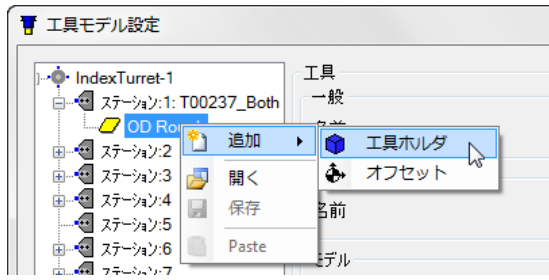
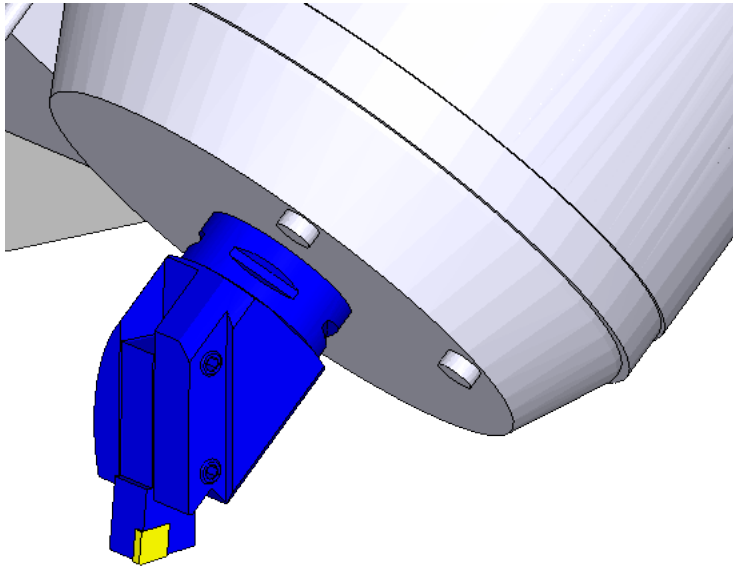


図 5. B軸ヘッドに取り付けられた工具ホルダー。



カスタム工具ホルダーのソリッドモデルをESPRITで作成し、STLモデルとして保存してから、[工具モデルの設定]で工具ホルダーとしてインポートすることができます。

図 6. ホルダーモデルはマシンに取り付けるオリエンテーションに合わせて配置します。ミル工具ホルダーは、Zベクトルが工具軸に沿う方向を指し、取り付け位置がグローバル座標系の原点になるよう配置します。

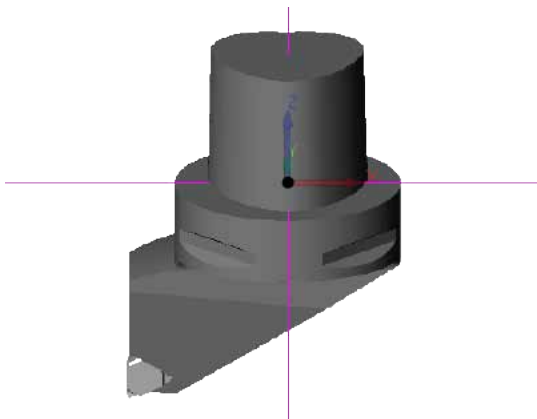
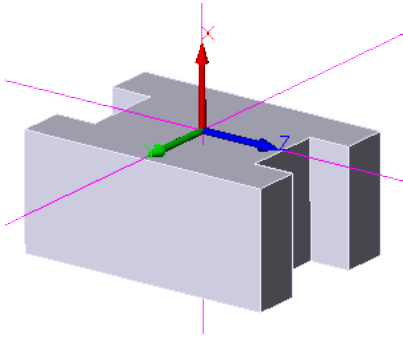
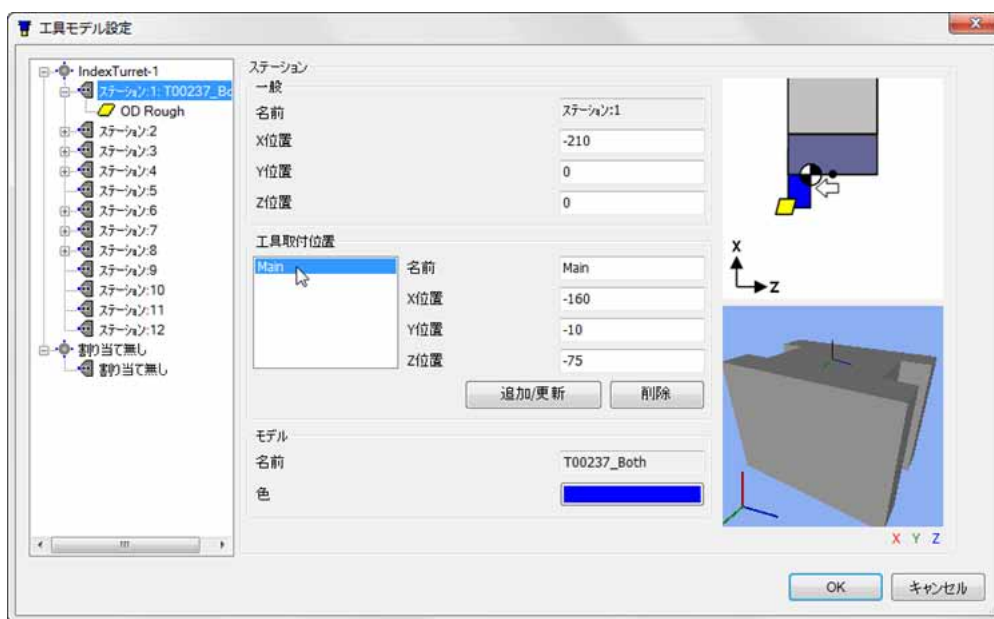


図 7. タレットホルダーにも同じ原理が適用されます。マシンに取り付けるオリエンテーションに合わせて、取り付け位置を原点に揃えてモデルを配置します。

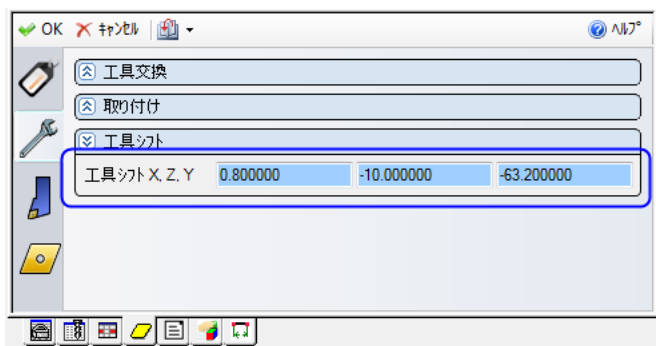


- ▶ モデルの位置は、便宜性と一貫性を保つために設定します。ホルダーの位置は [工具モデルの設定] ダイアログで調整できます。

図 8. ホルダーの位置およびホルダーに取り付けられる工具は、[工具モデルの設定] ダイアログで簡単に調整することができます。

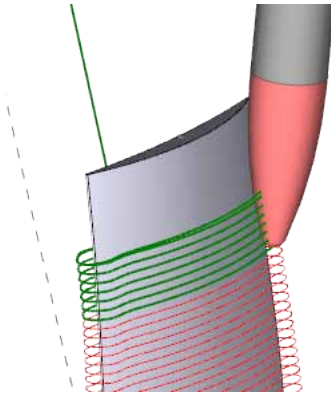


- ▶ [工具モデルの設定] 機能は、各工具に工具シフトのXYZ値を自動的に設定することにより、工具の位置を調整します。工具定義で工具シフトのXYZ値を手動により変更しないでください。代わりに [工具モデルの設定] で工具の取り付け位置を使用するか、工具にオフセットを追加します。工具長などの工具パラメータを修正する必要がある場合には、工具シフト値を再計算しなければなりません。これを行うには、そのような調整を加えた時点で「工具モデルの設定」ユーティリティを再起動してください。



バレル工具のサポートの強化

バレル工具は、通常使用される5軸スパイラル仕上げ加工サイクルで引き続き使用できますが、その他の5軸および3軸サイクルでも使用できるようになりました。



5軸スパイラル荒加工と仕上げ加工

- ・ 5軸ポート仕上げ加工
- ・ 5軸コンポジットミル加工
- ・ 5軸輪郭加工
- ・ 3軸等高線荒加工
- ・ 3軸等高線仕上げ加工
- ・ 3軸平行平面仕上げ加工
- ・ 3軸床仕上げ加工

次の表は、3軸および5軸の各フリーフォームサイクルでサポートされる工具について概要をまとめたものです。

5軸加工サイクルに対応する切削工具

工具シャンク	テーパなし			テーパ付き			アンダーカット			その他	
	ボール	ブルノーズ	エンドミル	ボール	ブルノーズ	エンドミル	アンダーカットミル	T-スロットミル	面取りミルタイプ4	バレル工具	カスタムミル
スパイラル荒加工と仕上げ加工	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
コンジット	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
輪郭加工	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
輪郭加工 (面取り)	✓	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗
インベラー加工	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗
ポート仕上げ加工	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
ポート荒加工	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
スワーフ (ルールド) プロフィール同期: [最適フィットになるようにTAを適合] 以外の全て	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗
スワーフ (サーフェス) プロフィール同期: [最適フィットになるようにTAを適合]	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

工具シャンク	テーパなし			テーパ付き			アンダーカット			その他	
	ボール	ブルノーズ	エンドミル	ボール	ブルノーズ	エンドミル	アンダーカットミル	T-スロットミル	面取りミルタイプ4	バレル工具	カスタムミル
チャンネルミル加工	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

3軸加工サイクルに対応する切削工具

工具タイプ	テーパなし			テーパ付き			アンダーカット			その他	
	ボール	ブルノーズ	エンドミル	ボール	ブルノーズ	エンドミル	アンダーカットミル	T-スロットミル	面取りミルタイプ4	バレル工具	カスタムミル
等高線荒加工	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
等高線仕上げ加工	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
平行平面仕上げ加工	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3D輪郭加工	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
床仕上げ加工											
放射状仕上げ加工	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
スパイラル仕上げ加工											
同心仕上げ加工	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗
曲線間仕上げ加工											
コーナー再加工	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
ペンシルトレーシング											

新しいProfitTurning加工法

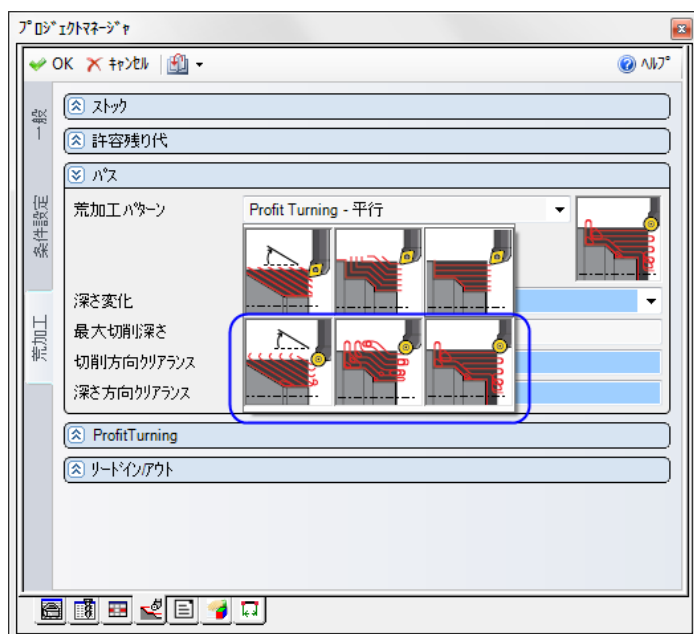
ProfitTurningは、トロコイド切削パスに従来の線形パスを組み合わせることで工具負荷を一定に保ち、難削材料の加工に最適な新しい旋盤荒加工方法です。トロコイドパスはシャープコーナーの従来の切削パスを置き換えるもので、工具が壁に接触する部分に円弧を挿入して、工具が切削のロールインとロールアウトを行えるようにします。新しい加工方法であるProfitTurningは、旋盤加工バランス荒加工に使用できるほか、溝切りの荒加工オプションとしても使用できます。

旋盤工具の新機能旋盤荒加工におけるProfitTurning

荒加工サイクルに、既存の加工方法に対応する3つの新しいProfitTurning加工法が追加されました。

- ・ ProfitTurning - 平行
- ・ ProfitTurning - オフセット
- ・ ProfitTurning - 角度

図 1. 荒加工サイクルで3つの新しいProfitTurning加工法を使用できます。



▶ ProfitTurning加工法は、工具の両側で切削が可能な円状のインサートを持つ切削工具のみをサポートします。選択できる工具は、フル円溝切り工具、円旋削インサート、または任意のフォーム/プロフィールのカスタム工具に限られます。

ProfitTurningは、サイクル時間を大幅に短縮し、工具の寿命を増加できる高速荒加工方法です。ProfitTurningによって、ツールパスに含まれるシャープな移動を丸み付き移動に置き換えることができます。工具方向の急激な変化を排除することで、一貫して高速な送り速度が可能になります。

図 2. ProfitTurningの平行加工法は、シャープコーナーにおけるツールパスのスムーズ化を行います。

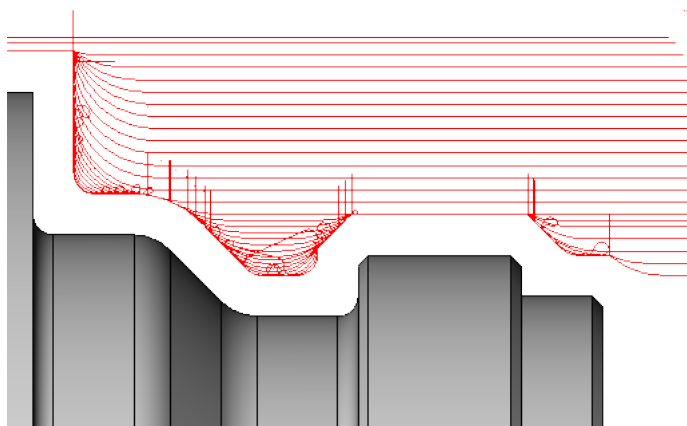


図 3. ProfitTurningのオフセット加工法は、コーナーでパートのオフセットをトロコイド動作に置き換えます。その場合、工具に適した切削力と切削方向を使用してください。

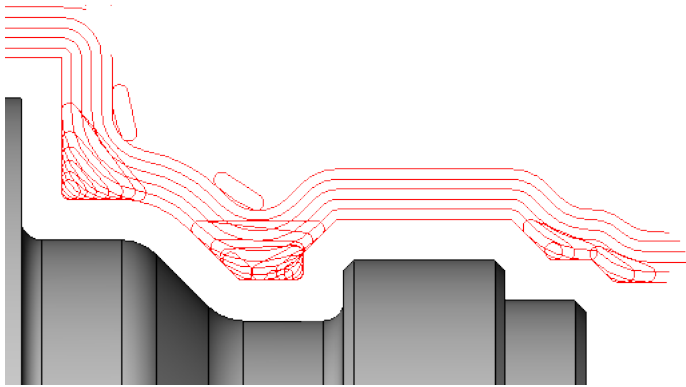
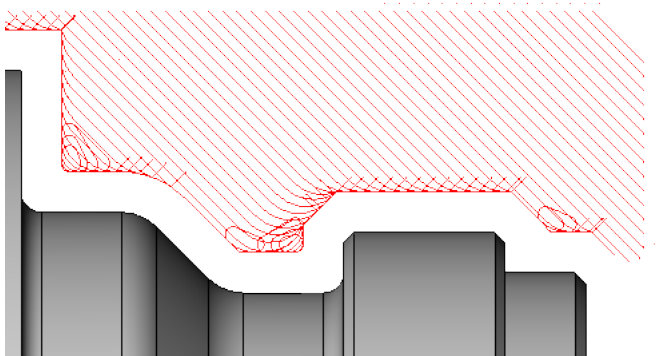
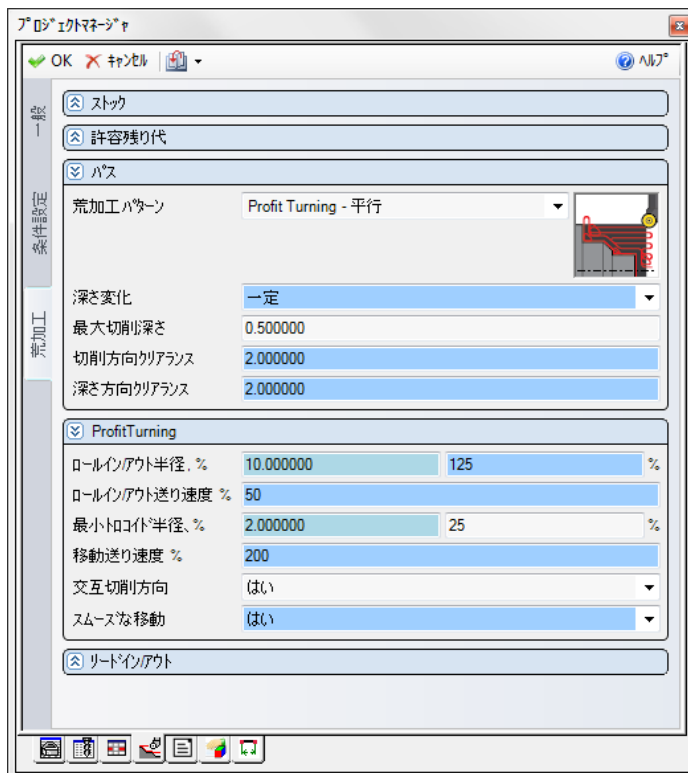


図 4. ProfitTurningの角度加工法は、ユーザー定義の角度で増分パスを作成します。



新しいProfitTurning加工法には、新しいテクノロジーパラメータが付随しています。

図 5. ProfitTurning加工法用の新しいProfitTurningテクノロジーパラメータ。



▶ ProfitTurning加工法を選択した場合、[リトラクト] グループの全パラメータが非表示になり、[深さのリトラクト%] は「110%」、[引き上げモード] は「フィーチャ沿い」にハードコードされます。バランスの悪いイン/アウト移動はジグザグモードで隣接するパスを接続する際に問題の原因となるため、これらのパラメータはシステムによって制御されます。

図 6. [ロールイン/アウト半径] は、線形の荒加工パスが壁に接触する全ての箇所にも円弧を追加します。

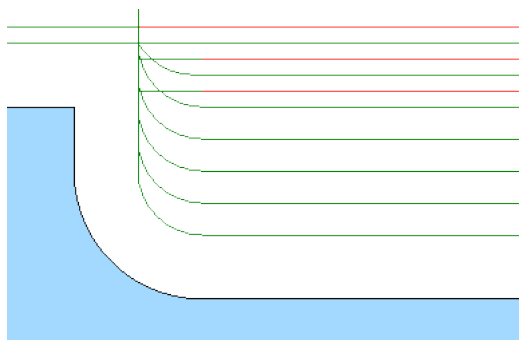


図 7. [ロールイン/アウト送り速度] は、ロールインおよびロールアウトの円弧に適用されます。

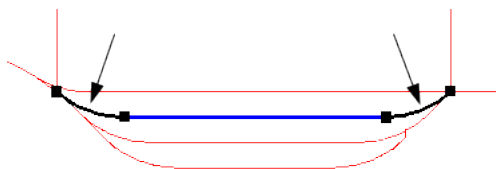


図 8. [交わり角度] は、工具が収まらない小さいコーナーでのトロコイド動作の大きさを制限します。

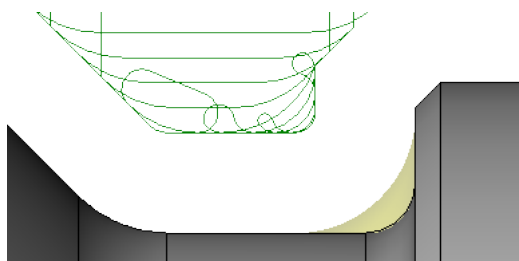


図 9. [移動送り速度 %] は、トロコイド動作の終点と次の移動の始点の間で直線移動に沿って送り速度を調整します。

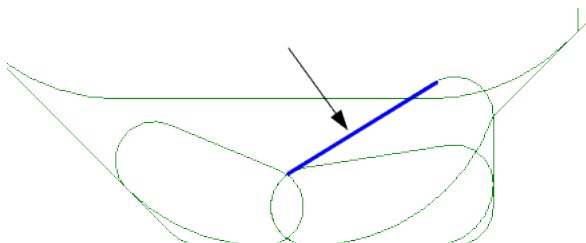


図 10. [交互切削方向] は、切削の方向を交互に切り替えてリトラクト移動を排除します。

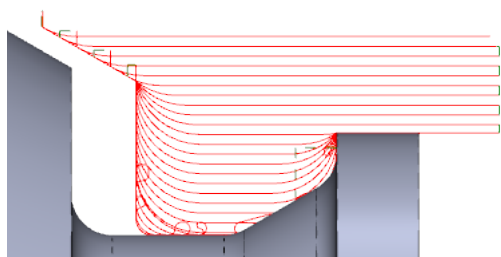
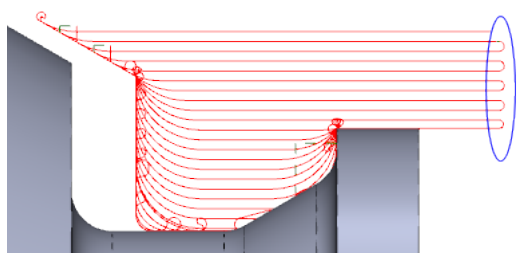


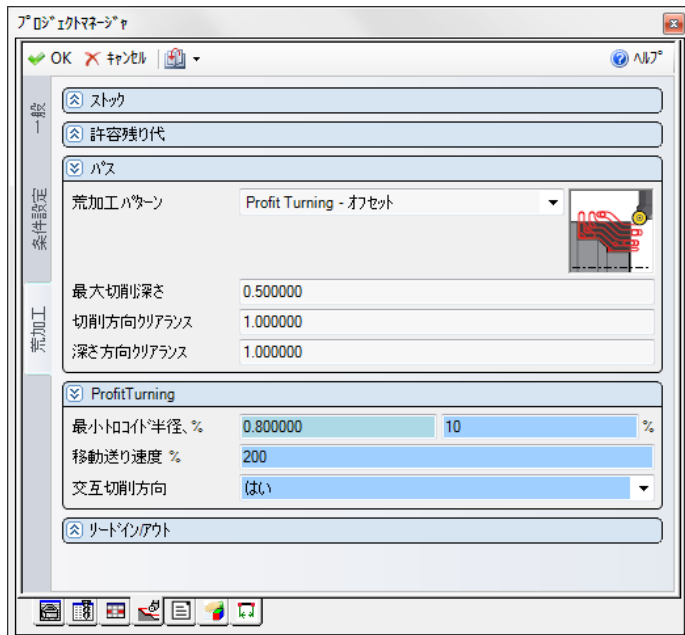
図 11. [スムーズ移動] は、交互のツールパスに含まれるブリッジ移動をスムーズな円弧で置き換えます。



オフセットツールパス

オフセットパスは固定されたオフセット値のみをサポートします。したがって、[X切削深さ]と[Z切削深さ]のパラメータが[最大切削深さ]で置き換えられます。

図 12. ProfitTurningのオフセット加工法のパラメータは若干異なります。



[切削方向クリアランス]は、平行移動ではなく、切削条件に基づいてクリアランス移動を作成するために機能強化されています。クリアランス移動には状況に応じて次の4つのタイプがあります。

- ・ 接するセグメント
- ・ 90度の正接円弧 + セグメント
- ・ 垂直セグメント
- ・ 可変度の正接円弧 + セグメント

図 13. 4タイプのクリアランスパス。



旋盤溝切り加工におけるProfitTurning

溝切りサイクルに新しいProfitTurning加工法が追加されました。この加工法は荒加工サイクルの「ProfitTurning - 平行」と似ています。

図 14. 荒加工パスの新しいProfitTurning加工法を、溝切りサイクルで使用できます。

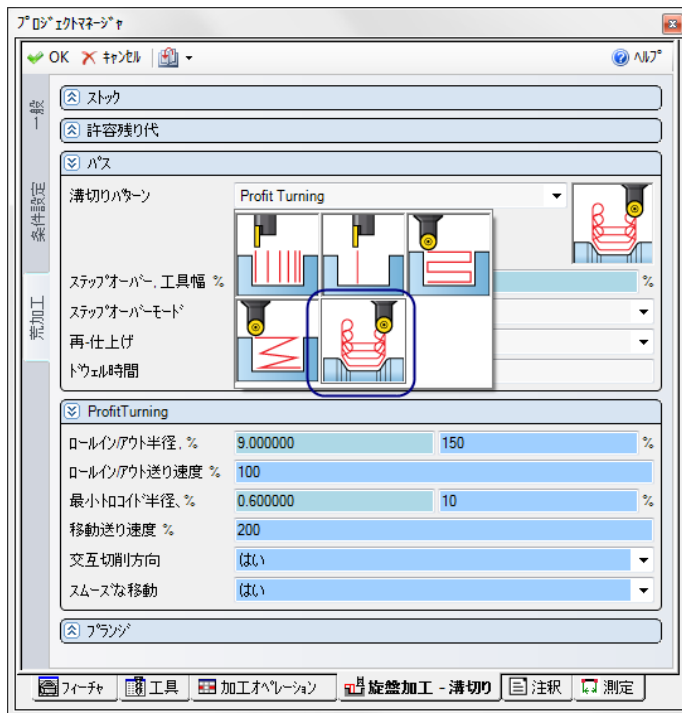
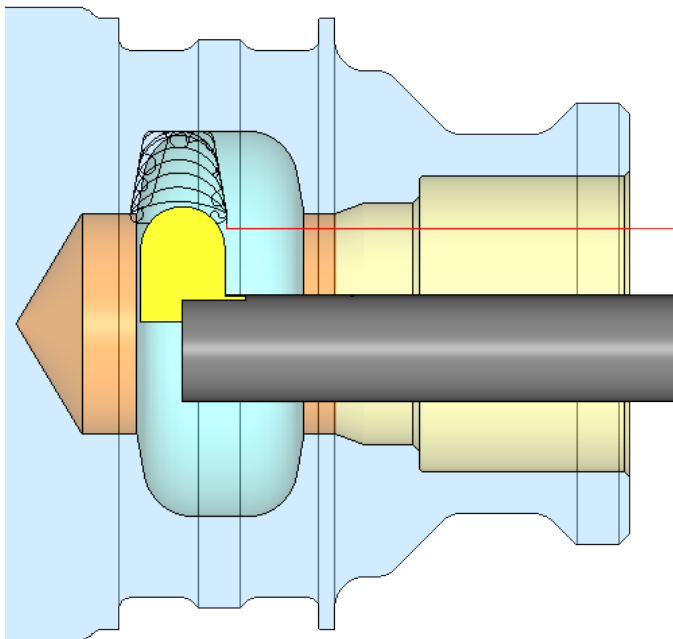


図 15. ProfitTurningによって、溝切りで丸みを持つ荒加工パスが生成されます。



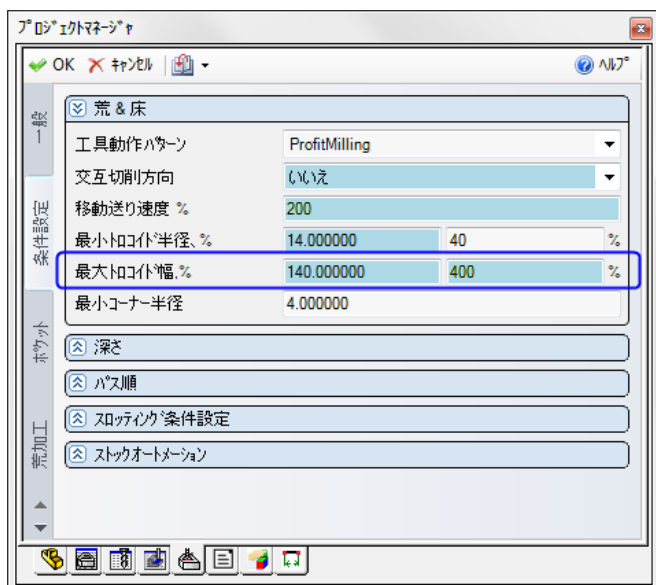
ミル加工の改善点

あらゆる寸法と形のポケットでのトロコイド動作に対応するため、ProfitMillingで幅を調整できるようになりました。さらに、高速ツールパスが不可能なポケットの場合、新しいオプションによって標準オフセットツールパスで工具負荷を一定に保つことができます。

ミル加工の新機能強化されたProfitMilling機能

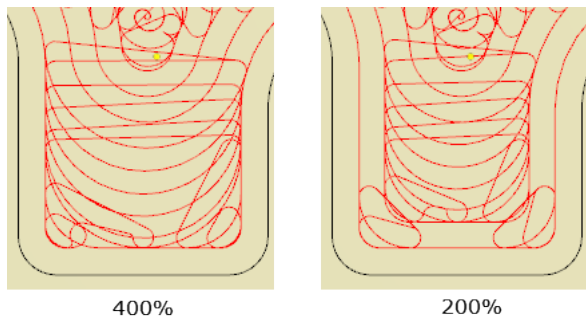
ProfitMilling条件設定が強化され、ツールパスにおけるトロコイド動作の幅を調整できる新しい設定が追加されました。この新しい設定は、ミルポケット加工、変動Z荒加工、およびモールド加工等高線荒加工で使用できます。

図 1. ProfitMillingツールパスの新しいオプション [最大トロコイド幅]。



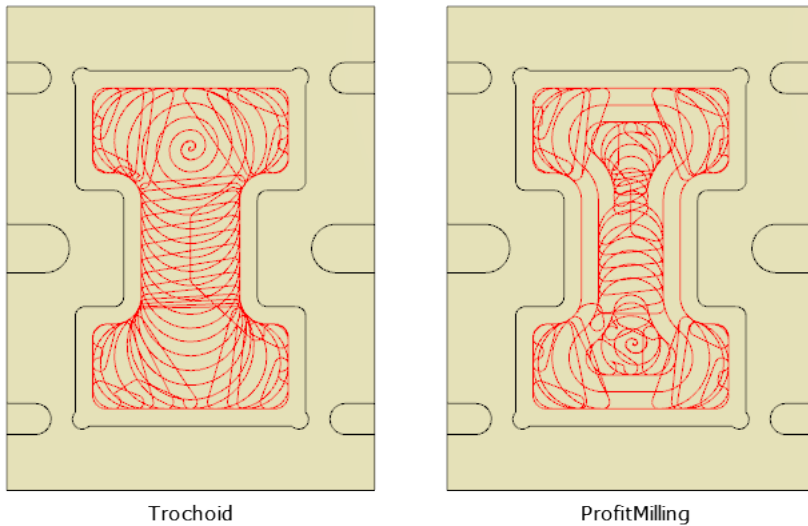
幅には実際の値または工具直径のパーセンテージを入力できます。

図 2. ツールパスでのトロコイド動作の量を調整できるようになりました。幅が大きいほどトロコイド動作も大きくなります。



以前のバージョンでは、工具が大きなスパイラルを描きながら外に移動できるため、トロコイドパターンは閉塞なしのポケットの方により適していました。新しい強化機能によって、ProfitMillingで大きなトロコイド動作を作成し、より多くの種類のポケットで高速加工のメリットを活用できます。

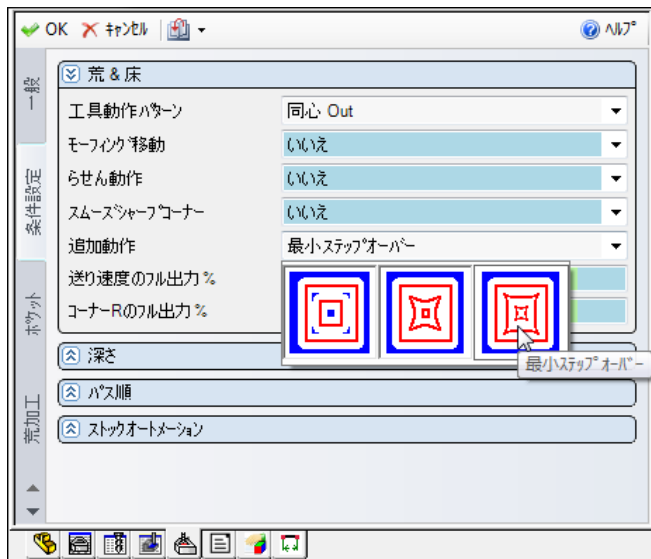
図 3. トロコイドおよび強化されたProfitMillingツールパスの比較。



ポケット加工と変動Z荒加工の新しいオプション [ステップオーバーの最小化]

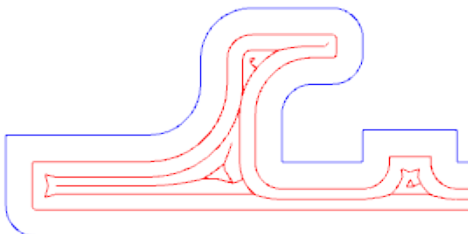
[追加動作] の設定に、標準の工具動作パターンにおける工具の交わり角度を一定に保つ「ステップオーバーの最小化」という新しいオプションが追加されました。

図 1. ポケット加工の新オプション「ステップオーバーの最小化」



[ステップオーバーの最小化] は、指定されたステップオーバーを可能な限り維持します。これは、工具の交わり角度を一定に保つ必要があり、高速加工サイクルが不可能な場合に便利です。

図 2. [ステップオーバーの最小化] で工具の交わり角度を一定に保ちます



▶ [ステップオーバーの最小化] は、トロコイドやProfitMillingなどの高速の工具動作では使用できません。

5軸加工の機能強化

ESPRIT 2016では複雑な壁面のスワーフミル加工機能が強化されました。一定あるいは調整可能な増分深さを適用するオプションを使用して、高さが変化する壁面上で工具が高から低へと移動する際の余分なプランジタイプの動作をなくします。また、ブレード加工サイクルは、適用対象となるパート図形の多様性を反映させて新たに「5軸スパイラル」という名称に変更されました。「5軸スパイラル荒加工」(旧称「ブレード荒加工」) および「5軸スパイラル仕上げ加工」(旧称「ブレード仕上げ加工」) は、ブレードの加工だけでなく、医療人工器官、エンジンコンポーネント、その他多くのパート図形の加工にも適しています。

5軸加工の新機能スワーフサイクルでの一定高増分加工条件

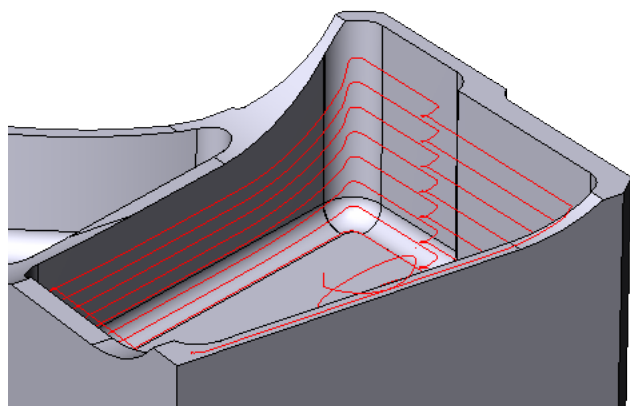
以前のバージョンでは、常に上部プロフィールから下部プロフィールへとパスを変形させることによって、スワーフサイクルの増分深さパスを分布させていました。新しい増分条件設定を使用すると、新しいオプションの「一定高」を選択することができます。

図 1. 新しい増分条件設定には「一定高」と「モーフィング」の2つのオプションがあります。



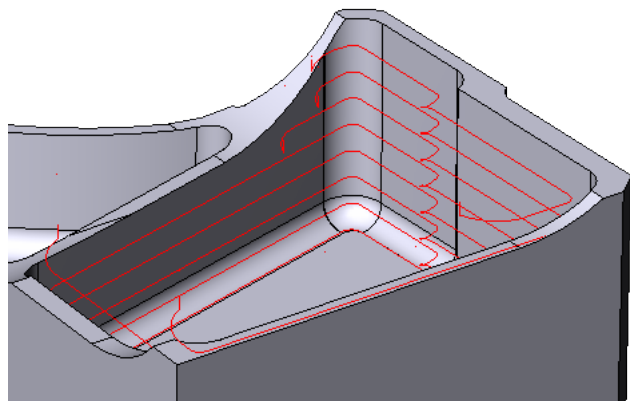
多くの複雑なパートでは、壁面の高さが一定していません。この高さの変化によって、ツールパスに望ましくないプランジコンポーネントが発生します。

図 2. モーフィングの増分パスは上部と下部のプロフィール間で調整を行うため、工具が高から低へ移動する際に不要なプランジタイプの動きが生じる場合があります。



新しい「一定高」オプションでは、全ての切削パスに一定した深さの増分が使用されます。

図 3. 一定高パスは、一定の深さ増分を使用します。



「ブレードサイクル」が「5軸スパイラル」に名称変更

「ブレード荒加工」と「ブレード仕上げ加工」という名称が、それぞれ「5軸スパイラル荒加工」と「5軸スパイラル仕上げ加工」という呼び名に変更されました。新しい名称は、これらのサイクルがブレード加工だけでなく医療人工器官、エンジンコンポーネントなど、さまざまなパートの加工に適用できることを反映しています。

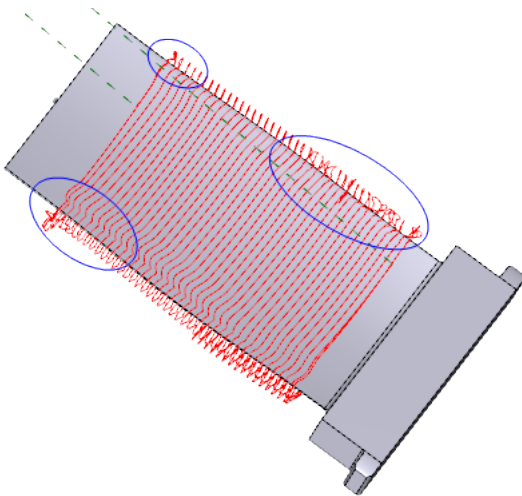
5軸スパイラル加工ではツールパスが以下の方法で最適化されます。

- ・ パートサーフェスに見られる欠点の修正とスムーズ化
- ・ エッジでの高速送り
- ・ ツールパスの簡単な定義

5軸スパイラル仕上げ加工におけるパートサーフェスのスムーズ化

5軸スパイラル仕上げ加工サイクルは、工具軸のオリエンテーションをパートのフェースまたはサーフェスに垂直であるとして計算します。これらのターゲットサーフェスに欠点がある場合、それが非常に小さい欠点であっても工具軸のオリエンテーションに有害をもたらし、不規則な結果が生じます。

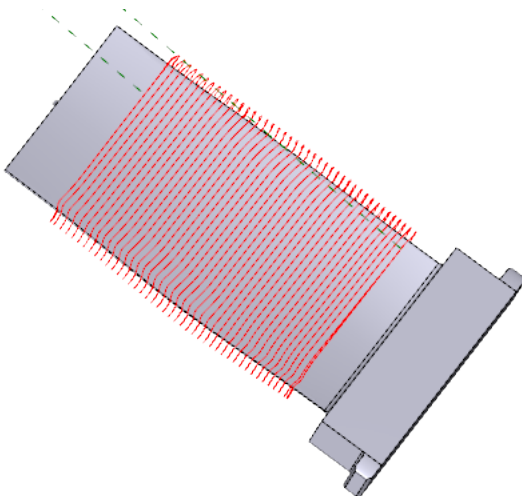
図 1. パートサーフェスにあるマイナーな欠点のために工具軸のオリエンテーションが不規則になる可能性があります。



CAMシステムによる修正が不可能な場合、こうした欠点のあるサーフェスを補正するにはCADシステムに戻ってパートの図形を修正する必要があります。

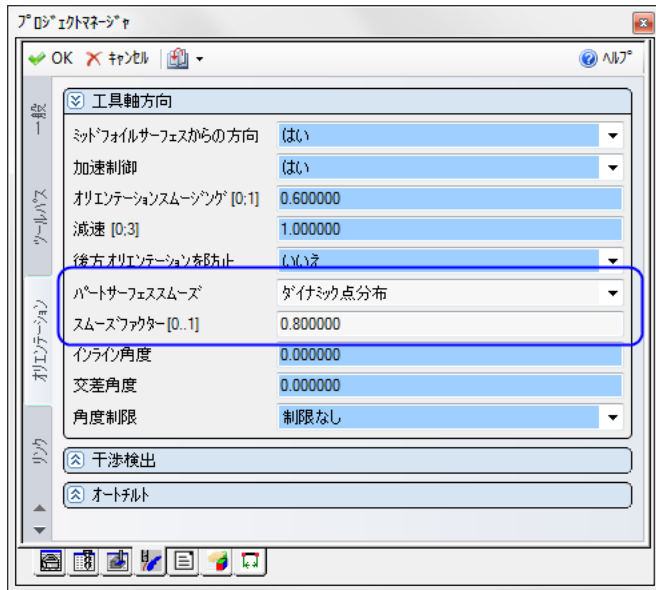
そこでESPRITには、垂直ベクトルがその全ての点で規則的になるよう、パートのフェースやサーフェスをスムーズ化、つまりリラックスさせる新しいオプションが追加されています。

図 2. 「パートサーフェスのスムーズ化」という新しいオプションでは、パートのサーフェスをスムーズ化し、その垂直ベクトルが全ての点で規則的になるように調整できます。



▶ スムーズ化は工具軸の動作のみに影響します。工具位置は常に元のパートフェース/サーフェス上に維持されます。

図 3. 新しい「パートサーフェスのスムーズ化」オプションは、[5軸スパイラル仕上げ加工] の [オリエンテーション] ページで使用できます。



次の2つのオプションがあります。

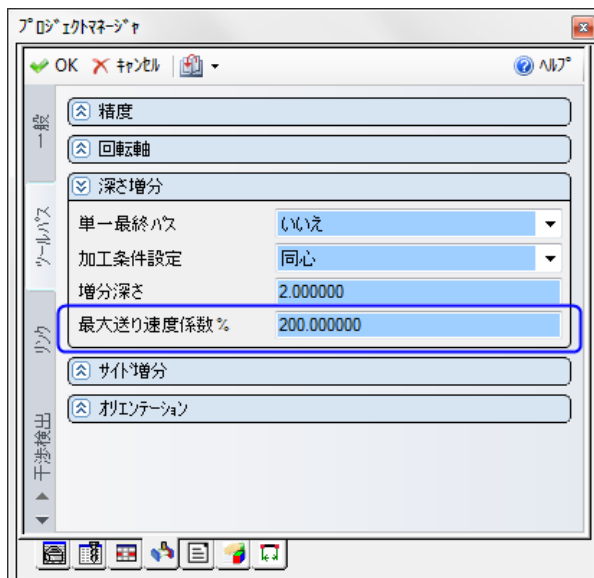
- ・ 動的な点分布: ツールパスの点の間の距離は可変です。この距離は、許容誤差値および加工するサーフェスの形に応じて、最大値 (点間の最大距離) から最小値の間で変化します。
- ・ 一様点分布: 点間の距離は一定しています。固定距離の値は、最悪なケースでのキャリパ歯許容差に従うように計算されます。これは工具がシャープエッジの周りを回転する場合です。その結果、通常は冗長な数の点が作成されます。[一様点分布] を有効にすると、隣接するパスの間で点が整列します。大半の場合、この機能によって工具がブレードエッジの周りを回転する際のサーフェスのアスペクトが改善されます。

[スムーズ化係数] は、スムーズ化の度合いを最小 0 から最大 1 の間で表します。

5軸スパイラル荒加工での高速送りクリアランス

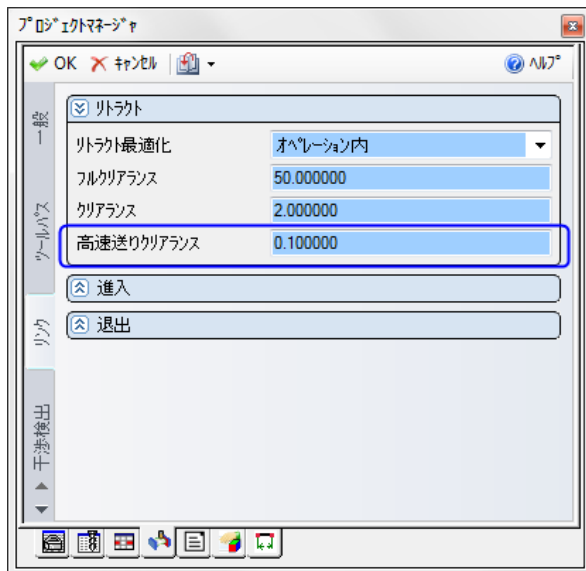
スパイラル荒加工サイクルでは、工具と材料との接触が失われたとき (非切削動作) に、送り速度を増すことができます。送り速度を増加させると、同心切削加工を使用した場合は特に、切削パスが中断される部分の加工時間を短縮できます。送り速度は [ツールパス] ページの [最大送り速度係数%] を使って調整します。

図 1. [最大送り速度係数%] は、工具と材料との接触が失われた際に送り速度を増加させる割合 (%) を入力します。



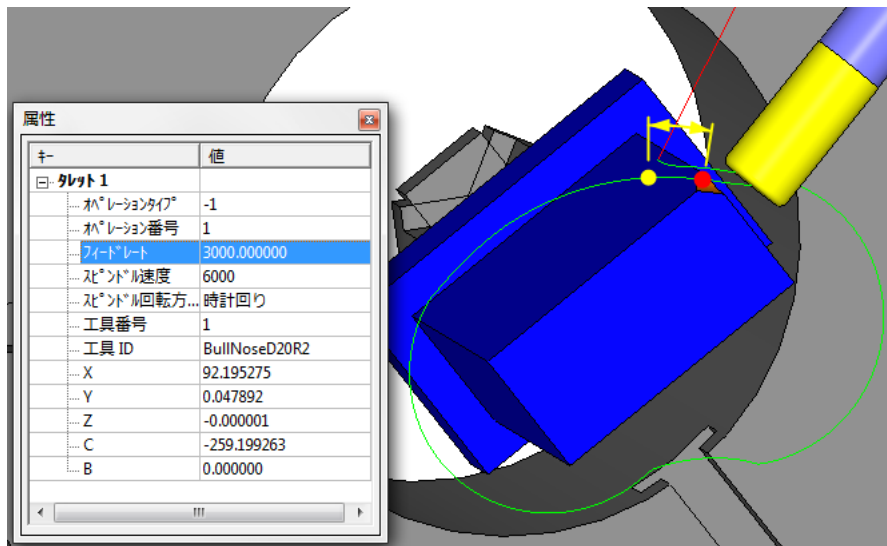
[リンク] ページに新しく追加された設定を使用して、安全距離を追加して工具が材料に近付く際にツールパスの非切削部分の高速送りから従来の作業送り速度まで動きを遅くすることができます。

図 2. 新しい [高速送りクリアランス] パラメータは、送り速度を切り替える際の安全距離を指定します。



このクリアランスによって、作業送り速度 (切削動作) から高速送り (非切削リンク動作) に切り替わる移動点の安全な高さが決まります。[高速送りクリアランス] は、パートではなくストックモデルから測定されます。

図 3. [高速送りクリアランス] は送り速度が切り替わる移動点から測定されます。

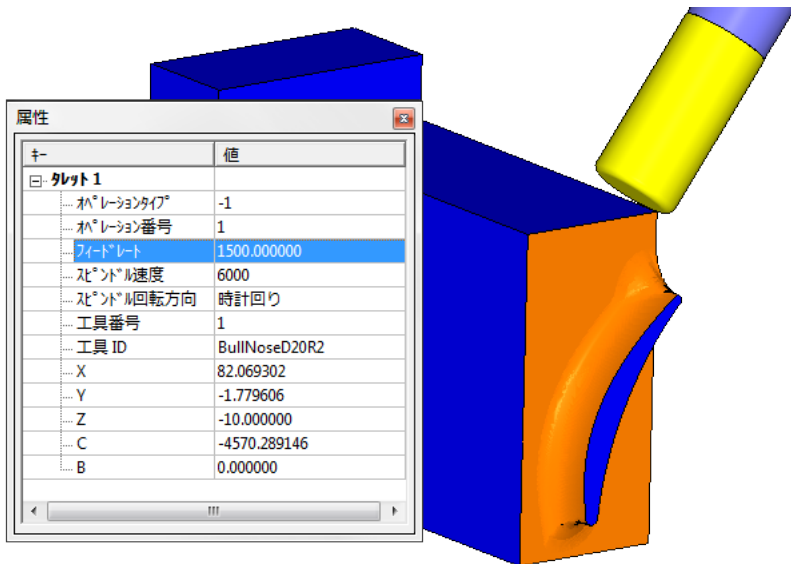


高速送りは早送りではなく、通常よりも高速で行われる送り動作のことです。高速送り速度は、[最大送り速度係数%] を使って計算されます。

$$\text{高速送り速度} = \text{オペレーション送り速度} \times \text{最大送り速度係数\%}$$

高速送り速度は、動作がより高速で行われるだけなので、送り速度の変化はツールパスまたはツールパス分析に表示されません。ただし、切削シミュレーションの際、工具が材料に近づく前後で送り速度値の変化を確認することはできます。

図 4. 送り速度の変化は、シミュレーション中に工具が材料に近付くときと、材料から離れるときに確認できます。

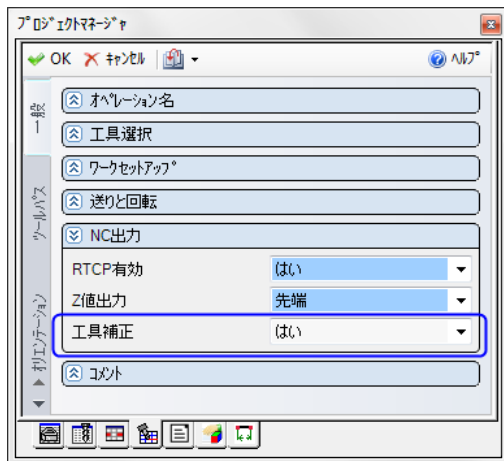


この変化に加え、スパイラル荒加工も安全距離に関係なく非切削動作を最小化するように改善されています。

3D工具補正

従来は3軸ツールパスでの使用に限られていた3D工具補正機能が、5軸ツールパスにも拡張されました。これは、工具補正の出力を有効化することで、Gコードを変更しなくても工具半径を変更できるようになる重要な機能強化です。

図 1. 工具補正は、インペラーを除く全ての5軸仕上げ加工サイクルで使用できます。

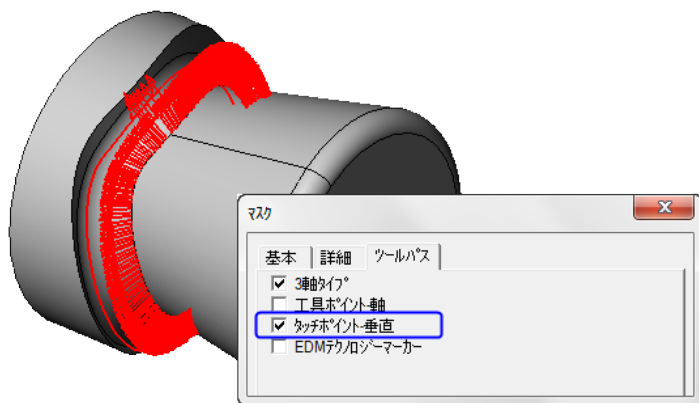


工具補正オプションは次の加工で使用することができます。

- ・ コンポジットミル加工 ([工具位置] が「曲線上の接触点」の場合のみ)
- ・ スワーフ
- ・ 5軸輪郭加工
- ・ スパイラル仕上げ加工
- ・ ポート仕上げ加工

工具補正が有効にされている場合、ツールパスは接触点座標と垂直ベクトルのセットとして出力されます。工具補正に関連するベクトルを表示するには、[マスク] ダイアログ (Ctrl-M) の [ツールパス] ページにある [タッチポイント-垂直] を選択します。

図 2. ツールパスベクトルは図表領域にも表示できます。



- ▶ 5軸のコンポジット加工およびポートミル加工サイクルでは、工具補正は ([ツールパス] ページにある) [工具位置] 設定の [曲線上の接触点] オプションに限定されます。5軸スワープ加工サイクルでは、[輪郭形状同期] に [最適フィットになるようにTAを適合] が設定されている場合、工具補正は使用されません。

使用するマシンフォーマットファイルには、X, Y, Z補正ベクトルを使用できなければなりません。接触点の定義に使用されるポストキーワードは、XSDim, YSDim, ZSDimです。垂直ベクトルの定義に使用されるポストキーワードは、ISDim, JSDim, KSDimです。

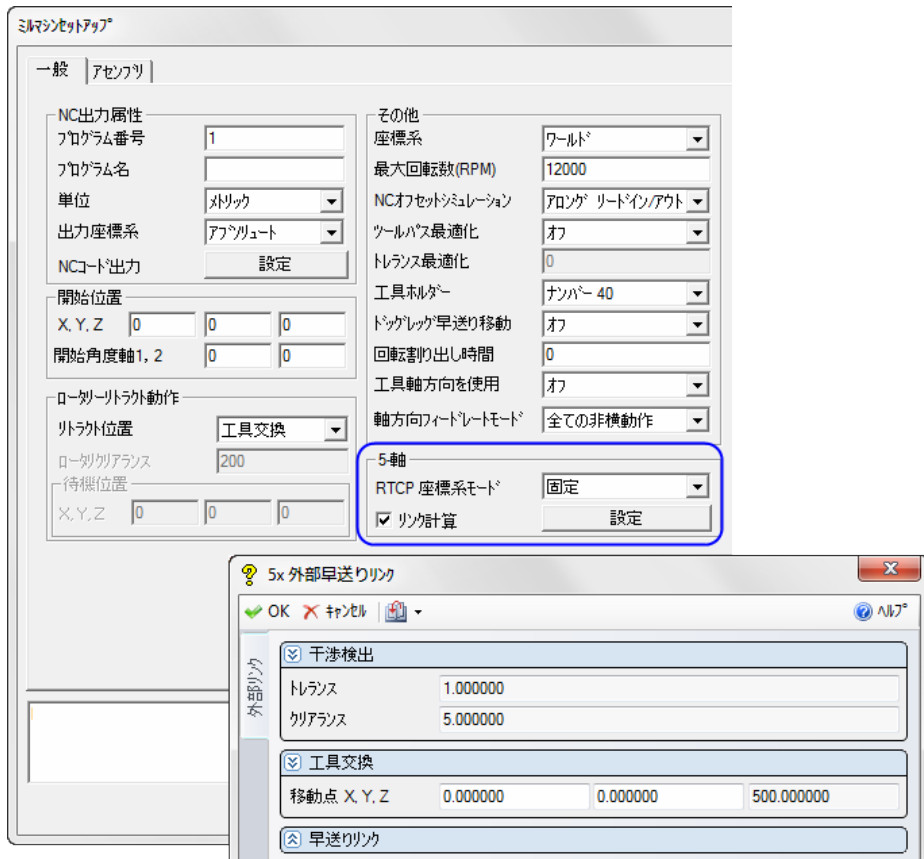
衝突のない外部リンク

ESPRIT 2016では、[マシンセットアップ] で有効にされた外部リンクの計算に、ソリッドボディ全体との衝突を避ける新しい計算方法が採用されています。

1つ以上の5軸加工オペレーションが使用される場合、オペレーションのフリーフォームフィーチャーで少なくとも1つのフェースが定義されている全てのソリッドについて、衝突のない外部リンクがシステムにより作成されます。

5軸加工オペレーションが、オペレーションマネージャの最初または最後のオペレーションである場合、移動点はこれらのソリッドの外になければなりません。そうでない場合、リンクはエラーとなります。

図 1. 外部リンクの移動点はソリッドモデルの外になければなりません。

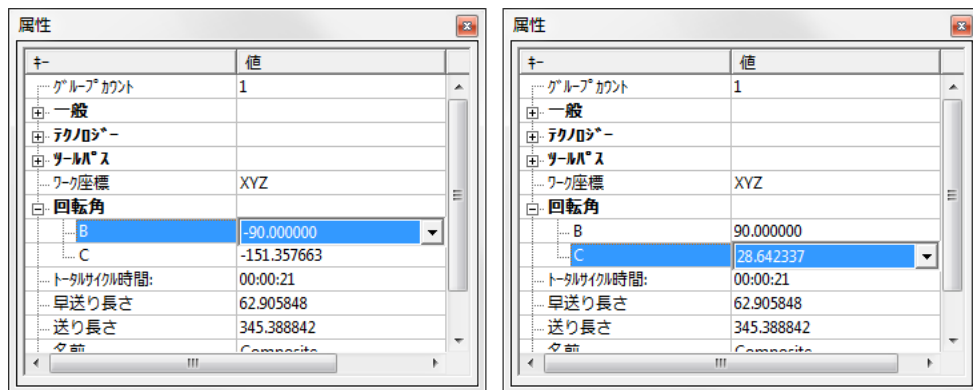


5軸外部リンクの回転ソリューション

5軸加工オペレーションの回転位置は、前のオペレーションの終了と次のオペレーションの開始における回転軸位置（最初/2番目のソリューション）によって決まります。ESPRITでは、オペレーション間の最小距離により回転位置が選択されます。2つのソリューションが考えられる場合など、パートおよび工具によっては、ESPRITで実行される自動回転計算が望ましくないことがあります。

5軸加工オペレーションには、属性マネージャで「回転角度」と呼ばれる新しい属性が追加されています。それぞれの回転軸は、次のオペレーションに達すると想定される回転位置に沿って表示されます。その後、工具および/またはパートに適した角度を選択できます。これは、次のオペレーションに達する方法が2つ以上ある場合に便利です。

図 1. 回転角度は、他の使用可能なオプションに変更できます。1つの回転軸を変更すると、他も自動的に更新されます。



ワイヤー放電加工

ワイヤー放電加工に、特に荒加工の切削を改善するための新しい加工方法が追加されました。

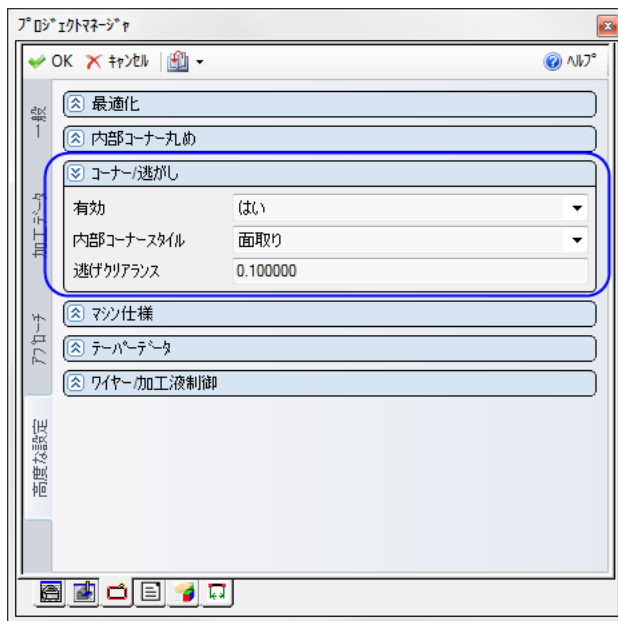
放電加工の新機能コーナーおよび進入確認の新しいリリース機能

許容誤差の小さい高精度スタンピング工具やその他のアプリケーションの場合、荒加工にコーナーリリースを追加することで、スキム加工でコーナーの送り速度を減速せずに、通常のワイヤーパスをなぞることが可能になります。また、進入/確認にリリースを追加すると、最終的な図形の精度に影響を与えずに切削を改善することもできます。

新しいオプションを使ってワイヤーパスを自動的に修正し、ダイや開いたプロフィールの進入および内/外部コーナーにリリースを追加できます。これらの新しいオプションはパンチには適用されません。

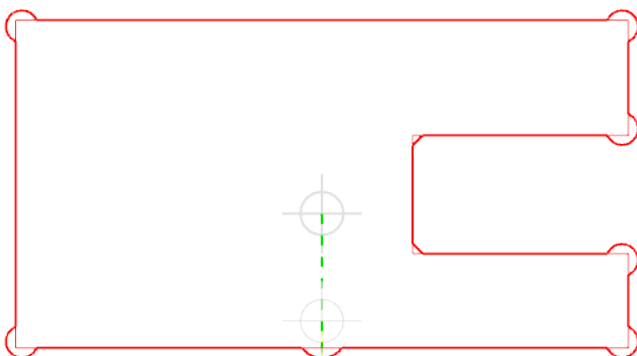
コーナー/確認リリースは、AgieVisionを除く全ての放電加工マシンタイプに導入されており、2軸輪郭加工オペレーションページの [高度な設定] タブで使用できます。

図 1. [高度な設定] タブの新しいオプションを使用すると、コーナーを修正してダイ切削にリリースを追加できます。



コーナー/確認リリースが有効にされると、荒加工ワイヤーパスの荒進入およびシャープな外部コーナーに半径が追加されます。ユーザーは、シャープな内部コーナーに半径またはチャンファーを追加することができます。

図 2. 荒加工の進入と内/外部コーナーにリリースが追加されます。



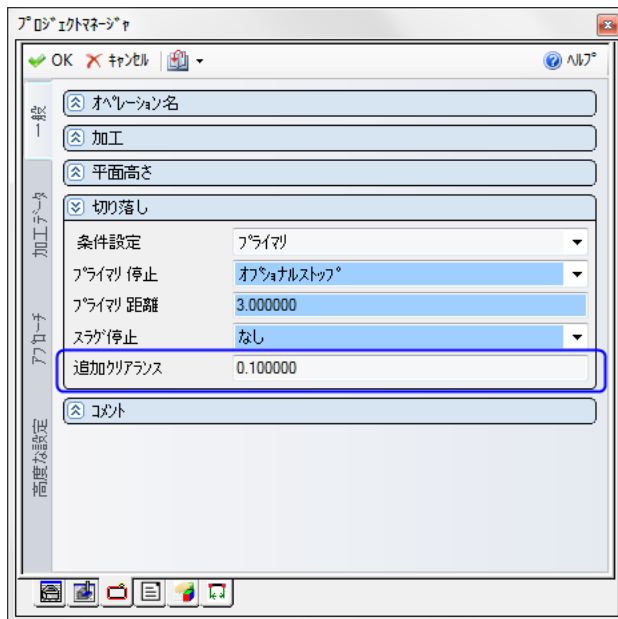
既に半径を持つ外部コーナーは、追加のクリアランスを足して修正されます。

切り落としの追加クリアランス

切り落としパスだけに追加のクリアランスを指定すると便利な場合があります。加工オペレーションのクリアランスだけでなく、切り落とし移動のみの追加のクリアランスを簡単に指定できるようになりました。

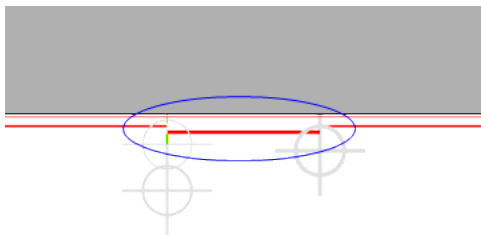
非切削パンチ (外部ブロック) または非切削ダイ (ガイド) のプロフィールの場合、メインフィーチャー (位置とオリエンテーション) は失わずにプロフィールが「破損」されることもあります。追加の研削の手間を省くため、切り落としパスのみに追加のクリアランスを指定すると便利な場合があります。

図 1. 切り落としのみに追加のクリアランスを指定できます。



加工条件設定に切り落としが含まれる場合、追加のクリアランスを加えたり取り除いたりできるのは、切り落としパスに対してのみです。

図 2. 切り落としに追加されたクリアランス。追加のクリアランスは全ての切削タイプ (オープン、ダイ、パンチ) に使用できません。



▶ [仕上げ 切り落とし] が [はい] に設定されている場合、追加のクリアランスは使用できません。

荒加工の半径を無視

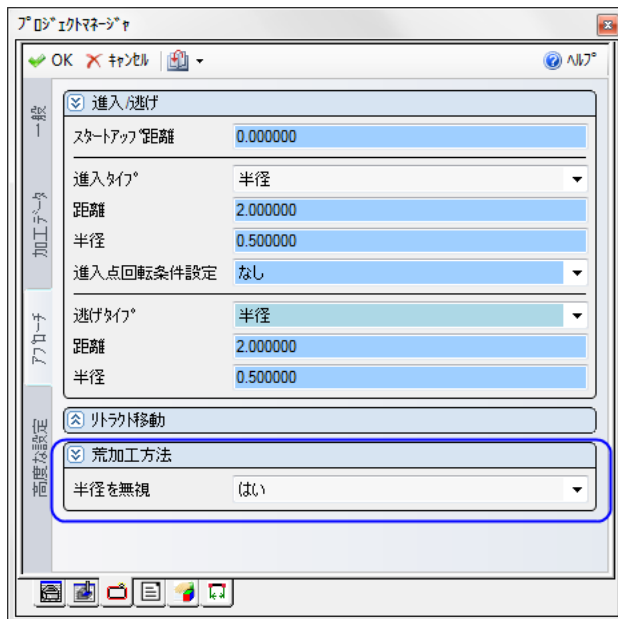
ダイに半径進入を使用する場合、ワイヤーの移動によって進入点に小さな突起 (不具合) ができることがあります。

この不具合を避けるには、荒加工パスの進入/逃げを、スキム加工パスの半径進入/逃げに影響を与えずに、直接移動に変換することができます。

ダイのプロフィールに次のいずれかの進入/逃げタイプが選択されている場合、新しいパラメータの [半径を無視する] を使用できます。

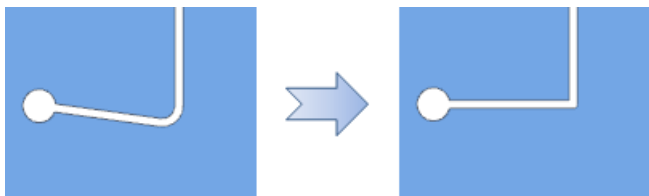
- ・ 半径
- ・ 垂直距離と円弧
- ・ 指定位置から円弧
- ・ 位置からの垂直距離/円弧

図 1. この新しい [半径を無視する] オプションは、2軸および4軸輪郭加工の [アプローチ] ページに追加されています。



[半径を無視する] を [はい] に設定すると、半径進入/逃げが、荒加工パスのみで直接进入/逃げに置換されます。その後のスキム加工には、加工オペレーションに定義されている半径進入が使用されます。

図 2. [半径を無視する] では荒加工パスのみに直接の進入/逃げが作成されます。



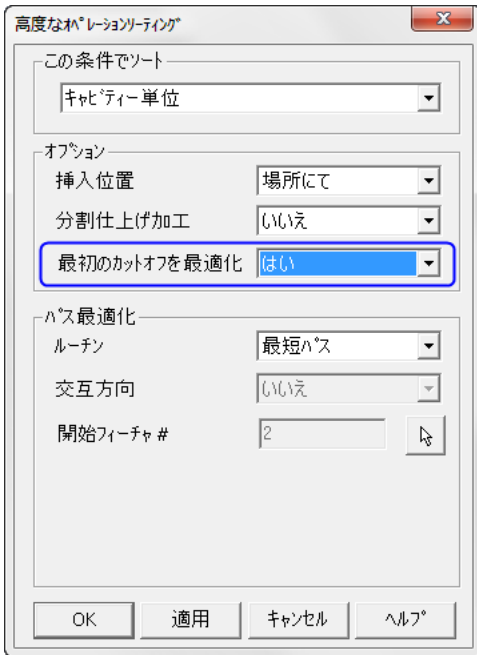
最初の切り落としの最適化

デフォルトでは、ワイヤー切削/ネジ切りの位置はワイヤーの再度ネジ切りが最小限になるように最適化されます。ただし、無人加工オペレーションをいくつかのパートに対して行う場合、自動最適化のためにマシンがネジ穴のない位置でワイヤーを再度ネジ切りしようとする場合があります。

そこで、[オペレーションの高度な並べ替え] に新しく追加されたオプションを使用して、ドリル加工済みのネジ穴がない位置ではワイヤーの再度ネジ切りを行わないように指定できます。

[最初の切り落としの最適化] という新しいオプションは、切り落としオペレーションの再度ネジ切りを、必ずドリル加工済みの穴がある初期位置で行います。ワイヤーパスは最適化されません。

図 1. [オペレーションの高度な並べ替え] に新しいオプション [最初の切り落としの最適化] が追加されました。



ポストプロセッサの新機能

ESPRITポストプロセッサのカスタマイズ、編集または開発を行う場合、ESPRIT 2016の新しい変更点を全て確認してください。

複数ファイルへの出力転送

ポストプロセッサごとに1つの出力ファイルを作成する代わりに、1つのポストプロセッサ内の出力を複数のファイルに転送できるようになりました。

この新しい機能にはさまざまな用途があります。

- ・ サブプログラムの出力時にプログラムごとに1つのファイルを生成する
- ・ 大規模な (3軸および5軸) 加工オペレーションの場合、オペレーションまたは工具ごとに1つのファイルを生成する
- ・ 放電加工用のテクノロジーファイルまたはコマンドファイルを生成する

BeginRedirect (FileNameを文字列で指定)

BeginRedirectは、ポストプロセッサの出力を指定された名前のファイルに転送する新しいキーワードです。

Ex_CycleStart : BeginRedirect (OperationFileName)

BeginRedirectは、他の命令を含まない個別の行に単独で指定する必要があります。FileNameにフルパスを指定しない場合、メインのNC出力ファイルと同じフォルダー内のファイルに出力されます。存在しないサブディレクトリがパスに含まれている場合、自動的にそのディレクトリが作成されます。

BeginRedirectは、MakeValidPathという新しい関数を実行して、FileNameから自動的に無効な文字を除外します。

- ▶ 同じファイルへの複数の転送命令がある場合、最初の転送によって既存のファイルが上書きされます。その後続く転送は、既に追加されている内容を上書きせずに、同じファイルに適用されます。

EndRedirect

EndRedirectは、NC出力の転送を停止する新しいキーワードです。出力は [NC出力ファイル] ダイアログで指定されている [NCファイル名] にリセットされます。

Ex_CycleEnd : EndRedirect

EndRedirectは、他の命令を含まない個別の行に単独で指定する必要があります。

MakeValidPath

MakeValidPathは、Windowsファイルパスで使用できない文字を自動的に削除する新しい関数です。MakeValidPathは、転送ファイルパスの有効性を保証するためにBeginRedirectキーワードによって自動的に実行されます。

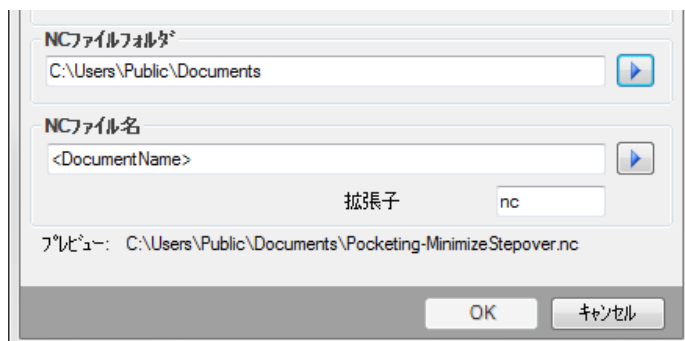
無効な文字の例:

- ・ < (より小さい)
- ・ > (より大きい)
- ・ : (コロン)
- ・ " (二重引用符)
- ・ / (前向きスラッシュ)
- ・ | (縦線)
- ・ ? (疑問符)
- ・ * (アスタリスク)

文字列としてのNCFileNameおよびNCFilePath

NCFileNameは、NC出力の定義からNCファイル名を返す新しいシステム変数です。

NCFilePathは、NC出力の定義からNCファイルフォルダーを返す新しいシステム変数です。



作成された全てのNCファイルの生成

Gp.exeは追加の引数を受け入れて、作成された全てのNCファイルのリストを返します。

コマンドラインから次を実行します。

Gp.exe C:\File.tmp Post.pst Output.nc FileName.txt

ヘリカル出力

ヘリカル出力が簡素化され、1より大きな回転を持つヘリカル移動を単一ブロックに出力できるようになりました。

新しいMaximumHelicalTurnsシステム変数

MaximumHelicalTurnsは、360度回転が何度許可されるかを設定します。たとえば、マシンによってはラインごとの回転数が99に制限される場合があります。

```
MaximumHelicalTurns      : 99
```

MaximumHelicalTurnsを使用すると、以下の設定を簡単に切り替えることができます。

- ・ ヘリカル出力なし (MaximumHelicalTurns=(0))
- ・ 最大360度のヘリカル移動 (MaximumHelicalTurns=(1))
- ・ 複数回転を持つ螺旋 (MaximumHelicalTurns=(99))

例

```
MaximumHelicalTurns      : 99
```

```
Ex_Setup                  : HelixOutput=(2)      ## 0=Helical motion interpolated with linear motions ; 1=Helical
motion up to 360deg ; 2= Full support with G2/3 P
                          : If (HelixOutput=0) MaximumHelicalTurns=(0)
                          : ElseIf (HelixOutput=1) MaximumHelicalTurns=(1)
                          : EndIf
```

新しい書式設定可能コード

AngleIncrementは、円移動の(終了角度 - 開始角度)です。1回より多く回転する場合、この値が360度より大きくなります。

```
AngleIncrement            : DA      N 4 Y N 3      N 4 Y N 3      1      1 N 0 0
```

HelicalFullTurnsは、ヘリカル移動内における360度回転の回数です。HelicalFullTurnsは、ヘリカル移動が360度を超える場合にのみ出力されます。

```
HelicalFullTurns         : TURNS= N 2 N N 0      N 2 N N 0      1      1 N 0 0
```

例

螺旋は回転数とは関係なく、Ex_Circle、Ex_CircleZX、Ex_CircleYZで出力されます。

```
Ex_Circle                 : IF (NextMisc(411)>0) ## HELICALFULLTURNS
                          :   "(Output Helix Commands)"
                          :   $OutputSequenceNumber WorkPlaneCode* CircleDirection* Absolute \\
                          :   XAbsolute YAbsolute ZAbsolute RotaryAxis1 RotaryAxis2 \\
                          :   XCenterAbsolute YCenterAbsolute ZCenterAbsolute \\
                          :   AngleStart AngleEnd AngleIncrement HelicalFullTurns \\
                          :   FeedUnit FeedRateCode
                          : ELSE ## regular circle
                          :   ...
                          : ENDIF
```

動的な配列

動的な配列 (1次元のみ) がサポートされるようになりました。配列を使用して、関連のある値のグループを保存できます。次の例では、固定サイズ (99、下限は 0) の配列を使って工具IDを保存します。このIDは後で検索して、干渉チェックを行い次の工具のIDを出力するために使用できます。

```
Variable          : ToolID(99) ToolCount TotalToolCount

ex_FirstToolChange : ToolCount=(ToolCount+1)
                   : OutputString("(CHANGING TO " + ToolID(ToolCount) + ")")
                   : T* M06*
                   : If (ToolCount < TotalToolCount)
                   :   OutputString("(Next TOOL IS " + ToolID(ToolCount+1) + ")")
                   :   T*(NextTool)
                   : Else
                   :   OutputString("(THIS IS THE ONLY TOOL USED)")
                   : EndIf
```

配列のサイズは動的に設定することが可能です。

```
Variable : MyArray1(*)      ## declares a dynamic array with a lower bound of 0

Variable : MyArray2(-4..*) ## declares a dynamic array with a lower bound of -4
```

動的配列に値を設定するには

```
Ex_... : MyArray1(4)=(52.1)  ## If array is not big enough, it is resized
        : MyArray2(-1)=(12.5) ## If array isn't big enough, it is resized
```

新しいLBound関数とUBound関数

Lbound関数とUBound関数は、配列のサイズを決定するために使用されます。

- ・ UBound関数を使用して、配列の次元の上限を見つけます
- ・ LBound関数を使用して、配列の次元の下限を見つけます

例:

```
Variable : MyVar

Ex_...   : MyVar=(LBound(MyArray1))
          : MyVar=(UBound(MyArray1))
```

新しいMinimum関数とMaximum関数

Minimum() とMaximum() は、引数として配列 (1つまたは複数) を受け取ることができます。動的な配列の場合、初期化されていない配列要素は計算から除外されます。固定サイズの配列である場合、配列の作成時に全ての要素が0.0に初期化されます。

例:

```
Variable : FixedArray(5) ## declares a fixed size array with a lower bound of 0
          ## all values initialized to 0.
          : DynArray(*) ## declares a dynamic sized array - initial size is 0.
          ## no values are ever initialized.

Variable : MyVar

Ex_... : FixedArray(0)=(12.4)
        : FixedArray(1)=(15.7)
        : FixedArray(4)=(9.3)

        : DynArray(0)=(12.4)
        : DynArray(1)=(15.7)
        : DynArray(4)=(9.3)

        : MyVar=(Minimum(FixedArray)) ## MyVar will be 0 - because all array elements were
          ## initialized to 0.0
        : MyVar=(Minimum(DynArray)) ## MyVar will be 9.3 - because no elements are
          ## pre-initialized.
```

実行時における値の変更

ポストプロセッサの実行時にマシンモード、書式設定可能コード、およびシンボリックコードの値を変更できるようになりました。

実行時におけるマシンモードの値の変更

例:

```
Ex_BeforeCycle: RPYAngleMode(YawPitchRoll)
```

SetSymbolicCode

実行時にシンボリックコードのシンボルを変更します。

構文:

```
SetSymbolicCode(シンボリックコード,"新規コード")
```

例:

```
SetSymbolicCode(SpindleCW, "M203")
```

SetFormatableSymbol

実行時に書式設定可能コードのシンボルを変更します。

構文:

```
SetFormatableSymbol(書式設定可能キーワード, シンボル)
```

例:

```
Ex_CycleStart : SetFormatableSymbol( XAbsolute , "XHP" )
```


SetFormatableInch、SetFormatableMetric、SetFormatableZero

インチ、メートル、およびゼロ値の出力における整数部および小数部の桁数を制御します。

構文:

SetFormatableInch (書式設定可能キーワード, 整数部桁数, 小数部桁数) SetFormatableMetric (書式設定可能キーワード, 整数部桁数, 小数部桁数) SetFormatableZero (書式設定可能キーワード, 整数部桁数, 小数部桁数)

例:

```
Ex_Setup : SetFormatableInch( XAbsolute , 4 , 6 )
```

SetFormatableMaximum、SetFormatableMinimum

書式設定可能コードで出力される値を制限します。書式設定可能コードの値が最大値を上回る場合、代わりにこの最大値が出力されます。

SetFormatableMinimumは、同様の方法で書式設定可能コードに最小値を設定します。

構文:

SetFormatableMaximum(書式設定可能キーワード, 最大値) SetFormatableMinimum(書式設定可能キーワード, 最小値)

例:

```
Ex_Setup : SetFormatableMaximum(InverseTimeFeed, 9999)
```

その他の新しいキーワード

ESPRIT 2016では、新しい機能強化が導入されただけでなく、新しいキーワードもいくつか追加されています。

ポストプロセッサの新しいキーワード:!!による出力の抑制

多くの場合、工具のスピンドル軸では軸が移動する必要がありますが、その動きはNCコードで明示的に指定されません。

軸がリンク動作の一部である場合、その位置の出力を省略しないでください。代わりに新しい抑制出力フォーマットを使ってその位置をポストします。

「!!」は特殊なキーワードで、下記の例のセクションで使用されるシンボリックコード、シンボリックスイッチ、書式設定可能コード、および書式設定可能スイッチに適用されます。

例:

```
AAxis          : C N 4 Y      N 3 N 4      Y N 3 1      1 N 0 0
```

```
CoolantOff     : M9
```

```
Ex_Before_Cycle : C!(0) ## This is the same as: NCOOutputOff C(0) NCOOutputOn
```

```
Ex_ToolCancel  : M9!   ## Coolant off code is not posted
```

ForceMisc

ForceMiscは新しいシステム変数です。ForceDimに似ていますが、ForceMiscはソフト例の出力を強制しません。

ForceMiscは、書式設定可能な各種登録値インデックス番号を引数に取ります。すると、その登録値に対応する書式設定可能な各種データが、値が変更されておらず、通常その出力がアスタリスク(*)で強制されなかった場合でも、次回の出力時に強制されます。

例: 工具変更の後で軸1のクランプコードの出力を強制します。

```
ForceMisc (401)=(1)
```

EspritVersion

EspritVersionは、ESPRITドキュメントのビルド番号を返す新しいシステム変数です。バージョンはNCコードで文字列として出力できます。

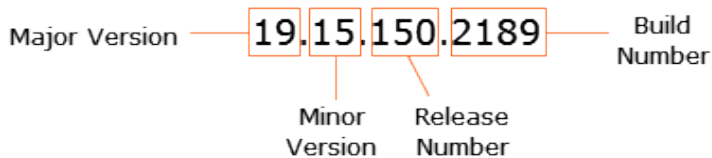
```
OutputString( "V(" + EspritVersion + ")" )
```

例:

```
Ex_StartCode          : ProgramNumber*  
                      : "(ESPRIT BUILD: " EspritVersion ")"
```

結果:

```
(ESPRIT BUILD: 19.15.150.2189)
```



FeedrateMode: 「距離/分」と「度/分」

RTCPまたは逆時間送り速度なしで回転軸の送りを行う際、NCコードの送り速度値は、大半のマシンでは「度/分」の度単位による角速度として解釈されます。NCコード用にF値が再計算されない場合、マシンに非常に悪い影響を及ぼすことがあります。

ESPRITで、新たにNCコード用のF値を計算できるようになりました。したがって、オペレーションページでプログラムされた送り速度は、工具の移動方向の切削速度に対応します。

このモードを使用する場合、回転軸が移動ブロック内で移動するたびに、NCコードの送り速度値がオペレーションページのプログラムされた値に一致なくなります。

新しいマシンモードのFeedrateModeは、送り速度値の出力方法を次のように制御します。

- ・ FeedrateModeがDistancePerMinuteに設定されている場合、特に計算は行われません。NCコードの送り速度は、オペレーションページのプログラムされた送り速度に一致します。NCコードの送り速度値は、常にパートに対する工具の線形送り速度です。
- ・ FeedrateModeがDistanceAndDegreePerMinuteに設定されている場合、ESPRITは次の式によってNCコードの送り速度値を計算します。回転軸が単独で移動する場合、出力値の単位は「度/分」です。
- ・ FeedrateModeが設定されていない場合、デフォルトでDistancePerMinuteが使用されます。

FeedrateMode = DistanceAndDegreePerMinuteの送り速度計算式

(1) を求めます: $f = DL / T$ (1)

ここで、

- ・ f (mm/分 | インチ/分) = 切削送り速度
- ・ DL (mm | インチ) = パートの切削長。これはESPRITでツールパス項目の長さです
- ・ T (s) = 切削時間

(2) を求めます: $F = L / T$ (2)

ここで、

- ・ F = NCプログラムでの送り速度
- ・ L = 制御装置の移動長 (パートの切削長とは異なります)

次を求めます: $L = \sqrt{DX^2 + DY^2 + DZ^2 + DA^2 + DB^2 + DC^2}$ (3)

ここで、

- ・ DX, DY, DZ (mm | インチ) = 軸ごとの移動長
- ・ DA, DB, DC (度) = 回転軸ごとの角度置換

(1) と (2) を組み合わせます:

$F = f * L / DL$

Fは、送り単位が (mm/分 | インチ/分) に設定されている場合、ESPRITがポストに書式設定可能コードの送り速度としてダンブする送り速度値です。

直線軸のみが移動した場合は $(L/DL) = 1$ および $F = f$ です。

回転軸が単独で移動する場合、Fの値は「度/分」となります。

InverseTimeFeedUnit

InverseTimeFeedUnitは、書式設定可能コードInverseTimeFeedで出力される値の単位を制御する新しいマシンモードです。

- ・ InverseTimeFeedUnitがInverseTimeFeedPerMinuteに設定されている場合、InverseTimeFeedの値は、その動きの完了にかかる時間の逆数 (単位:分) です。
- ・ InverseTimeFeedUnitがInverseTimeFeedPerSecondに設定されている場合、InverseTimeFeedの値は動きの完了にかかる時間の逆数 (単位:秒) です。
- ・ InverseTimeFeedUnitが設定されていない場合、デフォルトでInverseTimeFeedPerMinuteが使用されます。

GpAbort

GpAbortは、GPの実行を中断する新しい関数です。ポストが関数を読み取る際、GPの実行が中断されてESPRITにメッセージボックスが表示されます。OutputString関数と同様、メッセージは引数として入力します。引数には値、変数、変数の配列のいずれかを指定できます。

更新されたキーワード

ESPRITの変更点のため、いくつかのキーワードが更新されました。

更新されたポストプロセッサキーワード:シンボルの代わりに使用するポストキーワード

例に使用されているネイティブの書式設定可能/シンボリックコードをポストで使用できるようになりました。

現在でも、

```
Ex_Linear : G1 X Y Z B C F
```

のような例を記述できますが、ポストで新たに次を受け入れられます。

```
Ex_Linear : MotionLinear XAbsolute YAbsolute ZAbsolute BAxis AAxis Feedrate
```

ForceDimを使用した座標の出力のテスト

ForceDimはポストプロセッサの新しい関数ではありませんが、この関数を軸位置がリンク動作で出力されるかどうかを検出する条件として使用できるようになりました。

リンクの最初のZ動作で長さ補正の出力を取得することが必要になる場合がよくあります。

従来は、長さ補正が有効かどうかを検出するフラグを設定したうえで、DREGDIFF(3) が真の場合には長さ補正コードがEx_Link_Rapid_Positionに出力されていました。

リンクの構成内では、座標が変更されていない場合でもその出力を強制することができます。Zが強制され、しかも変更されない場合には、DREGDIFF(3) のテストは機能しません。また、変更を行うために、ポスト内でZ座標を変更する方法も望ましくありません。しかし、この問題を解決する方法があります。

Ejemplo:ForceDimを使用して軸座標が出力されるかどうかを判定する方法

例:

```
Ex_BeforeCycle          : LengthCompEnabled=0
```

```
Define OutputLengthComp \\  
  If (LengthCompEnabled=0 && millturncode=millcycle) If (FORCEDIM(3)) G43* H* LengthCompEnabled=(1) EndIf  
EndIf \\  
EndDefine
```

```
Ex_Link_Rapid_Position  : G G0 $OutputLengthComp G90 X Y Z B C
```

SpindleUnitとFeedRateCode

SpindleUnitが、工具回転のある全てのオペレーションでRPMに内部設定されるようになりました。SpindleUnitは、前回のオペレーションのスピンドル単位とは関係なく、正しいスピンドル単位を返します。

ミル加工、旋盤加工、インバース時間、値の強制/非強制のためのロジックを追加しなくても、FeedRateCodeが送り速度値の出力を強制するようになりました。

FeedRateCodeは、単位が変更されると送り速度値の出力を強制します。

- ・ G94 F500. (書式設定可能コードFeedRate)
 - ・ G95 F0.1 (書式設定可能コードChipFeedRate)
 - ・ G94 F500 ← 単位変更のため送り速度値が強制されます。以前のバージョンでは、書式設定可能コードFeedRateの値が前のオペレーションから変化しないので、値は強制されませんでした。
- ▶ InverseTimeFeedには例外が1つあります。InverseTimeFeedの値は、2つのブロック間で値が変化しない場合でも、常に強制的にインバース時間モードで出力されます。

新しい数学関数

新しい数学関数がいくつか追加されました。

ポスの新しい数学関数:Quotient (商)とRemainder (剰余)

Quotient (商) は、除算の結果得られる整数です。たとえば、 $40 \div 5$ を計算した結果の商は8です。

Remainder (剰余) は除算で割り切れなかった余りです。たとえば、 $43 \div 5$ を計算した結果の商は8、剰余は3となります。

QuotientとRemainderには、値と除数の2つの引数をコンマで区切って指定します。

構文:

Quotient(値,除数) Remainder(値,除数)

例:

```
: OutputString( Quotient( 3.141 , 2 ) ) OutputString( Remainder( 3.141 , 2 ) )  
: OutputString( Quotient( -3.141 , 2 ) ) OutputString( Remainder( -3.141 , 2 ) )  
: OutputString( Quotient( 3.141 , -2 ) ) OutputString( Remainder( 3.141 , -2 ) )  
: OutputString( Quotient( -3.141 , -2 ) ) OutputString( Remainder( -3.141 , -2 ) )
```

結果:

```
1 1.141 -1 -1.141 -1 1.141 1 -1.141
```

NormalizeAngle

NormalizeAngleには、角度と下限角の2つの引数をコンマで区切って指定します。

最初の引数は正規化の対象となる角度です。2番目の引数は、正規化の基準となる下限角度です。この関数は、下限角度から、下限角度に360を加えた値未満の範囲にある角度を返します。したがって、NormalizeAngle(360,0)の結果は0となります。

構文:

NormalizeAngle(角度,下限角度)

例:

NormalizeAngle(800,0) は、80を返します。NormalizeAngle(800,100) は、440を返します。

ClosestAngle

ClosestAngleは、 $\| \text{角度} - \text{基準角度} \| \leq 180$ であるような、基準角度に対して最も近い正規化された角度を返します。 $\| \text{角度} - \text{基準角度} \|$ のいずれかが180である場合、「基準角度+180」の角度が返されます。

構文:

ClosestAngle(角度, 基準角度)

例:

ClosestAngle(800,0) は、80を返します。ClosestAngle(800,100) は、80を返します。

MinimumとMaximum

Maximumは、値のリストから最も大きい値を返します。Minimumは、値のリストから最も小さい値を返します。

引数の数は制限されません。各引数には値、変数、変数の配列のいずれかを指定できます。

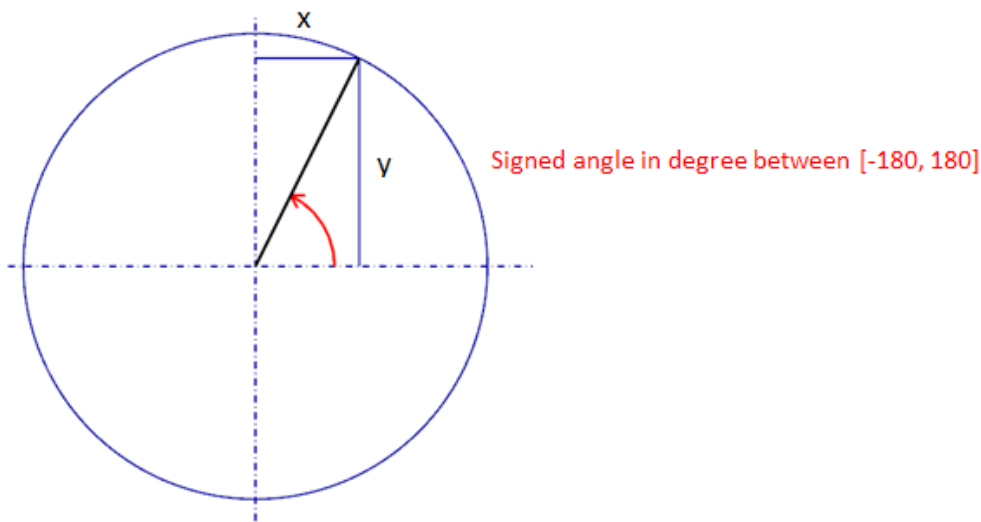
例:

Minimum(-10,80,-60,500) は-60を返し、Maximum(-10,80,-60,500) は500を返します。

Atan2

Atan2は、2つの引数を取る逆正接関数です。引数を1つではなく2つ使用する目的は、計算された角度の適切な象限を返すために、入力 of 符号に関する情報を集めることです。引数が1つのATN関数では、象限を返すことができません。

Atan2は、-180から+180度の間の値を返します。



構文:

Atan2(倍精度y, 倍精度x)

例:

Atan2(20,10)= 63.435 Atan2(-20,10)= -63.435 Atan2(-20,-10)= -116.565

ポストプロセッサの定数

ポストプロセッサには、内部宣言された定数が含まれています。これらはポストプロセッサ内でロジックを記述する際に使用できます。

次に、使用できる定数のリストを示します。

定数	値
PI	3.141592653589793238463
RadianToDegrees	57.2957795130823208768
DegreeToRadians	0.01745329251994329576924
InchToMm	25.4
MmToInch	0.03937007874015748031496