

Autodesk® Scaleform®

AMP ユーザーガイド

このドキュメントでは、Scaleform の「Analyzer for Memory and Performance」（AMP）の概要と、アプリケーションのプロファイリング方法について説明しています。

作者 Alex Mantzaris
バージョン 3.03
最終更新日 2013 年 4 月 10 日

Copyright Notice

Autodesk® Scaleform® 4.4

© 2014 Autodesk, Inc. All rights reserved. Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, Algor, Alias, AliasStudio, ATC, AutoCAD LT, AutoCAD, Autodesk, the Autodesk logo, Autodesk 123D, Autodesk CAM 360, Autodesk Homestyler, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSketch, AutoSnap, AutoTrack, Backburner, Backdraft, Beast, BIM 360, Burn, Buzzsaw, CADmep, CAiCE, CAMduct, CFdesign, Civil 3D, Cleaner, Combustion, Communication Specification, Configurator 360™, Constructware, Content Explorer, Creative Bridge, Dancing Baby (image), DesignCenter, DesignKids, DesignStudio, Discreet, DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWGX, DXF, Ecotect, ESTmep, Evolver, FABmep, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, ForceEffect, FormIt, Freewheel, Fusion 360, Glue, Green Building Studio, Heidi, Homestyler, HumanIK, i-drop, ImageModeler, Incinerator, Inferno, InfraWorks, InfraWorks 360, Instructables, Instructables (stylized robot design/logo), Inventor, Inventor HSM, Inventor LT, Kynapse, Kynogon, LandXplorer, Lustre, MatchMover, Maya, Maya LT, Mechanical Desktop, MIMI, Mockup 360, Moldflow Plastics Advisers, Moldflow Plastics Insight, Moldflow, Moondust, MotionBuilder, Movimento, MPA (design/logo), MPA, MPI (design/logo), MPX (design/logo), MPX, Mudbox, Navisworks, ObjectARX, ObjectDBX, Opticore, Pipeplus, Pixlr, Pixlr-o-matic, Productstream, Publisher 360, RasterDWG, RealDWG, ReCap, ReCap 360, Remote, Revit LT, Revit, RiverCAD, Robot, Scaleform, Showcase, Showcase 360 ShowMotion, Sim 360, SketchBook, Smoke, Socialcam, Softimage, Sparks, SteeringWheels, Stitcher, Stone, StormNET, TinkerBox, ToolClip, Topobase, Toxik, TrustedDWG, T-Splines, ViewCube, Visual LISP, Visual, VRED, Wire, Wiretap, WiretapCentral, XSI.

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

Autodesk Scaleform:の連絡先 :

ドキュメント名	AMP ユーザーガイド
住所	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
ウェブサイト	www.scaleform.com
Email	info@scaleform.com
電話	+1 (301) 446-3200
Fax	+1 (301) 446-3199

目次

1 はじめに	1
1.1 AMP サーバー	2
1.2 AMP クライアント	3
2 AMP の使い方	12
2.1 コネクションの確立	12
2.2 性能指標の解析	13
2.3 レンダリング指標の解析	14
2.4 メモリ消費の解析	14
2.5 オーバードロー プロファイリングの使用	19
2.6 バッチ プロファイリングの使用	20
3 プラットフォームとインテグレーションに関する注意	22
4 ビルドに関する注意	23
5 AMP サポートの追加	24
5.1 Debug ファイルの生成	24
5.2 アプリケーションのコントロール	24
5.3 コネクション・ステータス	25
5.4 マーカー	25
6 AMP 関連の FAQ	26
7 追加情報	27

1 はじめに

AMP™ は、Scaleform アプリケーションの CPU 利用率、グラフィックレンダリング、およびメモリ消費のそれぞれの状況をモニタする、Scaleform® のリモートプロファイリングシステムです。メモリや性能のボトルネックを特定できる情報がグラフ上にリアルタイムに表示されるほか、フレームごとの統計情報を参照してボトルネックの詳細な解析と特定が可能です。また、ActionScript 関数とソースコードのタイミングをフレーム単位に確認し、スクリプトの問題箇所を速やかに抽出することができます。

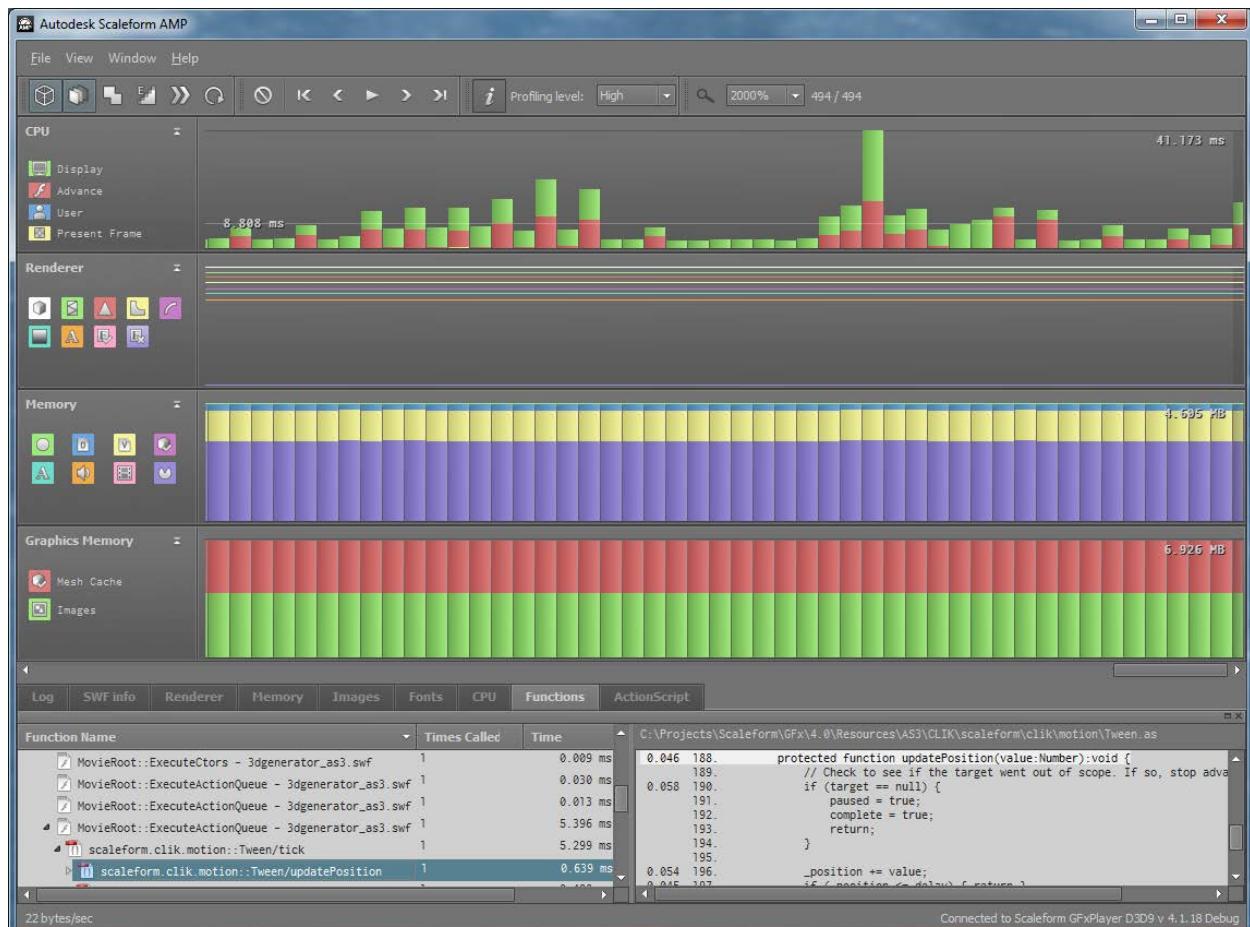


図 1: Scaleform AMP

すべての Scaleform アプリケーションは AMP サーバーとして動作します。すなわち、プロファイルからのリクエストをリスンし、各フレームの情報を集め、ネットワークを介してその情報をプロファイルに送信する役割を担います。プロファイルは AMP クライアントとして動作します。すなわち、プロファイルは特定の AMP サーバーと接続を行い、サーバーから送信される各フレームの情報を処理するとともに、プロファイル対象アプリケーションを制御するためにリクエストを送信します。

AMP は Scaleform 3.2 以上のバージョンで利用可能です。

1.1 AMP サーバー

プロファイル対象となる Scaleform アプリケーション（AMP サーバー）は、ネットワークを介して、ある量の情報をプロファイルラに送信します。関数プロファイリングのレベルに応じて、各 ActionScript 関数の経過時間と主要な C++ Scaleform 関数の経過時間が記録されます。ライン単位のソースコード・プロファイリング機能を ActionScript2 に有効にすると、各 ActionScript のバイトコード実行の経過時間が記録されます。（ActionScript3 ソースコードプロファイリングは各実行の経過時間を必要としません）。このほかに、レンダリング統計、詳細なメモリヒープとイメージ情報、Flash ファイル情報、および ActionScript マーカーが記録されます。なお、フレーム単位のこのような処理によって、とくにライン単位でのタイミング記録を有効にした場合に、アプリケーション性能の低下とメモリ消費量の増大を招くことがあります。

プロファイル対象のアプリケーションに対して AMP による介在を維持するために、すべての AMP サーバー情報はアプリケーションとは別のメモリヒープに保持されます。ヒープのサイズがプリセット上限を超えた場合、保留中のフレーム情報がネットワーク経由で送信されその後削除されるまで、アプリケーションは一時的に停止します。なお、Scaleform のシッピング・ビルドでは AMP サーバーは動作しません。なお、Scaleform のシッピング・ビルドは AMP サーバーとしては動作しません。また、いかなるビルドでも、必要に応じて、アプリケーションから AMP サポートを簡単に無効化することができます。

サーバー上の性能統計情報は Movie 単位に維持されます。そのため、特定の SWF ファイルに対応して複数のビューが存在する場合でも、問題箇所を簡単に特定することができます。また、別々のスレッド上で複数の Movies が動作している場合でも、ActionScript 実行に関して意味のある呼び出しグラフを生成することができます。

AMP サーバーはメモリ情報や統計情報をクライアントに報告する一方で、プロファイリングに必要な複数の機能をサポートするために、AMP クライアントによって制御されます。たとえばプロファイルラ側から、レンダリングのワイヤフレーム・モード、一時停止、早送りあるいは Flash ムービーの再実行、アンチエイリアスやストローク・モードのトグル設定、ローカライズ・フォントの変更、ベクターグラフィックスの曲線誤差といったリクエストを送信することができます。また、表示性能のボトルネットを確認するために、クライアント側から特殊な AMP レンダリング・モードを指定して、マスク、filtration、ピクセル・オーバードローをハイライトで表示させることもできます。

上記の各機能はあらかじめ Scaleform Player に実装されていますが、一部はアプリケーションごとに実装する必要があり、Scaleform インテグレーションの一部として実行します。Scaleform アプリケーションにこれら機能の一部が実装されていない場合に備えて、AMP サーバーはサポート済みアプリケーション制御機能の一覧を接続中のクライアントに報告します。また AMP サーバーは、視覚的なフィードバックのために、現在のステート（たとえばワイヤフレーム・モード）をクライアントに報告します。

さらに AMP サーバーは、Flash デバッグ情報をネットワーク経由でプロファイルラに送信する機能を備えます。そのため、クライアントがそれら情報にアクセスできない場合でも、ソースコードとライン単位でのタイミングを表示することができます。この送信を行う前に、SWD ファイル（ActionScript2）または AS ファイル（ActionScript3）を AMP が探し出せるディレクトリに配置しておかなければなりません。

1.2 AMP クライアント

リモート・プロファイラ（AMP クライアント）はスタンドアロン・アプリケーションです。AMP サーバーに接続してフレームごとにプロファイル情報を受信し、それら情報を性能の効率的な解析やメモリ・ボトルネックの特定ができるように、ユーザーに分かりやすい形式で表示します。プロファイル自体は Scaleform アプリケーションで、Scaleform CLIK を使って実装されています。

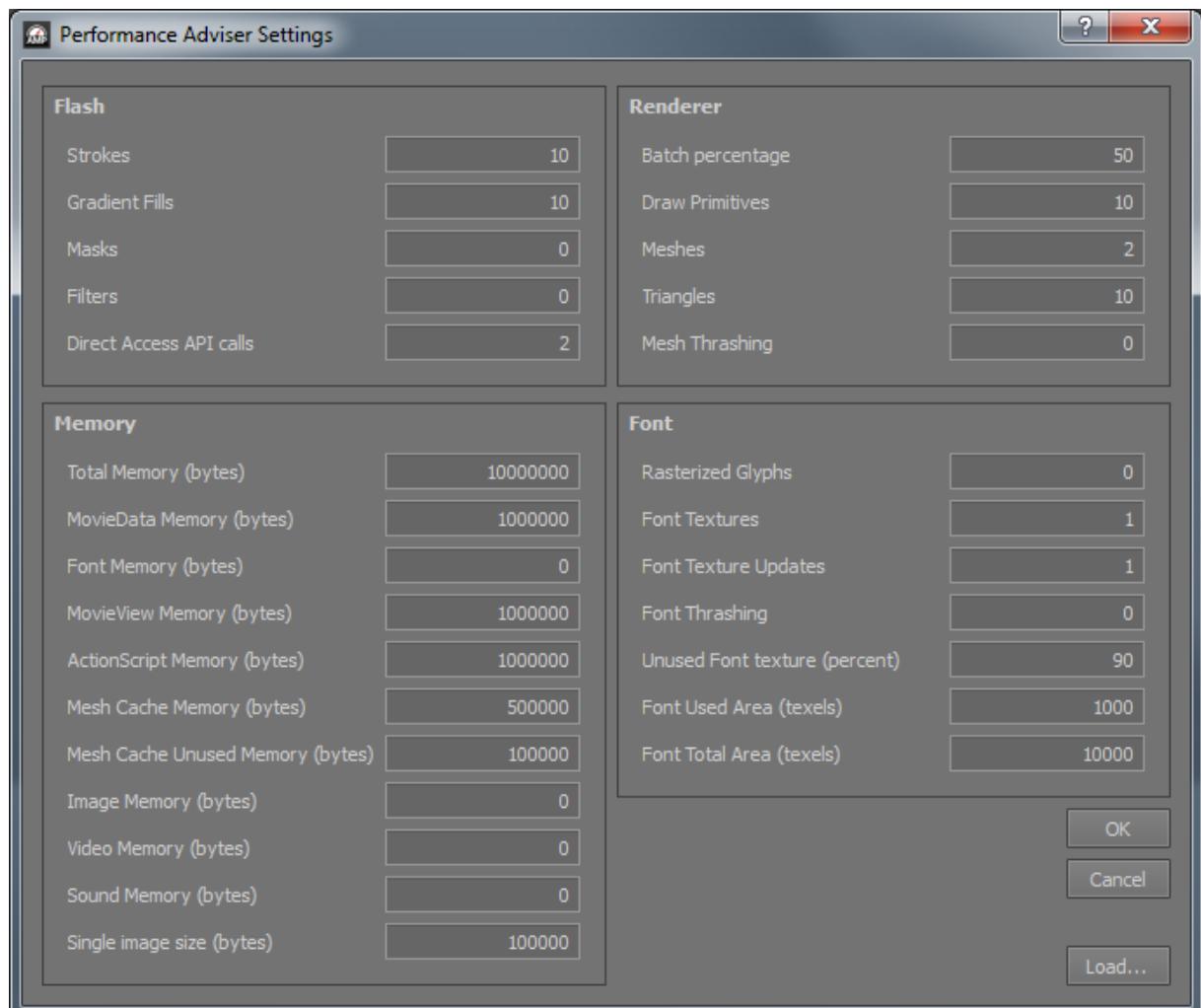
AMP クライアントには、開発した Flash 素材を複数の観点から開発者が確認し、問題箇所を特定できるような機能が登載されています。

- **CPU グラフ:**ゲームフレームごとの使用時間を次で示します：ActionScript (Advance)、Scaleform レンダリング (Display)、Direct Access API (User)、グラフィックスバッファースワッピング (PresentFrame)。このグラフは Scaleform に起因する CPU 利用率の一時的な上昇を特定したい場合に役立ちます。一時に上昇したフレームを選択し詳細統計を確認すれば、原因を特定することができるはずです。
- **レンダリンググラフ:**[Scaleform Best Practices Guide](#) (Scaleform のベストプラクティスガイド) ではレンダリング関連の分野でパフォーマンスに影響を与えるものを取り上げており、これには描画ライン、三角形、線の数が多すぎる、マスク、フィルター、ストローク、グラデーションの塗りつぶしが存在する、などがあります。こうした分野でのそれぞれのカウンターはレンダリンググラフに示されており、デベロッパーの方は Scaleform 内で何が起こっているかをより良く知ることができます。
- **メモリーグラフ:**Scaleform のメモリー使用量はメモリーグラフで把握されており、デベロッパーの方はカテゴリー別にシステムメモリーの使用量を素早くモニターできます。これにはレンダリング、画像、サウンド、フォント、動画データなどがあり、また全てのロードされたファイルも含まれています。
- **グラフィックスメモリーのグラフ:**Scaleform グラフィックスのメモリー使用量の多くは AMP でもトラックでき、ここからは画像とメッシュのキャッシュに使用される GPU メモリーを報告します。
- **Log タブ:**Log タブをクリックすると、接続中にプロファイルされたアプリケーションの生成した Scaleform ログが表示されます。
- **Performance Adviser (パフォーマンスアドバイザー)** : このタブは、Flash コンテンツのパフォーマンスを改善する方法についてのガイダンスを提供します。コンテンツ関係のプロパティーが、ユーザーの定義するしきい値を越した時に出る警告のリストです。パフォーマンスに関連するプロパティーには次のようなものがあります。
 - *Text Fields optimized for readability:* (読みやすくするためにテキストフィールドを最適化しました。) 大きいテキストフィールドに「Anti-alias for readability (アンチエイリアスで読みやすさを向上)」を使用すると、フォントのキャッシュには各文字の各フォントサイズに対して別個のグリフエントリーを含むようになります。テキストフィールドが「Anti-alias for animation (アニメーションのためのアンチエイリアス)」を使用する場合、フォントサイズにかかわらずエントリーは 1 つだけになります。
 - *Ratio of draw primitives to number of meshes* (メッシュの数に対する描画プリミティブの比率) : これはバッチ効率の指標です。バッチ処理が何もない最悪のケースの場合、各描画プリミティブに対して 1 つのメッシュがあります (比率は 1.0)。同じ描画プリミティブの 1 部としてメッシュが存在するようになると、これはバッチ処理効率の向上を示し、この比率が下がります。
 - *Number of draw primitives* (描画プリミティブの数) : 描画プリミティブの数が大きいと言う事は複雑なシーンであることを表し、レンダーするのにより長い時間がかかります。

- *Number of meshes* (メッシュの数) : メッシュの数が大きいことも、複雑なシーンであることを表します。
- *Number of triangles*: (三角形の数) : 三角形の数が多いと、メモリーを多用し、パフォーマンスも損なわれます。
- *Mesh cache thrashing*: (メッシュキャッシュスラッシング) : アイテムがキャッシュから取り除かれて、新しいエントリーのために場所を開ける、フレームごとの回数を示します。理想的にはこの数はゼロであることが望ましく、この場合レンダラーの必要とするものは全てキャッシュにあることになります。
- *Number of strokes* (ストロークの数) : 「ベストプラクティス」のガイドに示すように、Flash コンテンツ中のストロークはパフォーマンスヒットとなります。
- *Number of gradient fills* (グラデーション塗りつぶしの数) : 「ベストプラクティス」のガイドに示すように、グラデーション塗りつぶしもパフォーマンスヒットとなります。
- *Number of masks* (マスクの数) : 「ベストプラクティス」のガイドに示すように、マスクの使用はパフォーマンスヒットとなります。
- *Number of filters* (フィルターの数) : 「ベストプラクティス」のガイドに示すように、フィルターの使用はパフォーマンスヒットとなります。
- *Number of DirectAccess API calls* (直接アクセス API 呼び出しの数) : 直接アクセス API 呼び出しが多すぎると、パフォーマンスヒットとなります。
- *Number of rasterized glyphs* (ラスター化されたグリフの数) : クリフの数が多すぎるとメモリーの消費が増えます。
- *Number of font textures* (フォントテクスチャーの数) : プラットフォームによっては、フォントテクスチャーの種類が 1 つを超えるものもあります。
- *Number of font texture updates* (フォントテクスチャーアップデートの数) : 新しいフォントテクスチャーがフレームごとに生成される回数。これが多いとパフォーマンスが損なわれます。
- *Font cache thrashing* (フォントキャッシュスラッシング) : グリフがキャッシュから取り除かれて、新しいグリフのために場所を開ける、フレームごとの回数を示します。理想的にはこの数はゼロで、これはフォントキャッシュが十分大きく、全てのラスター化を保持できることを示します。
- *Unused font texture percentage* (使用しないフォントテクスチャーのパーセンテージ) : このパーセンテージが低いと、使用されていないフォントキャッシュのために予約されているスペースがあることを示します。
- *Used font cache area* (使用するフォントキャッシュエリア) : フォントとスタイルの種類が多すぎると、メモリの消費量が過大になります。
- *Total font cache area* (総フォントキャッシュエリア) : 全てのフォントテクスチャーのサイズ。フォントキャッシュはパフォーマンスを向上させますが、メモリーの消費量は増加します。
- *Total Memory* (総メモリー量) : Scaleform メモリーの総量。
- *MovieData Memory* (ムービーデータメモリー) : これはインポートされたムービーファイルのサイズに直接関連しています。
- *Font Memory* (フォントメモリー) : フォントメモリーは、テキストフィールドにフォントを共有させることで削減できます。
- *MovieView Memory* (ムービービューのメモリー) : GFx::Movie の使用するメモリーで、そのタイムラインとインスタンスデータが収められています。MovieView メモリーは、内部のガベージコレクションに依存します。
- *ActionScript Memory* (ActionScript メモリー) : ActionScript このコンテンツが多すぎると、メモリーの消費量が過大になります。

- *Mesh Cache Memory* (メッシュキャッシュメモリー) : メッシュキャッシュはパフォーマンスを改善しますが、メモリーの消費量を増加させます。
- *Unused Mesh Cache Memory* (使用しないメッシュキャッシュメモリー) : 使用しないメッシュキャッシュメモリーは、メモリーの消費量の増やすだけで、パフォーマンスは改善しません。
- *Image Memory* (画像メモリー) : 画像のために使用される、システムとグラフィックスカード両方のメモリー。
- *Video Memory* (ビデオメモリー) : このメモリーは GPU メモリーではなく、Scaleform ビデオに使用されるメモリーです。
- *Sound Memory* (サウンド メモリー) : サウンドに使用するメモリー。
- *Number of images exceeding a certain size* (一定のサイズを超過する画像の数) : ユーザーは画像のしきい値を設定でき、AMP はそのしきい値を超える画像の数をレポートします。

上に掲げる全てのしきい値を設定するにあたって、AMP は作業ディレクターで「adviser.xml」と言う名前の XML ファイルを読み込みます。この名前のファイルが見つからない場合、しきい値は全てゼロに設定されます。ユーザーは、Performance Advisor タブの右側にある Settings ボタンをクリックすることで、どのしきい値も変更できます。これをクリックすると次の設定ダイアログが表示されます。



OK をクリックすると設定が XML ファイルに保存され、次回実行時に AMP がこれを読み込みます。ユーザーはコンテンツタイプに応じて異なるアドバイザー設定を生成でき、これを設定ダイアログからロードして使用できます。「-adviser <AdviserFilePath>」をコマンドラインの引数として AMP を起動すると、プログラムに作業ディレクトリにあるデフォルトのアドバイザーファイル以外のものを読み込ませることができます。

- **SWF info タブ:**このタブではプロファイルされている全てのフラッシュファイルについての静的な情報が得られます。この情報にはファイル名、フラッシュバージョン、動画のディメンション、動画のフレームレートを含みます。
- **Renderer タブ:**このタブはグラフと同じ情報を、数字の形で表示します。このタブにはまたレンダリングに関する次のようなその他の統計もあります。メッシュとフォントキャッシュスラッシング（フレーム中に排除されたアイテム数）、フォント充足率（フォントキャッシュテクスチャーの全エリア中に占める全グリフのエリアのパーセンテージ）、フォントキャッシュ障害率（フレーム中に割り当てできなかった、またベクトル形状で置き換えられたグリフの数）、フレーム中にアップデートされたフォントキャッシュテクスチャーの数、現在のフォントテクスチャーの数。
- **Memory タブ:**このタブでは Scaleform フットプリント、Scaleform が要求しているメモリーの内どれだけを実際に使用するか、目的ごとにどのようにメモリーを割り当てるかをユーザーが詳細に決定できます。フルメモリーレポートモードでプロファイルする（AMP ツールバーの「i」ボタンでトグルする）場合、メモリーレポートはさらに詳細になり、ヒープごと、ファイルごと、目的ごとの内訳を表示します。
- **Images タブ:**各画像で消費しているメモリーを、そのディメンションとフォーマットと共に表示します。リストの各アイテムをクリックすると、その画像のサムネイルをプレビュー ウィンドウで表示します。画像のデータはプロファイル情報の一部としては送られず、要求に応じて送られることに注意してください。そのため、Scaleform がすでにアップロードした画像については画像が表示されない場合もあり得ます。AMP は現在 DXT 圧縮画像など全ての画像フォーマットのプレビューには対応していません。
- **Fonts タブ:**フォントキャッシュで消費する総メモリー量を、各 SWF のフォントリストと共に表示します。このタブのアイテムを選択すると、画像タブで画像がプレビューされるのと同じように現在のフォントキャッシュを表す画像が表示されます。
- **CPU タブ:**このタブはサブカテゴリーで Advance（詳細）、Display（ディスプレイ）、User times（ユーザー時間）に分割され、選択したフレームに対して使った時間のより詳細な概要を提供します。複数のフレームを選択した場合、値はこれらのフレームの平均値となります。
- **Functions タブ:**各関数で使用した時間を含むツリーコントロールを表示し、これには関数から呼び出された関数も含みます。複数のフレームを選択した場合、時間と呼び出し数はゲームフレームごとの平均値となります。ActionScript と C++ 関数は同じ呼び出しグラフにまとめられていますが、これはコードパスが ActionScript VM とその他のアプリケーションの間を動く可能性があるからです。各関数を展開すると、その関数から呼び出された関数を見ることができます。
- **ActionScript タブ:**このタブは Functions タブと同じような情報を表示しますが、ここでの情報は階層的ではなく、C++関数は含まれず、各関数での時間にはリスト中の他の関数での時間は含まれません。このタブは最もリソースを消費しているスクリプト関数はどれかを見るのに有用です。
- **Source ウィンドウ:**このウィンドウは Functions タブと ActionScript タブの一部です。AS3 の場合、ライン別のタイミングは常に AS ファイルに常駐するコードに対して有効です。SWF に埋め込まれている SA3 コードは、ソースコードが表示されません。。AS2 ではライン単位の時間取得はデフォルトで無効になっていて、有効にするには、ツールバーのドロップダウン メニューにある関数プロファイリング レベルを「High」に選択します。

AS2 での命令プロファイルリングは処理に時間を要し、プロファイル対象の Scaleform アプリケーションの動作を低下させるため、問題となる範囲をあらかじめ特定してから使うようにしてください。ある関数が 2 つ以上の関数から呼び出された場合、各ラインの隣りに表示される時間の合計は関数コールグラフ内で報告されている時間とは一致しません。その理由は、ライン単位タイミングは、どこからコールされたかには関係なく、関数内で費やされた合計時間を示しているためです。また、ライン タイミングは関数コール内部で費やされた時間は含みませんが、コールグラフ タイミングはコールされた関数内で費やされた時間を含みます。ソースコードとライン単位タイミングを表示するには、AMP サーバー (SWF の格納場所) 上に見つからない Flash デバッグ ファイルは AMP クライアントのワーキング ディレクトリ内に実行可能な状態として格納されていなければなりません。あるいは、AS3 ソース ファイルの場合は、ロードされた SWF と同じ相対パスに格納されていなければなりません。

- **File** メニュー：このメニューには以下のサブメニューがあります。
 - **Connection**：別のサーバーとの接続を確立する接続・ダイアログを開きます。
 - **Disconnect**：現在の接続を終了します。
 - **Debug Info Paths**：ファイルシステムの全てのパスをリストアップするダイアログを表示でき、ここでは AMP は AS2 では SWD ファイルと、AS3 では AS ファイルを検索します。パスは Add、Remove ボタンで追加または削除できます。
 - **Load Profile Frames**：以前に保存したプロファイル実行を再確認のために読み込みます。以前保存したプロファイル実行を再確認のために読み込みます。このメニューを選択するとファイル読み込みダイアログが表示されますので、プロファイルデータ ファイルを指定します。
 - **Save Profile Frames**：現在のプロファイル実行を外部記憶に保存します。保存したプロファイル実行は前述の「Load Profile Frames」サブメニューで再読み込みが可能です。このメニューを選択するとファイル保存ダイアログが表示されますので、プロファイル データを含んでいるファイルのパスおよび名前を指定します。
 - **Dump Memory Report**：AMP の作業ディレクトリにあるファイル "AmpMemReport.txt" に現在のプロファイルフレームを保存します。
 - **Exit**：アプリケーションを終了します。
- **View menu**：このメニューには次の選択肢があります。
 - **Graph Tooltips**：このメニューを選択した場合、AMP のグラフ上にマウスを置くと状況依存のポップアップヘルプがトグルされます。
 - **Time Units**：ここでは表示、詳細時間などの全ての時間量のレポート単位を選択できます。ミリ秒とマイクロ秒を選択できます。
 - **Memory Units**：Memory タブと Memory グラフでの単位を選択できます。バイト、キロバイト、メガバイトを選択できます。
- **Window menu**：このメニューでは上記のタブ (Log、SWF Info、Renderer、Memory、Images、Fonts、CPU、Functions、ActionScript) の何れでも表示、非表示を選択出来ます。ここにはまた Restore UI も選択でき、ユーザーがこれまでに AMP アプリケーションで行ったウィンドウのカスタマイズを削除します。
- **Help menu**：このメニューには現在次の 2 つのオプションがあります。User Guide では Scaleform.com のデベロッパーセンターでこの文書へのリンクがあり、About では AMP のバージョンとクレジットのスクリーンを表示します。
- **Application Control Toolbar**：プロファイル対象アプリケーションを制御する複数のボタンで構成される UI エレメントです。

-  **Wireframe** : ワイヤフレーム・モードでコンテンツをレンダリングするよう
に、プロファイル対象アプリケーションにリクエストを送信します。アプリケーシ
ョンがすでにワイヤフレーム・モードになっている場合は、このボタンをクリック
するとワイヤフレーム・モードはオフになります。
-  **Overdraw mode** : ピクセル・オーバードロー、マスク、フィルタの各領域
をハイライト表示するレンダリング・モードに移行するように、プロファイル対象
アプリケーションにリクエストを送信します。AMP サーバーは、ピクセル・オーバ
ードロー領域を緑（輝度はオーバードロー回数に比例）、マスク領域を赤、フィル
タ領域を青でレンダリングします。アプリケーションがオーバードロー モードで動
作している場合、このボタンをクリックすると、オーバードロー モードがオフにな
ります。
-  **Batch profiling mode** : それぞれのレンダー バッチを異なる色で描画する
レンダリング モードに移行するように、プロファイル対象アプリケーションにリク
エストを送信します。アプリケーションがすでにバッチ プロファイリング モードに
なっている場合は、このボタンをクリックすると、バッチ プロファイリング モード
がオフになります。
-  **Anti-aliasing** : アンチエイリアス・モードを循環するようにプロファイル対
象アプリケーションにリクエストを送信します。Scaleform のエッジ・アンチエイ

リアス法 (EdgeAA) 、ハードウェア・フルシーン・アンチエイリアス (HW

FSA) 、およびアンチエイリアスなし (None) の 3 種類のモードを循環します。

-  **Fast-forward** : Flash コンテンツを早送りで実行するようにプロファイル対象アプリケーションにリクエストを送信します。このモードにすると SFW はレンダリング・フレームと同じスピードで進み、ドキュメントの FPS 設定は無視されます。
アプリケーションがすでに早送りモードになっている場合は、このボタンをクリックすると早送りモードはオフになります。
 -  **Restart** : フラッシュコンテンツを再実行するようにプロファイル対象アプリケーションにリクエストを送信します。
-
- **Frame control ツールバー**: このツールバーには次のボタンがあります。
 -  **Clear** : 現在のプロファイル実行を破棄します。プロファイルフレームは廃棄され、これにより AMP クライアントメモリーに新しいデータを受け取る空きができるます。
 -  **First Frame** : 現在のプロファイル実行の先頭フレームを選択します、これにより最下部パネルのタブに表示されている情報をアップデートします。
 -  **Previous Frame** : 現在選択されているフレームのひとつ前のフレームを選択します、これにより最下部パネルのタブに表示されている情報をアップデートします。

-  **Pause** : メモリデータおよび性能データの送信を停止するように、プロファイル対象アプリケーションにリクエストを送信します。この操作を行ってもアプリケーション自体は停止しません。
-  **Next Frame** : 現在選択されているフレームの次のフレームを選択します、これにより最下部パネルのタブに表示されている情報をアップデートします。
-  **Last Frame** : 現在のプロファイル実行の最終フレームを選択してロックします。新しいフレームの情報を受信すると、つねに最終受信フレームを指すように選択が変わります、最下部パネルのタブに表示されている情報は継続的にアップデートされます。
- **Profiling ツールバー**: このエレメントにはプロファイル作成に使用するボタン群があります。
 -  **Detailed Profiling** (詳細なメモリーの内訳) : このオプションを選択した場合、AMP はフレームごとのより詳細なメモリー統計を収集し、これを Memory タブに表示します。このオプションを選択すると AMP は Images タブに、最大数の 10 個ではなく、画像の完全なリストを表示します。
 -  **Profiling Level**: 関数プロファイリングのレベルを、「Low」、「Medium」、「High」のいずれかに指定するドロップダウン メニューです。「Low」レベル プロファイリングを選択すると AMP は一部の上位レベル C++ 関数の時間のみを収集します。「Medium」レベル プロファイリングを選択す

ると、AMP は、「Low」レベル プロファイリングで対象となる C++ 関数に加えて、ActionScript 関数のプロファイリングを行います。「High」レベル プロファイリングを選択すると、AMP は一部の低位レベル C++ 関数の時間も収集します。さらに「High」レベル プロファイリングでは、ActionScript2 コードで費やされる経過時間をライン単位で取得して選択フレームのコードパネルに表示します。ソースライン単位での時間情報を見るにはデバッグ ファイル (SWD) が必要です。

ActionScript3 の場合はこれらタイミングは常に表示されます。

-  **Zoom ツールバー:** グラフの水平スケールを変更するドロップダウンコントロールがあります。ズームアウトすると問題箇所を見つけやすくなる一方で、ズームインするとその箇所のより詳しい様子がわかります。マウスホイールでもズームインおよびズームアウトが可能で、ドロップダウンメニューにはない高い倍率にも設定できます。ツールバーは現在選択されているフレーム番号も表示します。
- **Status バー:** プロファイルの下部には Status バーがあり、ここでは現在の接続とネットワーク経由で受け取る情報のビットレートを表示します。

AMP ではこの外見をカスタマイズできます。ツールバーとタブをドックする場所は、これらを新しい場所にドラッグして変更できます。タブとツールバーはこのアプリケーションからドック解除できます。グラフとタブのサイズは調節可能で、グラフは一部または全部を折りたためます。これらの UI の状態は AMP のプロファイルセッション中ずっと継続しますが、ウィンドウメニューの Reset UI を選択してデフォルトの状態にジオメトリーをリセットできます。

2 AMP の使い方

AMP プロファイルを使うには、メインとなる Scaleform インストール・ディレクトリ下の Bin/AMP にある *GFxAmpClient.exe* 実行ファイルを起動します。

2.1 コネクションの確立

クライアントの起動時にコネクション・ダイアログが自動的に表示されます。検出された AMP サーバーリストから選択するか、IP アドレスとポートを入力して指定します。プロファイル対象アプリケーションが起動中といった理由で接続すべき AMP サーバーがまだ起動していない場合は後者を選択してください。

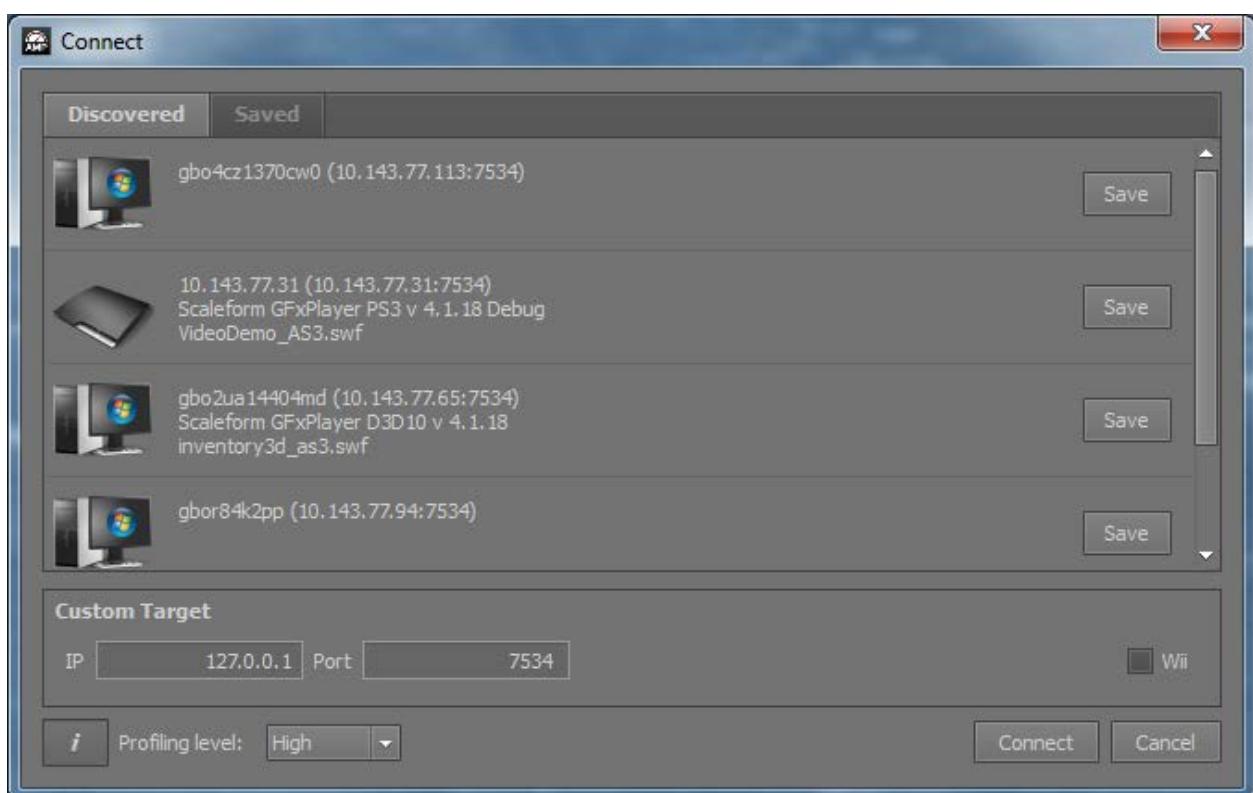


図 2 : AMP コネクション・ダイアログ

コネクション・ダイアログを任意のタイミングで開くには File メニューにある Connection サブメニューをクリックします。検出リスト中の各 AMP サーバーはプラットフォームごとのアイコンで表示されます（パソコン、Xbox360、PS3 など）。アイコンの隣りには、アプリケーションのタイトル、IP アドレスとリスン・ポート、および現在再生中の Flash コンテンツが表示されます。これらの情報の一部が表示されない場合は、該当する AMP サーバーがそのような情報を送信していないことを意味します。アプリケーション情報やコンテンツ情報のブロードキャストを含む AMP のフルサポート機能を任意の Scaleform アプリケーションに追加する方法については、このドキュメントで後述しています。コネクションを確立するには、検出サーバーをダブルクリックするか、サーバーを選択して「Connect」ボタンをクリックします。

接続ダイアログから接続時にプロファイルレベルと詳細なメモリー報告を選択できますので、接続と同時にこれらが確実に設定されるようにできます。このことは Scaleform のアプリケーションがプロファイルされる場合に特に重要です。

Wii 開発キット (NDEV) へのコネクションには上述のサーバー検出方法は使用しません。代わりに、NDEV に同梱される USB インタフェース (COM ポート) を使って Wii をパソコンに接続します。コネクション・ダイアログから Wii AMP サーバーへは、IP アドレス欄に「wii」と指定して接続します。

2.2 性能指標の解析

コネクションが確立されると、プロファイル情報がクライアントに送信され、グラフの描画が始まります。最上部グラフには、Scaleform のレンダリング (Display) 、Action Script の実行 (Advance) 、および Direct Access API (User) 実行に費やした CPU 時間がフレーム単位で表示されます。

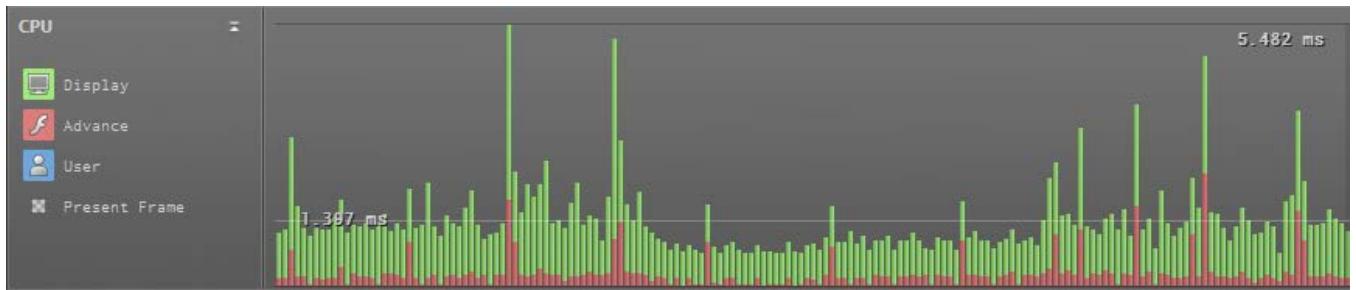


図 3: CPU グラフ

解析中に CPU のグラフを見る必要が無いときは、これを最小化またはサイズ変更してその他の UI エレメントに場所を割くことができます。グラフで表示されている数量の内関心の無いものがあれば、これはそのレジェンドの横にあるアイコンをクリックして削除できます。

CPU グラフに関連して、プロファイルバーの下部には CPU サマリー、関数、ActionScript タブがあります。CPU Summary タブにはグラフ情報がより詳細に表示され、Function タブには選択フレームのコールグラフが表示され、ActionScript タブには選択フレーム中に呼び出されたすべてのスクリプト関数が非階層形式で表示されます。

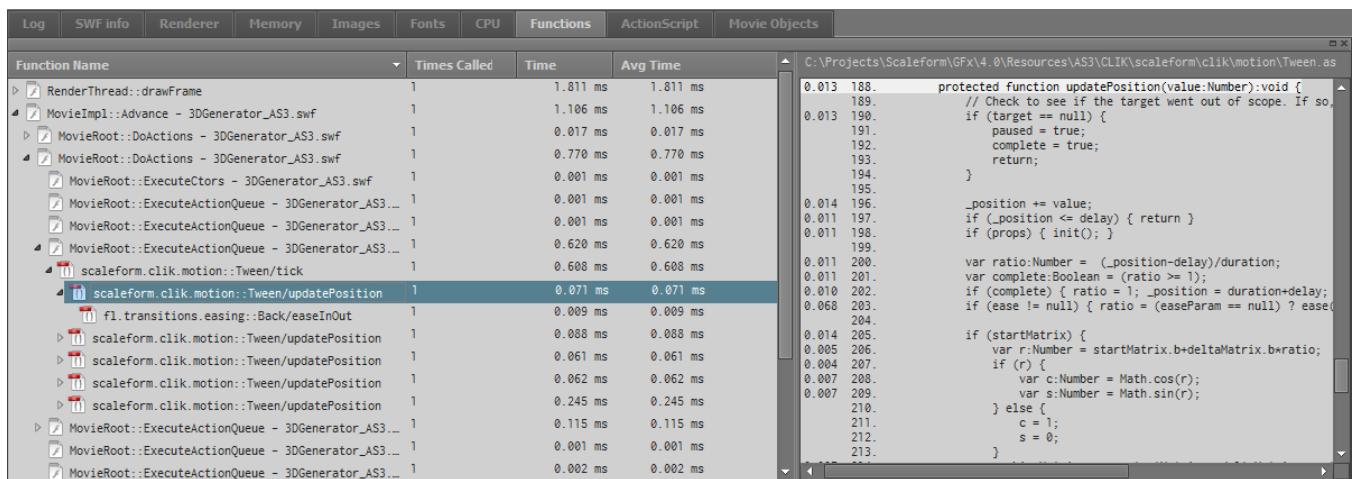


図 4 : CPU Summary タブ

CPU グラフ内にスパイクが存在する場合、CPU Summary タブを確認することでスクリプト性能のボトルネックの原因を突き止められる場合があります。一方、レンダリング性能のボトルネックは、レンダリング・グラフを調べるだけで判別できます。

2.3 レンダリング指標の解析

レンダリング・グラフには、選択フレームのレンダリング中に発生した描画プリミティブ数、トライアングル数、ライン数、使用マスク数、ストローク数、グラディエント・フィル数、メッシュ・キヤッッシュ内のグリフ数、およびフォント更新数が表示されます。性能向上を図るにはこれらの数をできるだけ少なくすることが望ましく、その詳細は [Best Practices Guide](#) で説明しています。

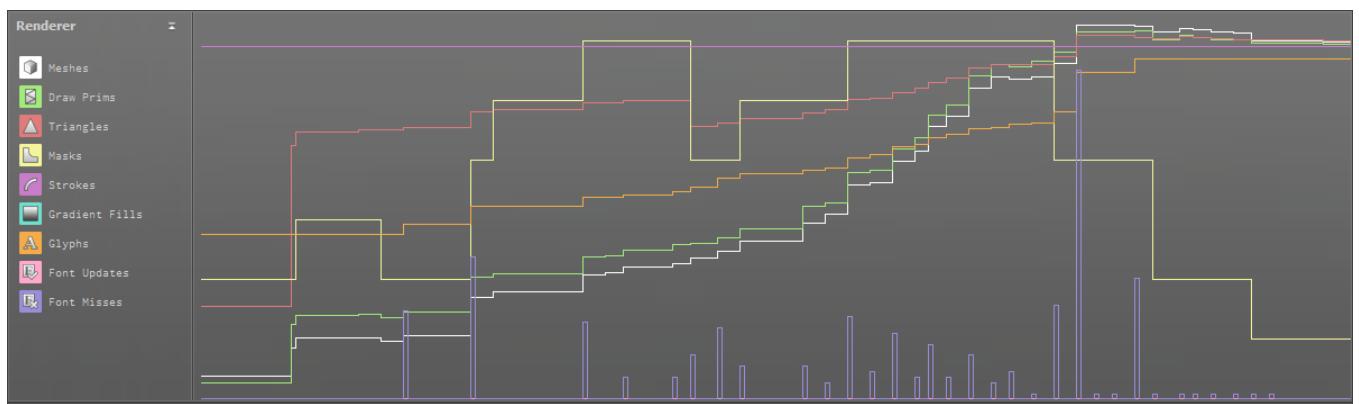


図 5: レンダリング・グラフ

レンダリンググラフに関連して、プロファイラーの下部にはレンダーサマリータブがあります。

Log		SWF info	Renderer	Memory
Meshes:	182			
Draw Primitives:	183			
Triangles:	5124			
Masks:	6			
Filters:	0			
Strokes:	21			
Gradient Fills:	0			
Rasterized Glyphs:	412			
Font Cache Updates:	0			
Font Textures:	1			
Font Cache Thrashing:	0			
Font Cache Fill:	28%			
Font Cache:	1048576 pixels			
Font Cache Used:	298294 pixels			
Font Failures:	0			
Font Cache Misses:	5			
Mesh Thrashing:	0			

図 6: Rendering Summary タブ

2.4 メモリ消費の解析

上から 3 番目のグラフは Scaleform システム、4 番目のグラフはグラフィックスのメモリー使用量を示します。

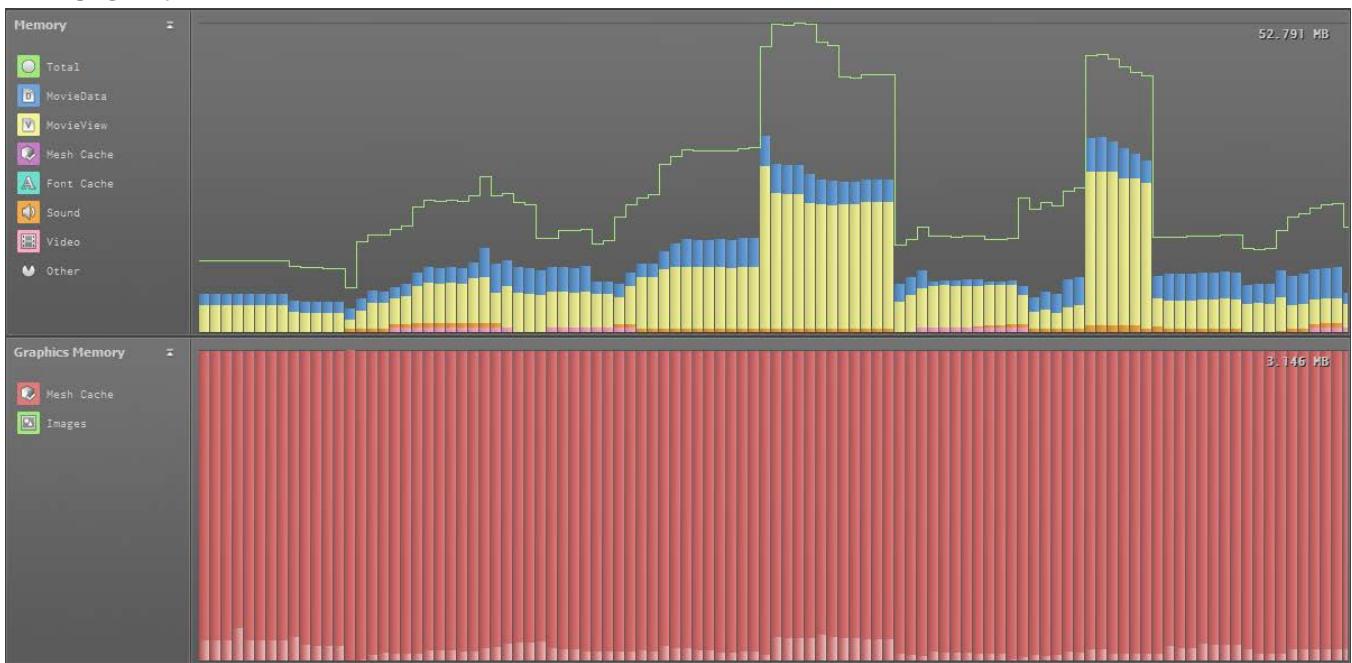


図 7: メモリ・グラフ

- **Total** : Scaleform が要求したトータルメモリです（フットプリント）。トータルメモリは線グラフとして描画され、以下にリストされるすべての項目と、オーバーヘッド、要求アロケーションの粒度、およびフラグメンテーションを含みます。いずれの項目にも未使用メモリは表示されないため、トータルメモリは使用済みメモリカテゴリの合計よりも大きくなります。
- **MovieData** : すべてのロード済み SWF ファイルまたは GFX ファイルによって占有されているメモリです。コンプレックス・ベクターグラフィクス・オブジェクト、エンベデッド・フォント、およびタイムライン・アニメーションはこの値に影響を与えます。
- **Movie view** : GFx::Movie インスタンスによって占有されているメモリで、タイムライン・アニメーションのサポート、オンスクリーン・オブジェクト、および ActionScript に割り当てられています。
- **MeshCache** : キャッシュされたシェイプメッシュ・データによって占有されているメモリで、ベクター・テッセレータとエッジ・アンチエイリアスによって生成されます。プラットフォームによってシステムまたはグラフィックスメモリーからメッシュキャッシュメモリーを割り当てるすることができます。実際に使用されるメモリーは暗めの色で表示されますので、より小さいメモリーキャッシュを設定できることが容易にわかり、メモリーの節約ができます。
- **Font Cache** : フォントキャッシュによって使用されているメモリです。
- **Sound** : エンベデッド・サウンドデータが使用するメモリです。エンベデッド・サウンド・サンプリングデータの個数、長さ、音質によって左右されます。
- **Video** : ビデオメモリ再生バッファが使用するメモリです。このバッファは再生開始で割り当てられ停止で解放されます。

- **Other** : Scaleform で使用中だが、上記カテゴリーに含まれないメモリー。メモリーのさらに詳細なカテゴリーの内訳は以下に述べるように詳細なメモリーレポートをトグルしてください。
- 画像（テキスチャー）にはグラフィックスメモリーを使用します。

メモリシステムの詳細と、設定、最適化、管理の各方法については、[Memory System Overview](#) ドキュメントを参照してください。

“Used Space”は GFx がアクティブに使用しているメモリーを表します。メモリータブの全ての値は、ビューメニューのメモリー単位選択で指定した同じ単位（バイト、キロバイト、またはメガバイト）です。。 Debug Data は GFx 内部でメモリーの追跡とメモリー、動作の統計用に使用しているメモリーを表します。 Memory タブには上記メモリーカテゴリの詳細が表示されます。特記のない限りすべての値の単位はバイトです。

AMP ツールバーの i ボタンで起動されるフルメモリーレポートモードでプロファイリングする場合、メモリーレポートがより詳しくなり、ヒープ別、ファイル別、目的別に区分して表示されます。 Scaleform ライブラリのデバッグバージョンは、ヒープ割り当てをメモリーカテゴリに分割します。このため、ユーザーは、Scaleform がメモリーを消費する方法を理解しやすくなります。メモリーカテゴリーは以下のようにレポートされます。

- **Renderer:** Scaleform レンダリングエンジンで割り当てられるメモリー。このカテゴリーは、以下のサブカテゴリーに分類されています。
 - **Buffers:** メッシュステージングバッファに割り当てられているメモリー。生成したメッシュがインスタント化されてバーテックスやインデックスバッファにコピーされる前に保管する一時キャッシュとして機能する
 - **RenderBatch:** 同時に描画されるレンダリングツリーのノードに関連するメッシュ、プリミティブ、またはレンダリングコマンドなどのエンティティのコレクションに割り当っているメモリー
 - **Primitive:** 同時に描画されるバーテックスコレクションを保持するためのメモリー
 - **Fill:** ハードウェアファイルで説明されるデータを保管および管理するためのメモリー
 - **Mesh:** メッシュキャッシュに割り当てられているメモリー。メッシュキャッシュは、バーテックスおよびインデックスデータのレンダラーをフィードする。
 - **MeshBatch:** メッシュキャッシュのバーテックスおよびインデックスバッファにキャッシュされた個別のデータの塊に割り当てられているメモリー
 - **Context:** レンダリングツリーの生存期間を管理するレンダリングサポートシステムが消費するメモリー。レンダリングツリーの論理的に独立したスナップショットを複数保持して、スレッドが互いに干渉しなくてもデータをレンダリングできるようにする
 - **NodeData:** レンダリングされたオブジェクトに適用されている現在のマトリックスや効果などのフレーム情報用のメモリー
 - **TreeCache:** レンダラーのシングラフに割り当てられているメモリー。レンダリングされたオブジェクトのツリー構造になっている
 - **TextureManager:** テクスチャ画像を管理するためのメモリー。実際の画像データは通常システムメモリーに保管されないため、このメモリーカテゴリにいれられている
 - **MatrixPool:** レンダラーが維持するマトリックスのプールが消費するメモリー
 - **MatrixPoolHandles:** マトリックスハンドラーの配列が消費するメモリー

- **Text:**テキストの管理とレンダリングに割り当てられているメモリー（フォーマット情報、スタイル、色情報、html構文解析、リッチテキストデータ、ラインバッファなど）
 - **Font:**フォントのレンダリングに関連するメモリー（グリフラスタライズ、フォントキャッシュストレージなど）
- **MovieDef:**ムービーエレメントのテンプレートを表す不变データに割り当てられているメモリー。以下のサブカテゴリがある
 - **CharDefs:**文字定義に使われるメモリー（ボタンやテキストなど）
 - **ShapeData:**形状定義に使われるメモリー
 - **Tags:**ActionScript タグに使われるメモリー
 - **FONTS:**フォントリソースに使われるメモリー
 - **Sounds:**サウンドデータに使われるメモリー
 - **ASBinaryData:**ActionScript バッファデータが消費するメモリー
 - **MD_Other:**その他の MovieDef データ（上のカテゴリを含まない）
- **MovieView:**タイムラインアニメーションサポート、オンスクリーンオブジェクト、ActionScript に割り当てられているメモリー。さらに以下のサブカテゴリに分類される
 - **MovieClip:**オンスクリーンアニメーションオブジェクトに使用されるメモリー
 - **ActionScript:**ActionScript コードおよびデータが消費するメモリー
 - **ASString:**ActionScript スクリプトが消費するメモリー
 - **Text:**テキスト関連のディスプレイオブジェクトが消費するメモリー
 - **XML:**XML オブジェクトが消費するメモリー
 - **MV_Other:**上述のカテゴリに当てはまらない Movie View メモリー
 - **VM:**ActionScript バーチャルマシンに割り当てられているメモリー。さらに以下カテゴリに分類される
 - **AS3 VM:**ActionScript 3 バーチャルマシンが消費するが、以下のサブカテゴリには属さない
 - **CallFrame:**ActionScript 実行時のコールスタックにある各エントリーに割り当てられているメモリー
 - **VTable:**ActionScript バーチャルテーブルが消費するメモリー
 - **SlotInfo:**ActionScript クラスメンバーを説明するストラクチャが使用するメモリー。各 ClassTraits および InstanceTraits には クラスまたはインスタンスのメモリーレイアウトを説明する SlotInfo ストラクチャがある。ClassTraits および InstanceTraits については以下の参考のこと。
 - **SlotInfoHash:**ActionScript のプロパティ検索を高速にするためのハッシュテーブルが使用するメモリー
 - **ClassTraits:**ActionScript クラスのメモリー配列を維持するために割り当てられているメモリー
 - **Class:**ActionScript オブジェクトの静的メンバーを維持するために割り当てられているメモリー
 - **InstanceTraits:**ActionScript オブジェクトの非静的メモリー配列を維持するために割り当てられているメモリー
 - **Instance:**ActionScript オブジェクトのインスタンス変数とメソッドを維持するために割り当てられているメモリー
 - **AbcFile:**バイトコードを表すデータ構造とその対応するデータに割り当てられているメモリー
 - **AbcConstPool:**メンバー、ストリング、名前空間、マルチネームなどの定数を保存するために使用されるメモリー。これは単純なバイトコードであるため、上述の AbcFile の一部でもある

- **VMAbcFile:**AbcFiles で生成したバイトコードを最適化して保管する内部データ構造を保持するメモリー。ファイルおよび別の情報が独立している
- **Tracer:**コード最適化に割り当てられているメモリー
- **IME:**Input Method Editor (IME) サポートが消費するメモリー

Log	SWF info	Renderer	Memory	Image
Memory			Size	
▲ Total Footprint			4849.402 KB	
▲ Used Space			4633.715 KB	
▷ Video Heaps			0.000 KB	
▷ Other Heaps			325.621 KB	
▲ Movie View Heaps			1226.094 KB	
▲ MovieView "3dgenerator_as3.swf"			1226.094 KB	
String			0.828 KB	
▷ Renderer			12.672 KB	
▷ MovieView			1176.781 KB	
General			35.812 KB	
▷ Movie Data Heaps			261.066 KB	
▲ Global Heap			2820.934 KB	
Video			264.176 KB	
String			20.633 KB	
Sound			0.141 KB	
▷ Renderer			2310.914 KB	
▷ MovieView			8.039 KB	
▷ MovieDef			116.676 KB	
Image			0.141 KB	
IME			9.914 KB	
General			90.301 KB	
▷ Debug Memory			8.000 KB	
▷ Unused Space			38.820 KB	
Debug Data			176.867 KB	

図 8: Memory

各カテゴリを展開すると、サブカテゴリか、（もし存在すれば）そのカテゴリで直接アロケーションを担っているメモリヒープが表示されます。展開を繰り返していくことで、特定フレームにおける Scaleform メモリの利用状況をより詳細に把握することができます。

画像タブはまたメモリープロファイルに関連づけられており、選択されたフレーム（複数可）にコードされた画像をそのサイズと共に表示します。

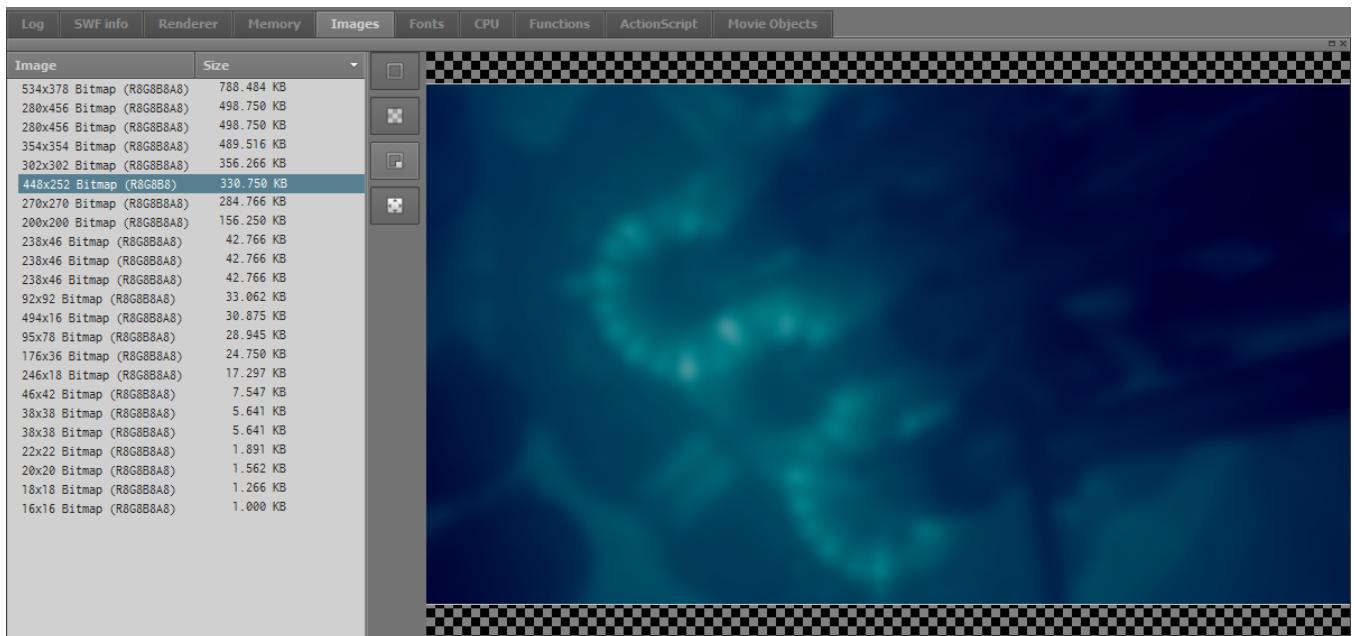


図 9: Images メモリタブ

このタブの情報から、対象とする Scaleform アプリケーションのメモリ消費が大きなイメージによって増大しているかどうかがすぐに判断できます。イメージリスト内のいずれかの項目をクリックすると、プレビュー ペインにイメージのサムネイルが表示されます。なおイメージ データは、要求があれば送信されますが、プロファーリング情報の一部として送信されることはありません。そのため、そのイメージが Scaleform によってアップロードされている場合は表示されない可能性があります。

メモリープロファイル関連の最後のタブは Fonts タブで、ここでは選択されたフレーム（複数可）のフォントに関する情報を表示します。



図 10: Fonts メモリタブ

このタブの情報から、各フォントタイプのエンベデッド・グリフの個数が SWF ファイル単位で分かれます。

エンベデッド・フォントは大きなメモリ容量を消費し、AMP では MovieData ヒープと CharDefs カテゴリのサイズ増加として現れます。これらは Heaps タブおよび Memory Summary タブからそれぞれ確認することができます。複数の SWF ファイル間でフォントを共有すれば、フォントメモリを節減できます。

2.5 オーバードロー プロファイリングの使用

プロファイル対象アプリケーションで問題が存在する可能性のある領域をハイライトする特殊なレンダリング・モードを、リモートから有効にすることができます。このレンダリング・モードをト

グルによって有効にするには、アプリケーション・コントロール・ツールバーにある該当ボタンをクリックします。

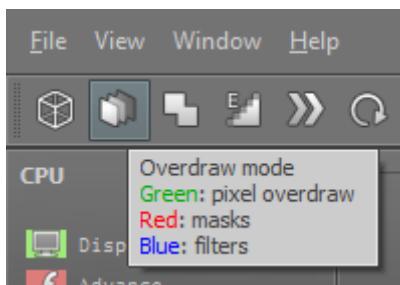


図 11: トグルボタン

このモードを有効になるとすべてのオブジェクトは緑色でレンダリングされます。緑はピクセル・オーバードロー（16 レイヤーでフル輝度）、マスクは赤、フィルタは青（マスクとフィルタは 8 レイヤーでフル輝度）です。これらは混色で表示されるため、たとえば明るい黄色は、多段のオーバードローとネストありマスクを示します。

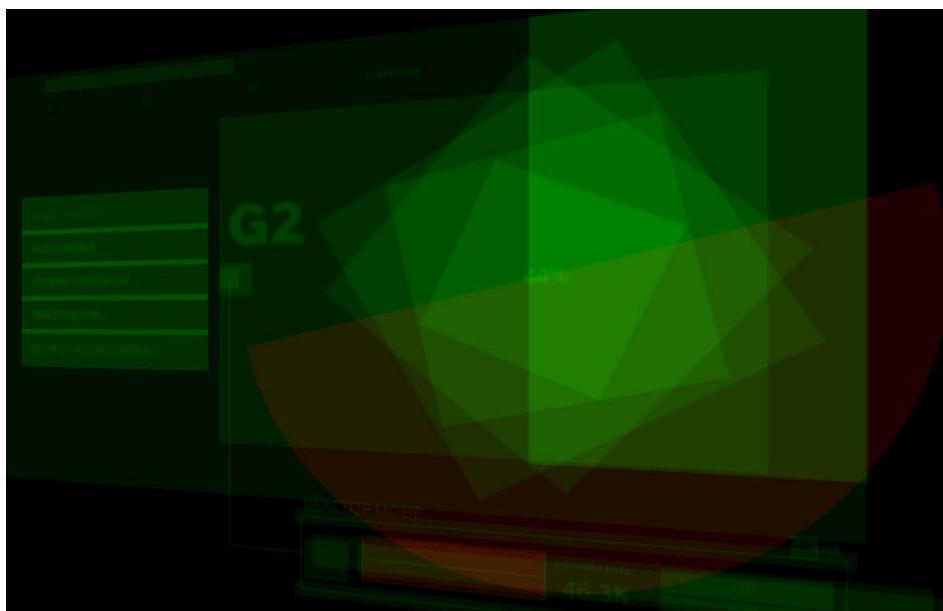
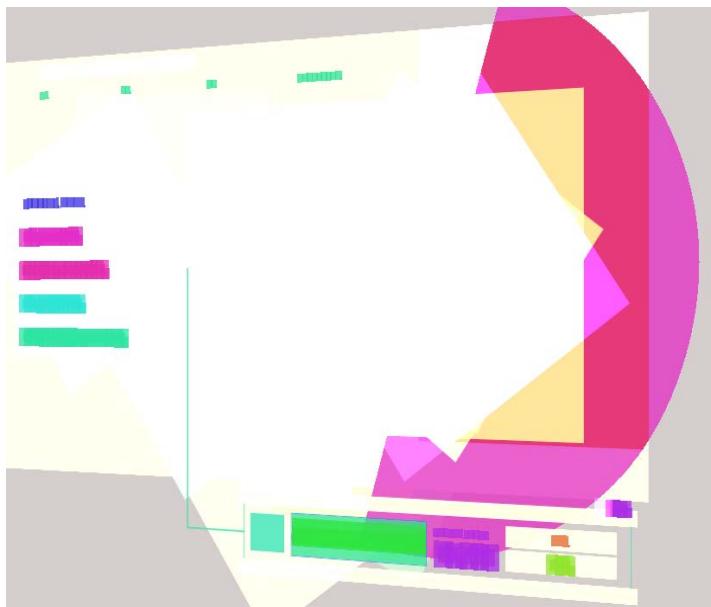


図 132: ハイライトレンダリング・モード

オーバードロー、マスク、およびフィルタのすべての性能に影響しますので、使用はできるだけ最小限に抑えてください。

2.6 バッチ プロファイリングの使用

AMPではバッチ プロファイリングという専用のレンダリング モードをサポートしています。このレンダリング モードを有効または無効にするには、アプリケーション ツールバーの該当ボタンをクリックします。



このモードを有効にすると各レンダリング バッチは異なる色でレンダリングされます。性能を高めるにはバッチ数をできるだけ減らす必要があります。

3 プラットフォームとインテグレーションに関する注意

1. AMP は通常 GFx::System オブジェクトの構築時、または GFx::System::Init 関数呼び出し時に初期化されます。また、Scaleform の初期化に Scaleform::System を使用した場合は、明示的に AMP を初期化、シャットダウンする必要があり、これにはそれぞれ GFx::AMP::Server::Init()、GFx::AMP::Server::Uninit()を呼び出します。
2. AMP サーバーは、現時点で、Windows、Xbox360、PS3、Linux、Mac、Wii、Android および iPhone/iPad、3DS、PS Vita、WiiU をサポートしています。AMP クライアントの自動検出機能を使えば接続は簡単です。自動検出を使わずに接続するには AMP サーバーの IP アドレスの知識が必要で、一部のプラットフォームは複数 IP アドレスを有するため、混乱しないよう注意が必要です。
 - Xbox360 の場合、デバッグ IP アドレスではなくタイトル IP アドレスを指定してください。タイトル IP アドレスは Xbox360 Neighborhood に同梱される開発キットまたはテストキットのプロパティを確認すれば分かります。
 - PS3 で開発キットが 2 個の IP アドレスを使用するように設定されている場合、デバッグ用に使用する IP アドレスではなく、アプリケーション IP アドレスを指定してください。
 - Android ではインターネットのパーミッションをアプリケーション レベルで有効にする必要があります。以下の記述を該当アプリケーションの AndroidManifest.xml に追加してください。

```
<uses-permission android:name="android.permission.INTERNET" />
```
 - WiiU への接続は WiFi 接続ではなく、ホスト PC ブリッジ経由で行ってください。ホスト PC はポート 6003 の自動検出でコンソールを検出します。接続要求があれば、通信は TCP/IP ソケットからホストブリッジへと切り替えられます。
 - Wii 開発キット (NDEV) へのコネクションは上述のサーバー検出方法は使用しません。代わりに NDEV に同梱される USB インタフェース (COM ポート) を使って Wii をパソコンに接続します。コネクションダイアログから Wii AMP サーバーへは、IP アドレス欄に「wii」と指定して接続します。重要 : hio.dll は AMP にロードされる可能性があるので RVL_SDK/X86/bin が PATH 環境内にあることを確認してください。
3. Unreal Engine 3 など一部のゲームエンジンは、自身の素材をパッケージ化するため、プロファイル対象の Flash コンテンツに対応する AS2 SWD ファイルを AMP サーバーが配置することができません。このような場合は、AMP サーバーに SWF と同じディレクトリ上から SWD ファイルを発見させてネットワーク経由で送信するのではなく、SWD ファイルを AMP クライアントのワーキングディレクトリに配置してください。

4 ビルドに関する注意

- AMP はデフォルトではすべての非シッピング・ビルドに組み込まれます。Scaleform から AMP を削除するには、GFXConfig.h 内で GFX_AMP_SERVER が未定義なことを確認し、Scaleform ライブラリをビルドします。Scaleform のソースにアクセスできない場合は、以下のコールによって AMP をランタイムにて無効化することができます。

```
Ptr<GFX::AMP::ServerState> serverState = *SF_NEW GFX::AMP::ServerState();
serverState->StateFlags &= GFX::AMP::Amp_Disabled;
AmpServer::GetInstance().UpdateState(serverState);
```

- AMP はプロファイラとの通信にネットワーク・ソケットを使用します。そのため、該当アプリケーションがプラットフォーム固有のネットワーク・ライブラリと正しくリンクされていることを確認してください（Windows なら Ws2_32.lib、Xbox360 なら Xnet.lib、PS3 なら libnetctl、Linux ならソケット、など）
- ネットワーク・ソケットを使うシステムが AMP だけの場合、AMP はソケット・ライブラリを初期化し終了時に解放します。一部のプラットフォームは、初期化回数のほうが解放回数よりも多い場合でも、解放してもライブラリには影響しません。この場合は問題ありません。ただし一部のプラットフォームでは、ライブラリの初期化回数のほうが解放回数よりも多い場合に、ネットワーク・ライブラリの解放はすぐに影響を及ぼし、アプリケーションの他の部分で使用しているソケット接続に対して AMP が干渉を与えるおそれがあります。この場合、ソケット初期化をスキップして過去に初期化されたソケット・ライブラリを使うように、以下のコールによって AMP の動作を強制しなければなりません。

```
AMP::Server::GetInstance().SetInitSocketLib(false)
```

このコールは AMP の初期化が完了する前に実行されなければなりません。

- AMP のメモリヒープは上限を設定することができます。リミットを超過すると、保留中のメッセージがプロファイラに送信されるまで、Scaleform は実行を一時的に停止します。上限は以下のコールで設定します。

```
AMP::Server::GetInstance().SetHeapLimit()
```

- AMP サーバーはデフォルトでポート 7534 にリスニング・ネットワークを作成します。異なるポートを指定するには以下をコールします。

```
AMP::Server::GetInstance().SetListeningPort()
```

- プロファイラ・クライアントとのコネクションが確立されるまで AMP の実行を一時的に停止するように設定することができます。起動中の統計情報を取得する場合に便利な機能であり、以下のコールを起動時に発行します。

```
AMP::Server::GetInstance().SetConnectionWaitTime
```

5 AMP サポートの追加

任意の Scaleform アプリケーションを AMP のリモート・プロファイリングに対応させるには、このセクションで概要を述べる以下のステップを実行します。AMP サーバーに全面的に対応した例としては Scaleform Player ソースを参照してください。

5.1 Debug ファイルの生成

AMP は ActionScript 2 Flash デバッグ情報（SWD ファイル）を使ってソースコードとライン単位のタイミングを表示します。Flash CS3 または CS4 からデバッガ（Ctrl+Shift+Enter）を起動して SWD ファイルを生成してください。SWD ファイルは、対応する SWF ファイルと同じディレクトリか、AMP クライアントのワーキングディレクトリに配置してください。

ActionScript 3 では SWD ファイルはありません。また、デバッグ情報は SWF ファイルに直接埋め込まれています。ソースコードを表示するには AS ファイルが必要です。Flash Studio からデバッガを実行して、デバッグ SWF ファイルを生成してください。

5.2 アプリケーションのコントロール

ワイヤフレーム モード、バッチプロファリング モードなど、プロファイル対象アプリケーションの設定をリモートから設定することができます。ほとんどの設定はアプリケーション固有であり、アプリケーションによって実装されます。これら機能が実装されていない場合は、単純に AMP クライアントから利用できません。

AMP から送信される app-control メッセージを扱うには以下の手順に従ってください。

- AMP::AppControlInterface に由来するカスタム app-control コールバック・クラスを実装し、AMP::MessageAppControl メッセージを AMP から扱えるように、HandleAmpRequest メソッドを上書きします。
- 起動時に AMP::Server::GetInstance().SetAppControlCallback をコールしてカスタムハンドラをインストールします。
- AMP::Server::GetInstance().SetAppControlCaps をコールしてサポートしている機能を AMP に知らせます。または、サポート機能を true にセットした AMP::MessageAppControl メッセージを使います。たとえば、

```
AMP::MessageAppControl caps;
caps.SetCurveToleranceDown(true);
caps.SetCurveToleranceUp(true);
caps.SetNextFont(true);
caps.SetRestartMovie(true);
caps.SetToggleAaMode(true);
caps.SetToggleAmpRecording(true);
caps.SetToggleFastForward(true);
caps.SetToggleInstructionProfile(true);
caps.SetToggleOverdraw(true);
caps.SetToggleBatch(true);
```

```

caps.SetToggleStrokeType(true);
caps.SetTogglePause(true);
caps.SetToggleWireframe(true);
AMP::Server::GetInstance().SetAppControlCaps(&caps);

```

5.3 コネクション・ステータス

AMP クライアントのステータスバーには接続アプリケーションの名前が表示されます。この情報は AMP::Server::GetInstance().SetConnectedApp をコールして取得します。また、AMP クライアントは現在のアプリケーションのステータスを表示しますが、そのためにはアプリケーションのステート変化に伴って通知が必要です。ステートの更新には以下の方法を使います。

- 先頭の引数を次の ServerStateType エニュメレーション・タイプのうちのいずれかに設定し、AMP::Server::GetInstance().SetState を発行します
 - Amp_RenderOverdraw
 - Amp_App_Wireframe
 - Amp_App_Paused
 - Amp_App_FastForward
- AMP::Server::GetInstance().SetAaMode
- AMP::Server::GetInstance().SetStrokeType
- AMP::Server::GetInstance().SetCurrentLocale
- AMP::Server::GetInstance().SetCurveTolerance
- または、AMP::ServerState オブジェクトを上記のすべての情報で満たし、AMP::Server::GetInstance().UpdateState に渡します

5.4 マーカー

プロファイル対象アプリケーション内でコールされた C++ または ActionScript コールに対応する CPU グラフの位置に、マーカーと呼ぶ特別なインジケータを表示することができます。ActionScript でマーカーを追加する方法は次のとおりです。

```
Amp.addMarker(1)
```

備考 : Amp.addMarker() コードを含む.as ファイルをコンパイルするとコンパイルエラーが発生することがあります。このエラーを回避するには Resources/AS2/CLIK ディレクトリ下で与えられるイントリンシック Amp クラスを使用してください。

C++ でマーカーを追加する方法は次のとおりです。

```
GFx::MovieImpl:: AdvanceStats->AddMarker(1)
```

このような変更によって、コールが実行されたフレーム上にマーカーが表示されます。将来は複数のマーカーを表示するために整数の引数をサポートする予定です。

6 AMP 関連の FAQ

このセクションにはデベロッパーの方が AMP のツールを使われるときに良くあるご質問をいくつか記載します。

1. AMP を動作させるのにゲーム側で最低限必要な要件は何ですか。

AMP がインープルされたビルド（GFX_AMP_SERVER が定義されている）。デフォルトでは、出荷されている GFx のビルドは全て AMP がインープルされています。また、ゲームが Platform::RenderThread::drawFrame を呼び出さない場合は、AMP が各フレームでアップデートされるように AmpServer::GetInstance().AdvanceFrame を呼び出してください。

2. AmpClient を使用してゲームに接続しようとすると、ゲームのフレームレートが非常に低くなってしまいます。AmpServer::AdvanceFrame() に多くの時間が取られています。これは通常の動作ですか。

多くのプロファイル同様、AMP に接続しているときは速度のペナルティーがあるのは普通です。速度への影響を軽減するには低いプロファイルレベルから始めてください。これは AMP クライアントツールバーの適切なドロップダウンコントロールから選択して設定します。こうしてどこで速度が落ちているのかを特定できたら、そこでのプロファイリングレベルを増加してさらに詳細に原因を決定してください。また、各フレームでの完全なメモリーレポートをリクエストすることも速度を遅くする大きな原因となりますので、こうしたレポートは実際に必要な時にだけ生成されるようにしてください。詳細なメモリーレポートをトグルするには AMP クライアントツールバーで小文字の「i」をオン、オフしてください。

3. ウィンドウズ、PS3、Xbox でゲーム実行中に、ゲームがそのディスカバリーインドウに AMP クライアントを発見した後でこれに接続できますが、統計は全て空白です。

AMP は、各フレームで情報を受け取らなければメモリーとパフォーマンスの統計を表示しません。もしゲームが Platform::RenderThread::drawFrame を呼び出さない場合は、AMP が各フレームでアップデートされるように AmpServer::GetInstance().AdvanceFrame を呼び出してください。

4. AMP がアップデートされるときにどうして 1 秒かそれ以上ポーズするのですか (AMP::Server::WaitForAmpConnection)。

AMP サーバーはこれがメモリーをたくさん消費しすぎる時には通常 1 秒間ポーズして、蓄積されたデータがクライアントに送られるようにします。これが継続して起こる場合、ユーザーのコンテンツにプロファイリング情報が多過ぎるため、Scaleform をポーズしなければ AMP がこれらを全てクライアントに送る時間がない状態になることがあります。

送るプロファイリング情報の量を削減するには、ソースラインタイミングと詳細メモリーリポート (AMP クライアントの「i」ボタン) をトグルしてオフにしてください。また、AMP クライアントで「profile level」のドロップダウンを低くすることもできます。最後に、この問題をトリガーするメモリースレッショルドを増加することもできます。このデフォルト値は 1MB に設定されていますが、これは GFx::AMP::Server::SetHeapLimit を呼び出して変更できます。

5. プロファイラー使用時にはどれほどのメモリーオーバーヘッドがありますか。

デフォルトでは AMP ヒープの限度は 1MB ですが、これはハードリミットではないので、少々超過しても構いません。ゲーム中のこの限度は AmpServer::SetHeapLimit を呼び出して変更可能です。限度が大きいほど、データが AMP クライアントに送られている最中に GFx がポーズされなければならない可能性は低くなります。

7 追加情報

AMP のより詳しい情報については以下のリソースを参照してください。

- [AMP Use Cases](#) ドキュメント
- [Scaleform Reference](#) ドキュメント – コードとクラスの詳細
- [AMP Forum](#) – 質問やサポートが可能なコミュニティ