DAIKIN

One More Solution

qube! クイックガイド *Maya Jobtype・コマンドライン編*

第6.0版 – v7.0.x対応

ダイキン工業株式会社

ご注意

このクイックガイドは、本製品の使用許諾契約書に基づいて使用することができます。
製品に付属するすべての資料の全部または一部を、ダイキン工業株式会社の書面による許可を得ることなく複写、複製、転用することはできません。
記載内容は、予告なく変更することがあります。

～ 目次 ～

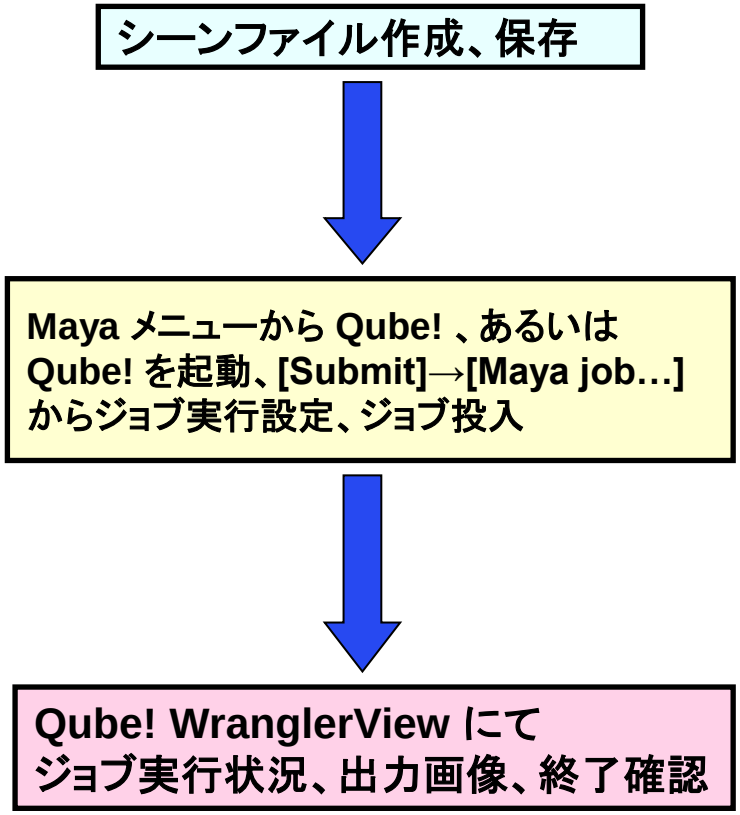
1. はじめに

本ガイドは、Qube! クイックユーザーガイド Maya編です。

詳細につきましては、Maya については Mayaオンラインヘルプ、また Qube! に関しましては、http://docs.pipelinefx.com 、または、GUI の [Help]→[Qube User Guide](pdf) 、または、[スタート]→[プログラム]→[PipelineFX]→[User Manual](pdf)を参照ください。

2. ジョブ投入フロー (Maya編)

ジョブ投入フローの概要は以下の通りです。詳細説明については、左側で表示しましたページにてご覧いただけます。



Maya メニューでの操作 → 6 ～ 23ページ

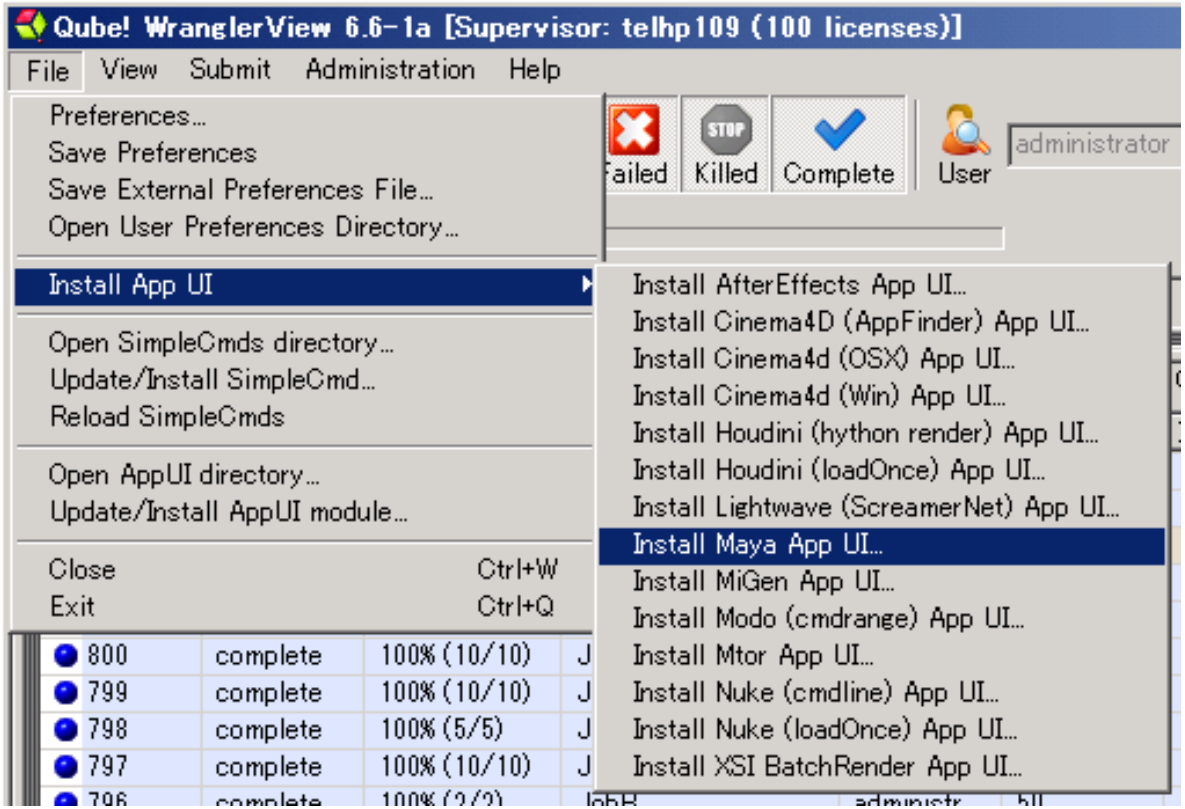
Qube! からの操作 → 24 ページ

25ページ

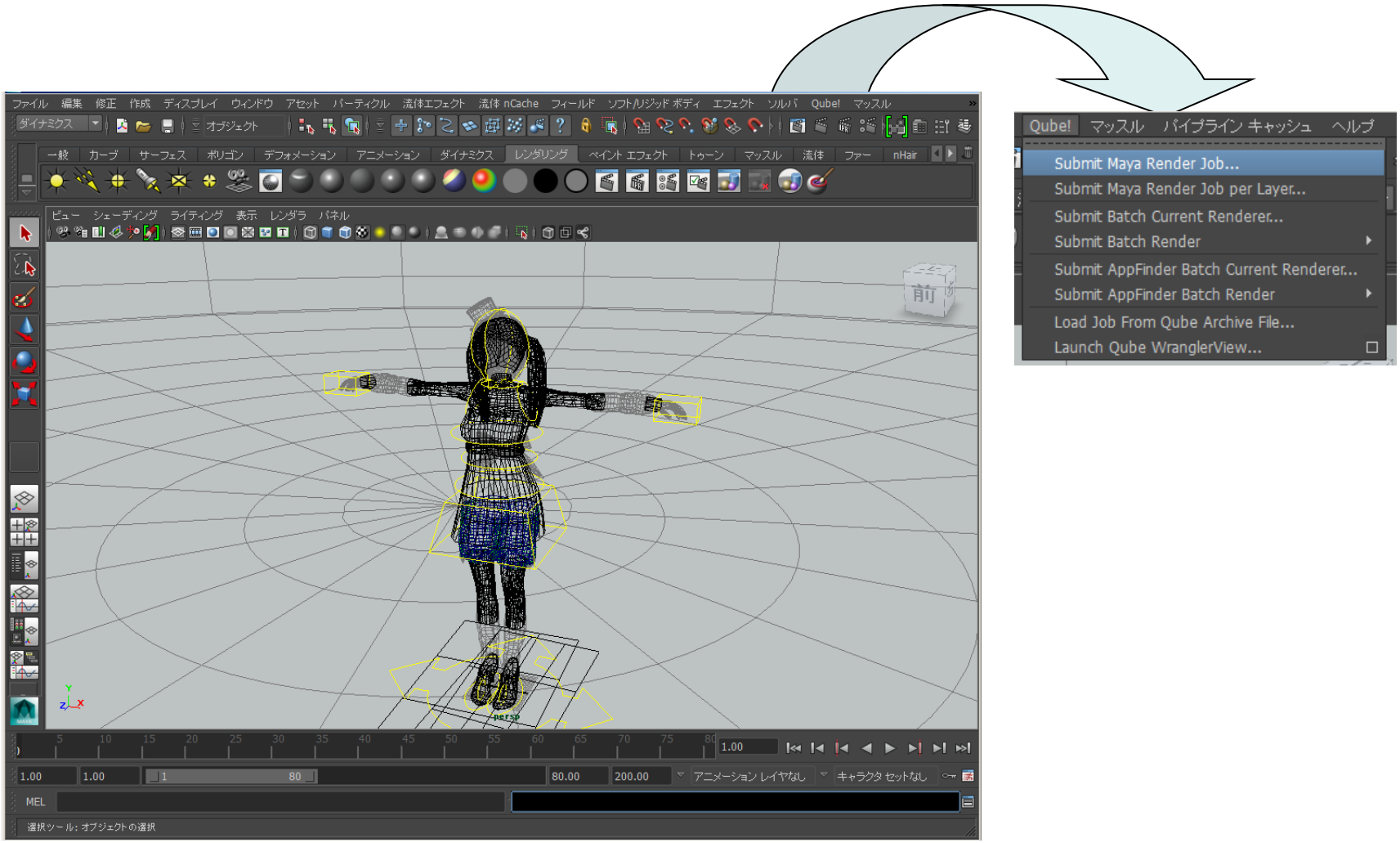
2－1. Maya GUI からの投入

レンダリングジョブを実行する前に、対象となるシーンの作成、保存を済ませます。

データ(プロジェクト、シーン、テクスチャ)の保存先としては、Qube! クライアント、Qube! Worker ホスト両方がアクセス可能となるネットワークドライブへ保存します。
Maya で Qube! を使用する前に、Qube!GUIの[File]→[Install App UI]から[Install ‘Maya’ App UI]をインストールします。
C:¥Users¥<username>¥Documents¥maya¥scripts にプラグインがインストールされますので、Mayaの各バージョンで共通に使用できます。

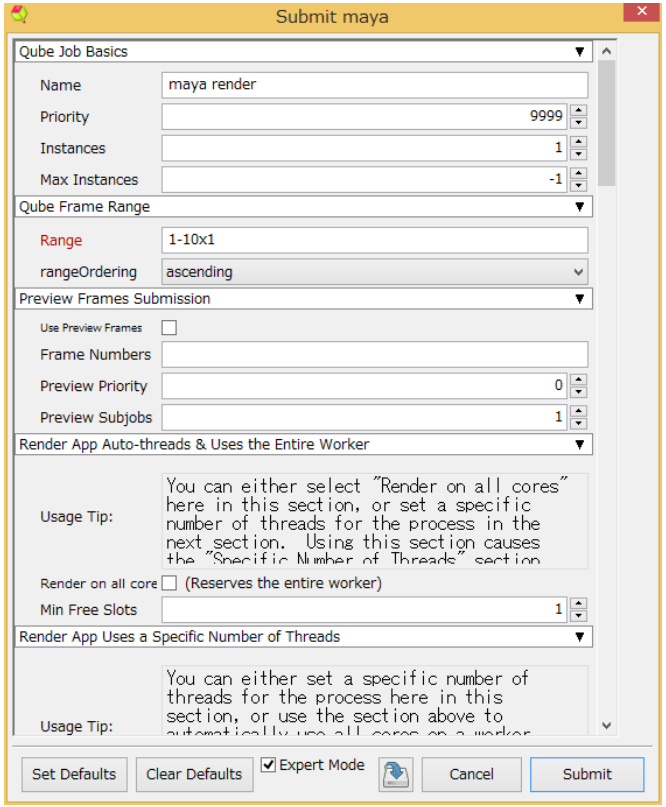


データ保存の完了後、メニューの [Qube!]→[Submit XXXX...] をクリックします。Submit ダイアログが起動します。



以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



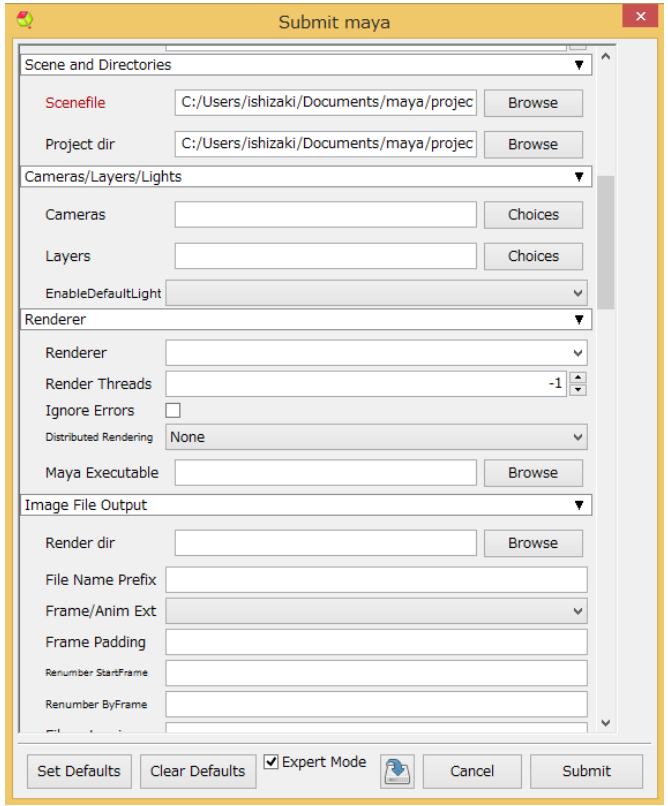
<Submit Maya Render Job / Submit Maya Render Job per Layer>
ダイナミックアロケーションを使用したジョブ投入を行います。
フレーム毎にワーカーを割り当てますので、Submit Batch Renderにあるような Execution指定(Chunks設定)はありません。
Submit Maya Render Job per Layerでは有効なレイヤ数分、Submitダイアログが表示されます。

== Qube Job Basics ==
Name ・・・ 実行ジョブの名称です。一意である必要はありません。
Priority・・・デフォルトは 9999 です。ジョブの優先度としては最低値となります。(値が低い方が優先度が高くなります。i.e. 9999 < 9998 ← ジョブ優先度)
Instances ・・・ 同時に実行されるタスク数。デフォルトは 1 です。
Max Instances・・・SmartShareが有効なとき（supervisorのqb.conf内の **supervisor_smart_share_mode** が”jobs”）、この数を上限に、可能な数までInstance数が自動的に拡張されます。(0:SmartShareを使用しない、-1:上限無し。ただし、supervisorのqb.conf内の **supervisor_max_cpus_limit** で制限可）
== Qube Frame Range ==
Range ・・・ レンダリングするフレーム範囲を指定します。(例：1-100 or 1-100x3 or 1,3,7,10)
1-100x3の x3はステップ値を表します。
rangeOrdering ・・・ 投入するフレームの順番を指定します。ascending（昇順）、descending（降順）、Binary（最初,最後,真ん中,残り..の順）
== Preview Frames Submission ==
Use Preview Frames ・・・ プレビュー用のフレームを投入します。
Frame Numbers ・・・ プレビュー用にレンダリングするフレーム番号を指定します。指定しない場合は、最初,最後,真ん中のフレームをレンダリングします。
Preview Priority・・・プレビュー用のフレームの優先度を指定します。デフォルトは0で最優先になります。
Preview Subjobs ・・・ プレビュー用に投入するサブジョブ数を指定します。
== Render App Auto-threads & Uses the Entire Worker ==
Render on all core・・・ワーカーのすべてのコアを使用します。
Min Free Slots・・・ワーカーがジョブを受けるために必要な空きスロットの最低数を指定します。自動的にReservationsフィールドのhost.processor=n+ に反映されます。
== Render App Uses a Specific Number of Threads ==
Slots = Threads ・・・ コア数がそのままSlot数として設定されている場合（デフォルト）、それぞれのスレッド用にワーカーのスロットを確保します。
Specific Thread Count ・・・ レンダースレッド数を指定します。Render on all coreがONの場合、この設定は無視されます。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。

<Submit Maya Render Job / Submit Maya Render Job per Layer>(Cont'd 1)



== Scene and Directories ==

Scenefile・・・レンダリングするシーンファイル名を指定します。基本的にUNCパスで記述します。

Project dir・・・プロジェクトフォルダへのパスを指定します。一時ファイルが置かれますので、書き込み可能なフォルダを指定します。指定しない場合は、システムのテンポラリフォルダ(Windowsの場合、環境変数:TEMP)が使用されます。

== Cameras/Layers/Lights ==

Cameras・・・レンダリング対象のカメラのリストを記述します。(セパレータ:空白)
Submit Maya Render Job per Layerで投入すると、レイヤの設定に基づき、自動的に設定されます。

Layers・・・レンダリング対象のレイヤのリストを記述します。(セパレータ:空白)
Submit Maya Render Job per Layerで投入すると、自動的に設定されます。

EnableDefaultLight・・・デフォルトライトを有効にします。

== Renderer ==

Renderer・・・ 使用するレンダラを選択します。指定しない場合は、シーン内の設定が使用されます。

Render Threads・・・ 各サブジョブで使用するスレッド数を指定します。-1では、host.processorsの設定に従い、0ではワーカーのすべてのコアを使用します。

Ignore Errors・・・ レンダリング時に発生したエラーメッセージを無視します。OFFの時、エラーになるとfailureとなります。エラーにはなるが、レンダリング画像は正常な場合など、無視して続行できます。

Distributed Rendering・・・ 分散レンダリングを使用します。(Mental rayのサテライト、V-RayのDR)

Maya Executable・・・ Mayaの実行ファイルを指定します。(Windowsの場合、mayabatch.exe)

== Image File Output ==

Render dir・・・ 出力ファイルのフォルダへのパスを指定します。 基本的にUNCパスで記述します。

File Name Prefix・・・ イメージファイル名のプリフィックスを指定します。

Frame/Anim Ext・・・ フレームと拡張子のフォーマットを指定します。

Frame Padding・・・ フレームパディングの桁数を指定します。4の場合、name.XXXX.extとなります。

Renumber StartFrame・・・ ファイル名に付加されるフレーム番号のスタートフレーム番号(オフセット数)を指定します。

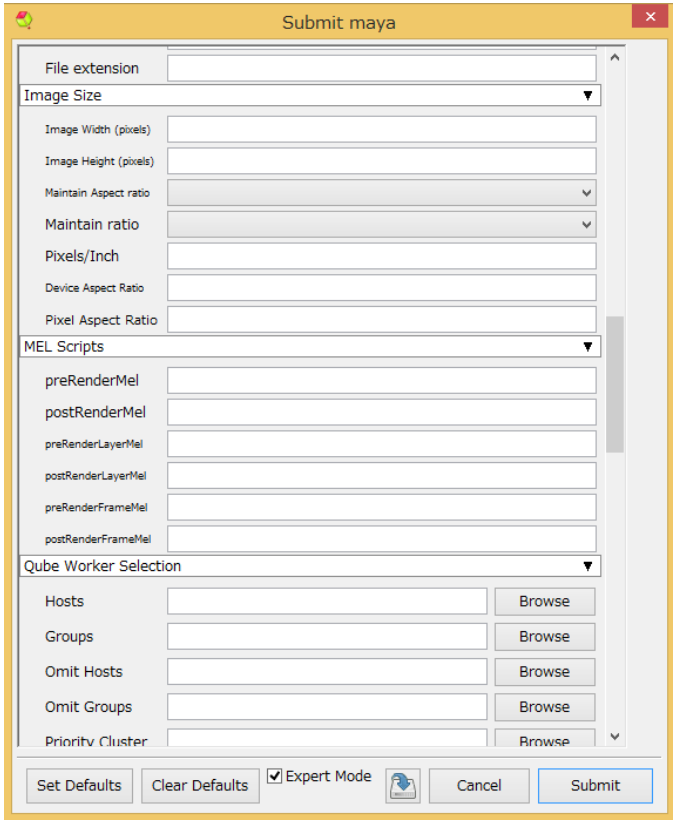
Renumber ByFrame・・・ ファイル名に付加されるフレーム番号のステップ値を指定します。

File extension・・・ ファイル名に付加される拡張子を指定します。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。

<Submit Maya Render Job / Submit Maya Render Job per Layer>(Cont’d 2)



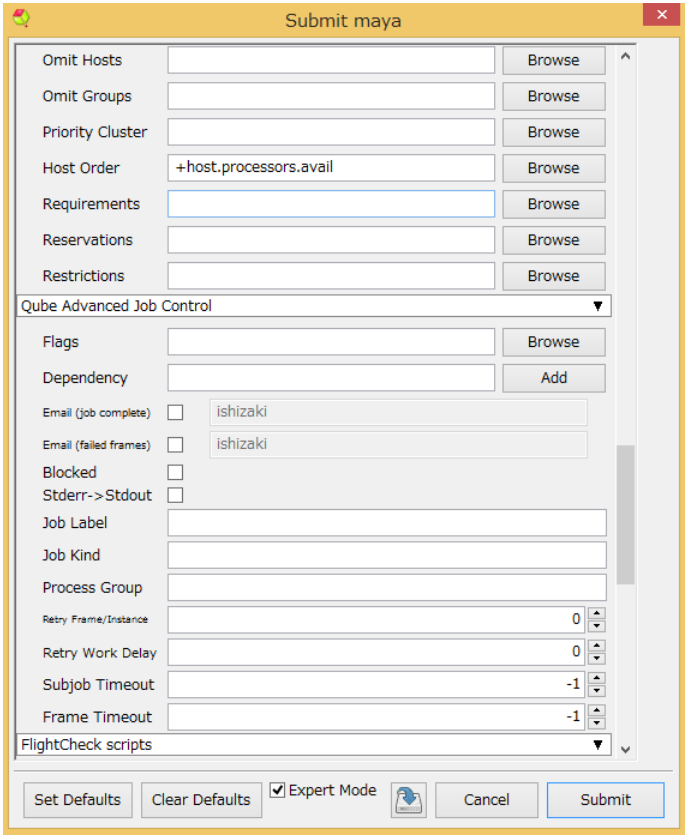
== Image Size ==
Image Width(pixels)・・・ 出力ファイルのイメージ幅(pixels)を指定します。
Image Height(pixels)・・・ 出力ファイルのイメージ高さ(pixels)を指定します。
Maintain Aspect ratio・・・ 出力ファイルのアスペクト比を維持します。
Maintain ratio・・・ 維持するアスペクトを指定します。(Pixel Aspect/Device Aspect)
Pixels/Inch ・・・ 解像度を指定します。
Device Aspect Ratio・・・ デバイスのアスペクト比を指定します。
Pixel Aspect Ratio・・・ ピクセルのアスペクト比を指定します。

== MEL Scripts ==
preRenderMel・・・ シーンがレンダリングされる前に実行されるMELを設定します。
postRenderMel・・・ シーンがレンダリングされた後に実行されるMELを設定します。
preRenderLayerMel・・・ レンダーレイヤがレンダリングされる前に実行されるMELを設定します。
postRenderLayerMel・・・ レンダーレイヤがレンダリングされた後に実行されるMELを設定します。
preRenderFrameMel・・・ フレームがレンダリングされる前に実行されるMELを設定します。
postRenderFrameMel・・・ フレームがレンダリングされた後に実行されるMELを設定します

== Qube Worker Selection ==
Hosts・・・ジョブ実行ホストを指定します。
Groups・・・ジョブ実行Workerグループを指定します。
Omit Hosts・・・ジョブ実行を抑制するホストを指定します。
Omit Group・・・ジョブ実行を抑制するWorkerグループを指定します。
Priority Cluster・・・ジョブ実行時の優先Workerクラスタを指定します。
Host Order・・・ジョブ実行時のホスト順番の優先度条件を指定します。+が付くと高くなり、-が付くと低くなります。
Requirements・・・ジョブ実行要件式を定義します。OSの種類、同種のジョブのみの投入制限ができます。(Job Kind欄参照)
Reservations・・・ジョブ実行要件指定を定義します。プロセッサ数とメモリが指定できます。+が付いている場合は、使用可能なすべてのプロセッサを使用します。
Restrictions・・・ジョブ実行抑制クラスタを指定します。指定したクラスタのワーカーのみで実行するよう抑制できます。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit Maya Render Job / Submit Maya Render Job per Layer>(Cont’d 3)

== Qube Advanced Job Control ==

Flags･･･ジョブ実行フラグを指定します。

Dependency･･･実行ジョブ依存関係の指定式を定義します。i.e. ジョブ A が終わった後に実行する

Mail(job complete)･･･チェックを入れた場合、右側の Email Address フィールドに指定したメールアドレス（コンマで複数指定可）にジョブ終了通知メールを送信します。(Success or Failure)

Mail(failed frames)･･･チェックを入れた場合、右側の Email Address フィールドに指定したメールアドレス（コンマで複数指定可）にジョブ異常終了通知メールを送信します。(Success or Failure)

Blocked･･･ ジョブを”Block”された状態で投入します。直ちに実行したくないときに使用し、手動で開始できます。

Stderr->Stfout･･･Stderrに出力されるエラーをStdoutにリダイレクトします。

Job Label･･･ ジョブ識別のためのラベルを指定します。Process Group内でユニークな名前である必要があります。

Job Kind･･･ ジョブを識別するための任意の情報を設定します。Requirements欄の「Only 1 of a “kind” of job」が指定された際に、ジョブの種類を区別するために参照されます。

Process Group･･･ ジョブを組織的にまとめるためのグループ名を設定します。デフォルトはjobidです。Job Labelと組み合わせてユニークな名前にする必要があります。

Retry Frame/Instance･･･ フレーム/ジョブインスタンスがfailしたときに、リトライする回数を指定します。-1 を指定すると、studioのデフォルト値を使用します。（Preferences - Studio Defaults)

Retry Work Delay･･･ failしたframeに対して自動的にリトライをかける前の待ち時間を秒で指定します。

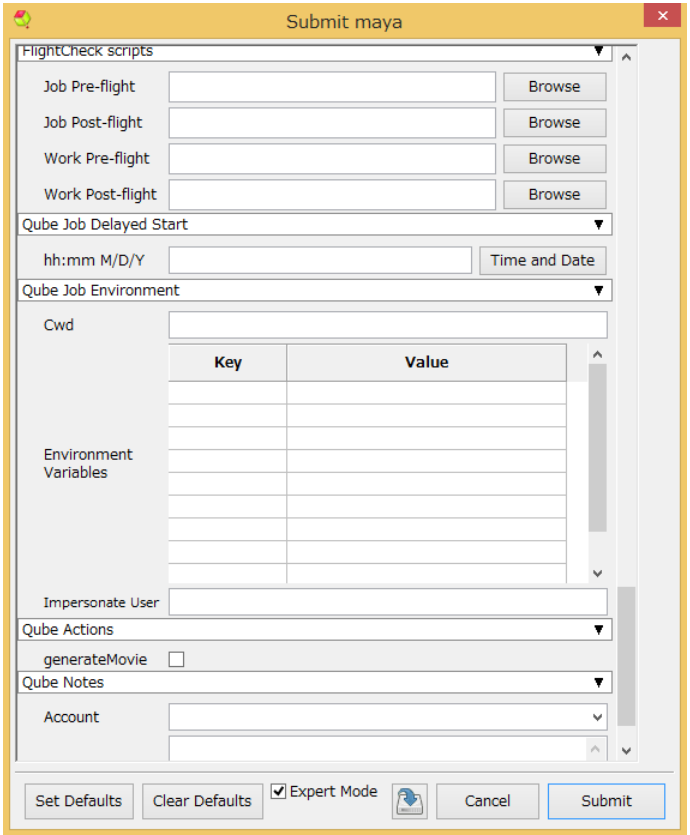
Subjob Timeout･･･ サブジョブがタイムアウトしてKillされる時間を秒で指定します。時間がかかりすぎるジョブを強制終了させることができます。-1 はこの機能を無効化します。

Frame Timeout･･･ フレーム単位の計算がタイムアウトしてKillされる時間を秒で指定します。時間がかかりすぎるジョブを強制終了させることができます。-1 はこの機能を無効化します。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。

<Submit Maya Render Job / Submit Maya Render Job per Layer>(Cont’d 4)



== FlightCheck scripts ==　※各スクリプトでゼロ以外を返すと、ジョブが”failed”になります。
Job Pre-flight･･･ 各ワーカーで、ジョブインスタンスが実行される前に実行されるスクリプトを指定します。
Job Post-flight･･･ 各ワーカーで、ジョブインスタンスが実行された後に実行されるスクリプトを指定します。
Work Pre-flight･･･各ワーカーで、各フレームまたはアジェンダが実行される前に実行されるスクリプトを指定します。
Work Post-flight･･･各ワーカーで、各フレームまたはアジェンダが実行された後に実行されるスクリプトを指定します。

== Qube Job Delayed Start ==
hh:mm M/D/Y･･･ ジョブを実行する時間を指定します。

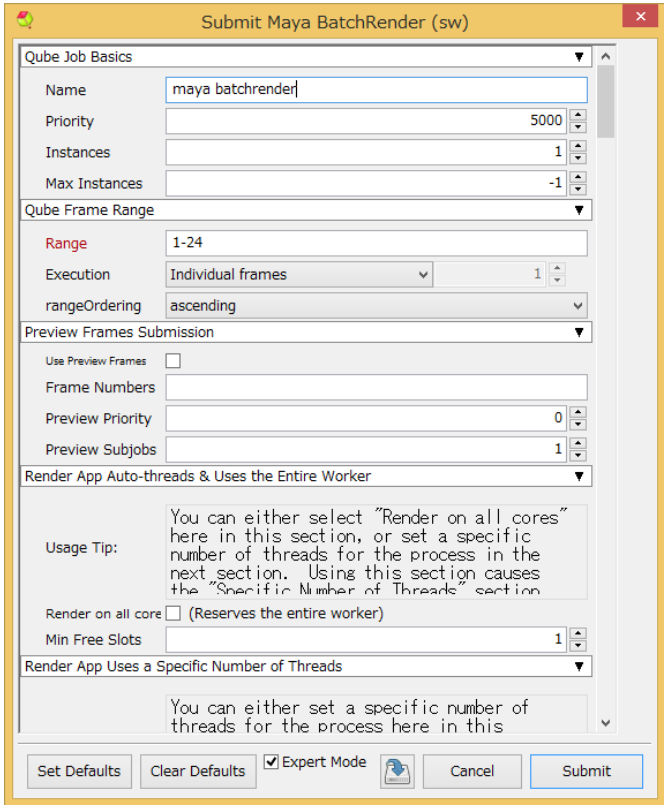
== Qube Job Environment ==
Cwd･･･ ジョブ実行時のカレントワークディレクトリを指定します。
Environment Variables･･･ ジョブ実行時に使用する環境変数を設定します。既存の環境変数に上書きできます。
Impersonate User･･･ ジョブを投入する際、指定したユーザとして投入します。デフォルトはカレントユーザです。指定する場合は、Qube! WranglerViewのUser Permissionsタブで、Impersonate権限を与える必要があります。

== Qube Actions ==
generateMovie･･･ 出力イメージからムービーを作成するためのジョブにリンクを追加します。ffmpegなどへリンク可能です。

== Qube Notes ==
Account･･･ジョブタグによる、任意のアカウントやプロジェクトデータを設定します（ユーザ定義）。Performance Chartsやジョブ検索などに使用できます。
Notes･･･ このジョブについて、コメントを記述します。


または

･･･クリックするとジョブ投入設定をファイルとして保存できます。
保存した設定でのSubmitは、Submit - [Job from file…]でそのファイルを選択します。



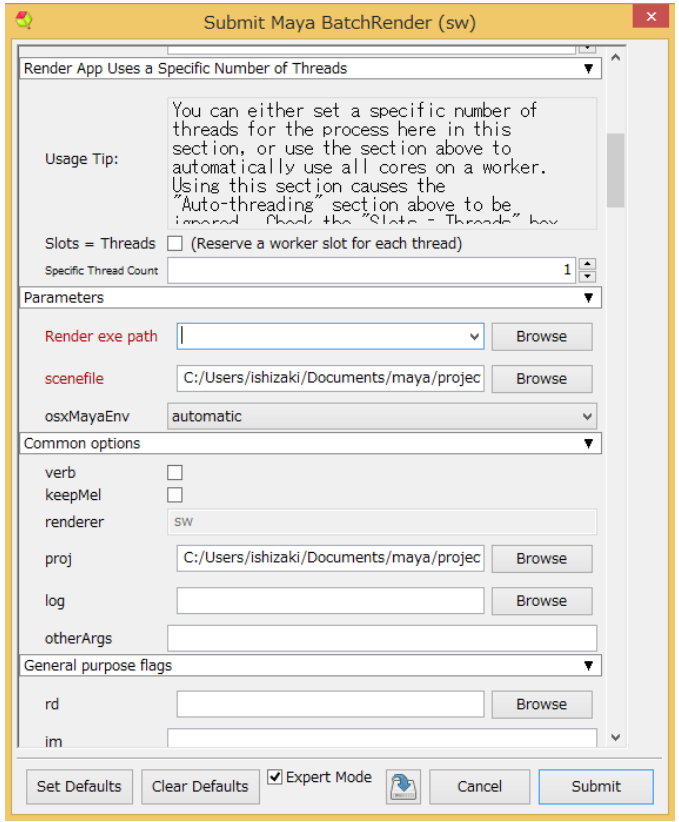
下線項目は必須入力項目です。

<Submit Batch Current Renderer... / Submit Batch Render >
Render.exeコマンドを使用して、数フレームまとめて各ワーカーに投入します。
各ワーカーのレンダリング性能や負荷に差が少ない場合は、Submit Maya Render Job（ダイナミックアロケーションを使用）よりも効率が良いことがあります。
また、レンダラで使用可能なレンダリングパラメータの設定もできますが、パラメータはレンダラによって異なりますので詳細は各レンダラのドキュメントなどをご参照ください。

== Qube Job Basics ==
Name ・・・ 実行ジョブの名称です。一意である必要はありません。
Priority・・・デフォルトは 9999 です。ジョブの優先度としては最低値となります。(値が低い方が優先度が高くなります。i.e. 9999 < 9998 ← ジョブ優先度)
Instances ・・・ 同時に実行されるタスク数。デフォルトは 1 です。
Max Instances・・・SmartShareが有効なとき（supervisorのqb.conf内の **supervisor_smart_share_mode** が"jobs"）、この数を上限に、可能な数までInstance数が自動的に拡張されます。(0:SmartShareを使用しない、-1:上限無し。ただし、supervisorのqb.conf内の **supervisor_max_cpus_limit** で制限可）
== Qube Frame Range ==
Range ・・・ レンダリングするフレーム範囲を指定します。(例：1-100 or 1-100x3 or 1,3,7,10)
1-100x3の x3はステップ値を表します。
Execution・・・ フレーム範囲をどのように分割するかを指定します。（Individual frames:1フレーム単位、Chunks with n frames: nフレーム単位、Split into n partitions: n個のかたまりに分割）
rangeOrdering ・・・ 投入するフレームの順番を指定します。ascending（昇順）、descending（降順）、Binary（最初,最後,真ん中,残り..の順）
== Preview Frames Submission ==
Use Preview Frames ・・・ プレビュー用のフレームを投入します。
Frame Numbers ・・・ プレビュー用にレンダリングするフレーム番号を指定します。指定しない場合は、最初,最後,真ん中のフレームをレンダリングします。
Preview Priority・・・プレビュー用のフレームの優先度を指定します。デフォルトは0で最優先になります。
Preview Subjobs ・・・ プレビュー用に投入するサブジョブ数を指定します。
== Render App Auto-threads & Uses the Entire Worker ==
Render on all cores ・・・ 各インスタンスにつき、各ワーカーで使用できる最大数のコアが割り当てられます。
Min Free Slots ・・・ ここで指定する数以上のオープン状態のスロットを持つワーカーが割り当て対象になります。Render on all coresがONの場合、有効になります。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit Batch Current Renderer... / Submit Batch Render >(Cont’d 1)

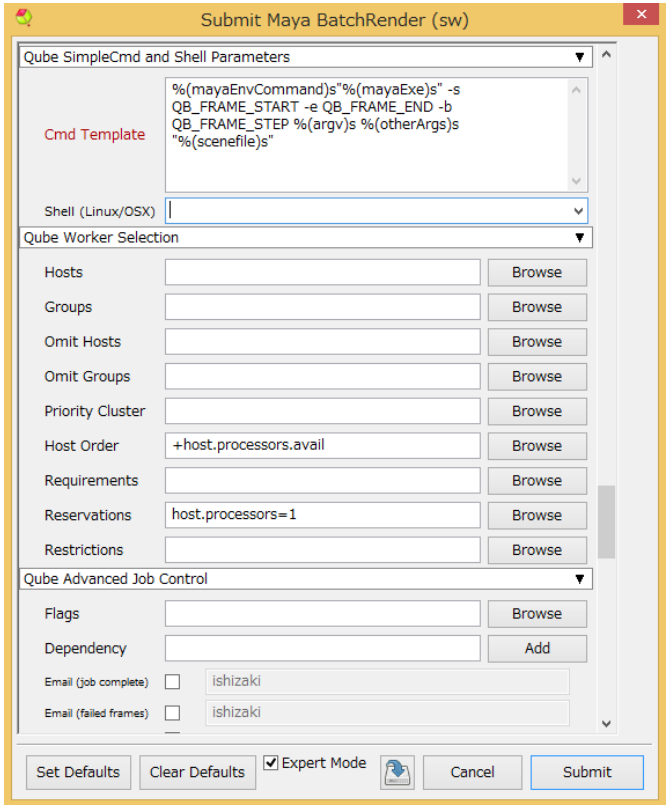
== Render App Uses a Specific Number of Threads ==
Slots = Threads ・・・ ワーカーが、スロット数=コア数 に設定されている場合、各スレッド用にワーカースロットを確保します。
Specific Thread Count ・・・ Slots = ThreadsがOFFの時、レンダースレッド数を指定します。

== Parameters ===
Render exe path・・・ Render.exeファイルをフルパスで指定します。
Scenefile ・・・レンダリングするシーンファイル名を指定します。基本的にUNCパスで記述します。
osxMayaEnv・・・MAYA_LOCATION等の環境変数を設定するために、MayaENV.shを実行します。(OSX版Maya2010では必須）

以下はレンダラが「Mayaソフトウェア」の場合、表示される項目です。詳細はRender.exeのオプション仕様等を参照してください。
== Common options ==
== General purpose flags ==
== Anti-aliasing quality ==
== Raytracing quality ==
== Field Options ==
== Motion Blur ==
== Render Options ==
== Memory and Performance ==
== Render Layer and Passed ==
== Mel callbacks ==
== Others ==

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit Batch Current Renderer... / Submit Batch Render >(Cont’d 2)

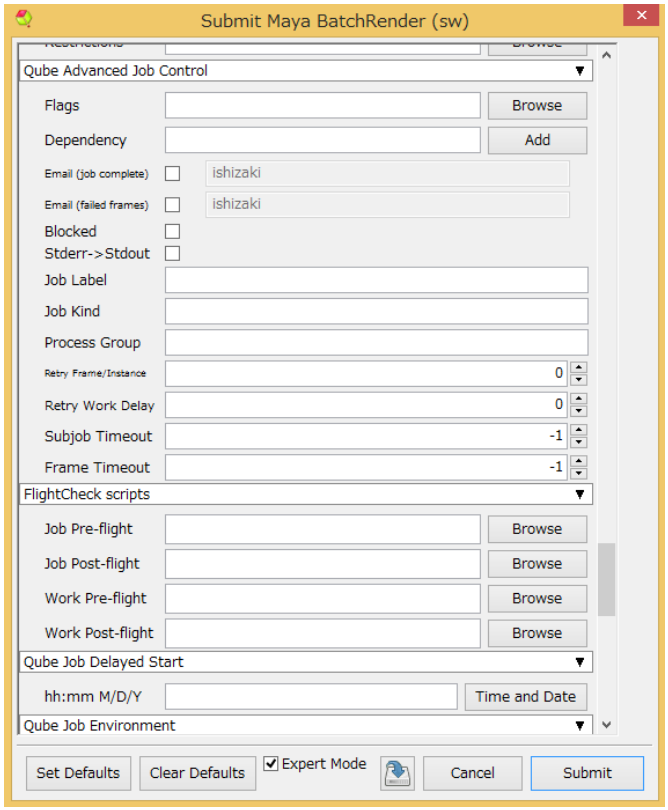
== Qube SimpleCmd and Shell Parameters ==
Cmd Template･･･ 投入するコマンドを生成するためのテンプレートです。
Shell(Linux/OSX)･･･ Linux/OSXでコマンド投入時に使用されるシェルを指定します。デフォルトは /bin/sh です。

== Qube Worker Selection ==
Hosts･･･ジョブ実行ホストを指定します（コンマで羅列可）。

Groups･･･ジョブ実行Workerグループを指定します（コンマで羅列可）。

Omit Hosts･･･ジョブ実行を抑制するホストを指定します（コンマで羅列可）。

Omit Group･･･ジョブ実行を抑制するWorkerグループを指定します（コンマで羅列可）。
Priority Cluster･･･ジョブ実行時の優先Workerクラスタを指定します。
Host Order･･･ジョブ実行時のホスト順番の優先度条件を指定します。+が付くと高くなり、-が付くと低くなります。
Requirements･･･ジョブ実行要件式を定義します。OSの種類、同種のジョブのみの投入制限ができます。（Job Kind欄参照）
Reservations･･･ジョブ実行要件指定を定義します。ジョブスロット数とメモリが指定できます。+が付いている場合は、使用可能なすべてのジョブスロットを使用します。
Restrictions･･･ジョブ実行抑制クラスタを指定します。



下線項目は必須入力項目です。

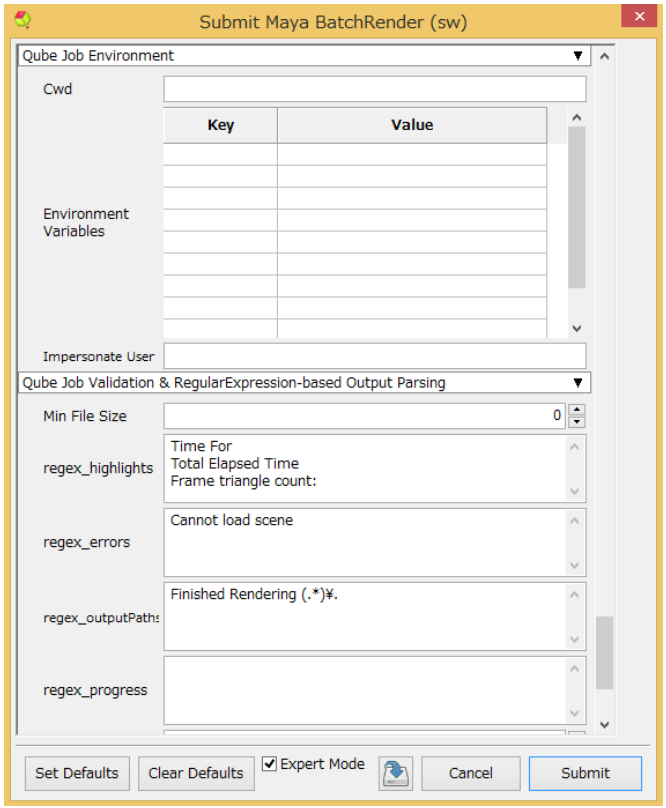
<Submit Batch Current Renderer... / Submit Batch Render >(Cont'd 3)

== Qube Advanced Job Control ==
Flags・・・ジョブ実行フラグを指定します。
Dependency・・・実行ジョブ依存関係の指定式を定義します。i.e. ジョブ A が終わった後に実行する
Mail(job complete)・・・チェックを入れた場合、右側の Email Address フィールドに指定したメールアドレス(コンマで複数指定可)にジョブ終了通知メールを送信します。(Success or Failure)
Mail(failed frames)・・・チェックを入れた場合、右側の Email Address フィールドに指定したメールアドレス(コンマで複数指定可)にジョブ異常終了通知メールを送信します。(Success or Failure)
Blocked・・・ ジョブを"Block"された状態で投入します。直ちに実行したくないときに使用し、手動で開始できます。
Stderr->Stfout・・・Stderrに出力されるエラーをStdoutにリダイレクトします。
Job Label・・・ ジョブ識別のためのラベルを指定します。Process Group内でユニークな名前である必要があります。
Job Kind・・・ ジョブを識別するための任意の情報を設定します。Requirements欄の「Only 1 of a "kind" of job」が指定された際に、ジョブの種類を区別するために参照されます。
Process Group・・・ ジョブを組織的にまとめるためのグループ名を設定します。デフォルトはjobidです。Job Labelと組み合わせてユニークな名前にする必要があります。
Retry Frame/Instance・・・ フレーム/ジョブインスタンスがfailしたときに、リトライする回数を指定します。-1 を指定すると、studioのデフォルト値を使用します。(Preferences - Studio Defaults)
Retry Work Delay・・・ failしたframeに対して自動的にリトライをかける前の待ち時間を秒で指定します。
Subjob Timeout・・・ サブジョブがタイムアウトしてKillされる時間を秒で指定します。時間がかかりすぎるジョブを強制終了させることができます。-1 はこの機能を無効化します。
Frame Timeout・・・ フレーム単位の計算がタイムアウトしてKillされる時間を秒で指定します。時間がかかりすぎるジョブを強制終了させることができます。-1 はこの機能を無効化します。

== FlightCheck scripts == ※各スクリプトでゼロ以外を返すと、ジョブが"failed"になります。
Job Pre-flight・・・ 各ワーカーで、ジョブインスタンスが実行される前に実行されるスクリプトを指定します。
Job Post-flight・・・ 各ワーカーで、ジョブインスタンスが実行された後に実行されるスクリプトを指定します。
Work Pre-flight・・・各ワーカーで、各フレームまたはアジェンダが実行される前に実行されるスクリプトを指定します。
Work Post-flight・・・各ワーカーで、各フレームまたはアジェンダが実行された後に実行されるスクリプトを指定します。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit Batch Current Renderer... / Submit Batch Render >(Cont’d 4)

== Qube Job Delayed Start ==
hh:mm M/D/Y・・・ ジョブを実行する時間を指定します。

== Qube Job Environment ==
Cwd・・・ ジョブ実行時のカレントワークディレクトリを指定します。
Environment Variables・・・ ジョブ実行時に使用する環境変数を設定します。既存の環境変数に上書きできます。
Impersonate User・・・ ジョブを投入する際、指定したユーザとして投入します。デフォルトはカレントユーザです。指定する場合は、Qube! WranglerViewのUser Permissionsタブで、Impersonate権限を与える必要があります。

== Qube Job Validation & RegularExpression-based Output Parsing ==
Min File Size・・・ 出力ファイルのサイズチェックについて、最小サイズ値を指定します。0ではチェックを行いません。**(**regex_outputPaths指定時）
regex_highlights・・・ stdout/stderrからのメッセージでハイライト表示する項目を指定します。
regex_errors・・・ stdout/stderrからのメッセージでfatal errorと認識する文字列を指定します。
regex_outputPaths・・・ stdout/stderrからのイメージファイルのパスを認識する文字列を指定します。
regex_progress・・・ stdout/stderrからのin-frame/chunkの進捗を認識する文字列を指定します。
regex_maxLines・・・ stdout/stderrからの一致パターンのために保持する最大行数を指定します。

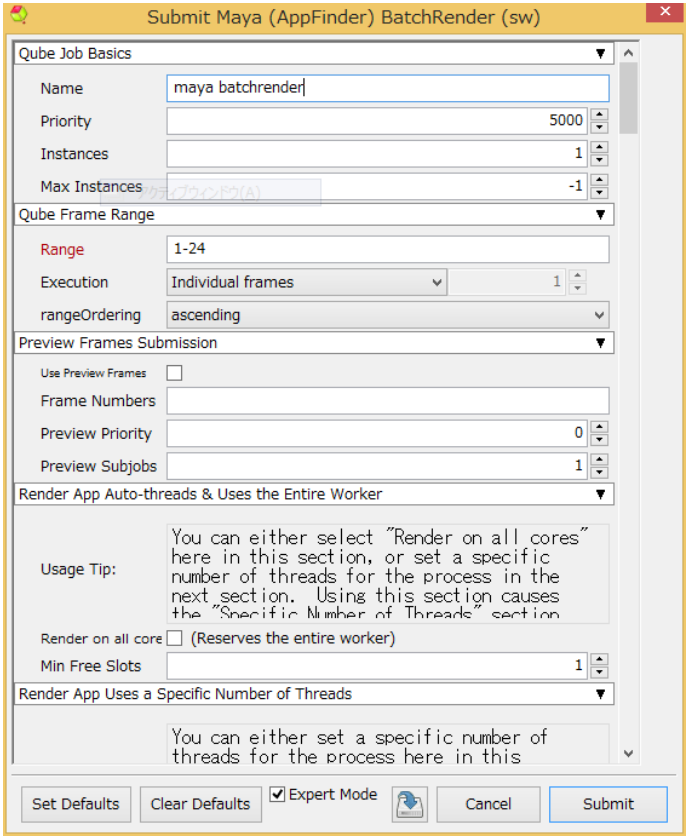
== Qube Actions ==
generateMovie・・・ 出力イメージからムービーを作成するためのジョブにリンクを追加します。

== Qube Notes ==
Account・・・ジョブタグによる、任意のアカウントやプロジェクトデータを設定します（ユーザ定義）。Performance Chartsやジョブ検索などに使用できます。
Notes・・・ このジョブについて、コメントを記述します。


または

・・・クリックするとジョブ投入設定をファイルとして保存できます。

保存した設定でのSubmitは、Submit - [Job from file…]でそのファイルを選択します。



下線項目は必須入力項目です。

<Submit AppFinder Batch Current Renderer... / Submit AppFinder Batch Render >

この投入メニューでは、指定したバージョンのアプリを自動的に見つけて実行できます。ワーカーOSの混在環境やインストールパスが統一されていないワーカー群に投入する際に便利です。
ユーザはインストールパスを意識する必要が無いというメリットもあります。
レンダラで使用可能なレンダリングパラメータの設定もできますが、パラメータはレンダラによって異なりますので詳細は各レンダラのドキュメントなどをご参照ください。

※デフォルトパスは、以下のファイルで定義されていますが、変更は推奨されていません。
C:¥Program Files¥pfx¥qube¥api¥python¥qb¥backend¥appDefaultPaths.py
各ワーカーのworker_path_mapを使用することを推奨します。

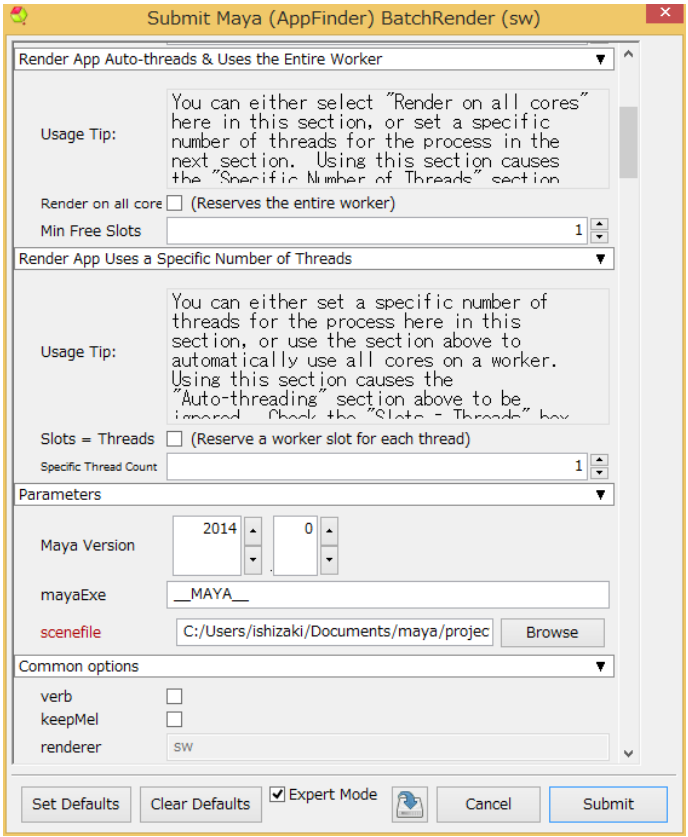
== Qube Job Basics ==
Name ・・・ 実行ジョブの名称です。一意である必要はありません。
Priority・・・デフォルトは 9999 です。ジョブの優先度としては最低値となります。(値が低い方が優先度が高くなります。i.e. 9999 < 9998 ← ジョブ優先度)
Instances ・・・ 同時に実行されるタスク数。デフォルトは 1 です。
Max Instances・・・SmartShareが有効なとき（supervisorのqb.conf内の **supervisor_smart_share_mode** が"jobs"）、この数を上限に、可能な数までInstance数が自動的に拡張されます。(0:SmartShareを使用しない、-1:上限無し。ただし、supervisorのqb.conf内の **supervisor_max_cpus_limit** で制限可）

== Qube Frame Range ==
Range ・・・ レンダリングするフレーム範囲を指定します。(例：1-100 or 1-100x3 or 1,3,7,10)
1-100x3の x3はステップ値を表します。
Execution・・・ フレーム範囲をどのように分割するかを指定します。（Individual frames:1フレーム単位、Chunks with n frames: nフレーム単位、Split into n partitions: n個のかたまりに分割）
rangeOrdering ・・・ 投入するフレームの順番を指定します。ascending（昇順）、descending（降順）、Binary（最初,最後,真ん中,残り..の順）

== Preview Frames Submission ==
Use Preview Frames ・・・ プレビュー用のフレームを投入します。
Frame Numbers ・・・ プレビュー用にレンダリングするフレーム番号を指定します。指定しない場合は、最初,最後,真ん中のフレームをレンダリングします。
Preview Priority・・・プレビュー用のフレームの優先度を指定します。デフォルトは0で最優先になります。
Preview Subjobs ・・・ プレビュー用に投入するサブジョブ数を指定します。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit AppFinder Batch Current Renderer... / Submit AppFinder Batch Render > (Cont’d 1)

== Render App Auto-threads & Uses the Entire Worker ==
Render on all cores ・・・ 各インスタンスにつき、各ワーカーで使用できる最大数のコアが割り当てられます。
Min Free Slots ・・・ ここで指定する数以上のオープン状態のスロットを持つワーカーが割り当て対象になります。Render on all coresがONの場合、有効になります。
Reservationsフィールドのhost.processor=n+ に反映されます。

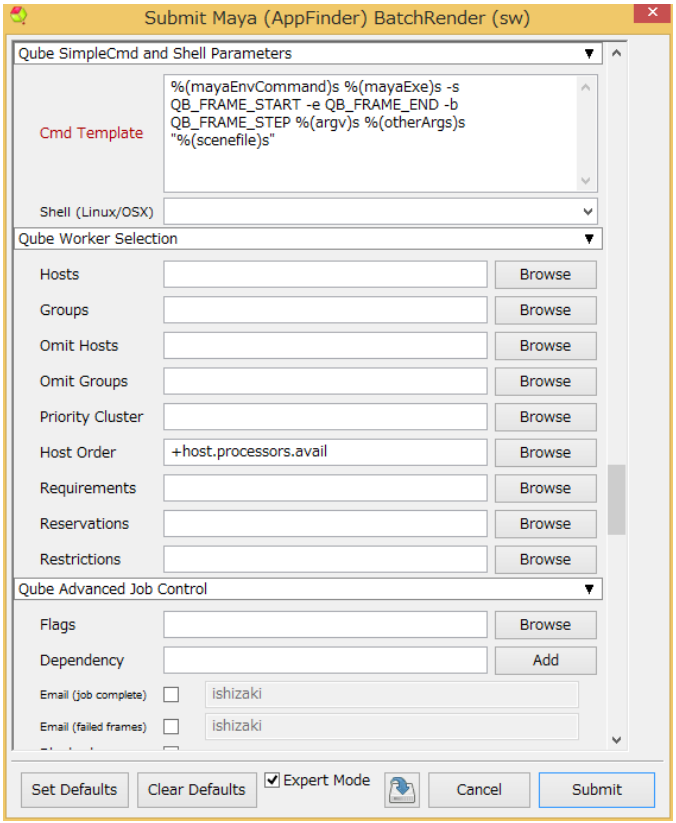
== Render App Uses a Specific Number of Threads ==
Slots = Threads ・・・ ワーカーが、スロット数=コア数 に設定されている場合、各スレッド用にワーカースロットを確保します。
Specific Thread Count ・・・ Slots = ThreadsがOFFの時、実行できるスレッド数を指定します。

== Parameters ===
Maya Version・・・ バージョン番号を指定します。
mayaExe・・・ Mayaのアプリケーションテンプレートを指定します。
Scenefile ・・・レンダリングするシーンファイル名を指定します。基本的にUNCパスで記述します。

以下はレンダラが「Mayaソフトウェア」の場合、表示される項目です。詳細はRender.exeのオプション仕様等を参照してください。
== Common options ==
== General purpose flags ==
== Anti-aliasing quality ==
== Raytracing quality ==
== Field Options ==
== Motion Blur ==
== Render Options ==
== Memory and Performance ==
== Render Layer and Passed ==
== Mel callbacks ==
== Others ==

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。

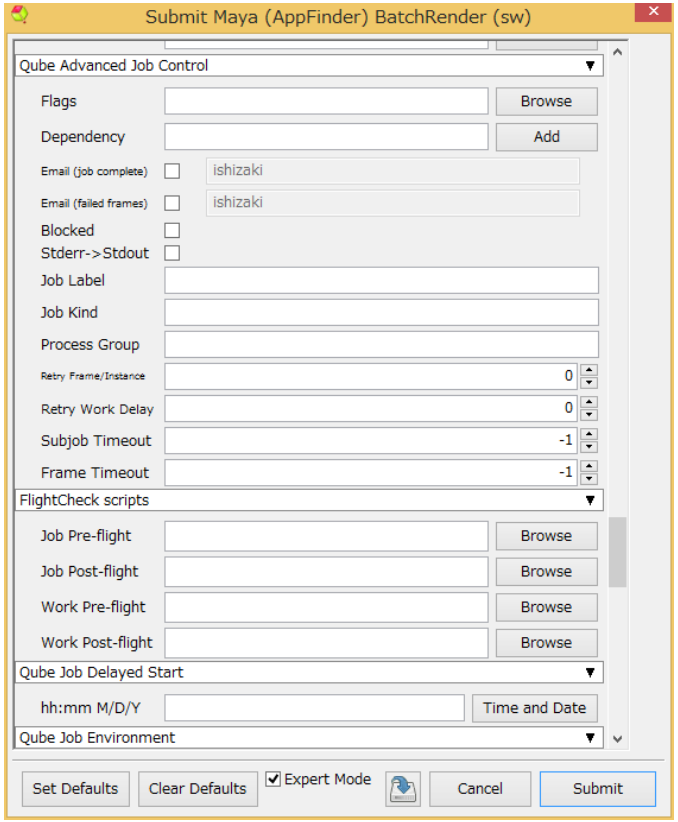


<Submit AppFinder Batch Current Renderer... / Submit AppFinder Batch Render > (Cont’d 2)

== Qube SimpleCmd and Shell Parameters ==
Cmd Template･･･ 投入するコマンドを生成するためのテンプレートです。
Shell(Linux/OSX)･･･ Linux/OSXでコマンド投入時に使用されるシェルを指定します。デフォルトは /bin/sh です。

== Qube Worker Selection ==
Hosts･･･ジョブ実行ホストを指定します（コンマで羅列可）。
Groups･･･ジョブ実行Workerグループを指定します（コンマで羅列可）。
Omit Hosts･･･ジョブ実行を抑制するホストを指定します（コンマで羅列可）。
Omit Group･･･ジョブ実行を抑制するWorkerグループを指定します（コンマで羅列可）。
Priority Cluster･･･ジョブ実行時の優先Workerクラスタを指定します。
Host Order･･･ジョブ実行時のホスト順番の優先度条件を指定します。+が付くと高くなり、-が付くと低くなります。
Requirements･･･ジョブ実行要件式を定義します。OSの種類、同種のジョブのみの投入制限ができます。（Job Kind欄参照）
Reservations･･･ジョブ実行要件指定を定義します。ジョブスロット数とメモリが指定できます。+が付いている場合は、使用可能なすべてのジョブスロットを使用します。
Restrictions･･･ジョブ実行抑制クラスタを指定します。

下線項目は必須入力項目です。



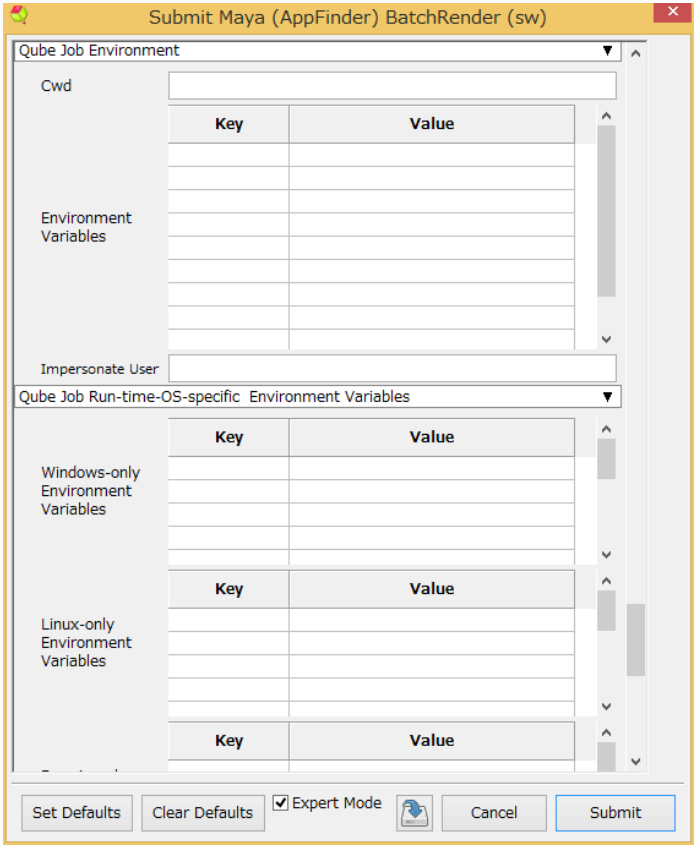
<Submit AppFinder Batch Current Renderer... / Submit AppFinder Batch Render > (Cont'd 3)

== Qube Advanced Job Control ==
Flags･･･ジョブ実行フラグを指定します。
Dependency･･･実行ジョブ依存関係の指定式を定義します。i.e. ジョブ A が終わった後に実行する。
Mail(job complete)･･･チェックを入れた場合、右側の Email Address フィールドに指定したメールアドレス(コンマで複数指定可)にジョブ終了通知メールを送信します。(Success or Failure)
Mail(failed frames)･･･チェックを入れた場合、右側の Email Address フィールドに指定したメールアドレス(コンマで複数指定可)にジョブ異常終了通知メールを送信します。(Success or Failure)
Blocked･･･ ジョブを"Block"された状態で投入します。直ちに実行したくないときに使用し、手動で開始できます。
Stderr->Stfout･･･Stderrに出力されるエラーをStdoutにリダイレクトします。
Job Label･･･ ジョブ識別のためのラベルを指定します。Process Group内でユニークな名前である必要があります。
Job Kind･･･ ジョブを識別するための任意の情報を設定します。Requirements欄の「Only 1 of a "kind" of job」が指定された際に、ジョブの種類を区別するために参照されます。
Process Group･･･ ジョブを組織的にまとめるためのグループ名を設定します。デフォルトはjobidです。Job Labelと組み合わせてユニークな名前にする必要があります。
Retry Frame/Instance･･･ フレーム/ジョブインスタンスがfailしたときに、リトライする回数を指定します。-1 を指定すると、studioのデフォルト値を使用します。(Preferences - Studio Defaults)
Retry Work Delay･･･ failしたframeに対して自動的にリトライをかける前の待ち時間を秒で指定します。
Subjob Timeout･･･ サブジョブがタイムアウトしてKillされる時間を秒で指定します。時間がかかりすぎるジョブを強制終了させることができます。-1 はこの機能を無効化します。
Frame Timeout･･･ フレーム単位の計算がタイムアウトしてKillされる時間を秒で指定します。時間がかかりすぎるジョブを強制終了させることができます。-1 はこの機能を無効化します。

== FlightCheck scripts == ※各スクリプトでゼロ以外を返すと、ジョブが"failed"になります。
Job Pre-flight･･･ 各ワーカーで、ジョブインスタンスが実行される前に実行されるスクリプトを指定します。
Job Post-flight･･･ 各ワーカーで、ジョブインスタンスが実行された後に実行されるスクリプトを指定します。
Work Pre-flight･･･各ワーカーで、各フレームまたはアジェンダが実行される前に実行されるスクリプトを指定します。
Work Post-flight･･･各ワーカーで、各フレームまたはアジェンダが実行された後に実行されるスクリプトを指定します。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit AppFinder Batch Current Renderer... / Submit AppFinder Batch Render > (Cont’d 4)

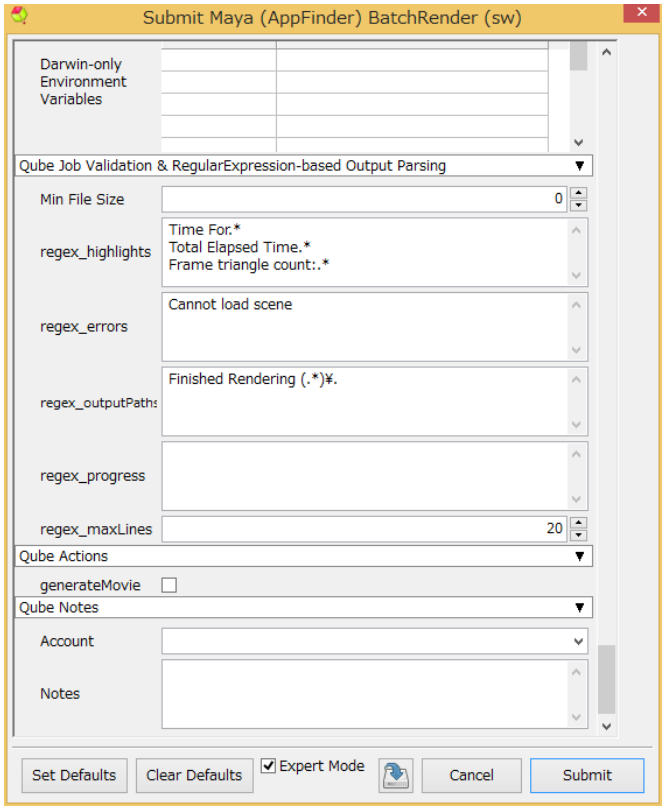
== Qube Job Delayed Start ==
hh:mm M/D/Y・・・ ジョブを実行する時間を指定します。

== Qube Job Environment ==
Cwd・・・ ジョブ実行時のカレントワークディレクトリを指定します。
Environment Variables・・・ ジョブ実行時に使用する環境変数を設定します。既存の環境変数に上書きできます。
Impersonate User・・・ ジョブを投入する際、指定したユーザとして投入します。デフォルトはカレントユーザです。指定する場合は、Qube! WranglerViewのUser Permissionsタブで、Impersonate権限を与える必要があります。

== Qube Job Run-time-OS-specific Environment Variables ==
Windows-only Environment Variables・・・ Windows上でジョブ実行時に使用する環境変数を設定します。既存の環境変数に上書きできます。
Linux-only Environment Variables・・・ Linux上でジョブ実行時に使用する環境変数を設定します。既存の環境変数に上書きできます。
Darwin-only Environment Variables・・・ Darwin（Mac OS X)上でジョブ実行時に使用する環境変数を設定します。既存の環境変数に上書きできます。

以下のジョブ投入メニューは、”Expert Mode” で表示させたものです。***下線項目は必須入力項目です。***

指定が必要な全ての項目に入力した上で、”Submit” ボタンをクリックし、ジョブを投入します。



<Submit AppFinder Batch Current Renderer... / Submit AppFinder Batch Render > (Cont’d 5)

== Qube Job Validation & RegularExpression-based Output Parsing ==
Min File Size・・・ 出力ファイルのサイズチェックについて、最小サイズ値を指定します。0ではチェックを行いません。**(**regex_outputPaths指定時）
regex_highlights・・・ stdout/stderrからのメッセージでハイライト表示する項目を指定します。
regex_errors・・・ stdout/stderrからのメッセージでfatal errorと認識する文字列を指定します。
regex_outputPaths・・・ stdout/stderrからのイメージファイルのパスを認識する文字列を指定します。
regex_progress・・・ stdout/stderrからのin-frame/chunkの進捗を認識する文字列を指定します。
regex_maxLines・・・ stdout/stderrからの一致パターンのために保持する最大行数を指定します。

== Qube Actions ==
generateMovie・・・ 出力イメージからムービーを作成するためのジョブにリンクを追加します。

== Qube Notes ==
Account・・・ジョブタグによる、任意のアカウントやプロジェクトデータを設定します（ユーザ定義）。Performance Chartsやジョブ検索などに使用できます。
Notes・・・ このジョブについて、コメントを記述します。

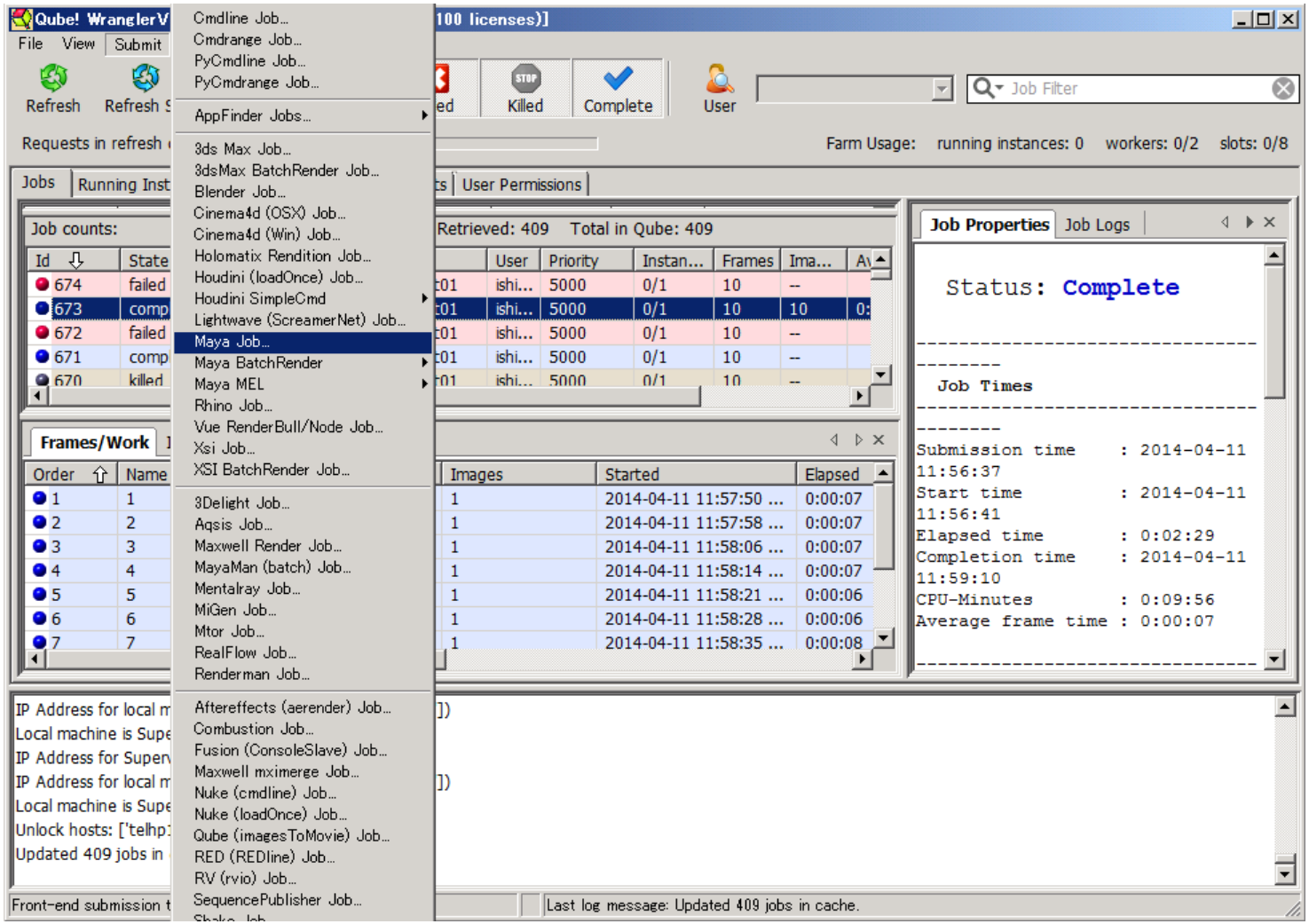

または

・・・クリックするとジョブ投入設定をファイルとして保存できます。

保存した設定でのSubmitは、Submit - [Job from file…]でそのファイルを選択します。

2－2. Qube! WranglerView からの投入

[スタート]→[プログラム]→[PipelineFX]→[Qube!] をクリック、Qube! WranglerView を起動、[Submit]→[Maya Job…] を選択します。設定項目は、Maya GUIからのものと同じです。



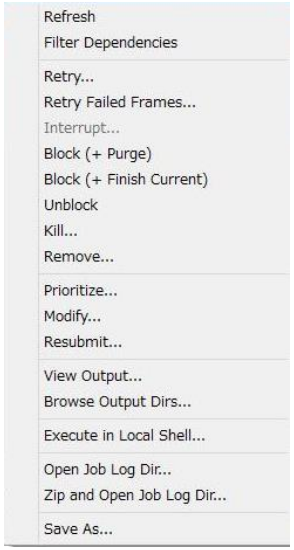
2－3. ジョブ投入後のジョブ確認方法について

投入したジョブが完了したかについては、完了通知メール送信設定をされている場合でも、必ず Qube! GUI を起動し確認するかたちになります。([スタート]→[プログラム]→[PipelineFX]→[Qube!])

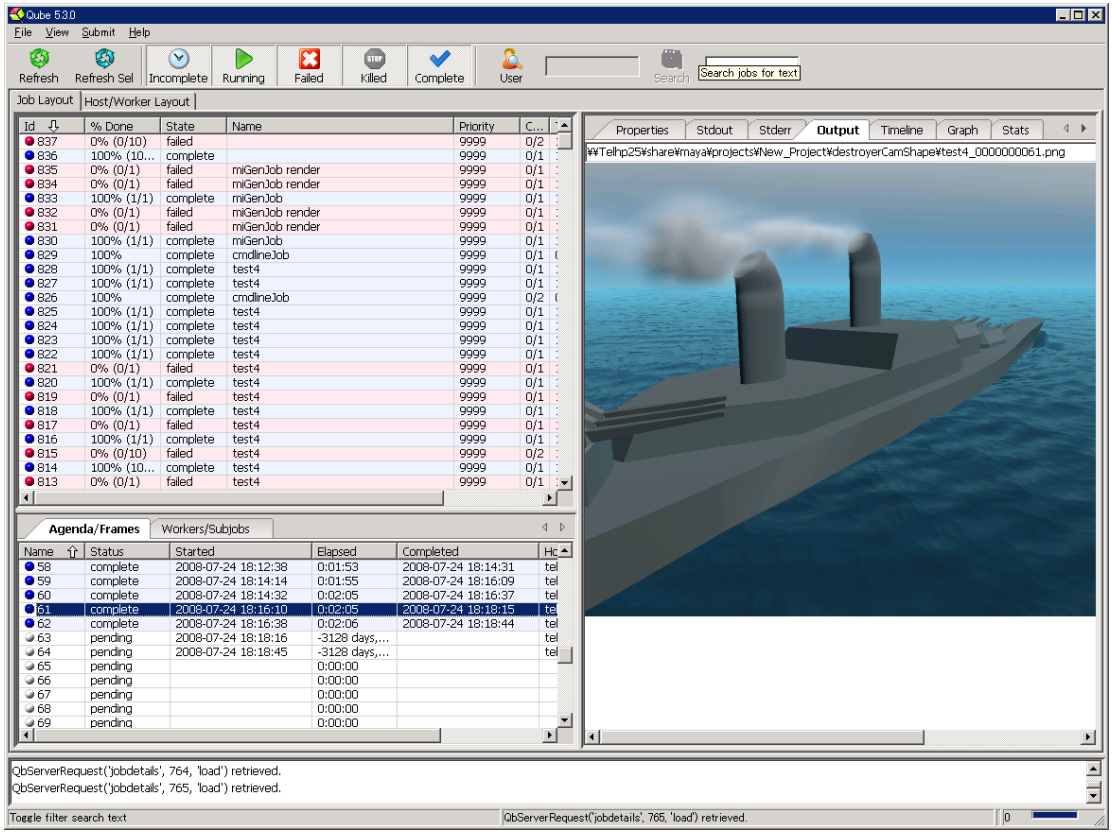
“Jobs” タブで実行中、あるいは実行完了したジョブの概要をブラウズします。

また各タブをクリックする事で、[Job Properties] (ジョブ概要)、[Stdout] (ジョブ標準出力)、[Stderr] (ジョブエラー出力)、[Output] (出力画像確認※) などが行えます。ジョブの状態は “State” 表示で “running” (実行中)、”complete” (完了)、”failed” (失敗) であるか確認します。

実行中のジョブをキャンセルする場合には、ジョブ ID 項目を選択の上、マウス右クリックし、メニューの “Kill” を選択します。



Frames/Work から目的の Work を選択の上、左側　ウインドウの”Output” タブをクリックする事で、画像表示をさせます。対応画像フォーマット以外を表示させるには、File > PreferencesのImage Viewerにて外部ビューアが指定できます。



※ 対応画像フォーマット～ TIFF(標準フォーマット)、PNG、JPG、BMP、GIF

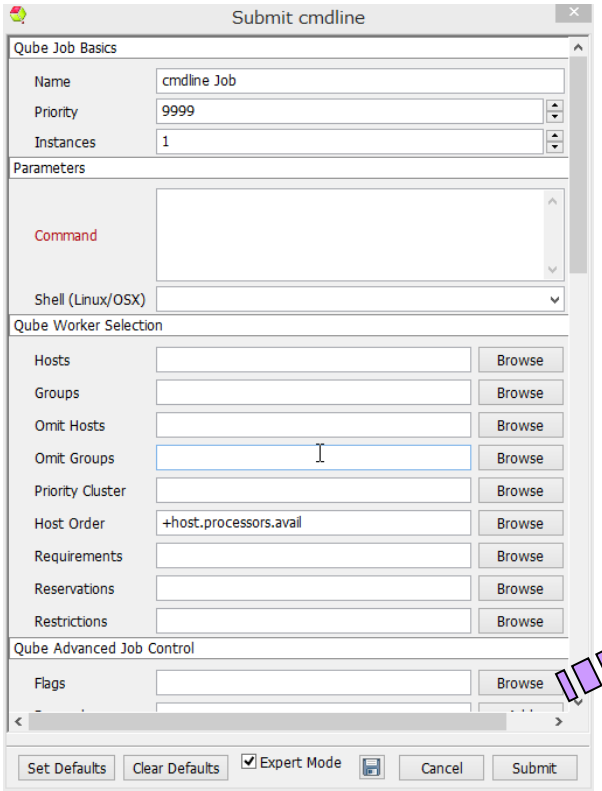
3. コマンドラインジョブ投入フロー

3－1. コマンドラインジョブ

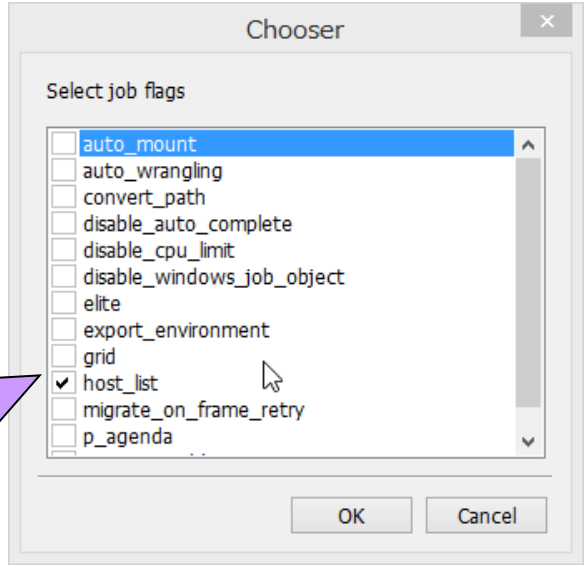
Qube! GUI からレンダリングジョブ以外に使用可能なコマンドジョブの実行が可能です。この機能を活用する事により、レンダーファームの管理～リブート、シャットダウン、システム設定ファイルの差し替え、ソフトウェアのインストールなど～が行えます。※

[Submit]→[Cmdline Job…] をクリックすると以下のウインドウが開きます。

Command フィールドに実行コマンドラインを入力、Hosts で実行ホストの指定、あるいは Flags フィールドで Browse ボタンを押し以下にある Chooser ウインドウで “host_list” にチェックを入れた上でジョブ実行を行うと、全ての Worker ホストで指定されたコマンドを実行します。

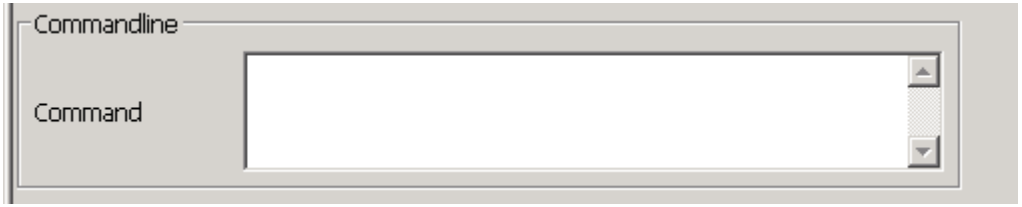


※ 実行条件は、proxy mode が User である場合、Administrators あるいは Superuser 権限でのジョブ投入、また proxy である場合は Proxy ユーザーが Administrators あるいは Superuser 権限 である事が必要になります。(ドメインの場合は、Domain Admins権限が必要です)



3－2. 応用編 ～ Qube！ジョブとしてのインストールパッケージ処理

コマンドラインジョブを応用していただくことにより、レンダーファームホストのソフトウェア管理工数を削減する事も可能です。共有ファイルシステムへインストールパッケージを保存し、特定のレンダーファームホストに対して、無人インストールを行うかたちで対応可能です。
実行にあたっては、Command フィールドに直接コマンドラインを記述する、あるいはそれらの記述を行っている実行バッチを指定する、などの方法となります。



各プラットフォーム毎の代表的なインストールコマンドとそのオプション指定について、以下に示します。

詳細はそれぞれのコマンドのヘルプをご参照ください。

-Windows (msiパッケージ)の場合-

msiexec /i "Z:¥XYZ.msi" /qn /liwearcmov Z:¥install.log

-OS X (インストーラーパッケージ) の場合-

/usr/sbin/installer -pkg /foo/share/XYZ.pkg -target "/Volumes“

-Linux (rpmパッケージ)の場合-

rpm –ivh /foo/share/XYZ.rpm

4. トラブルシューティング

■ failedになる、pendingのまま進まないなど、うまくレンダリングができない場合

ログにエラーメッセージ等が表示されることもありますが、基本的な事項として先ずは以下の件をご確認ください。
•UACがオフになっているか
•“Interactive Services Dialog Detection Service” が存在している場合、停止状態になっているか
•MSE(Microsoft Security Essentials)を使用している場合は、C:¥ProgramData¥Pfx¥Qube を除外登録しているか（登録していない場合、supervisorとworker間の通信が不安定になる現象が報告されています）

•各ワーカーから、同じパスでシーンやテクスチャ、出力フォルダ等にアクセスできるか。基本的にUNCパスで記述しますが、各ワーカーでパスが異なる場合は、worker_drive_mapやworker_path_map等の設定が必要となります。

4. トラブルシューティング(Cont'd 1)

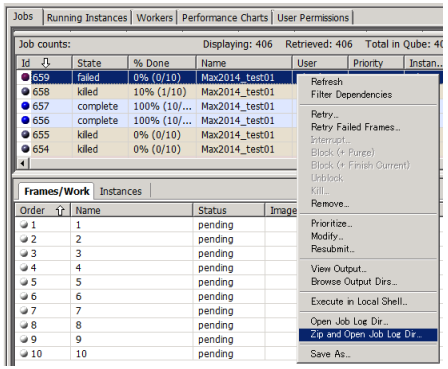
■ エラー解析に必要なログについて

● ジョブログ(http://docs.pipelinefx.com/display/QUBE/Job+logs)

位置：スーパーバイザマシン上の、C:¥ProgramData¥Pfx¥Qube¥logs¥job

各ジョブ単位で出力されるログです。投入時の設定、Qube!およびレンダラからの情報が出力されています。

スーパーバイザにて、ジョブを右クリック > Open Job Log Dir... でフォルダへアクセスでき、Zip and Open Job Log Dir... にて、Zipファイルにアーカイブ化できます。メールに添付して送付する際などに便利です。

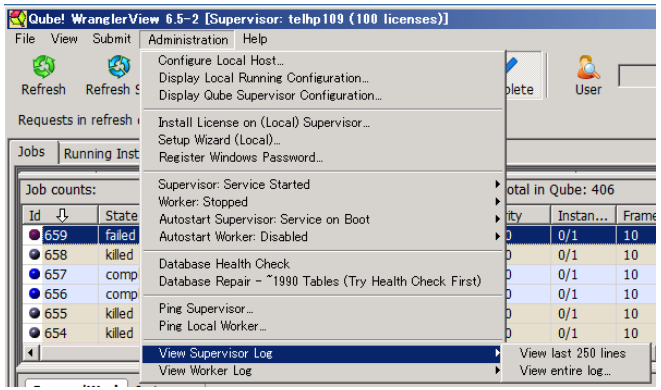


●Supervisor Logs(supelog)(http://docs.pipelinefx.com/display/QUBE/Supervisor+Logs)

位置：スーパーバイザマシン上の、C:¥ProgramData¥Pfx¥Qube¥logs¥supelog

スーパーバイザが出力するログです。各ジョブの割り振り状況やワーカーからのエラー、リトライ情報などファーム全体のマネジメントに関する情報が出力されています。

スーパーバイザにて、Qube! WranglerView > Administration > View Supervisor Log にて見ることができます。放っておくと大きなファイルになりがちですので、定期的に作成し直すことをお勧めします。（ファイル名を変更したり、削除したりすると、自動的に作成されます）



4. トラブルシューティング(Cont'd 2)

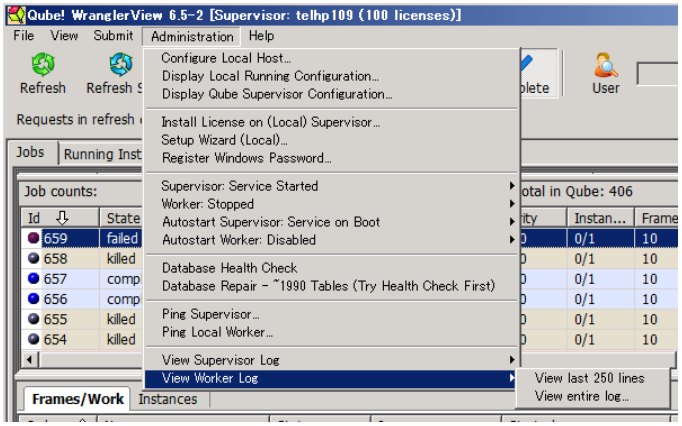
■ エラー解析に必要なログについて

● Worker Logs(http://docs.pipelinefx.com/display/QUBE/Worker+Logs)

位置：ワーカーマシン上の、C:¥ProgramData¥Pfx¥Qube¥logs¥workerlog

各ワーカー単位で出力されるログです。ワーカーのステータスや割り振られたジョブに対し、本ワーカーがどのようにレンダリング資源を割り当てたかなどの情報が出力されています。

ワーカーにて、Qube! WranglerView > Administration > View Worker Log にて見ることができます。放っておくと大きなファイルになりがちですので、定期的に作成し直すことをお勧めします。（ファイル名を変更したり、削除したりすると、自動的に作成されます）



■ Qube! サポートサイトもご参照ください。
https://www.comtec.daikin.co.jp/DC/prd/qube/support/
FAQ: https://www.comtec.daikin.co.jp/faq-qube/
開発元サポートサイト（英語）：https://www.pipelinefx.com/supportpage/

Qube!クイックガイド *Maya Jobtype・コマンドライン編*　第6.0版 – v7.0.x対応

2020年 9月 24日

ダイキン工業株式会社 電子システム事業部　営業部　MCグループ
https://www.comtec.daikin.co.jp　qb-support@daikin.co.jp
〒104-0028
東京都中央区八重洲二丁目2番1号　東京ミッドタウン八重洲　八重洲セントラルタワー