

リリースノート

バージョン 6.4.0

2009年1月16日

株式会社東陽テクニカ
ソフトウェア・ソリューション

文書バージョン 1.00

1. はじめに

本書では、Windows 2000/XP(x86)/XP(x64)/Vista(x86)/Vista(x64)、Solaris 10(SPARC)、HP-UX 11i(PA-RISC)、Redhat Enterprise Linux 3(x86) 用の Imagix 4D バージョン 6.4.0 について記述してあります。本書には、ソフトウェアをシステムにインストールする前にお読みいただきたい重要な情報が含まれています。

2. サポートするプラットフォームおよびオペレーティングシステム

Imagix 4D バージョン 6.4.0 がサポートするプラットフォームおよびオペレーティングシステムは、以下の通りです。

- ・ x86 プロセッサ上で動作する Windows 2000/XP/Vista
- ・ x64 プロセッサ上で動作する Windows XP x64 Edition/Vista x64 Edition
- ・ SPARC プロセッサ上で動作する Solaris 10
- ・ PA-RISC プロセッサ上で動作する HP-UX 11i
- ・ x86 プロセッサ上で動作する Redhat Enterprise Linux 3 (RHEL 3)

3. 使用上の注意点

Imagix 4D バージョン 6.4.0 を日本語環境でご利用頂く場合の注意事項についてご説明します。

- ・ 日本語のディレクトリ名に保存されているソースファイル、あるいは、フルパス名のいずれかに日本語のディレクトリが含まれるソースファイルを解析することができません。
(回避策) 英数字のディレクトリにコピーをとって、コピーされたファイルを使って、解析を行ってください。
- ・ プロジェクト名に日本語を利用することができません。
(回避策) プロジェクト名には、英数字を使用してください。
- ・ プロジェクトを日本語のディレクトリに作成すると、解析時にエラーになります。
(回避策) プロジェクトは、日本語を含まないディレクトリに作成してください。
- ・ 解析および表示をサポートしている日本語のエンコーディングは、Windows ではシフト JIS、Solaris、HP-UX、RHEL 3 では EUC のみです。これら以外のエンコーディングのソースファイルを解析する際は、事前に、サポートされているエンコーディングに変換してから解析を実施してください。
- ・

4. Imagix 4D リリースノート 6.4.0 - 5.4.5 からの変更点

表記法

当リリースノートは一連のセクションに分けられており、各セクションではツールの各分野における変更点を説明しています。変更点は、重大度に応じて2つのカテゴリに分けられます。カテゴリ/重大度はインデントのレベルにより次のように示されます。

重要な新機能/変更

その他の変更および修正

バージョン番号が記述されていない場合、変更は最初の 6.0 リリースで導入されたものです。

C/C++ソースコードアナライザ

-encmac オプションを追加しました。このオプションは外側のマクロ展開のどこかで展開されているマクロについて、複数行からなるマクロ定義での行番号を追跡します。これらは、マクロ同士の相互参照の構成に使用されます。これによってマクロ定義本体の中が色付けにより強調され、Function Calls with Macros ビューからその場所にナビゲートすることができます。(C627 - 6.0.0)

123ull のような整数リテラルを符号なしの long long 型リテラルとするサポートを追加しました。(C654 - 6.0.0)

パラメータ宣言において多数の同じ名前が使用されているソースコードでのアナライザの速度が大幅に向上しました。この現象が起きる代表的な例は Microsoft のインクルードファイルです。(C656 - 6.0.1)

GNU C 拡張機能の __gnuc_va_list を可変パラメータのリストに追加しました。(C667 - 6.0.0)

extern "C" 変数宣言における型名の省略が可能になりました (Microsoft 拡張機能)。(C670 - 6.0.1)

for ループに対して生成される制御フロー情報が改善され、静的なバウンドがあるためにループに 1 回以上進入していることを Imagix 4D データフローアナライザに伝えるようになりました。(C671 - 6.0.2)

sizeof で括弧を付けずに単純な型名を使用可能とする Microsoft 拡張機能を追加しました。例：s = sizeof mytype;(C680a - 6.1.0)

アドレスによる共用体の定義を可能とする IAR コンパイラ拡張機能を追加しました。例：union {int a; float b;} @ 0x1234;(C684 - 6.1.0)

-uaggr オプションを追加しました。このオプションは同じ名前を持つ最上位レベルのクラスの出現ごとに一意の相互参照情報を生成し、実質的にそれらを異なるクラスにします。-uaggr オプションにより、同じ名前を持つクラスを別個のものとして扱うべき箇所での名前の衝突を避けることができます。これは、複数のプログラム、DLL、または共有ライブラリにまたがっている Imagix 4D プロジェクトに対応しています。一般に、これらを別々のプロジェクトに入れること、そして Imagix 4D のスーパープロジェクト機能を使用してグローバル・ビューを取得することが推奨されます。(C689 - 6.1.0)

-trkarray オプションを追加しました。このオプションは静的な配列要素の相互参照情報を追跡します。これらの要素は Imagix 4D のディスプレイにおいて array[1] などの別々の変数として表示され、配列変数に含まれません。(H07-5 - 6.2.0)

システムインクルードファイルに定義されている型について、追加の相互参照情報を作成しました (システムインクルードファイルに関して他のデータが収集されていない場合)。これにより、Imagix 4D はより完全に型の関係をグラフに表示できるようになりました。以前はこれらの型はデータベースから省略されていました。(C692 - 6.2.1)

関数内部の#include からのコードを Imagix 4D のファイルエディタにおいてタグ付けできるように、追加の相互参照情報を作成しました。(C713 - 6.2.0)

-msrcfx オプションを追加しました。このオプションは、インクルードファイルが複数回インクルードされ、その都度異なるソースコードを生成する場合に、追加の相互参照情報を生成します。このオプションは通常、考慮されるすべての条件パスに対してデータを生成するオプションと共に使用されるでしょう。これは相互参照情報を増大させることがあり、異なる定義設定に基づいて新しいコードを生成するためにインクルードファイルを故意に複数回インクルードする場合にのみ意義があるため、注意して使用すべきです。(C719 - 6.2.1)

ISO C 言語 1999 標準規格に従って、__VA_ARGS_の取り扱いを追加しました。(C730 - 6.2.1)

Microsoft 拡張機能の dllexport および dllimport を追加しました。クラスの名前が複数の DLL 間で同一であることを強制するので、-uaggr(C689)と共に使用される場合に、特に有用です。(C734x - 6.3.0)

123ui8 や 123i16 など、型に固有の整数リテラルのための Microsoft DDK 拡張機能を追加しました。(C746 - 6.3.0)

ラベルおよびラベル式のアドレス (計算された'goto') に対応した、GNU C/C++拡張機能を追加しました。(C758 - 6.3.1)

extern "C"の前で Microsoft declspec を使用可能とする Microsoft 拡張機能を追加しました。(C761 - 6.3.1)

テンプレートパラメータに依存する typename を解析する際の、未定義の typename に関する誤った警告を除去しました。(C631 - 6.0.0)

未定義のマクロ使用についての相互参照情報を除去しました。そのようなマクロ名が後に異なって定義される可能性があるため、そのような相互参照情報は誤解を招きかねません。(C655b - 6.0.0)

2 つの C++の関数が同一であることを解決するために使用されるパラメータのチェックを変更しました。現在では、パラメータ型はすべて潜在型に削減されています。以前は、中間の typedef が使用可能でした。また、これは C++関数に対して生成された相互参照データベース内の改変された名前を変更し、完全なパラメータ型が Imagix 4D のディスプレイに表示される場合にユーザから可視となります。(C655 - 6.0.0)

インクルードファイルと同じ名前を持つディレクトリがあった場合に発生していた、「再オープンできない」というエラーメッセージを除去しました。(C666 - 6.0.0)

クラス名または型名の解決を修正し、後に中間の名前空間のエイリアス(例: namespace n2 = n1; using namespace n2;) を付けられるようにしました。(C673 - 6.0.1)

クラス内の無名の共用体メンバを、それを囲むスコープ内で可視にしました。(C673 - 6.0.1)

マクロ展開における一致しないパラメータに関する誤った警告を除去しました。(C680c - 6.1.0)

型名の解決を修正しました。コンテキストで型名が必要とされている場合、ルックアップを型名に限定することにより、それ以外の方法では隠される型名が認識されるようにしました。(C680d - 6.1.0)

クラスの解決を修正しました。基底クラス名が関数または変数によって隠されている場合、コンテキストによりそれがクラスでなければならないと判定されるため、依然として解決の必要があります。(C681 - 6.1.0)

#ifdef ディレクティブでのマクロ使用箇所を示す行番号の情報を修正しました。(C682 - 6.1.0)

インクルードファイルからの 1 行からなる関数が制御フロー情報に複数回記録されないようにし、フローチャートの表示が Imagix 4D で正しく動作するようにしました。(C683 - 6.1.0)

複数個のファイルに広がった構文エラーに関する行情報を改善しました。(C694 - 6.1.0)

シンボル型の解決を修正しました。C++の場合、クラス名が同じ有効範囲内の同名の変数、関数、または列挙子によって隠蔽されるようにしました。また、enum タグ名が相互参照

情報において同じ有効範囲内にある変数名とは異なるようにし、Imagix 4D がそれらを別々に表示できるようにしました。(C697 - 6.1.0)

クラスメンバのテンプレート関数の識別方法を修正しました。例：
object.templatefunction<int, int>(x, y);(C701a - 6.2.0)

展開を行う前に一連のマクロ再定義が発生する、特定の状況下でのマクロ展開を修正しました。(C701b - 6.2.0)

再定義されたクラスメンバがトークン並び-> Classname <... にかかわる場合、テンプレートのインスタンス化がソースコードの解釈の 1 つであるとみなされるように解析を修正しました。(C701c - 6.2.0)

C++においてネストされた extern "C"コードに現れる関数に対して正しいリンケージ名が生成されるように、解析を修正しました。この動作は、これらの関数が C の関数本体と正しく相互参照されるために必要です。(C708 - 6.2.0)

構文エラーが見つかった後での不要なエラーメッセージのカスケード表示を減らしました。(C711 - 6.2.0)

sizeof classname の直後に for 文が続く箇所での誤った構文エラーを修正しました。(C715 - 6.2.1)

function<T1>(exp, constructor<T2>(var)); のような、特殊化されたテンプレート関数のネストされた呼び出しに関する解釈を修正しました。(C726 - 6.2.1)

どのファイルが関数ポインタを割り当てているかを追跡するための相互参照情報を作成しました。(C729 - 6.2.1)

システムインクルードファイル内のマクロの再定義であるマクロについて、マクロ情報を作成しました(システムインクルードファイルに関して他のデータが収集されていない場合)(C733 - 6.3.0)

初期化子の関数名の最初にある余分な"*"に対する制御フロー情報を修正しました。これにより、データフローレポートでこのようなケースが正しく処理されます。(C739- 6.3.0)

ある 16 進リテラルがマクロと混在している箇所でのマクロ展開を修正しました。(C749 - 6.3.0)

ネストされた再帰的な型に対する sizeof の計算のパフォーマンスを最適化しました。(C763 - 6.3.1)

変換であるか型仕様であるかがあいまいな int (* const... の解決を改善しました。(C767 - 6.3.2)

struct 型の前方宣言をインクルードするパラメータ型に関する相互参照情報を修正しました。(C770 - 6.3.2)

wchar_t を明示的な事前定義の型として Imagix 4D の相互参照情報に追加しました。これにより、wchar_t 型であるプログラム要素のユーザ検索が可能となりました。(C775 - 6.4.0)

プリプロセッサディレクティブにより生成されたソースコードの省略部分の始点と終点を調整しました。現在、囲んでいる#ディレクティブ行は解析対象とみなされます。これにより、ファイルエディタにおいてグレーで影を付けられる行が変化します。(C777 - 6.4.0)

テンプレート実引数の終端についての識別方法が修正されました。以前は、ある種の複雑かつネストされたテンプレートのインスタンス化では、解析においてテンプレート実引数の終端が不明となることがありました。これによって"abandoning template arguments" という偽のエラーメッセージと、後続するいくつかの副次的なエラーが発生していました。(C779 - 6.4.0)

その他のデータ収集

../imagix/user/cc_cfg ディレクトリには、gnu gcc コンパイラの様々なバージョンに適用可能な.inc コンパイラ構成ファイルがいくつか含まれています。gcc コンパイラは構成可能な範囲が広いので、これらの.inc ファイルでさえもユーザの環境に合わせて構成する必要があります。コンピュータの環境固有のビルド向けに最適化された.inc ファイルを自動的に作成する、新しいユーティリティの imagix-gnu が追加されました。(6.3.0)このユーティリティが特定の Windows 環境に対応するために修正されました。(6.3.1)

ソースアナライザのエラーや警告メッセージに関する技術サポートとのやり取りを容易にするために、メッセージとそれらが関連するソースコードと基盤となる Imagix 4D の環境に関するさまざまな情報を 1 つにまとめる関数が追加されました。このまとめられた情報は Imagix 社または販売代理店の技術サポートに送ることができます。この関数には、Analyzer Results ウィンドウの File ->Export...メニューからアクセス可能です。情報は../your_project.4D/analyzer_log ファイルに書き込まれます。2つのバージョンのログファイルを作成できます。デフォルトのバージョンでは実際のソースコードを示し、マスクされたバージョンではシンボル名がすべて一般的な名前に置き換えられます。(6.1.0)

gnu_xxxx.inc 群を改善しました。(6.0.2)

MSVC Workspace のデータ収集アプローチを使用する際に特定のワークスペース/ソリューションの構成(Debug や Release など)を選択する機能を追加しました。(6.1.0)

MSVC Project または MSVC Workspace のデータ収集アプローチを使用する際の/F1 オプションのサポートを追加しました。(6.1.0)

msvc_win.inc 内の MSVC バージョン番号を修正しました。(6.1.0)

arm_cross.inc を改善しました。(6.1.0)

新しい iar_arm.inc を追加しました。(6.2.0)

iar_8051.inc、iar_avr.inc、iar_cr16c.inc、および iar_hc12.inc を改善しました。(6.2.0)

IBM Visual Age コンパイラ用に新しい ibm_aix.inc を追加しました。(6.2.1)

MSVC Project のデータ収集アプローチを使用する際の.dsp パス名におけるスペースのサポートを修正しました。(6.2.1)

metrowerks_hiware.inc を改善しました。(6.2.1)

Windows 環境の特定の条件下で、解析結果を analyzer_log ファイルにエクスポートできないという問題を解決しました。(6.3.0)マクロ展開が失敗するという問題を解決しました。(6.3.1)

新しい freescale_hcxx.inc を追加しました。(6.3.0)

gnu_xxxx.inc 群を改善しました。(6.3.1)

Windows DDK のサポートを msvc_win.inc に追加しました。(6.3.2)

MSP430 を ti_cross.inc に追加しました。(6.3.3)

MSVC Project または MSVC Workspace のデータ収集アプローチを使用する際、Visual Studio のプロジェクト定義において Exclude From Build の設定が可能になりました。(6.3.3)

プロファイルデータのインポートの際に、すでに無効となったデータの表示および進捗バーをクローズする処理を修正しました。(6.3.3)

統合されたディスプレイウィンドウ

Imagix 4D の 6.x バージョンにおいて最も際立った変更点は、以前のユーザインターフェース表示の大部分が 1 つのディスプレイウィンドウに統合されたことです。

File Editor、Flow Chart、Calculation Tree、Metrics Tool、および様々な Report など、以前は固定されていなかったディスプレイの多くが従来の Graph ウィンドウと結合され、統合ディスプレイウィンドウの Main パネルが形成されました。

また、従来の Project パネルおよび Symbol パネルも拡張されました。それらのパネルには、File Browser、Class Browser、Use Browser、Grep Tool など以前は固定されていなかった他のディスプレイのいくつかが含まれています。さらに、新しい機能も導入されています。

ディスプレイを一箇所にまとめるほか、新しい Main パネルでは同種のウィンドウの複数インスタンス（10 個まで）をサポートしています。例えば、最大 10 個の Graph、最大 10 個の Flow Chart、および最大 10 個の Calculation Trees を同時に開くことが可能になりました。以前は、複数インスタンスは File Editor においてのみサポートされていました。

デフォルトでは、様々なディスプレイの複数のインスタンスがノートウィジェット内にタブとして現れます。ノートは、横並びまたは縦並びの 2 つのノートに分割可能です。ノートが 2 つに分割されているとき、個々のタブ付きディスプレイを一方からもう一方へとドラッグすることができます（タブ名を使用）。

MDI ディスプレイメカニズムは、タブ付きのノートに代わる手段として利用可能です。

ノートまたは MDI による方法のどちらかを使用して、実際のディスプレイを切り離し、統合ディスプレイウィンドウとは別に表示させることができます。

この変更の一部として、あるディスプレイに固有のメニュー項目はそのディスプレイに移動したため、メインメニューの一部ではなくなりました。従来のメニュー項目のほとんどは、最終的に Graph ウィンドウに組み込まれました。

ツール全体にわたってマウスとキーボードの割り当てが変更され、より直感的で一貫性のあるものとなりました。マウスの右クリックによるコンテキストセンシティブメニューの項目は、追加のディスプレイウィンドウに対応して拡張されました。

Graph ウィンドウ

統合ディスプレイウィンドウの変更の一部として、それぞれ独自のメニューを持つ 10 個の Graph ウィンドウを同時に表示させることが可能になりました。その時々で、これらのうち 1 つがアクティブな Graph ウィンドウとみなされます。アクティブウィンドウでは、強調されるシンボルが黄色で表示され、他のウィンドウではそれらが白で表示されます。Graph ウィンドウを使用する際は常に、それがアクティブウィンドウになります。

「モード」の概念を Graph ウィンドウから削除しました。以前は UML Diagram モードおよび Control Flow モードで提供されていたグラフディスプレイは、単にビューとして View メニューにより提供されるようになりました。(6.4.0)

多くのグラフィックディスプレイおよび従来の Browse モードの動作は、現在は Symbols パネルの Cross Reference タブから使用できます。

ビュー可能なシンボル型および関係型を選択する主要なメカニズムは、従来の Set View ダイアログから、View メニューの下にある一連の事前定義のビューに変わりました。Set View ダイアログは現在、メニュー項目 Other... から利用可能です。カスタムビューでの名前空間の包含をクラスとは別に制御できるように、新しいシンボル型が追加されました。Set View メニューから定義されたビューを保存して、事前定義ビューのリストに追加することができます。

6 つの標準の事前定義ビューが追加されました。

File Calls はソースコード内のファイルおよびそれらの間の呼び出し関係を示します。これらの呼び出しはファイルメンバ間の呼び出しが抽象化されたものです。例えば、FileA.c にある FuncA が FileB.c にある FuncB を呼び出す場合、FileA.c は FileB.c を呼び出すとみなされます。この機能は、制御フローの観点からどのファイルが他のどのファイルに関係しているかを診断する際に役立ちます。(6.4.0)

Namespaces and Classes は名前空間とクラスとの間にあるコンテナと継承関係を示します。以前は、名前空間はクラスおよびテンプレートと組み合わせて様々なビューに表示されていました。現

在は名前空間は別に扱われ、このビューにのみ表示されます。このビューは名前空間の内容が理解できるように最適化されます。

Class Calls はソースコード内のクラスおよびそれらの間での呼び出し関係を示します。これらの呼び出しはクラスメンバ間の呼び出しが抽象化されたものです。例えば、ClassA::FuncA が ClassB::FuncB を呼び出す場合、ClassA は ClassB を呼び出すとみなされます。この機能は、制御フローの観点からどのクラスが他のクラスのどれに関係しているかを診断する際に役立ちます。(6.4.0)

Class HasType はデータタイプとして他のクラスで使用されているクラスを示します。これはクラス間のデータ相互依存性のいくつかを診断する際に役立ちます。(6.4.0)

Function Calls with Set Variables は関数呼び出しおよび関数での変数のセットを示します。これは変数のセットと読み取りの両方が表示される Function Calls with Variables ビューを補完するビューです。(6.4.0)

Function Calls with Macros は関数呼び出しおよび関数でのマクロの使用を示します。これはソフトウェアのマクロ定義に何が隠蔽されているかを理解する際に役立ちます。

File Build、Classes and Calls、Class Use、Type Use の 4 つの事前定義ビューが削除されました。

Traverse 機能に、1 組の新しい機能として Unique Up および Unique Down が追加されました。これらは Full Up および Full Down と同様の機能ですが、グラフ内の他のシンボルからも到達可能であるシンボルを除外します。(6.1.0)

Filter 機能に、Hide Library が追加されました。これはグラフからライブラリ関数を削除する簡単な方法を提供します。

新しい関数として Analyze が追加されました。Analyze ダイアログはコンテキストセンシティブメニュー（シンボルの右クリックで表示）の Analyze...メニュー項目から利用可能です。Analyze ダイアログには実行可能な解析のリストが含まれており、そのうち大部分については新しいグラフウィンドウが表示されます。この機能により、関心のある情報を表示するグラフの作成が大幅に迅速化および簡素化されます。Analyze 機能のごく一部の機能をサポートしていた AutoQuery 機能は削除されました。(6.0.0 - 6.4.0)

手動によるグラフィックレイアウトの持続性が大幅に拡大されました。グラフィックレイアウトに対するユーザの最適化は、グラフの周囲にシンボルをドラッグすることによって常に可能です。この最適化は新しいシンボルがグラフに追加される時にも持続するようになりました。新しい機能の Automatic Layout が Edit メニュー項目およびアイコンに追加されました。この機能は自動レイアウトに復元します。(6.3.0)

いくつかのビューについて、以前は View メニューから利用可能であった表示フォーマットのオプションが、新しい Display メニューから利用可能になりました。UML Diagram ビューでは、これらのコントロールは以前は View->Set View メニュー項目で提供されていました。構成関連のビューでは、これらはメトリックス情報をグラフで色分けするためのコントロールになっています。(6.4.0)

新しいコントロールにより、あるレポート（特に Functions Not Used In Tasks レポート）からグラフへと情報を色分けして示すことが可能になりました。(6.3.0)

マウスの右クリックによるコンテキストセンシティブな操作が拡張されました。拡張された操作の中には、グラフの空白部分またはグラフ内の関係をクリックしたときの新しい操作が含まれます。

グループメンバと構造体メンバの衝突を解決しました。(6.2.0)

ブックマークおよび元に戻す/やり直しの機能を拡張し、ビューをまたがった変更に対応するようになりました。(6.2.1)

structure ビューグラフでのシンボルの位置をブックマークに追加し、復元されたブックマークにグラフィックレイアウトに対するユーザの最適化が含まれるようにしました。(6.3.0)

Set View が同一セッションの複数のプロジェクトから開かれたときに発生していた障害を解決しました。(6.3.3)

Manage Groups ダイアログの外観と動作を改善しました。(6.3.3)

グラフでの元に戻す/やり直す操作において、グループの処理を修正しました。(6.3.3)

Control Flow ビューのレイアウト順序を改善しました。順序には最上位のシンボルだけではなく、前のカラムのすべてのシンボルが考慮されるようになりました。(6.4.0)

UML Diagram Set View ダイアログのメンバ有効範囲の定義を変更し、パラメータを含めるようにしました。これにより、メンバのフィルタリング設定はパラメータとローカル変数の両方に作用します。(6.4.0)

Control Flow ビューおよび UML Diagram ビューでのズーム/パンの機能を拡張しました。白またはグレーの背景をもつグラフでパンしたときの見やすさが改善されました。(6.4.0)

UML Class Diagram ビューで Aggregates 関係の Information オーバレイにフォーカスしたときのクラッシュを解決しました。(6.4.0)

Edit -> Automatic Layout のアイコンとメニューを同期させました。(6.4.0)

Control Flow ビューでのブックマークの復元を修正しました。(6.4.0)

Select -> Find and Filter -> Add ダイアログでのディレクトリ名のマッチングを改善しました。(6.1.0)ダイアログの Options 部分の分かりやすさが改善されました。(6.4.0)

Flow Chart ウィンドウ

統合ディスプレイウィンドウの変更の一部として、それぞれ独自のメニューを持つ 10 個の Flow Chart ウィンドウを同時に表示させることが可能になりました。デフォルトでは、各 Flow Chart は Main パネルの一部として表示されます。

複雑なプログラムロジックを持つ関数で操作する際の Flow Chart を簡素化するために、Flow Chart をユーザ指定のデシジョンポイントで折りたたむことにより、図の一部を隠すことができます。この機能はマウスの右ボタンクリックによるコンテキストメニューによって制御されます。出力される図では、該当するデシジョンポイントおよびそれに対応するデシジョンの終結点より内側にあるどのロジックも表示されません。(6.2.0)

Highlight メニューに、強調の新しい選択肢として Editor Selection を追加しました。(6.0.0)強調表示を修正しました。(6.1.0)

ElseIf 構文の偽の行を取り除きました。(6.3.0)

Calculation Tree

統合ディスプレイウィンドウの変更の一部として、それぞれ独自のメニューを持つ 10 個の Calculation Tree を同時に表示させることが可能になりました。デフォルトでは、各 Calculation Tree は Main パネルの一部として表示されます。

データフロー解析により生成される Calculation Tree ディスプレイに、新しい Display 設定として、関数間での割り当てフローに関する情報を表示するか抑止するかの設定が追加されました。表示する場合は、フローパスから現在の場所へと呼び出される関数のどのステートメントも、現在の値に影響を与えていればリストされます。また、現在の計算に関係していれば、グローバル変数のファイルレベルの初期化もリストされます。(6.3.3)

より大規模なプロジェクトでの Calculation Tree の使用を可能にするために、拡張仮想メモリスキームが追加されました。詳しくは、当リリースノートの Flow Check のセクションをご覧ください。(6.3.0)

C++のクラスメンバに対応するために、Calculation Tree が拡張されました。(6.4.0)

Variable Table タブのデータ関連の障害を解消しました。(6.2.1)

Main パネル内の 2 つのノート間での、Calculation Tree タブのドラッグ操作を修正しました。(6.3.0)

Calculation Tree のユーザインターフェースを改善し、Calculation Tree ディスプレイ全体の中で内側のノート/タブを使用するようにしました。(6.2.1)改善および修正を行いました。(6.3.2)

複数の Calculation Tree ディスプレイ間でのデータの衝突を解消しました。(6.3.2)

Print ダイアログを同期させ、Calculation Tree ディスプレイを修正しました。(6.3.2)

Calculation Tree ディスプレイを生成する際のパフォーマンスを改善しました。複雑なプロジェクトでの計算時間に関する警告を追加しました。(6.3.3)

Variable Dependencies タブのレイアウトおよび情報表示を改善しました。(6.3.3)

Calculation Tree の変更に合わせて、ツールチップを更新しました。(6.4.0)

File Editor ウィンドウ

デフォルトでは、各 File Editor は Main パネルの一部として表示されるようになりました。

ファイル内でのシンボルへのスクロールを容易にするため、ファイルメンバのリストが追加されました。このリストはスライドアウトのウィジェットであり、File Editor アイコンバーの右に表示される新しいアイコンから利用可能です。Tools -> Options (General タブ)ダイアログにチェックボックスがあり、これによってスライドアウトをクリックする必要があるか、または単にカーソルをアイコンの上に置けばよいかを制御します。

File Editor ディスプレイの左側に行番号を表示できるようになりました。行番号表示の有無は、メニュー項目の Options -> Layout -> Show Line Numbers により制御されます。(6.3.0)行折り返しが有効である場合には、行の付番に誤りが生じるため、機能が無効にされました。(6.3.1)

一部のレポートでは、レポート結果に関連するソースコードの行を、File Editor の左側に示すことができます。Functions Not Used in Tasks レポートおよび Recursive Functions レポートでは、この表示はメニュー項目の Options -> Report Data により制御されます。Skipped Lines レポートでは、この表示はメニュー項目の Options -> Data Display -> Code Skipped by Source Analyzer により制御されます。(6.3.0)

Control-S のサポートを追加しました。Control-S を押すと File Editor の内容が保存されるようになりました。

Home キーと End キーの動作を、標準的な用法に一致するように変更しました。(6.1.0)

Save As 操作におけるファイル名の問題を解決しました。(6.1.0)

カット/コピー/ペーストのクリップボード操作を改善しました。(6.1.0)

プロジェクトデータの更新または再生成におけるファイル保存の操作を修正しました。(6.2.1)

読みやすさを向上させるため、様々なシンボル型の表現に使用される色を少し変更しました。さらに、ライブラリ関数を示す色が標準の関数を示す色と少し異なるようにしました。また、ローカル変数を示す色が標準の変数を示す色と少し異なるようにしました。(6.2.2)

Control-F のサポートを追加しました。Control-F を押すと、File Editor の一番上にある検索入力フィールドにフォーカスが移動するようになりました。Control-F を押した時点でテキストが強調表示されていれば、そのテキスト文字列が検索に使用されます。(6.3.1)

ファイルメンバリストのスライドアウトウィジェットに、ファイル順にするかアルファベット順にするかのユーザ選択が追加されました。(6.4.0)

Automount パス名またはデータベースに存在しないファイルに対する File -> Open のサポートが追加されました。(6.4.0)

プリプロセッサディレクティブにより生成されたソースコードの省略部分の始点と終点を調整しました。現在、囲んでいる#ディレクティブ行は解析対象とみなされます。これにより、ファイルエディタにおいてグレーで影を付けられる行が変化します。(6.4.0)

File Editor の変更に合わせて、ツールチップを更新しました。(6.4.0)

プロジェクトパネル

統合ディスプレイウィンドウの左側のカラム全体をカバーするように Project パネルの領域が拡張されました。Project パネルタブを表示するために 2 つのノートが提供されており、2 つのタブを同時に表示することができます。Display タブは (タブ名の) ドラッグ&ドロップにより、一方のノートから他方のノートへと移動させることができます。

従来の File Browser は Project パネル内の File Index に統合されました。従来の File Browser にあった Members タブ、Relationships タブおよび Metrics タブは、File Index の右上隅にある新しい >>> アイコンから、スライドアウトタブとして利用できます。Tools->Options (General タブ) ダイアログにチェックボックスがあり、これによってスライドアウトをクリックする必要があるか、または単にカーソルをアイコンの上に置けばよいかを制御します。フォーカスしているファイルのソースチェック違反を表示する新しいタブとして、Source Checks が追加されました。

新しいタブとして Namespace Index が追加されました。以前は、名前空間はクラスおよびテンプレートと組み合わせて様々なビューに表示されていました。現在は、名前空間は別に処理されます。

従来の Class Browser は Project パネル内の Class Index に統合されました。従来の Class Browser にあった Members タブ、Relationships タブおよび Metrics タブは、Class Index の右上隅にある新しい >>> アイコンから、スライドアウトタブとして利用できます。Tools->Options (General タブ) ダイアログにチェックボックスがあり、これによってスライドアウトをクリックする必要があるか、または単にカーソルをアイコンの上に置けばよいかを制御します。

以前は固定されていなかった Grep Tool が Project パネルに統合されました。

新しいタブとして Database Lookup が追加されました。これは Grep ツールを補完するもので、テキスト文字列ではなくシンボル名のプロジェクト全体での検索をサポートします。名前検索ではワイルドカード文字(*)を使用できます。

Graph Symbols タブが改善され、可視のシンボルの表示に加えて、選択されたシンボルだけを表示するか、あるいはシンボルの数を要約して表示する機能が加わりました。ユーザインターフェースでは Graph ウィンドウを最大 10 個までサポートするようになったため、Graph Symbols タブにはアクティブな Graph ウィンドウに関する情報が表示されます。

Home キーと End キーの動作を、標準的な用法に一致するように変更しました。(6.1.0)

タイミングの問題によって時として発生していた、Graph Symbols ディスプレイが Graph ウィンドウと同期しなくなる現象を解決しました。(6.2.0)

プロジェクトレベルの複雑度メトリックスを Project Summary タブに追加しました。(6.2.0) 平均複雑度の計算を修正しました。(6.2.1)

動作しなくなっていた、grep シンボルの強調表示を修正しました。(6.3.0)

構造体と共用体を、Namespace Index の Members スライドアウトタブに追加しました。(6.3.3)

Grep Tool タブ内の grep 文字列入力フィールドのスクロールを改善しました。(6.3.3)

Graph Summary タブの印刷出力の内容を修正しました。(6.3.3)

スライドアウトタブがメインウィンドウの後ろに隠れてしまわないようにしました。(6.4.0)

シンボルパネル

Symbol パネルの領域を拡張し、統合ディスプレイ・ウィンドウの下部のほとんどがカバーされるようにしました。Symbol パネルタブを表示するために 2 つのノートが提供されており、2 つのタブを同時に表示することができます。Display タブは (タブ名の) ドラッグ&ドロップにより、一方のノートから他方のノートへと移動させることができます。

Symbol パネルにフォーカスするための新しい方法が追加されました。ツールの任意の場所で、シンボルの上でマウスの右ボタンを (クリックするのではなく) 長く (0.5 秒以上) 押し続けると、Symbol パネルがそのシンボルにフォーカスします。従来のコンテキストメニューによる Symbol パネルにフォーカスする動作は依然として存在します。

従来の Declaration タブが改善され、フォーカスされたシンボルの完全なソースコードを示すようになりました。例えば、関数が Symbol パネルでフォーカスされると、タブには関数の完全な定義が表示されます。タブの名前が Source Code に変更されました。

Cross Reference タブが追加されました。このタブではフォーカスされたシンボルのファンインとファンアウトがグラフィック表示されます。このタブは Browse モードにおいては従来の Graph ウィンドウと同様に動作し、フォーカスと緊密にリンクしており、Symbols タブの残りの要素のフォーカスを決定付けるかそれに従います。グラフの表示オプションは、グラフの背景の空白部分を右クリックすることによって表示されます。

従来の Use Browser は、Usage タブとして Symbol パネルに統合されました。以前は、固定されていない別のディスプレイウィンドウでした。従来のユーザ設定のいくつかを削除し、タブがより自動的に機能するようになりました。(6.0.0)関数ポインタの位置の解析が改善されました。(6.2.1)システム関数ポインタのサポートを追加しました。(6.3.0)

History タブが追加されました。これにより、Symbol パネルで前にフォーカスしていた場所に戻ることができます。(6.3.0)

File Location タブが改善され、フォーカスされたシンボルの定義と共に宣言の場所も示すようになりました。

Cross Reference グラフ内にメトリック値によって偽りの色付けが表示されるのを解消しました。(6.2.1)

Usage タブを、他のシンボルの宣言に使用されるシンボルに対して適切なファイルからソースコードの行を表示するように修正しました。(6.3.3)

Usage タブおよび History タブの印刷処理を修正しました。(6.3.3)

Metrics ウィンドウ

統合ディスプレイウィンドウの変更の一部として、従来の Metrics Tool の様々なディスプレイが、Main パネルの一部であるディスプレイとしてビュー可能になりました。

ファイルレベルでは、6 つのメトリックスが追加されました。それらは Functions in File、Total Cyclomatic Complexity、Average Cyclomatic Complexity、Maximum Cyclomatic Complexity、Halstead Program Volume、Maintainability Index です。(6.2.0)

新しいメトリックスディスプレイとして Namespace Metrics が追加されました。以前は、名前空間はクラスおよびテンプレートと組み合わせてメトリックスディスプレイに表示されていました。現在は名前空間は別に処理され、メンバ数を示すソフトウェアメトリックスの小さい集合だけを実際に名前空間に使用できます。(6.2.0)

クラスレベルでは、7 つのメトリックスが追加されました。それらは Internal Methods Called、External Methods Called、Chidamber/Kemerer Class Coupling、External Methods and Coupling、Average Cyclomatic Complexity、Maximum Cyclomatic Complexity、Halstead Program Volume です。(6.2.0)

いくつかのクラスメソッドに対し、代替の計算方法が追加されました。Lack of Cohesion of Methods メトリックスには、Chidamber/Kemerer メソッドペア共有計算および Li/Henry/Hitz/Montazeri の別の依存グラフ計算が、Henderson/Sellers の標準計算のオプションとして利用可能です。Internal Methods メトリックスおよび External Methods メトリックスには、さらに別の計算オプションが利用可能です。これらの計算方法は Tools -> Options(Metrics タブ)ダイアログを通して選択されます。(6.2.0, 6.2.1)

関数レベルでは、5 つのメトリックスが追加されました。これらは、McCabe Decision Density、McCabe Essential Complexity、McCabe Essential Density、Knots、Halstead Program Volume です。(6.2.0)

クラスメソッドを Response for Class メトリックスの計算に追加しました。(6.2.0)

Chidamber/Kemerer Class Coupling、Chidamber/Kemerer Lack of Cohesion of Methods、および External Methods Called の各メトリックスの計算を変更しました。(6.2.1)

スクロール中のメトリックスの表示更新を修正しました。(6.2.1)

ステートメントによりソートする際の問題を解決しました。(6.2.1)

Java コードに対する Depth of Inheritance Tree メトリックスの計算における問題を解決しました。(6.3.1)

Java .class ファイルに対して Knots メトリックスを計算しようとする問題を解決しました。(6.3.1)

フローチェックレポート

拡張された仮想メモリスキームを使用するすべてのフローチェックレポートでメモリ容量を増やし、関数間のデータフロー解析に必要とされる相当量のメモリ要件に対処しています。Imagix 4D では、データフロー解析での中間的な計算に最大 64GB のディスク格納領域を使用できるようになりました。以前のフロー解析は、50K から 100K のステートメントと関数を持ちサイクロマチック複雑度が 200 未満であるプロジェクトにしか現実的には適用できませんでした。この改善により、関数のサイクロマチック複雑度が 200 を超えない限り、100K から 500K のステートメントを持つさらに大規模なプロジェクトを処理することができます。関数がより複雑である場合でも解析は実行可能ですが、より大きいプログラムでは実行時間が数時間かかる可能性があります。500K 以上のステートメントおよび多数の複雑な関数を持つプロジェクトでは、解析が完了しない可能性が高いです。可能であれば、それらをより小規模なプロジェクトに分割することをお勧めします。(6.3.0)

Task Flow Checks メニューにある新しいレポートの Functions Not Used in Tasks は、直接呼び出されない関数、または定義したタスクから関数ポインタにより呼び出されない関数をすべてリストします。タスクを全く定義していない場合、すべてのルート関数がタスクであるとみなされ、このレポートにはどの関数もリストされません。このレポートには Display 設定があり、それによってライブラリ関数のリスト表示の有効/無効を切り替え、名前、ファイル、またはディレクトリ名などによってリストの順序を変えることができます。(6.3.0)

Mismatched Critical Regions レポートでは、従来提供されていた関数呼び出しの始点または終点がない領域に加え、あいまいなステートメントおよび関数を表示するようになりました。あいまいなステートメントとは、1つの制御フロー内のクリティカルな領域により保護され、別の制御フローで保護を解除されている可能性があるステートメントです。同様に、あいまいな関数とは1つの制御フローケースで保護され、別の制御フローで保護を解除されている可能性がある関数全体です。これらのリストは一致しない領域に関するもう1つのビューをユーザに提供します。この情報を使用して、クリティカルな領域の保護呼び出しを追加することによりこれらの制御フローを改善すべきかどうかを決定することができます。

Variables Set in Multiple Tasks レポートおよび Reentrant Functions レポートは特定の用途に関係するクリティカルな領域の名前を、保護インジケータの隣にリストします(下のスクリーンショット参照)。識別しやすくするため、クリティカルな領域の名前はレポートの始めにリストされます。またこれらのレポートではステートメントが保護されるかどうかを判定する際、すべてのクリティカルな領域を考慮します。以前は、最後に定義されたクリティカルな領域だけが使用されていました。(6.2.0)

```

File
  Variable (Line)
    Task
      Line Number of Usage
        Critical Region
          User of Variable

sample600_2.c
  glob (5)
    t600_2a
      10 U (CR1) f600_2b sample600_2.c (10)
      10 P (CR2) f600_2b sample600_2.c (10)
    t600_2b
      10 U (CR1,CR2) f600_2b sample600_2.c (10)
    t600a
      15 P (CR1) t600a sample600.c (15)
      15 U (CR2) t600a sample600.c (15)
    t600b
      19 P (CR1) f600n sample600.c (19)
      19 U (CR2) f600n sample600.c (19)

```

すべてのフローレポート、特に Uninitialized Variables Read、Mismatched Critical Regions、および Out of Step (Z) Variables では、C/C++ソースアナライザによって提供される、for ループに関する追加情報を使用します。静的な境界が存在していて、for ループに1回以上入っていることをアナライザが判定すると、フロー解析では無条件にループに進入します。これは while ループと do ループのフローの違いに似ており、1つめは最初に条件があり、2つめはただちにループに進入します。(6.0.2)

Variable Flow Between Tasks レポートでは、複数の潜在的な関数呼び出しを通してセットされる変数について、複数回セットされるとみなします。これは"SS"により示されます。以前は、これらが単一のセットを示す"S"でフラグされることがありました。同様にこのレポートでは、呼び出された関数内の別の割り当てがフロー解析に含まれるため、いくつかの変数フローの分類を"R,S"または"RR,SS"から"R,*"に変更しました。(6.0.2)

Reentrant Functions レポートでは、同じ関数の複数回の出現を解消し、レポート内の冗長部分を減らしました。(6.1.0)

いくつかのレポートにおけるメモリ使用量を削減しました。(6.2.0)

Out of Step (Z) Variables レポートは、呼び出し階層のほか行番号に基づいて関数の並びをソートします。以前は、同じレベルにある関数呼び出し(if文またはswitch文での分岐など)が、ランダムに順序付けされることがありました。(6.3.1)

ソースの複数箇所に宣言を持つ変数について、レポートが正しく宣言の1つをリストするようになりました。以前は、レポートでは間違ったファイルからの行番号で変数をリストすることがありました。

Uninitialized Variables Read レポートは、変数が実際には現在の使用へのパス上で呼び出された関数によってセットされる箇所について、以前は偽の肯定をリストしていました。

Reentrant Functions レポートに、複数のクリティカルな領域に関する報告を追加しました。(6.4.0)

その他のレポート

すべてのレポートに適用される新しいオプションを使用すると、レポートの最初の起動時に実際にレポートが実行されないように設定できます。このオプションを指定してレポートを起動すると、結果となるレポートは空白で表示されます。これによってユーザは、レポートが実際に実行される前にレポートのメニュー項目を通してレポートの設定を変更することができます。集中的に計算が行われるレポートでは、これによって大幅に時間を節約できます。(6.3.0)

Include Analysis レポートの出力結果は、シンボルの従属性の基本計算を改善したことにより大幅に改訂されました。(6.3.3)

新しいレポートとして Similar Functions が追加されました。このレポートはある評価基準に基づいて、類似する構成体を持つ関数を特定します。類似性はユーザ設定により定義され、名前、相互参照ファンアウト、および一連のメトリックス(McCabe Cyclomatic Complexity、Halstead

Program Difficulty、および Number of Statements) のさまざまな組み合わせを含みます。例えば、このレポートはデフォルトでは、90%の類似した名前と 90%の類似したファンアウトを持つか、90%の類似した名前と 90%の類似したメトリックス値を持つか、90%の類似したファンアウトと 90%の類似したメトリックス値を持つ関数をリストします。このレポートは、類似したコードの統合をめざしたリファクタリングの努力をサポートします。またこれは、類似した関数でバグが発生したためにバグ修正の適用の必要があると思われる関数を特定するのに役立ちます。(6.2.0)

新しいレポート関数として Import Report が追加されました。レポートを特殊な 4dr フォーマットで保存し、のちのセッションで Import Report 機能を使用してそれを再オープンすることにより、計算が集中的に行われるレポートの再生成に要する時間を節約することができます。また、Import Report レポートにより他のツールからのデータを Imagix 4D にインポートした後、Imagix 4D を使用して、他のツールにより発生したソフトウェアの問題を調べることができます。

Import Report を通してインポートされるレポートに期待されるフォーマットを変更しました。(6.1.0)

一部のスクロール操作を改善しました。(6.3.3)

Similar Functions レポートにおける相互参照の類似に関する計算を修正しました。(6.3.3)

Class Summary 合計から名前空間を除去しました。(6.3.3)

ソースチェックの Potential Static Function から非関数を除去しました。(6.3.3)

-msrc オプションを使用したときのソースチェックの重複エントリを除去しました。(6.3.3)

ソースチェックレポートにおいて Omitted として示される行の整合性を改善しました。(6.4.0)

Include Analysis レポートにおける var スタイル変数の解析を修正しました。(6.4.0)

その他のユーザーインターフェースおよび情報の表示

Help メニューから利用可能な入門ツアーが 6.x の機能を示すように更新され、ツールにロードされた任意のプロジェクトと共に機能するように改善されました。その結果、ユーザは Imagix 4D の機能が自分のコードに対してどのように動作するかを調べることができます。

Information オーバーレイに表示されるプレースメントおよびグループの情報が改善されました。(6.1.0)

Windows の内部メニューに対するツールチップの掲示を修正しました。(6.1.0)

メインウィンドウ起動時の大きさと位置の設定に、より一貫性をもたせました。(6.3.2)

Tools メニューから複数ウィンドウが起動される現象を修正しました。(6.3.3)

メニューおよびアイコンの有効/無効の切り替えを改善しました。(6.3.3)

ローカルおよびパラメータについて、Information オーバーレイに表示される情報を改善しました。(6.4.0)

Information オーバーレイにタスク情報を追加しました。(6.4.0)

Options ダイアログ、コンテキストセンシティブヘルプ、ツールチップ、およびツアーからモードに関する記述を削除しました。(6.4.0)

ドキュメントの生成

名前空間のドキュメント出力に新しいセクションが追加されました。以前は、名前空間はクラスおよびテンプレートと組み合わせてクラスセクションに表示されていました。現在は、名前空間は別に処理されます。

Internet Explorer 7 での大きい.png ファイルの表示を修正しました。(6.0.0)

ドキュメントのクラスセクションが不適切に省略されないようにしました。(6.1.0)

HTML ドキュメント上のソースコードセクション間のナビゲーションを改善しました。
(6.1.0)

変数および関数ポインタに対してどのサブセクションが生成されるかを修正しました。
(6.2.0)

HTML ドキュメントの一番上に Struct セクションが二重に表示されないようにしました。
(6.2.0)

ライブラリ型のサポートを追加しました。(6.2.1)

DocGen シートに、相互参照グラフのレイアウトおよびコンテンツ用のコントロールを追加しました。(6.2.1)

手動でフローチャートの失敗を通知する必要性をなくしました。(6.2.1)

相互参照レイアウト、コンテンツ制御および新しいメトリックスに対応してサンプル.dgn シートを更新しました。(6.3.0)

プロジェクトの全体より小さい範囲をカバーするドキュメントについて、ソースコードの断片を修正して正しいファイルから抽出されるようにしました。(6.3.3)

ソースアナライザの-vuid オプションを使用して部分的または分散型の HTML ドキュメントを生成する際の、参照の衝突を修正しました。(6.3.3)

'enum'リテラルについて、読み取られる箇所を示すように用途の情報を変更しました。
(6.4.0)

出力の印刷

Main パネルのディスプレイには、ローカルの File メニューを通して利用可能な独自の Print ダイアログが用意されています。(6.0.0)一番上のメニューバーの Print ダイアログが、追加のディスプレイを扱うために改善されました。(6.2.0)Print ダイアログが改善されました。.png ファイルについて、グレースケール出力が選択できるようになりました。グラフの背景を逆にするオプションが追加されました。フォーマットを変更するとファイルの接尾語が自動的に更新されるようになりました。(6.4.0)

Visio .vdx フォーマットのサポートを追加したことにより、Visio2003 以降のバージョンの Microsoft Visio へのエクスポートが可能になりました。(6.3.1)Calculation Tree に.vdx フォーマットを追加しました。(6.3.2).

.png 出力に、16 ビットの日本語の文字に対するサポートを追加しました。(6.1.0)

Windows の.png 出力で、Arial の文字を使用できるようになりました。(6.2.1)

プリンタへの出力について、用紙サイズの最適化機能を改善しました。(6.3.0)

.png 出力およびプリンタへの出力について、白い背景を持つグラフでの色の変換を修正しました。(6.3.2)

.vdx 出力に、SJIS に対するサポートを追加しました。(6.3.2)

.csv 出力について、正方形とテキストがあるグラフの印刷を修正しました。(6.3.3)

.ps 出力について、ボックスのサイズ変更用の線がある Control Flow ビューの印刷を可能にしました。(6.4.0)

.txt 出力について、Analyzer Results ウィンドウの印刷を修正しました。(6.4.0)

.vdx 出力について、HTML 特殊文字を含む名前の印刷を修正しました。(6.4.0)

.vdx 出力について、Graph ウィンドウのコメントの印刷を修正しました。(6.4.0)

.png 出力およびプリンタへの出力について、Graph ウィンドウのコメントにおける日本語の文字の印刷を修正しました。(6.4.0)

.png 出力について、コメントと幅の広い文字列を持つ Flow Chart の出力における関数名およびコメントの位置を修正しました。(6.4.0)

プリンタへの出力について、フォントサイズの計算を改善することにより文字が重ならないようにしました。(6.4.0)

バッチモード/コマンドラインでの起動

ブックマークデータのオープンに関連する問題を解決しました。(6.2.0)

ファイルの編集内容の保存が `cmmdRegenerateProjectData` に干渉しないようにしました。(6.2.2)

`cmmdOpenProject` を同一コマンドファイル内で複数回起動できるようにし、複数のプロジェクトを連続的にオープンできるようにしました。(6.3.2)

相互参照の印刷処理が `cmmdDocument` 出力に干渉しないようにしました。(6.3.2)

名前空間メトリックスを、デフォルトの `cmmdMetrics` の起動により生成される出力に追加しました。(6.3.3)

`cmmdMetrics` 出力のカラム名から偽りのカンマを除去しました。(6.3.3)

位置を示す引数が与えられていない場合の `cmmdMetrics` コマンドおよび `cmmdReportMultiple` コマンドの出力位置を修正しました。(6.4.0)

インストール

Vista 環境下でのインストールを簡素化しました。(6.0.0)

`imagix-licsppt.bat` コーティリティにおけるディレクトリ名の中のスペースの取り扱いを修正しました。(6.1.0)

ライセンス認可メカニズムにおいてユーザ名の大文字/小文字を区別しないようにしました。(6.1.0)

MAC アドレスの読み取りのサポートを Vista の外国語バージョンに追加しました。(6.3.2)

ソフトウェア使用許諾契約も含め、日本語サポートのインストールを改善しました。(6.4.0)

5. 株式会社東陽テクニカへのお問合せ

株式会社東陽テクニカは、Imagix Corporation の日本国内における総代理店です。
日本国内でのお問合せについては、株式会社東陽テクニカへお願いします。

株式会社 東陽テクニカ
ソフトウェア・ソリューション

郵便番号 〒103-8284

住所 東京都中央区八重洲 1-1-6

電話 03-3245-1248(直通)

FAX 03-3246-0645

E-Mail(営業窓口) ss_sales@toyo.co.jp

E-Mail(サポート窓口) ss_support@toyo.co.jp

E-Mail(ライセンス窓口) ss_lic@toyo.co.jp

6. Imagix Corporation へのお問合せ

以下に示す Imagix Corporation に直接お問合せいただくこともできます。

電話 +1 805 781 6002

FAX +1 805 781 6003

E-Mail(営業窓口): sales@imagix.com

E-Mail(サポート窓口): support@imagix.com

7. 保証の免責

株式会社東陽テクニカおよび Imagix Corporation は、作成時に細心の注意を払って本書を正確に記述することに努めておりますが、本書内の誤りや欠落に関する責任は負いかねます。また、本書内の記述通りに製品が動作することを表現かつ意味する保証書として作成された文書ではないことをお断りしておきます。

8. 著作権について

本書には著作権があり、全体または一部を問わず、どのような方法(電子的またはその他)によっても、Imagix Corporation の文書による事前の承諾なしには、複製、再生、公開、転記、翻訳、縮小を行うことはできません。

9. 商標

本書内に示した他の全ての製品やサービスには、それらを販売する企業で選定された商標、サービスマーク、製品名が含まれます。