



One More Solution



簡易インストールガイド (Ver.7.0-x)

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製、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。

Qube! は、PipelineFX, LLC.の登録商標です。

Audodesk 3ds Max、Autodesk Maya、Autodesk SoftimageはAutodesk, Inc. の登録商標です。

After Effects はAdobe Inc. の登録商標です。

その他、会社名、商品名は一般に各社の商標または登録商標です。なお、文章中ではTM マークおよびR マークは明記していません。



～ 目次 ～

1. はじめに	4
2. システム構成について	
2-1. 構成図	5
2-2. Qube! システムの各コンポーネント	6
3. システム要件	
3-1. システム要件詳細	7
3-2. Qube! システムインストール事前確認事項	
(1) Qube! システムのネットワーク構成での注意点について	8
(2) 必須インストールモジュール及びインストール順番	9
(3) サポートCGアプリケーションと対応状況一覧	10
4. インストール手順	
4-0-1. (準備)MySQLからPostgreSQLへのデータ移行	11
4-0-2. (準備)Perl, Pythonのバージョン確認とバージョンアップ	11
4-1. Windows	12-19
4-2. Mac OS X	20
4-3. Linux	21-22
4-4. Job Type	23
OS プラットフォーム毎の Job Type インストール方法について	
5. 設定手順	24-30
6. qb.confの設定(Advanced)	31-40
7. 補足資料	41-44

1. はじめに

本ガイドは、Qube! インストール簡易ガイドです。

詳細につきましては、Qube! WranglerView の [Help]→[Qube! Installation Guide](pdf)

または、[スタート]→[プログラム]→[PipelineFX]→[Installation Manual](pdf)を参照ください。

ダイキン工業Qube! 製品情報サイト: <https://www.comtec.daikin.co.jp/DC/prd/qube/>

ダイキン工業Qube! サポートサイト: <https://www.comtec.daikin.co.jp/DC/prd/qube/support/>

ダイキン工業Qube! FAQサイト: <https://www.comtec.daikin.co.jp/faq-qube/>

Pipeline FX社ホームページ: <https://www.pipelinefx.com/>

Qube! インストーラダウンロードサイト: <https://www.pipelinefx.com/downloadversions/>

Pipeline FX社サポートサイト: <https://www.pipelinefx.com/supportpage/>

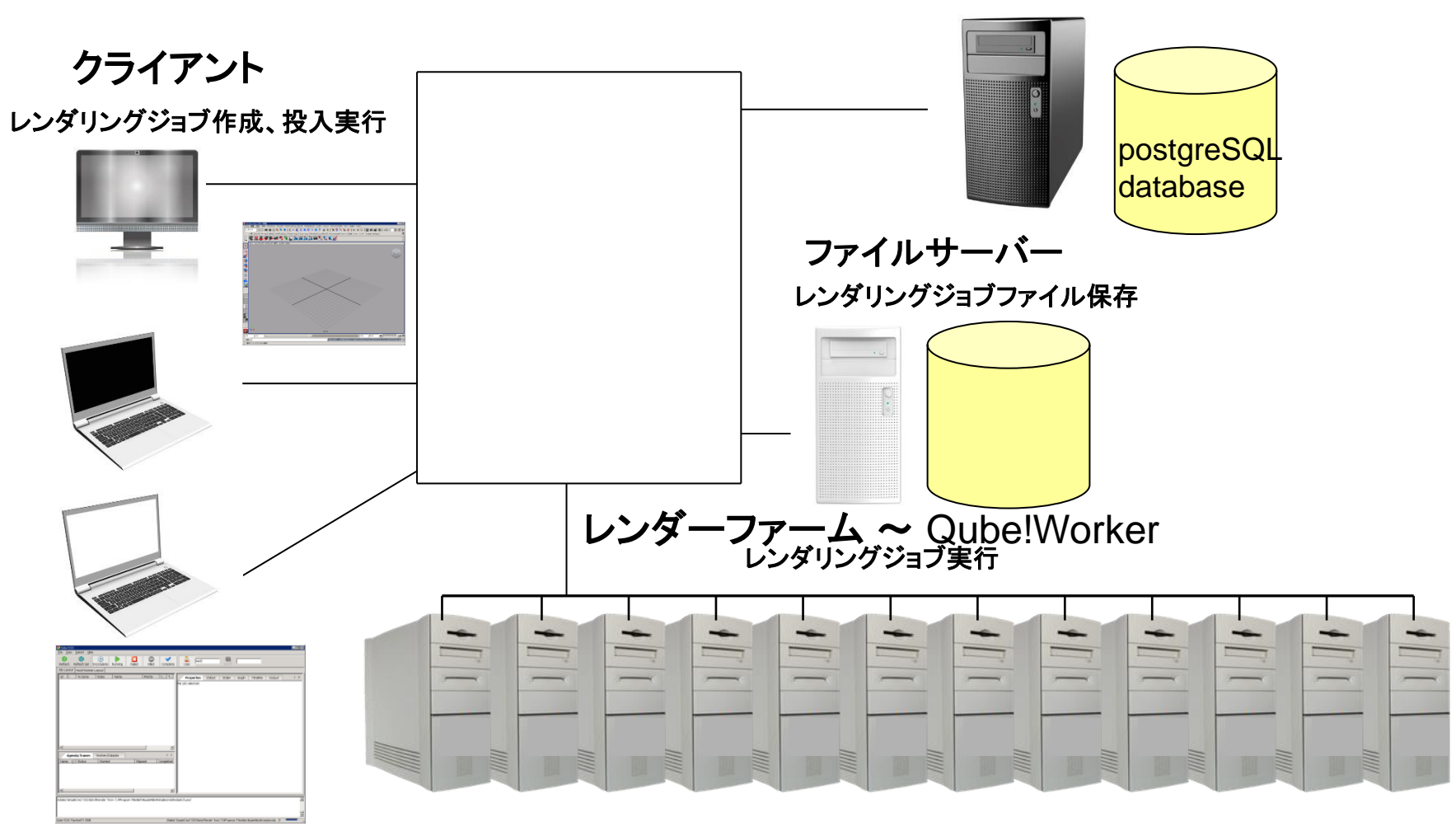
※最新のドキュメントやテクニカルフォーラム、ナレッジベース等にアクセスできます。

Qube! Tube(ビデオデモサイト): <https://www.pipelinefx.com/qubetube/>

2. システム構成について

2-1. 構成図

Qube! システム構成の例です。



2-2. Qube! システムの各コンポーネント

◆Qube!Supervisor ～ レンダリングジョブ管理

Qube!システム下で 1 台にインストールします。

クライアントから投入されたジョブをレンダリングファームへ送ります。

Qube! WranglerView/Artist View を通して各クライアントからのジョブの操作、監視機能も提供します。

更にファームを構成するジョブ実行ホストの稼働状態を監視する事も行います。
ライセンスを管理しています。

◆Qube!Worker ～ レンダリングジョブ実行

バックグラウンドで稼働するプログラムで、Qube!システム下でレンダリングジョブを実行するホストに、Jobtype と呼ばれる各CG アプリケーションに対応したプラグインと同時にインストールします。

Qube! WranglerView/Artist Viewで稼働状況を確認できます。

◆Client ～ レンダリングジョブ作成、投入

CGアプリケーションをインストールしているワークステーション上にインストールします。
GUI、あるいはコマンドラインインターフェースがあり、GUI は専用ツール、また Jobtype と呼ばれる各CG アプリケーションに対応したプラグインと同時にインストールします。

3. システム要件

3-1. システム要件詳細

対応 OS、及びハードウェア要件は以下の通りとなっております。

最新情報は、開発元 (<https://www.pipelinefx.com>) の[PRODUCT]>>[Tech Specs]ページをご参照ください。

(注)Windowsマシンをご使用の場合は、次の設定を行います。

1)コントロールパネル:Worker, Clientにおいては、UAC(ユーザアカウント制御設定)を「無効」にします。

※インストール後も「無効」のままにしておくことを推奨します。(アプリ/レンダー側で常時、「無効」が推奨されていることが多いためです。)

2)コンピュータの管理:Workerにおいて、[Interactive Services Detection]サービスが動作していれば停止させ、[スタートアップの種類]で「無効」にします。

3)ローカル コンピュータ ポリシー:Supervisorにおいて、Windows Defenderの「リアルタイム保護」が有効になっていれば、無効に設定することをお勧めします(P.44 補足資料(4)をご参照ください)。これは、負荷が重く、Supervisorのレスポンスに大きな影響があるためです。セキュリティ保護のためには、インターネット接続無しでの運用、もしくはサードパーティー製のウィルス対策ソフトをご検討ください。

	Operating Systems	Hardware Minimum
Supervisor	<ul style="list-style-type: none"> - Red Hat Enterprise Linux (RHEL) 6.7 – 7.5 - CentOS 6.7 – 7.5 - Ubuntu 16.04 - Windows 8.1, 10 - Windows Server 2012R2, 2016, 2019 - Mac OS X 10.9(Marvericks), 10.10(Yosemite), 10.11(EI Capitan), 10.12(Sierra), 10.13(High Sierra), 10.14(Mojave), 10.15(Catalina) 	<ul style="list-style-type: none"> -64 bit Intel, AMD multi-core processor 4 cores (8 cores 推奨) -4GBメモリ(最少) 8GB以上を推奨 ・25 workers以上ではSSDを推奨
Worker	CG アプリケーションの要件に準じます。	CG アプリケーションの要件に準じます。
Client	同上	同上



3-2. Qube! システムインストール事前確認事項

(1) Qube! システムのネットワーク構成での注意点について

・ユーザー認証

小規模のファームの場合、"qubeproxy" という代理ユーザーでのジョブ実行が可能です。
大規模であり、ドメイン環境、また LDAP、NIS で構成されたネットワークであれば、それぞれの認証スキームに登録されたユーザーでのジョブ実行が可能です。それぞれ proxymode として、前者は proxy、後者は user として設定します。ネットワーク認証を使用する場合、クライアント、レンダーファームのホスト共に同じドメインを利用ください。

・ネットワーク接続

レンダーファームとファイルサーバーの接続はギガビットの利用を推奨します。

・サブネット

Supervisor マシンと Worker マシン は同一サブネット上での使用を推奨します。

※スーパーバイザの自動検出を使用せず、qb_supervisorとqb_domain(P.25)を設定すれば同一サブネットである必要はありません。

・ファイアーウォール

Supervisor と Worker、クライアントの接続にあたってファイアーウォールの利用は推奨しません。
使用する場合はインストール/設定後、Supervisor 上では2つの tcp/udp ポート(50001, 50002)、および、postgreSQLで使用するtcpポート(50055)、Worker 上では1つの tcp/udp ポート(50011)を除外する設定が必要になります。(Qube!で使用するポートは、Qube! 設定で任意のポート番号の指定が可能です)

-ファイルサーバーの利用について-

・アクセス

全てのジョブ実行ユーザーからリード/ライトアクセスが必要になります。
proxymode が proxy の場合(標準設定)、“qubeproxy” というユーザーでのリード/ライトアクセスが必要になります。

・ライセンス

ファイルサーバーが Windows の場合、サーバーのクライアントアクセスライセンスが Worker とクライアントを合わせた数以上のライセンスが必要になります。(一部のCGアプリケーションをのぞく)

・ファイル共有

通常、レンダーファームホストでのジョブ実行後、そのホスト自身に結果ファイルを生成します。
そのため、レンダーリングデータの保存はローカルにマウントしたネットワークドライブへ行きます。

(2) 必須インストールモジュール及びインストール順番

予め弊社からお渡ししたインストールメディア、あるいは、Pipeline FX社Webサイト(<https://www.pipelinefx.com/downloadversions/>)、同社 ftp サイト(<ftp://ftp.pipelinefx.com>) ※1から目的バージョンのモジュールをダウンロードしておいてください。

※1: ご購入時発行のアクセス用アカウント(ユーザーID、パスワード)が必要です。

以下、それぞれのコンポーネント毎に必要なインストールモジュールをまとめます。インストール順番は、必須ソフトウェアモジュールをインストール後、Qube!関連をインストールしてください。上記サイトのqube-installerをご使用になりますと、各モジュールのダウンロード、インストールはほぼ自動で行うことができます。

64ビットOSへは 64ビット版モジュールのインストールを行います。

	必須ソフトウェアモジュール								Qube! Supervisor	Qube! Worker	Jobtype 注2	Qube! Wrangler View	Qube! Artist View 注4
	CG Software	postgre SQL 注1	Perl (ver5.26. 1) 注1	Python (ver2.6- 2.7)注1		Qube! Core	VC++ 2010 再配布 パッケージ (x86)注3	VC++ 2015 再配布 パッケージ (x64)					
Supervisor		○		○		○	○	○	○			○	
Worker	○		○	○		○	○			○	○	○	
Client	○					○	○				○	○	△

(注1) PostgreSQL / Perl / Python/Pywin32 について

1. PostgreSQLは、Windows/Mac OS X/Linux とともに、QubeSupervisorインストーラによりインストールされます。
2. Windowsでは、Active State のサイト<https://www.activestate.com/> からPerl/Python をダウンロードし、インストールします。
3. Linux/Mac OS X では、Perl/Python は通常はOSにバンドルのものがお使いになれますが、Pythonはバージョンをご確認いただき、2.6 もしくは 2.7 をご使用ください。

※ いずれもフリーバージョンで問題ありません。64ビットOS では 64 ビット版の Perl/Python をご利用ください。

(注2) Jobtypeインストーラは、3dsMAX,Maya,Modo,Mtor,Houdini,Nuke,VRED,redshift,Clarisse用のみです。

(注3) WranglerViewで使用します。32bitアプリのため、32bit版をインストールします。

(注4) 機能的には、WranglerViewからArtist用に最低限の機能に絞ったものとなりますので、必須ではありません。

◆ CG ソフトウェアについて

Qube! インストール順にかかわらず、利用アプリケーションのインストール、また必要に応じて各ライセンス設定を行っておきます。

(3) サポートCGアプリケーションと対応状況一覧

各プラットフォーム毎に利用CGアプリケーション対応の以下の Jobtype をインストールします。

この表に記述の無いもの、および最新情報は、開発元 (<http://www.pipelinefx.com/forum/>, <https://www.pipelinefx.com/> - Tech Specs) ページをご参照ください。

	3ds Max	Maya	Modo	Houdini	Cinema 4D	Arnold (kick)	After Effects	nuke			
Windows 32bit	○	○		○	○	○	○	○			
Windows 64bit	○	○	○	○	○	○	○	○			
Mac OS X		○	○	○	○	○	○	○			
Linux		○	○	○		○		○			
対応バージョン (Qube! 7.0-x)	2010 2011 2012 2013 (32/64bit) 2014 2015 2016 2017 2018 (64bit)	2010 2011 2012 2013 (32/64bit) 2014 2015 2016 2017 2018 (64bit)	8~12	9~16	R10~ 20	4~5	CS5, CS5.5 CS6 CC CC2014 CC2015 CC2015.3 CC2017 CC2018	5.0~ 11.2			

4. インストール手順

※Qube! v7.0より前のバージョンからv7.0以降へのバージョンアップの際の準備項目です。

4-0-1. (準備)MySQLからPostgreSQLへのデータ移行

Qube! v7.0では、データベースが、従来のMySQLからPostgreSQLへ変更されています。このため、そのままバージョンアップを行いますと、履歴や統計などのデータがすべて失われてしまいます。これらのデータを引き継ぎたい場合は、**supervisorをバージョンアップする前に**、以下のサイトのステップを実行してください。引き継ぐ必要が無い場合は、このステップは省略していただいて構いません。

- MySQLからPostgreSQLへのデータ移行手順

<https://www.comtec.daikin.co.jp/page.jsp?id=26117&QubeFaqid=79>

(原文:英語) Transferring data from MySQL to PostgreSQL

<http://docs.pipelinefx.com/display/QUBE/Transferring+data+from+MySQL+to+PostgreSQL>

<作業概要>

1. 現在のファームを非アクティブにします。
2. MySQLからデータを、.csvへエクスポートします。
3. Supervisorをバージョンアップします。
4. .csvを、PostgreSQLにインポートします。
5. 新しいSupervisor環境で、ファームをアクティブ化します。

4-0-2. (準備)Perl, Pythonのバージョン確認とバージョンアップ(Windows版supervisorのみ)

現在、インストールされているPerl, Pythonのバージョンをご確認ください。V7.0以降では以下のバージョンが必要となります。

Perl: ver5.26.1 **※特にPerlは、Qube v6.x での対応バージョンよりアップされております。ご注意ください。**

Python: ver2.6 ~ 2.7

4. インストール手順

4-1. Windows

(注)あらかじめActive Perl / Python をインストールしておきます。(対象: ClientおよびWorker、SupervisorはPythonのみ)

(注)あらかじめ.NET Framework 3.5 以上をインストールしておきます。ほとんどの場合、最初からインストールされていますが、Windows Serverではインストール作業が必要な場合があります。 <http://docs.pipelinefx.com/display/QUB065/.NET+Framework+3.5>

(注)あらかじめVisual C++ 2010 再配布パッケージ(x86)をインストールしておきます。WranglerViewで使用しますが、既にインストールされていれば、省略可能です。

ダウンロードサイト : <https://www.microsoft.com/ja-jp/download/details.aspx?id=5555>

(注)supervisorには、Visual C++ 2015 再配布パッケージ(x64)をインストールしておきます。既にインストールされていれば、省略可能です。

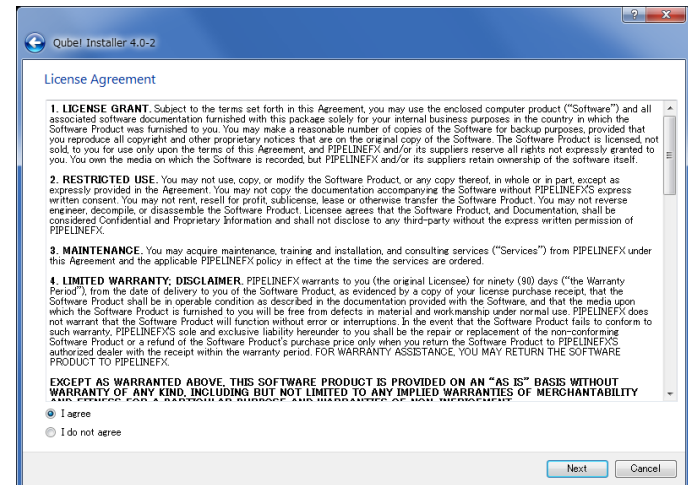
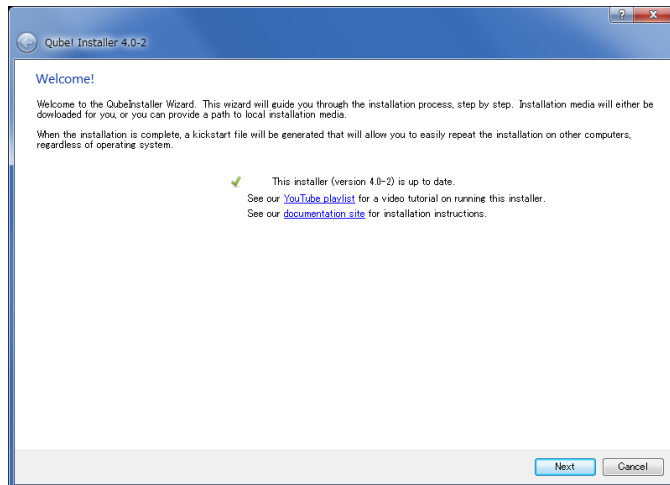
ダウンロードサイト : <https://www.microsoft.com/ja-JP/download/details.aspx?id=53840>

4-1-1. Qube! インストーラの実行(Supervisor, Worker, Clientすべてのマシンで実行します。)

(1) Administratorもしくは、Administrator権限のあるアカウントでログオンします。

(2) 準備した“qubeInstaller.4.0-2.Win32-x64.exe”をダブルクリックしインストーラを起動します。”Next”をクリックします。

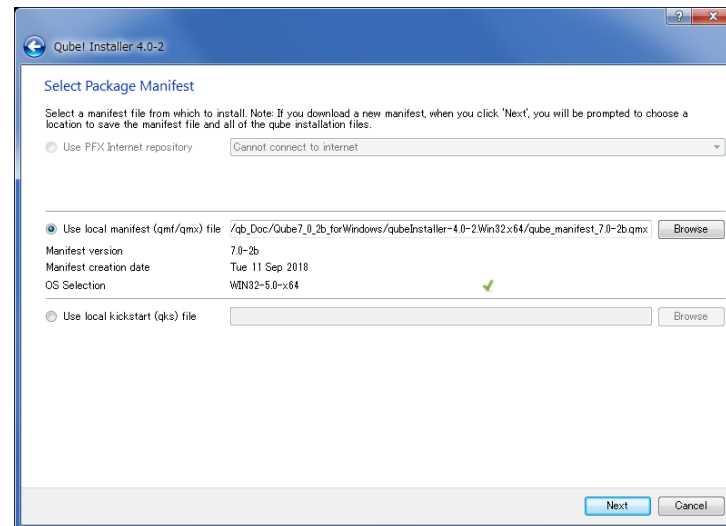
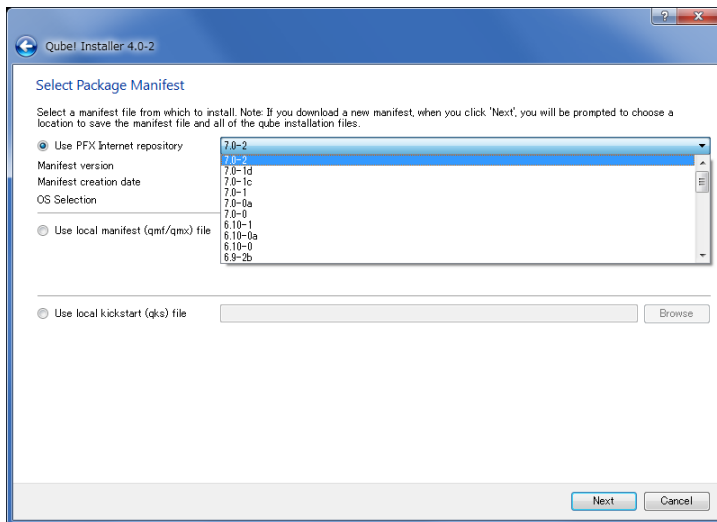
(3) “I agree”を選択し、”Next”をクリックします



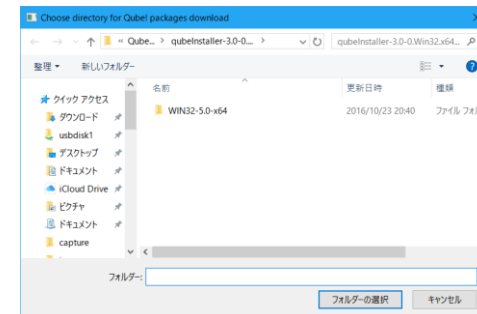
(4) マニフェストファイル(インストールするパッケージ情報を記述したファイル)を取得します。インターネット経由で取得する場合は、“Use PFX Internet repository”を選択します。既に取得済みの場合は、“Use local manifest(qmf) file”を選択し、“Browse”ボタンでそのファイルを指定します。

一通り、インストール作業を行うと、kickstartファイル(.qks)が作成されます。複数台のマシンに同じ設定でインストールしたい場合に、“Use local kickstart (.qks) file”で指定すると、この後の手順を省略してインストールできるほか、コマンドラインでのインストールも可能になりますので、大量のWorkerやClientに対する作業にも便利です。

(コマンド形式) > /path/to/qubeInstaller.version.os.exe /path/to/qube-kickstart.qks

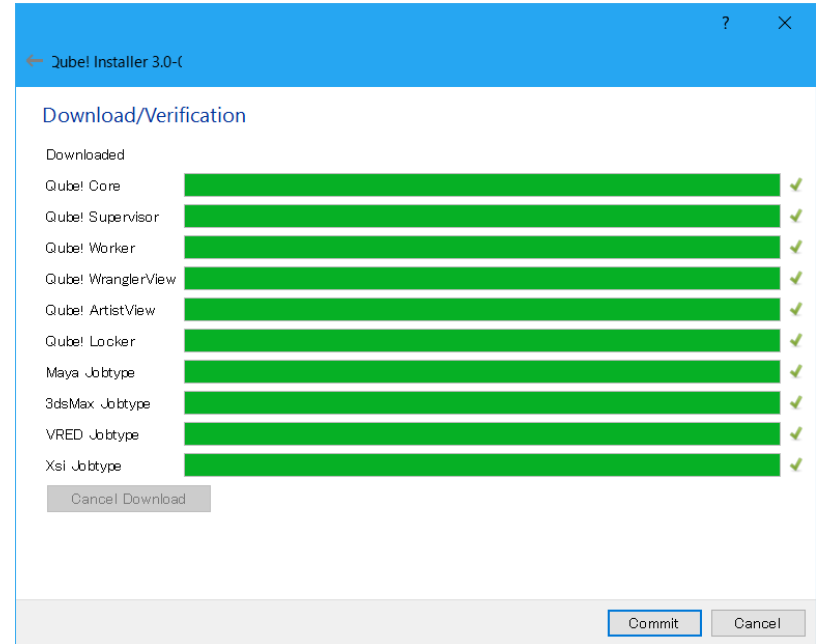
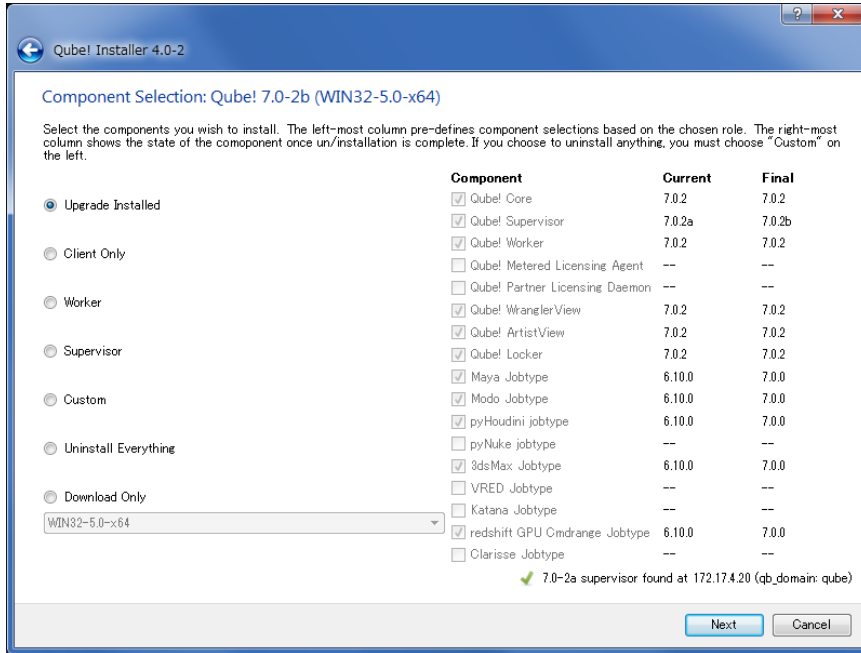


(5) “Use PFX Internet repository”を選択した場合は、ダウンロードしたインストーラを置く、フォルダを指定します。



4-1-2. 旧バージョンからのバージョンアップ(1)

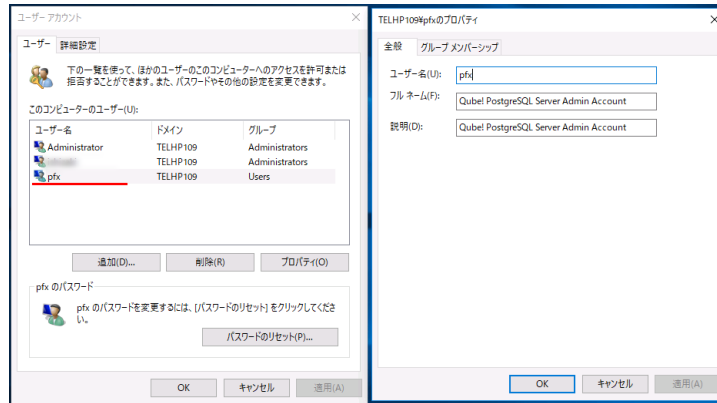
(1) "Update Installed"を選択し、"Next"をクリックします。旧バージョンは自動的にアンインストールされ、同じ構成で新しいバージョンがインストールされます。



4-1-2. 旧バージョンからのバージョンアップ(2) ※supervisorのみの確認作業と設定です。

(2) バージョンアップ後は、データベースが従来のMySQLからPostgreSQLに変更されます。以下を確認、設定します。

① pfxアカウントの確認



ユーザーアカウントに、PostgreSQL用のpfxユーザーアカウントが作成されているのを確認します。②のpostgresql-pfxサービスは、このアカウントで起動されます。通常はログインする必要はありません。

ユーザー名 : pfx

パスワード : Qub3!R0x!

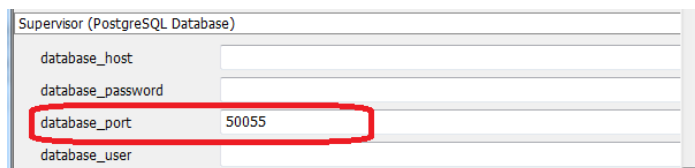
※最後から3文字目は、オーではなくゼロです。

② postgresql-pfx サービスの確認

タスクマネージャの「サービス」タブにて、従来のqubesupervisorサービスの他に、新たに、postgresql-pfxサービスが [実行中] となっていることを確認します。



③ データベースアクセスポート番号の設定

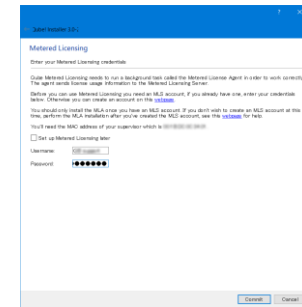
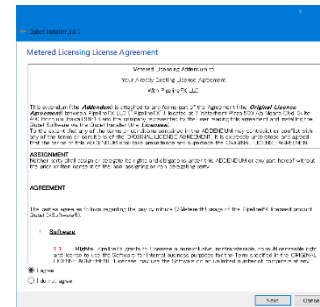
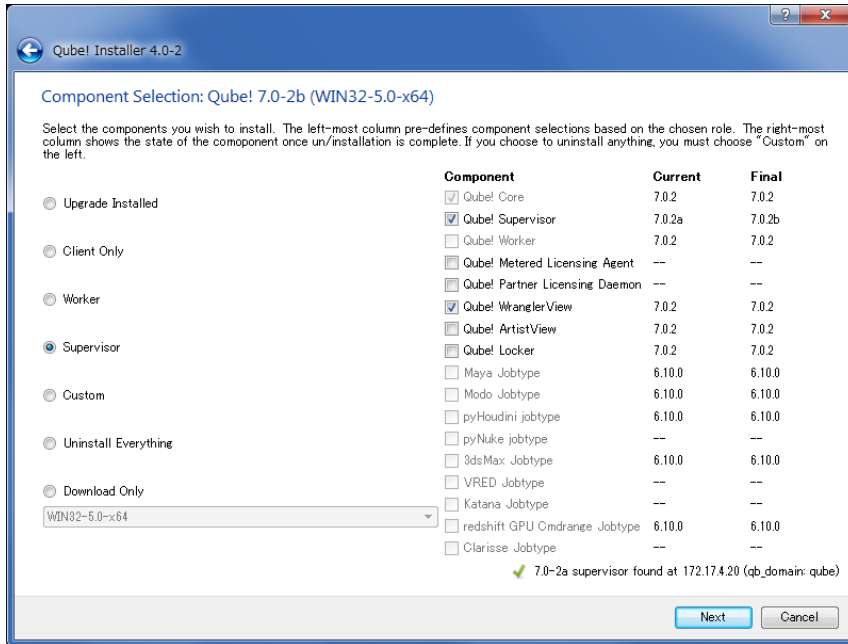


Qube Configurationにて、database_portを、50055に変更します (MySQLでは、3300)。

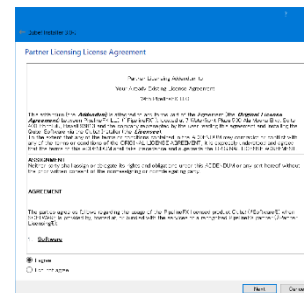
4-1-3. Qube! Supervisor のインストール

- (1) Component Selectionの画面で、“Supervisor”を選択し、“Next”をクリックします。インストールが開始されます。

- (2) Qube! Metered License Agentを選択した場合、ライセンス同意書が表示されます。同意できれば“I agree”を選択します。その後、Metered License アカウントのUsername と Passwordを入力する画面が出てきます。まだ作成していない場合は、後で作成できます。



- (3) Qube! Partner Licensing Daemonを選択した場合、ライセンス同意書が表示されます。同意できれば“I agree”を選択します。



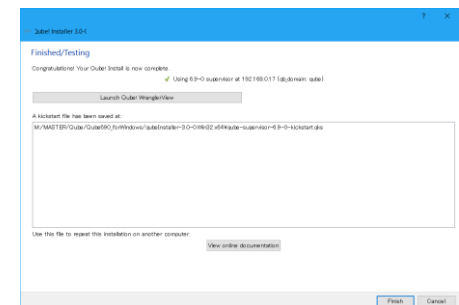
※初めてsupervisorをセットアップされる場合は、Qube! Metered Licensing Agent、およびQube! Partner Licensing Daemonはチェックを外し、インストールされないことをお勧めします。(後でインストール可能です)

<詳細>

<http://docs.pipelinefx.com/display/QUBE/Metered+Licensing>

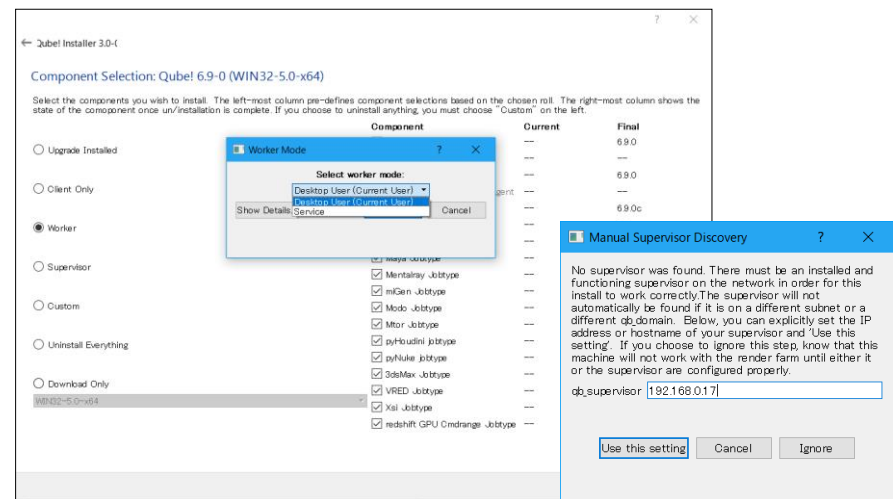
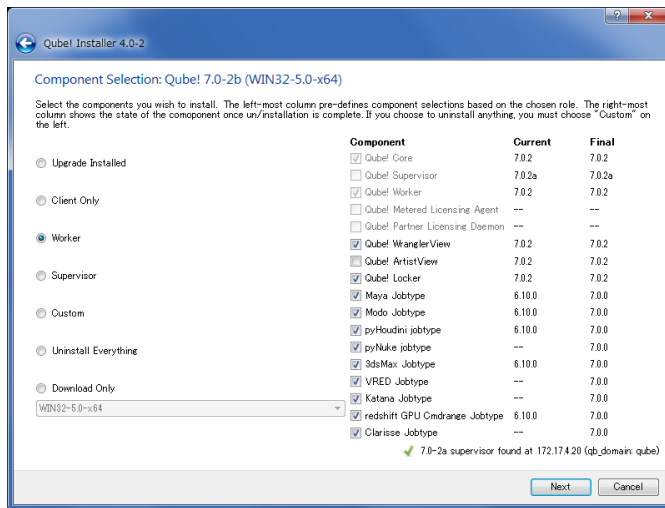
- (4) インストールが終了するとテスト画面が表示されます。問題がある場合はメッセージが表示されます。“Finish”をクリックしインストールを完了します。

完了後マシンを再起動させます。



4-1-4. Qube! Worker のインストール

- (1) Component Selectionの画面で、“Worker”を選択し、“Next”をクリックします。インストールが開始されます。Workerのジョブ実行モード(Desktop User/Service)の選択が可能です。実行モードについては、後述の「5. 設定手順-(3)」をご参照ください。後で変更も可能です。



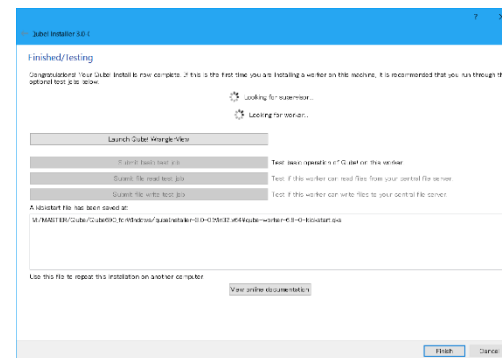
- (2) インストールが終了するとテスト画面が表示されます。

問題がある場合はメッセージが表示されます。

“Finish” をクリックしインストールを完了します。

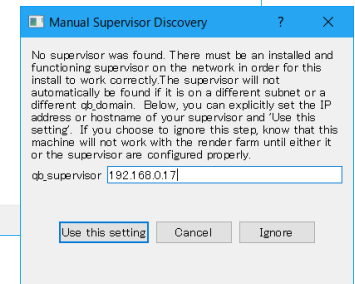
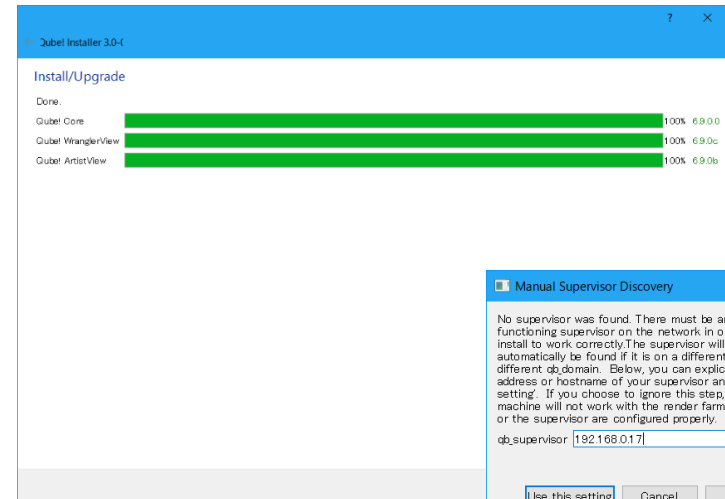
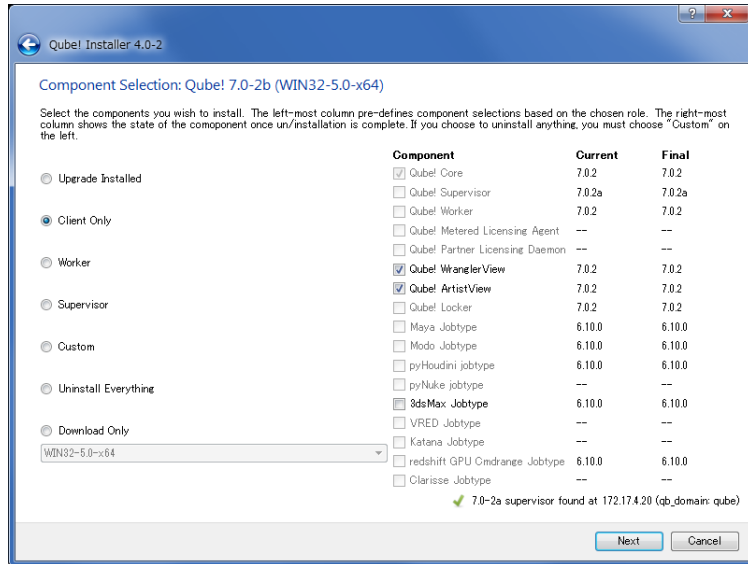
完了後マシンを再起動させます。

Supervisorが見つからない場合、ここで、supervisorのマシン名またはIPアドレスを入力できます。Ignoreにて、後で設定することもできます。



4-1-5. Qube! Client Onlyのインストール

- (1) Component Selectionの画面で、“Client Only”を選択します。必要なJobtypeも選択できます。“Next”をクリックします。インストールが開始されます。

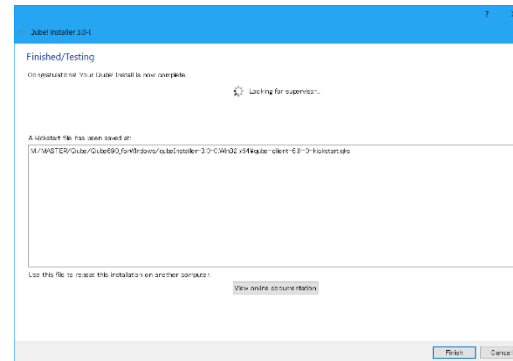


- (2) インストールが終了するとテスト画面が表示されます。

問題がある場合はメッセージが表示されます。

“Finish” をクリックしインストールを完了します。

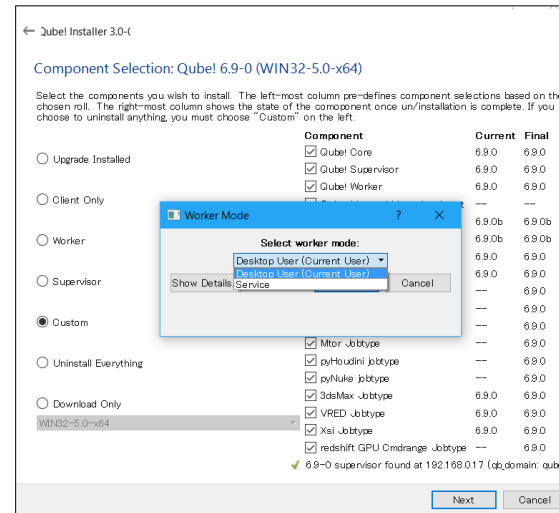
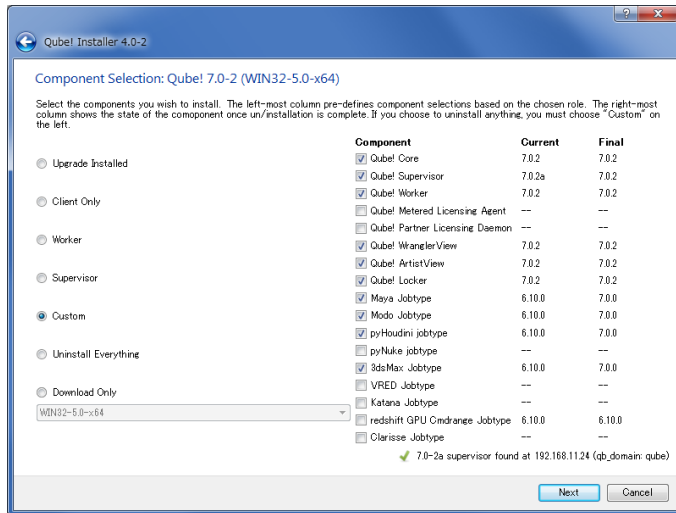
完了後マシンを再起動させます。



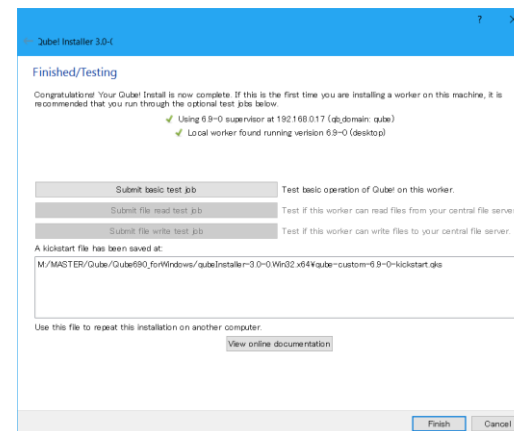
Supervisorが見つからない場合、ここで、supervisorのマシン名またはIPアドレスを入力できます。Ignore!にて、後で設定することもできます。

4-1-6. Qube! Customのインストール

- (1) Component Selectionの画面で、“Custom”を選択し、必要なものを選択し、“Next”をクリックします。インストールが開始されます。Workerのコンポーネントが含まれる場合には、Workerのジョブ実行モードが選択できます。(後で変更できますので、ここではデフォルトのままでも結構です⇒P.24参照)



- (2) インストールが終了するとテスト画面が表示されます。問題がある場合はメッセージが表示されます。“Finish”をクリックしインストールを完了します。完了後マシンを再起動させます。



4-2. Mac OS X

4-2-1. Qube! インストーラの実行 (Supervisor, Worker, Clientすべてのマシンで実行します。)

- (1) root もしくは管理者権限でログインします。
- (2) 準備したインストーラーファイル “qubeInstaller.<ver>.MacOSX-x86_64.dmg” をダブルクリックし、インストーラーを起動します。
- (3) 以後の手順は、Windows版と同じです。4.1.1.-(3)以降の手順に従ってインストールしてください。

4-2-2. QubeGUI/QubeWranglerViewの起動

/アプリケーション/pfx/qube以下のqubeアイコンをダブルクリックします。

※QubeGUIは、6.4以降、QubeWranglerViewという名前に変更になりました。



4-3. Linux

日本語パスが混在している場合、Qubeインストーラが正常にフォルダを認識しないため、英語環境で作業を行ってください(インストール作業後は日本語に戻していただいても構いません。)。また、rootでログインして作業してください。

4-3-1. インストール前準備

- (1) selinuxを”disable”にします。(supervisor, worker, client)

マシンが再起動しても設定を保持するように /etc/selinux/configを直接編集します。

```
# vi /etc/selinux/config  
SELINUX=disabled
```

- (2) (Supervisorマシン、workerマシン) Python, Perlは、OSにバンドルされているものを使用します。

Xinetdのみ別途インストールします。

(例)

Cent OS 7.5: 「端末」から、rootユーザにて、以下のコマンドを実行

```
# yum install xinetd
```

- (3) (クライアントマシン)

Python, Perlは、OSにバンドルされているものを使用します。各CGアプリケーションのインストール、設定を行っておいてください。



4-3-2. Qube! インストーラの実行(Supervisor, Worker, Clientすべてのマシンで実行します。)

- (1) 旧バージョンがインストールされている場合は、予めアンインストールをしておきます。
- (2) rootでログインの上、準備した“qubeInstaller.<ver>-linux”を実行します。
※“linux”の部分は各ディストリビューションの名称毎に置き換えてください。
- (3) 後の手順は、Windows版と同じです。4.1.1.-(3)以降の手順に従ってインストールしてください
- (4) /etc/bashrc ファイルをエディタで開き、次の行をファイルの最後に追加し、保存して閉じます。

```
export QBDIR=/usr/local/pfx/qube[1][SEP]  
export PATH=$PATH:$QBDIR/bin:$QBDIR/sbin
```

4-3-3. Qube!GUI/Qube!WranglerViewの起動

/usr/local/pfx/qube/bin/qube を実行します。

※QubeGUIは、6.4以降、Qube!WranglerViewという名前に変更になりました。



4-4. Job Type

各OS プラットフォーム毎に使用するアプリケーション用のJob Type をWorkerマシン、クライアントマシンにインストールします。

セットアップ時に一緒にインストールされていれば必要ありません。JobTypeの追加インストール時のみ実行してください。

Windows

旧バージョンをアンインストールしておきます。

インストーラーをダブルクリックし、画面の指示に従いインストールを進めます。

Mac OS X

インストーラーファイルをダブルクリックし、マウントされたフォルダ内にある“pkg”アイコンをダブルクリックしインストーラーを起動、インストールを進めます。

Linux

旧バージョンをアンインストールしておきます。

rpm コマンドでインストールします。 i.e.) rpm -ivh <Job type file>

5. 設定手順

(1) ライセンスを設定します。

この設定は Supervisor マシンで実行します。

- ・V6以降のライセンスはSupervisorマシンのMacアドレスを元に発行されます。
- ・通常、0.1バージョンごとに(v6.9やv6.10、v7.0など)ライセンスは異なります。

ただし、ポイントリリース(v6.9、v6.9-1、v6.9-2など)は同じライセンスでご使用いただけます。

弊社からお送りします Qube! ライセンスファイルをインストールします。(ファイル名 : qb.lic)

ライセンスファイルを各OSの所定の場所へ置きます。

Windows の場合 : C:\%programData%\pfx\qube

Linux & Mac の場合 : /etc

ライセンスインストール後、supervisorを再起動し(qubesupervisorサービスを再起動)、ライセンス数を確認します。

Qube!WranglerView([Administrator]→[Ping Supervisor]) または、以下コマンドでご確認ください。(各プラットフォーム共通)

qbping

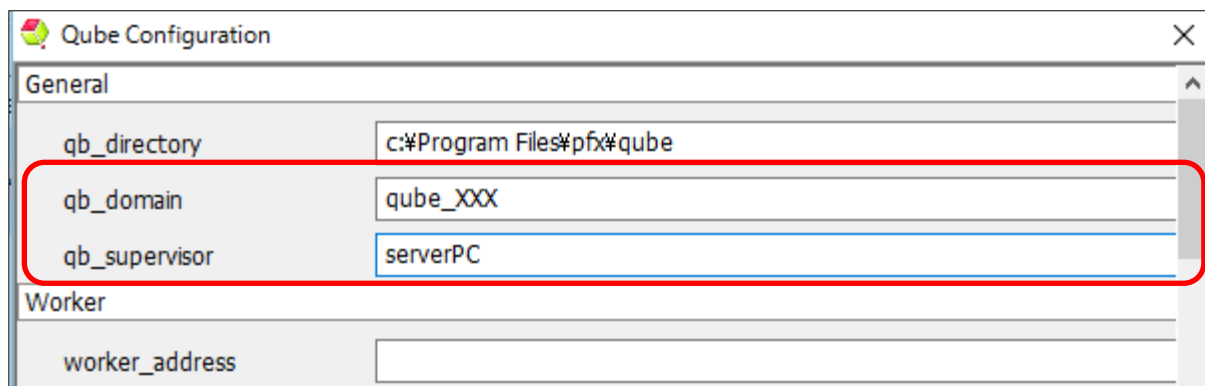
```
supervisor - active - tag: <address> x.x-x bld-x-x-xxxx-xx-xx-x <OS> - - subjob - 0/X licenses.
```

※ 上記 0/X licensesの分母部分がライセンス数となります。

ライセンスをインストールしていない場合、または認識できていない場合は“0”となります。

うまく認識されない場合は、エンコードや改行等の問題である可能性がありますので、Supervisorマシン上で、Qube!WranglerViewを管理者権限で起動し、[Administration] > [Install License on (local) Supervisor] にてライセンスインストールをお試しください。

(2) Supervisor ホスト、ドメイン指定を行います。



この設定は **Supervisor**、**Worker**、**client (GUI)**に対して行います。以下は管理者権限での操作になります。
設定完了までWranglerViewの起動が遅い場合があります。

<Windows の場合>

Qube! WranglerViewを管理者権限で起動し、[Administration]→[Configure Local Host]にて、Supervisor (qb_supervisor にホスト名かIPアドレスで設定)、Supervisor Domain (qb_domain に任意名を設定)を設定します。

※[Configure Local Host]がグレイアウトしている場合は、権限の問題である可能性があります。Qube!起動アイコンを右クリックし、「管理者として実行」で起動してください。

<Mac の場合>

Finder から、[アプリケーション]→[pfx]→[Qube! Preferences] を選択、起動します。

<Linux の場合>

Qube! WranglerViewの[Administration]→[Configure Local Host] または、直接設定ファイル (/etc/qb.conf) を編集します。

(3) ジョブ実行モードの設定を行います。

ジョブの実行モードは、Serviceモード(proxy と user)、Desktop Userモードの3種類があります。この設定は Workerに対していずれかに設定する必要があります。以下は管理者権限での操作になります。

Serviceモード(proxyとuser)は、Qube!WranglerViewの[Administration > Worker > Start as Service]、および[Administration > Configure Local Host] のproxy_execution_modeにて、Desktop Userモードは、[Administration > Worker > Start as Desktop User] にて行います。

<Service: proxy モード> ※proxy_execution_mode: proxy

代理ユーザー(qubeproxy) によるジョブ実行になります。ユーザー認証をネットワーク認証していない小規模ネットワークでの設定です。(Windows Workgroup など)

デフォルト: Username: **qubeproxy** (表示名: Qube! Proxy) Password: **Pip3lin3P@\$swd**

<Service: user モード> ※proxy_execution_mode: user

ログオン(ログイン) ユーザー権限でジョブを実行します。ネットワーク認証(Windows ドメイン、NIS、LDAP) での認証スキームを利用できます。Windowsドメイン認証を使用する場合のみ、Qube! システムへ実行ユーザーのパスワードを登録する作業が必要になります。

このモードの場合、クライアント側でQube! WranglerViewの[Administrator > Register Windows Password]よりジョブ実行ログオンユーザーのパスワード登録(初回のみ)を行います。((5)パスワード登録(Windows環境)を参照ください。)

<Desktop Userモード> (デフォルト)

ワーカーの表画面にログインしているユーザー権限でジョブを実行します。GPUを使用するようなハードウェアレンダに対応でき、ジョブの実行ユーザIDやドライブマウントをDesktop ユーザの環境をそのまま引き継ぐため、比較的簡単に初期設定が可能であるというメリットがあります。また、AfterEffectsや3ds Maxなどでは、アプリケーションやプラグインにより、ハードウェアレンダ以外にも、Desktop Userモードのほうが導入しやすい場合(もしくはそうでないと動作しない)があります。

例えば、After Effectsで、Quicktimeコーデックを使用したジョブを投入する場合はDesktop Userモードにする必要があります。

このモードで運用する場合は、意図するユーザアカウントにて、ワーカーにログインし、必ず表画面で運用している状態にしてください。裏でログインしている状態では正常な動作は保証されません。

【設定方法】(Desktop UserモードからServiceモードへの切り替え、またはその逆)

管理者権限を持ったユーザーでQube! WranglerViewを起動し、以下の設定を行います。

1. 現在のモードを停止します。

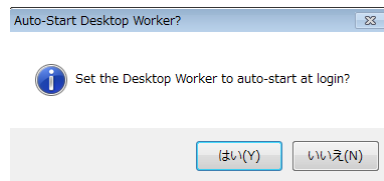
Administration > Worker:[現在のモード] > Stop (現在のモードを停止します)

2. 新しいモードで起動します。

Administration > Worker:Stoped > Start as the Desktop User または、Start as a Service

※Desktop Userに切り替えた際、以下のようなダイアログが表示されたら「はい」をクリックします。

※Administration > Autostart Worker:[現在のスタートアップ設定] にて、自動起動の設定が可能です。



(4) Job Type の設定を行います。

Jobtypeインストーラは、3dsMAX,Maya,Modo,Mtor,Houdini,Nuke,VRED,redshift,Clarisse用のみです

各 CG アプリケーション GUI メニューから Qube! ジョブ投入ダイアログを表示させるための設定を行います。

(管理者権限での操作が必要です。)

3dsMax ※3ds Maxでは、clientおよびworkerに、この作業が必要です。インストーラ側で既に実施済みであればこの作業は不要です。Clientおよびworkerに、3dsMax本体およびJob Type インストール後、3dsMax を起動し、“Qube!” メニューがアクティブであることをご確認ください。

アクティブにならない場合は、C:¥Program Files¥pfx¥jobtypes¥_3dsmax¥loadqube.mzp を

C:¥Program Files¥Autodesk¥3ds Max <ver>¥scripts¥Startup にコピーしてください。

Workerは、Desktop Userモードで運用されることをお勧めします。

After Effects

Client側では、Qube! WranglerViewの[File > Install App UI]から[Install Aerender(AppFinder) App UI]をインストールします(インストール先: <AE inst dir>¥ Support Files¥Scripts)。After Effects を起動し、[一般設定]で[スクリプトによるファイルへの書き込みとネットワークへのアクセスを許可]を有効にします。

[ファイル]→[スクリプト]→ [Qube_XXX]メニューがあることをご確認ください。

(CS3 バージョンは After Effects メニューからのジョブ投入がサポートされていません)

Worker側では、CS6以降、After Effectは非ロイヤリティベアリングモードで使用しますので、アクティベートの必要は無くなりました。この場合、ae_render_only_node.txtという空ファイルを以下のフォルダに置きます。

・Windowsの場合: C:¥Users¥<username>¥Documents または、C:¥Users¥Public¥Documents¥Adobe

・Macの場合: /Users/<username>/Documents/ または、/Users/Shared/Adobe/

Workerは、Desktop Userモードで運用されることをお勧めします。

Fusion

Qube! WranglerViewからJobを投入します。

LigthWave3D

Qube! WranglerViewの[File > Install App UI]から[Install Lightwave(ScreamerNet) App UI]をインストールします。

Maya

Worker側は、Maya本体をそのままインストールしておきます。Mayaソフトウェアレンダラは、ライセンスは必要ありませんが、Arnoldレンダラはライセンスが必要です。

また、Worker側は、Job Type をインストールし、Client側では、Qube! WranglerViewの[File > Install App UI]から[Install Maya App UI]を実行します。

必要なファイルが、以下のフォルダにインストールされます。(各バージョン共通で使用可)

Windowsの場合: C:¥Users¥<username>¥Documents¥maya¥scripts

Macの場合: /Users/<username>/Library/Preferences/Autodesk/maya/scripts

Shake

Qube! WranglerViewからJobを投入します。

Nuke

Qube! WranglerViewの[File > Install App UI]から[Install Nuke(cmdline) App UI]をインストールします。

Cinema4d

Worker側は、Cinema4D本体をそのままインストールしておきますが、コマンドラインレンダラのライセンスが必要です。

Client側は、Qube! WranglerViewの[File > Install App UI]から[Install Cinema4d(OSX) App UI]または[Install Cinema4d(Win) App UI]をインストールします。OSX/Winのどちらをインストールするかは、ワーカーのOSではなく、自マシンのOSに依存します。

(インストール先: <C4D inst dir>¥library¥scripts もしくは、Scripts > User Scripts > Script Folder のフォルダ)

modo

Worker側は、modo本体をそのままインストールしておきます。コマンドレンダラ(modo_cl.exe)のライセンスは不要です。

また、Worker側は、Job Typeをインストールし、Client側は、Qube! WranglerViewの[File > Install App UI]から[Install Modo (cmdrange) App UI]をインストールします。(インストール先: C:¥Users¥<username>¥AppData¥Roaming¥Luxology¥Configs)

英語版と日本語版はインストールパスが異なるため(フォルダ名に“j”が付加される)、投入時、modo_cl.exeへのパスを修正する必要があります。面倒な場合は、各ClientとWorkerにて、シンボリックリンクを設定しておく便利です。

(例): mklink /D "C:¥Program Files¥Foundry¥Modo¥12.1v1" "C:¥Program Files¥Foundry¥Modo¥12.1v1j"

(5)パスワード登録 (Windows環境のみ)

この作業はWranglerViewが起動する任意のPC上で行います。ジョブ投入する全てのユーザーでの設定が必要になります。

設定は初回一度だけでOKです。(設定は、supervisorに保持されます)

Windowsドメインネットワークを使用し、proxy モードを“user”と設定している場合、ジョブ実行前にQube! システムへのパスワード登録を行います。

- (1) Supervisor で設定したドメインのドメインユーザーでログオンします。
- (2) Qube!WranglerViewの[Administrator > Register Windows Password] を起動します。
- (3) ログオンユーザーのパスワードを入力、確認入力し OK をクリックします。



The image shows a Windows-style dialog box titled "Password Registration". It contains three text input fields: "Windows Username", "Windows Password", and "Confirm Windows Password". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

6. qb.confの設定(Advanced)(1)

Qube Configuration

General

qb_directory c:\Program Files\pfx\qube

qb_domain qube_XXXX

qb_supervisor 192.168.0.16

Worker

worker_address

worker_boot_delay 0

worker_check_interval 1800

worker_cluster

worker_cpus 0

worker_description

Drive	Network Path (UNC)	Username	Password

worker_drive_map

worker_flags auto_mount,remove_logs Choices

worker_flight_check_path Browse

worker_groups

worker_heartbeat_interval 90

OK Cancel

== General ==

qb_directory ... Qube!のインストールフォルダ。

qb_domain ... 複数のsupervisorが存在しているファームで、supervisorのドメインを分けるのに使用します。ドメイン毎に任意の名前を設定します。

qb_supervisor ... subervisorのマシン名、またはIPアドレスを設定します。指定しない場合は同じサブネット上で自動的に検出します。

== Worker ==

worker_address ... ワーカーとして認識させるIPアドレスを記述します。指定しない場合は、このワーカーの持つすべてのIPアドレスが対象となります。

worker_boot_delay ... システムの初期化のためにワーカーのスタートアップの待ち時間を秒で指定します。OS Xワーカーの場合、OS Xネットワークの起動に時間がかかるための待ち時間設定に便利です。

worker_check_interval ... プロセスが走っているかどうかをチェックする間隔を秒で指定します。デフォルトは、1800です。

worker_cluster ... ワーカーが属するクラスタを設定します。

⇒ 補足資料(1)参照

worker_cpus ... ワーカーで使用できるCPUコア数を設定します。デフォルトは0で、自動的にCPUコア数を検出します。

worker_description ... ワーカーのコメントを記述します。

worker_drive_map ... (Windowsのみ)プロセスが実行される前に自動的にドライブにマウントするネットワークパスを記述します。worker_flagsフィールドにauto_mountフラグが必要です。

worker_flags ... ワーカーのコンフィグパラメータを設定します。

dedicated/dynamic/auto_mount/remove_logs/load_profileをコンマで区切って指定します。

worker_flight_check_path ... ジョブの前後で実行するプログラムを置くパス(flight check path)を指定します。複数指定するには、コンマで区切ります。デフォルトは、\$QBDIR/flightCheck です。

worker_groups ... ワーカーの属するグループを設定します。複数グループに属するには、グループ名をコンマで区切ります。

worker_heartbeat_interval ... ワーカーがスーパーバイザに送信するハートビートパケットの時間間隔を秒で指定します。デフォルトは90です。

qb.confの設定(Advanced)(2)

Qube Configuration

worker_heartbeat_interval 90

worker_host_domain

worker_idle_threads 4

worker_job_types

worker_journal_location Browse

worker_log_timeout 10

worker_logfile Browse

worker_logmode remote

worker_logpath Browse

worker_max_clients 256

worker_max_threads 8

Remote Path	Local Path

worker_path_map

worker_pidfile Browse

worker_port 50011

worker_post_interval 43200

OK Cancel

worker_host_domain ... ワーカーが所属するウィンドウズのドメインを指定します。デフォルトはローカルコンピュータのドメインが使用されます。

worker_idle_threads ... 負荷最小時のワーカースレッドのアイドル数を指定します。

worker_job_types ... ワーカーが実行を許可するジョブタイプのリストを明示的に記述します。デフォルトは、worker_template_path以下に存在するジョブタイプを自動的に検出します。

worker_journal_location ... ワーカーのjournalファイルを置くパスを記述します。Journalファイルは実行中のジョブに関する様々な情報を含むファイルです。デフォルトは、(Windows)C:\ProgramData\%Pfx%\Qube\logs\worker.jnl です。

worker_log_timeout ... logmodeが"remote"の時、supervisorに通常のログ出力やエラーログを送信する前の待ち時間を設定します。デフォルトは10です。

worker_logmode ... ワーカーログモードを設定します。"remote"の時は、ジョブログはsupervisorへ送信され、"mounted"の時は、worker_logpathで設定されたフォルダへ出力されます。このフォルダは、supervisorからも共有でマウントされます。

worker_logpath ... ワーカーが出力する標準およびエラーのログを出力するパスです。デフォルトは、(Windows)C:\ProgramData\%Pfx%\Qube\logsです。

worker_max_clients ... このワーカーにアクセスするクライアントの最大値を設定します。ワーカーサービスが止まるときなどにクライアント側に送信するメッセージのために参照し、値を増やすとオーバーヘッドを少なくできますが、使用されなくなったメモリの開放が遅くなるデメリットがあります。

worker_max_threads ... ワーカーがメッセージ類を処理する際の最大スレッド数を設定します。ワーカーのパフォーマンス調整に使用でき、増やすと、必要とするメモリも増加します。通常の使用では、特に変更する必要はありません。デフォルトは8です。

worker_path_map ... ワーカー側でパス変換を行います。異なるプラットフォームが混在しているファームで便利です。JobのFlagsで、"convert_path"が設定されているジョブや、pyCmdline, pyCmdrange, AppFinderジョブで使用されます。

worker_pidfile ... ワーカーがプロセスをコントロールするために、プロセスIDを記述するためのパスを記述します。デフォルトは、(Windows)C:\ProgramData\%pfx%\qube\logs\workerpid です。

worker_port ... ワーカーサービスのために割り当てるTCP/UDPポート番号です。Supervisor側にも設定する必要があります。デフォルトは、50011です。

worker_port_interval ... ワーカーが何も実行していない時に、supervisorにStatusを送信する間隔を設定します。デフォルトは12時間(43200秒)です。

qb.confの設定(Advanced)(3)

worker_process_timeout ... ワーカーがプロセスをチェックするためのタイムアウト(秒)を設定します。

worker_properties ... ワーカーに与えるプロパティを記述します。host.os, host.cpusなど既に組み込まれているものもあります(ユーザガイド参照)。ここで記述したプロパティを、ジョブ投入時の"Restriction"や"Host Order"等で指定することにより、条件付けが可能です。

worker_resources ...ジョブ投入時の"Reservations"で指定できる、ワーカーのリソースの記述を行います。Host.memory, host.processorsが設定でき、自身のリソース情報を上書きできます。

worker_restrictions... ワーカーで実行するジョブをクラスタ名をベースにして制限します。worker_clusterで"/private"、worker_restrictionsで"/private"と設定すると、クラスタ名"/private"で投入されたジョブのみ実行可能となります。"*"を付加して、1階層下、"+"を付加して、その階層以下という指定も可能です。

worker_template_path... ワーカーがジョブ実行時に参照するライブラリ類がインストールされているパスを指定します。OS X/Linuxではコロン、Windowsではセミコロンで区切って複数パスが指定できます。ローカルのjobtypeのパス(デフォルト)は必ず含めておきます。

== Worker (User) ==

proxy_account... proxyモードのとき、ワーカーにログインの際に使用されるアカウント。デフォルトでは"qubeproxy"でジョブが実行されます。

proxy_execution_mode... ワーカーの実行モードを指定します。"proxy"の場合、proxy_accountのアカウントで投入され、"user"では投入ユーザのアカウントで投入されます。

proxy_location... proxyアプリの位置を指定します。Windowsではデフォルトで、%QBDIR%\%sbin%\proxy.exe が使用されます。

proxy_nice_value... 投入されたジョブの優先度を調整します。例えば、クライアント兼ワーカー機の場合、大きな値を設定して、投入されたジョブのプライオリティを下げることができます。-20から20の範囲で設定します。

proxy_password... proxyモード時のproxy_accountのパスワードを設定します。qhashコマンドで暗号化した値を設定します。

qb.confの設定(Advanced)(4)

The screenshot shows the 'Qube Configuration' dialog box with the 'Supervisor (Commonly Changed)' section expanded. The settings are as follows:

Parameter	Value
supervisor_default_priority	9999
supervisor_default_security	submit_job,submit_callback,kill,remove,modify,preempt,bl
supervisor_flags	heartbeat_monitor,host_recontact,retry_busy,running_tr
supervisor_global_resources	
supervisor_highest_user_priority	0
supervisor_host_policy	open
supervisor_idle_threads	8
supervisor_language_flags	auto_wrangling,dependency,mail,perl,post,python,qube
supervisor_logfile	
supervisor_logpath	
supervisor_max_clients	256
supervisor_max_threads	32

Below this section, other categories are visible but not expanded:

- Supervisor (Smart Share)**
 - supervisor_default_max_cpus: -1
 - supervisor_max_cpus_limit: 100
 - supervisor_smart_share_mode: jobs
 - supervisor_smart_share_preempt_policy: aggressive
- Supervisor (User Job Limits)**

== Supervisor (Commonly Changed) ==

supervisor_default_priority ... ジョブの優先度を設定せずに投入した場合のデフォルトのジョブ優先度を設定します。

supervisor_default_security ... パーミッションを明示的に設定されていないユーザに適用されるパーミッションを設定します。

supervisor_flags ... スーパーバイザのグローバル設定フラグをマスクで設定します。

supervisor_global_resources ... ファーム内のリソース数(通常、ライセンス数)を設定します。例えば、global.maya=10 のように設定すれば、ファーム内で10ライセンスを超えない範囲でMayaジョブが実行でき、超えた分はpendingされます。

supervisor_highest_user_priority ... 非管理者ユーザが投入するジョブの最も高い優先度値を設定します。(値としては最小値)

supervisor_host_policy ... ファームに参加させるワーカーをどのように決めるか設定します。

open: 制限は無く、どのワーカーでも参加できます。

restricted: supervisor_worker_configfileにて記述された名前もしくはアドレスで設定します。

restrictedbyname: supervisor_worker_configfileにて記述された名前で設定します。

restrictedbyaddress: supervisor_worker_configfileにて記述されたIPアドレスで設定します。

supervisor_idle_threads ... スーパーバイザの最小スレッド数を設定します。これにより、ネットワークメッセージ用にそのスレッド数分確保します。使用メモリ容量に影響しますのでパフォーマンスの改善に役立ちます。

supervisor_language_flags ... コールバックコードで使用できる言語を設定します。

supervisor_logfile ... スーパーバイザのログファイルを設定します。Windowsの場合、デフォルトは、C:\ProgramData\pfx\qube\logs\supelog

supervisor_logpath ... ジョブログを保存するフォルダを設定します。Windowsの場合、デフォルトは、C:\ProgramData\pfx\qube\logs

supervisor_max_clients ... dyingになる前にサービスに割り当てるネットワークメッセージのスレッドの最大値を設定します。これにより、再投入の際のスレッドのオーバーヘッドを減少できる反面、未使用メモリの解放が遅くなります。

supervisor_max_threads ... スーパーバイザスレッドの最大値を設定します。増やすとスーパーバイザのパフォーマンスを改善できますが、使用メモリも増加します。

qb.confの設定(Advanced)(5)

== Supervisor (Smart Share) == ⇒補足資料(2)参照

supervisor_default_max_cpus・・・ SmartShareによって拡張されるインスタンス数のデフォルトの最大数を設定します。ジョブ投入時のMax Instancesで明示的に指定されない場合は、この値が使用されます。-1または*(アスタリスク)を設定すると、“制限無し”となります。

supervisor_max_cpus_limit・・・ SmartShareによって拡張されるインスタンス数の最大値を設定します。ジョブ投入時のMax Instances値に関係なく、すべてのジョブに適用される上限値となります。

supervisor_smart_share_mode・・・SmartShare機能を使用するかどうかを設定します。

jobs: SmartShareを使用します。
none: SmartShareを使用しません。

supervisor_smart_share_preempt_policy・・・ SmartShareで拡張されたインスタンスに対し、Supervisorが使用する割り込み方法を設定します。
passive: 現在のプロセスが終了するまで待って、割り込みを行う。
aggressive: 現在のプロセスを直ぐに終了させて割り込みを行う。(デフォルト)
disabled: 割り込みを行わない。

== Supervisor(User Job Limits) ==

supervisor_default_pgrp_subjob_limit・・・ 同時に実行するプロセスグループのサブジョブの最大値を設定します。-1は制限なしを意味します。例えば、Rendermanなどでは、1つのショット内に、同じプロセスグループで複数のサブジョブから構成されています。

supervisor_default_user_subjob_limit・・・ すべてのプロセスグループ間で、同時に実行するサブジョブ数の最大値を設定します。-1は制限なしを意味します。

supervisor_pgrp_subjob_limits・・・ 特定のユーザにプロセスグループのサブジョブ数のリミット値を設定します。root=-1,eric=10,.. のように設定します。指定しないユーザには、supervisor_default_pgrp_subjob_limitの値が適用されます。

supervisor_user_subjob_limits・・・ 特定のユーザにサブジョブのリミット値を設定します。root=-1,jimi=20,.. のように設定します。指定しないユーザには、supervisor_default_user_subjob_limitの値が適用されます。

qb.confの設定(Advanced)(6)

The screenshot shows the 'Supervisor (Email)' section of the Qube Configuration dialog. It contains several input fields for configuring email settings:

- mail_administrator: [Empty text box]
- mail_domain: [Empty text box]
- mail_host: [Empty text box]
- mail_job_status_format: [Empty text box] with a 'Browse' button to its right.
- mail_license_status_format: [Empty text box] with a 'Browse' button to its right.
- mail_port: [Empty text box]
- mail_subjob_status_format: [Empty text box] with a 'Browse' button to its right.
- mail_work_status_format: [Empty text box] with a 'Browse' button to its right.

Below this section are other configuration sections: 'Supervisor (Auto-Wrangling)' with 'aw_activation_work_count' (5) and 'aw_job_migrate_max' (3); 'Supervisor (PostgreSQL Database)' with 'database_host', 'database_password', 'database_port' (50055), and 'database_user'; and 'Supervisor (Advanced)' with 'supervisor_address' and 'supervisor_backuppath'.

== Supervisor(Email) ==

mail_administrator・・・ supervisorが送付するメールユーザアカウントを記述します。システムのクリティカルイベント、ライセンスの通知などのemailが送付されます。

例: admin@mydomain.com

mail_domain・・・ドメイン名を設定します。ユーザがフルでアドレス名を指定していない場合、Supervisorが、ユーザ名にこのドメイン名を結合して、メールアドレスとします。例: pipelinefx.com

mail_host・・・ emailを送信するのに使用するSMTPサーバーを設定します。このサーバーはパスワード無しをサポートし、送信先SMTPサーバーである必要があります。例: mail.pipelinefx.com

mail_job_status_format・・・ ユーザにジョブステータスを送付する時に使用されるフォーマットを含むファイルを指定します。

Windowsデフォルトでは、C:\Program Files\pfx\qube\etc\job.mail です。

mail_license_status_format・・・ ライセンス期限などの警告メッセージを送付するときに使用されるフォーマットを含むファイルを指定します。

Windowsデフォルトでは、C:\Program Files\pfx\qube\etc\license.mail です。

mail_port・・・ SMTPサーバーと通信するために使用されるポート番号を設定します。デフォルトは、25です。

mail_subjob_status_format・・・ ユーザにサブジョブステータスを送付する時に使用されるフォーマットを含むファイルを指定します。

Windowsデフォルトでは、C:\Program Files\pfx\qube\etc\subjob.mail です。

mail_work_status_format・・・ ユーザにワークステータスを送付する時に使用されるフォーマットを含むファイルを指定します。

Windowsデフォルトでは、C:\Program Files\pfx\qube\etc\work.mail です。

== Supervisor(Auto-Wrangling) == ⇒ 補足資料(2)参照

aw_activation_work_count・・・ ワーカーで、この設定回数だけfailすればauto-wranglingがアクティブ状態になります。

aw_job_migrate_max・・・ ワーカーに不具合等がある場合、auto-wranglingにて、ジョブをfailにするか、ブロックするかを決定する前に、ジョブを自動で他のワーカーに移行させようとしています。ここでは移行を試行する最大数を設定します。

qb.confの設定(Advanced)(7)

The screenshot shows the 'Qube Configuration' window with two sections:

- Supervisor (PostgreSQL Database)**
 - database_host: []
 - database_password: []
 - database_port: 50055
 - database_user: []
- Supervisor (Advanced)**
 - supervisor_address: []
 - supervisor_backuppath: []
 - supervisor_default_cluster: /
 - supervisor_default_hostorder: []
 - supervisor_heartbeat_interval: 360
 - supervisor_heartbeat_timeout: 1080
 - supervisor_job_flags: [] (Choices)
 - supervisor_log_flags: admin, callback, job, mail, subjob, user, work (Choices)
 - supervisor_manifest: [] (Browse)
 - supervisor_manifest_flags: [] (Choices)
 - supervisor_max_postboard_retry: 8
 - supervisor_max_priority: 9999
 - supervisor_max_worker_queue: 8

Buttons: OK, Cancel

== Supervisor (PostgreSQL Database) ==

database_host ... PostgreSQLデータベースサーバーが動作しているマシンを指定します。Supervisor機と異なるマシンで動作している場合にのみ設定します。

database_password ... データベースへのパスワードを設定します。qbhashコマンドで暗号化した文字列を設定します。

database_port ... Supervisorがデータベースサーバーと通信する際に使用されるポート番号を設定します。デフォルトは、50055です。

database_user ... SupervisorがPostgreSQLデータベースにアクセスするユーザのIDを設定します。デフォルトは、qube です。

== Supervisor(Advanced) ==

supervisor_address ... Supervisorが使用するIPアドレスを限定します。複数のIPアドレスを持つマシンの場合、使用したい方のIPアドレスを記述します。記述しない場合は、マシン持つすべてのIPアドレスが使用されます。

supervisor_backuppath ... Supervisorがデータベースをバックアップするディレクトリを指定します。Supervisorマシンが落ちるとき、postgresqlをコピーし、Supervisorの現在の状態をリカバーできるようにします。

supervisor_default_cluster ... デフォルトのクラスタを設定します。投入するジョブにクラスタが指定されていない場合、ここで指定されたクラスタに投入されます。デフォルトは、/ (ルート)です。

supervisor_default_hostorder ... ジョブが投入されるホストの順番を設定します。host.processors, host.memory, host.processor_speedなどが使用でき、+ または - で要求度を強くしたり、弱くしたりできます。

supervisor_heartbeat_interval ... ワーカーがsupervisorにステータスを送信する時間間隔を秒で設定します。supervisor_heartbeat_timeoutで設定した時間内にレスポンスが無いワーカーは"down"とマークされます。

supervisor_heartbeat_timeout ... ワーカーがsupervisorとsupervisor_heartbeat_intervalの時間間隔で通信する際にタイムアウトと見なされる時間を秒で設定します。タイムアウトと見なされたワーカーは"down"とマークされます。

qb.confの設定(Advanced)(8)

Parameter	Value	Control
supervisor_heartbeat_timeout	1080	
supervisor_job_flags		Choices
supervisor_log_flags	admin, callback, job, mail, subjob, user, work	Choices
supervisor_manifest		Browse
supervisor_manifest_flags		Choices
supervisor_max_postboard_retry	8	
supervisor_max_priority	9999	
supervisor_max_worker_requeue	8	
supervisor_max_worker_retry	64	
supervisor_migrate_timeout	1200	
supervisor_policy_binding	Internal	
supervisor_policy_library	none	
supervisor_port	50001	
supervisor_preempt_policy	passive (kill and replace running job after work completes)	
supervisor_queue_binding	Internal	
supervisor_queue_library	cluster	
supervisor_tmppath		Browse
supervisor_verbosity	admin, auth, callback, command, error, file, host, info, job, licen	Choices
supervisor_worker_configfile		Browse

supervisor_job_flags... 様々なジョブフラグを設定します。ファームに投入されたすべてのジョブに適用されますので、ここでは最低限のものを設定し、ジョブ投入時のFlagsフィールドでジョブ単位の項目を設定します。

supervisor_log_flags... supervisorがログ出力するイベントをフラグで設定します。
supervisor_manifest... マニフェストログの置かれるパスを設定します。このログには、ジョブに関するすべてのイベントやアクションが記録されています。Windowsのデフォルトは、C:\ProgramData\pfx\qube\logs\manifestlog です。

supervisor_manifest_flags... supervisorがマニフェストログに出力するイベントをフラグで設定します。job, subjob, workが使用できます。

supervisor_max_postboard_retry... supervisorが”post”クライアントと通信する際にリトライする最大数を設定します。”post”クライアントは、GUIやスクリプトなどのクライアントプログラムに設定されたコールバックからイベントを受け取るものを意味します。

supervisor_max_priority... ジョブの優先度の最大値を設定します。(優先度としては最も低くなります)

supervisor_max_worker_requeue... supervisorがワーカーに可能なCPU数分だけジョブを割り当てるようにリトライする最大値を設定します。

supervisor_max_worker_retry... supervisorがワーカーにジョブを割り当てるのをあきらめるまでにリトライする最大値を設定します。”Down”とマークされているワーカーに対し、定期的にこの設定数までのリトライを行います。

supervisor_migrate_timeout... ジョブが他のホストに移行される前のタイムアウト値を設定します。

supervisor_policy_binding... supervisor_policy_libraryが参照している言語を設定します。Internalもしくは、Perl が設定可能で、デフォルトはInternalです。

supervisor_policy_library... supervisorが使用するキューイングのポリシーを記述したライブラリファイルへのパスを設定します。デフォルトは”none”です。

supervisor_port... supervisorが使用するポート番号を設定します。すべてのワーカーおよびクライアントで統一する必要があります。デフォルトは、50001です。

supervisor_preempt_policy... Supervisorが使用する割り込み方法を設定します。supervisor_smart_share_preempt_policyはSmartShareで拡張されたインスタンスが対象ですが、ここではプライマリのインスタンスが対象となりますので、デフォルトはpassiveとなっています。

passive: 現在のプロセスが終了するまで待つ、割り込みを行う。(デフォルト)

aggressive: 現在のプロセスを直ぐに終了させて割り込みを行う。

qb.confの設定(Advanced)(9)

Qube Configuration

supervisor_preempt_policy: passive (kill and replace running job after work completes)

supervisor_queue_binding: Internal

supervisor_queue_library: cluster

supervisor_tmppath: [Browse]

supervisor_verbosity: admin,auth,callback,command,error,file,host,info,job,licen [Choices]

supervisor_worker_configfile: [Browse]

Client (Submission defaults)

client_account: []

client_cluster: []

Drive	Network Path (UNC)	Username	Password

client_drive_map: []

client_host_domain: []

client_job_flags: auto_mount [Choices]

client_priority: []

client_restrictions: []

Client (Configuration)

client_lnpath: [Browse]

OK Cancel

supervisor_queue_binding... supervisorのキューイングアルゴリズムのバインドタイプを設定します。

Internal: Qube!内蔵のキューイングアルゴリズムを使用します。(デフォルト)

Perl: スクリプト化されたキューイングアルゴリズムを使用します。

supervisor_queue_library... supervisorのキューイングアルゴリズムが定義されているバイナリまたはスクリプトファイルを指定するか、あらかじめ用意されたタイプを指定します。

cluster: Qube!のクラスタリングアルゴリズムに基づきます。(デフォルト)

priority: プライオリティ値に基づきます。

queue: FIFOに基づきます。(投入した順番)

supervisor_tmppath... supervisorが出力するテンポラリファイルのフォルダを設定します。Windowsでは、WindowsデフォルトユーザTempパスになります。

supervisor_verbosity... supervisorログに出力される項目を設定します。

supervisor_worker_configfile... supervisorがワーカーを集中管理するためのワーカーコンフィグファイルを指定します。Windowsでは、C:\ProgramData\%Pfx%\qube\qbwrk.conf がデフォルトです。

== Client(Submission defaults) ==

client_account... 投入ダイアログやコマンドにかかわらず、このマシンから投入されたすべてのジョブにデフォルトの"account"を設定します。

client_cluster... クライアントとして使用時、ジョブ投入ダイアログでクラスタが指定されていない場合、このクラスタ設定がデフォルトとして使用されます。

client_drive_map... (Windows Only)このクライアントから投入されたジョブにドライブマップが含まれていた場合、ワーカー側にauto_mountフラグが設定されていれば、そのドライブをネットワークパス(UNCパス)に変換します。

client_host_domain... クライアントのWindowsのドメイン名を使用せず、クライアントメッセージを送りたい場合に使用します。

client_job_flags... クライアントから投入されたすべてのジョブに適用されるジョブフラグを設定します。デフォルトでは、auto_mountのみが設定されています。

client_priority... ジョブ投入時にプライオリティが設定されていない場合に適用されるデフォルトのプライオリティ値です。

client_restrictions...ジョブ投入時にrestriction(制限)が設定されていない場合、デフォルトとして、このrestriction が適用されます。worker_restrictionsも参照してください。

qb.confの設定(Advanced)(10)

== Client(Configuration) ==

client_logpath・・・ジョブログの出力先がネットワーク上の共有フォルダに設定されていた場合、このパスにより、supervisorを介さずに、Client側から直接アクセスすることができます。

The screenshot shows the 'Qube Configuration' dialog box. The 'Client (Configuration)' section is expanded, showing the 'client_logpath' field with a 'Browse' button next to it. Other fields in this section include 'client_account', 'client_cluster', 'client_drive_map' (a table), 'client_host_domain', 'client_job_flags' (with a 'Choices' button), 'client_priority', and 'client_restrictions'. The 'Client (Submission defaults)' section above it contains 'client_account', 'client_cluster', and a table for 'client_drive_map' with columns 'Drive', 'Network Path (UNC)', 'Username', and 'Password'. The 'Supervisor' section at the top includes 'supervisor_queue_binding', 'supervisor_queue_library' (set to 'cluster'), 'supervisor_tmppath' (with a 'Browse' button), 'supervisor_verbosity' (with a 'Choices' button), and 'supervisor_worker_configfile' (with a 'Browse' button). 'OK' and 'Cancel' buttons are at the bottom.

7.補足資料(1)

・ クラスタ(階層指定)

各ワーカーを、クラスタと呼ばれる階層に分類、ユーザー、グループ、場所、プロジェクトなどを基準に優先度を設定することが可能。特定のクラスタを指定して投入されたジョブは異なるクラスタに投入されたジョブより優先度が高くなります。ファームのキャパシティを最大に利用しつつ、部署やプロジェクトでの割り当てを維持することができます。

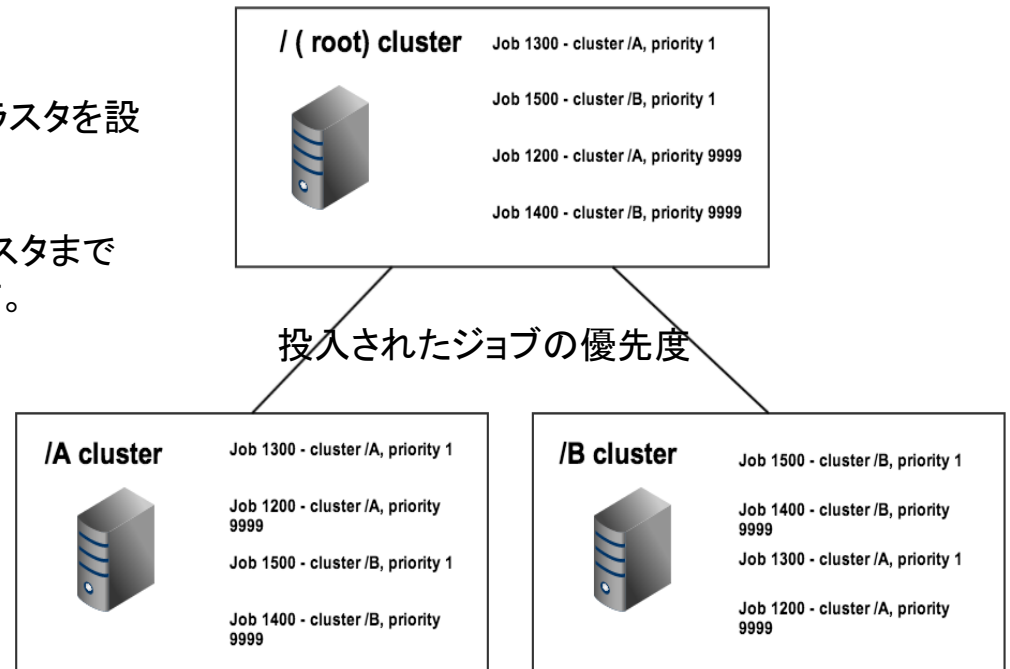
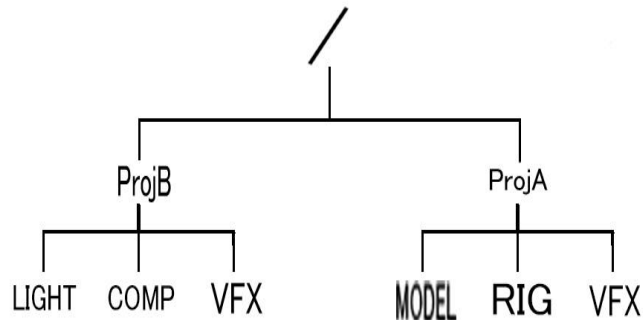
各ワーカーのqb.confのworker_clusterに所属するクラスタを設定します。

例: /ProjA/VFX

worker_restrictionに受付を制限するクラスタを設定します。

例: /ProjA || /ProjA/*

/ProjA または /ProjA の1つ下のクラスタまでが指定されたジョブのみが受付可能です。



7.補足資料(2)

- Auto-Wrangling

障害のあるジョブやノードを、組み込み実装されたロジックにより検出し、それらを自動的にブロックしたり、特定のマシンのみで起こるのであれば他のマシンで再レンダーするなど、状況に応じて、適切なジョブの割り振りを行います。同時にユーザやQube!管理者にメールすることも可能です。

- Smart Share

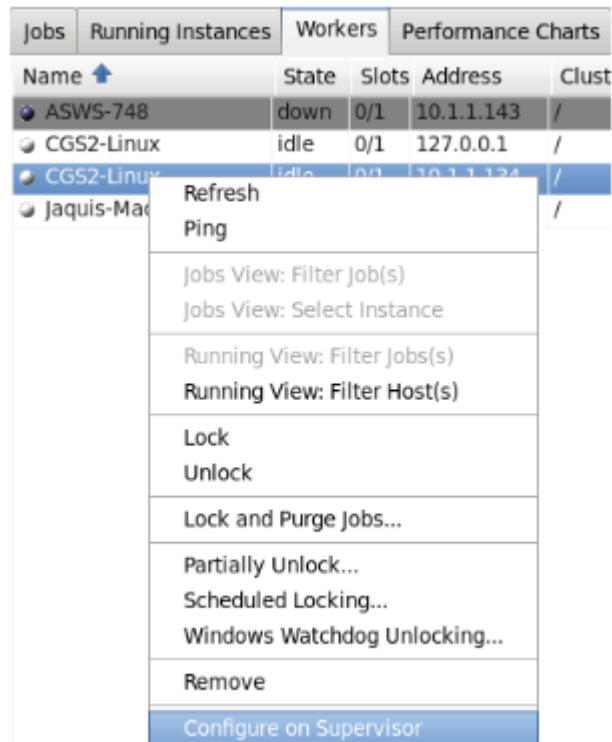
ファームに空きリソース(スロット)がある場合、実行中のジョブのインスタンス数を自動的に増やしてやり、空きリソースを有効利用する機能です。またその際、プライオリティが同じであれば全てのジョブで公平に空きリソースを共有(つまり等分割)します。

例えば、まったくジョブが走っていない100台のマシンが有り、そこにジョブを Instances = 1, Max Instances = 100で投入すると、100台全てを使用してレンダーリング開始するよう拡張されます。拡張されたインスタンスの優先度は低く設定されます。ここに、2つ目のジョブを、やはり同じプライオリティ、同じInstances指定で投入します。すると、最初のジョブは50台、2つ目のジョブも50台でレンダーリングします。3つ目を投入すれば33台ずつでレンダーリングする、という分割になります。

7.補足資料(3)

・ ワーカーの集中管理

スーパーバイザマシン上で、各ワーカーのqb.conf内の項目の集中管理が可能です。各ワーカーのqb.confをいちいち開いて編集する必要がなくなり、便利です。



スーパーバイザマシンにて、Qube!の'admin'権限があるアカウントで、WranglerViewを起動します。Workersタブ内で、ワーカーを選択(複数可)し、右クリックで、「Configure on Supervisor」を選択します。表示されるworker_XXX項目のうち、以下の項目以外が設定可能です。

worker_address
 worker_boot_delay
 worker_journal_location
 worker_lookup
 worker_max_threads
 worker_pidfile
 worker_port

設定したファイルは、以下に置かれます。

Linux & OS X: /etc/qbwrk.conf

Windows: C:\ProgramData\pfx\qube\qbwrk.conf

7.補足資料(4)

Windows Defenderの「リアルタイム保護」の無効化

Supervisorマシン上で、Windows Defenderの「リアルタイム保護」が有効になっていれば、無効に設定することをお勧めします。これは、負荷が重く、Supervisorのレスポンスに大きな影響があるためです。具体的には以下のような症状が発生することがあります。セキュリティ保護のためには、インターネット接続無しでの運用、もしくはサードパーティー製のウイルス対策ソフトをご検討ください。

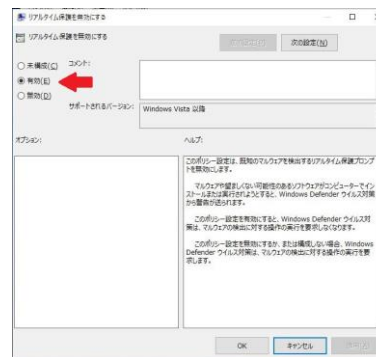
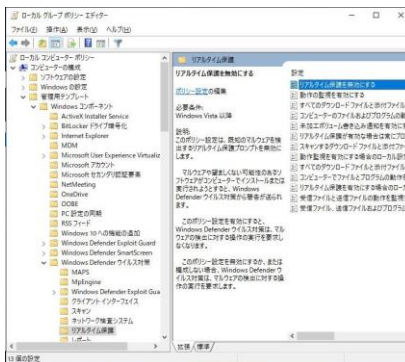
・Supervisor上でWranglerViewを起動すると、以下のようなログが残ることがある

INFO: Local machine is Supervisor = False

- ・Refreshボタンを押すと、リフレッシュに非常に時間がかかる。また、Job欄のリフレッシュがかからず、ホワイトアウトする
- ・Workerが稼働中にもかかわらず、“down”と表示される
- ・Jobを投入すると、投入されるべきWorkerや空きスロットに投入されない
- ・1つのInstance内に、failedになるフレームが発生する(他はComplete)

<無効化の方法>

- ① Windowsキー + Rにて、「ファイル名を指定して実行」より、gpedit.mscコマンドを実行し、ローカルグループポリシーエディタを起動
- ② [コンピュータの構成 > 管理用テンプレート > Windowsコンポーネント > Windows Defenderウイルス対策 > リアルタイム保護] にアクセス
- ③「リアルタイム保護を無効にする」にて、[有効] に設定、「適用」ボタンをクリック



※[有効]に設定すると無効化されます。



Qube!簡易インストールガイド(Ver.7.0-x) 第7.02(改定)版

2020年 9月

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