



JProfiler

スタート ガイド



エクセルソフト株式会社

2017年5月9日

はじめに

JProfiler は、アプリケーションのボトルネックやメモリ リークを解消し、パフォーマンスの最適化を実現するオールインワン Java 解析ツールです。

通常の解析機能に加え、解析データのリアルタイム表示、CPU 解析、スナップショット データの保存や比較、5 つの異なるビューでオブジェクトのセットを分析できるヒープ ウォーカーなどの機能を提供します。

JProfiler の直感的な UI 操作による解析が可能で、解析結果におけるビジュアル要素が豊富であるため、高度な解析技術や知識を持たないユーザーにも扱いやすい製品です。

多くのプラットフォームや IDE、アプリケーション サーバーをサポートしているので、多様な環境でのパフォーマンス向上を図ることができます。

本スタート ガイドは、JProfiler を初めて利用される方を対象に作成しております。

基本的な操作方法、各解析ツールの使用方法について、サンプルを元に説明しています。

なお、サンプルを用いた解析結果について、必ずしも同じ結果になるとは限りませんので、予めご了承ください。

JProfiler の評価版は、製品版と同等の機能が提供されており、アクティベーション（キーの登録作業後）から 10 日間ご使用いただけます。

すべての ej-technologies の製品名は、ej-technologies の商標または登録商標です。

その他記載の商標、サービス マーク、商品名は各社の商標または登録商標です。

内容




インストール手順.....	5
クイックスタート ダイアログ.....	9
➤ デモ セッションおよび既存のセッションの解析.....	9
➤ 実行中の JVM へのアタッチ.....	9
➤ アプリケーション サーバーのローカルまたはリモートでの解析.....	9
➤ スナップショットを開く.....	9
JProfiler スタート センター.....	11
➤ セッションを開く.....	11
➤ クイック アタッチ.....	11
➤ 新しいセッション.....	12
➤ スナップショットを開く.....	13
デモ セッションの実行.....	14
➤ 各種デモ セッション.....	15
解析における各種機能.....	17
➤ メモリ解析.....	17
➤ ヒープ ウォーカー.....	17
➤ CPU 解析.....	20
➤ VM テレメトリ情報.....	23
➤ JEE およびプローブ.....	24
問題の解決.....	25
➤ パフォーマンスの問題.....	25
➤ 過度のメモリ消費.....	25
➤ メモリ リーク.....	25
➤ 検出が困難なバグ.....	28
➤ 品質保証.....	28
アプリケーション サーバーの統合.....	29
➤ アプリケーション サーバーの統合手順.....	29

IDE 統合.....	31
➤ IDE の統合手順.....	31
JProfiler のライセンスについて	33

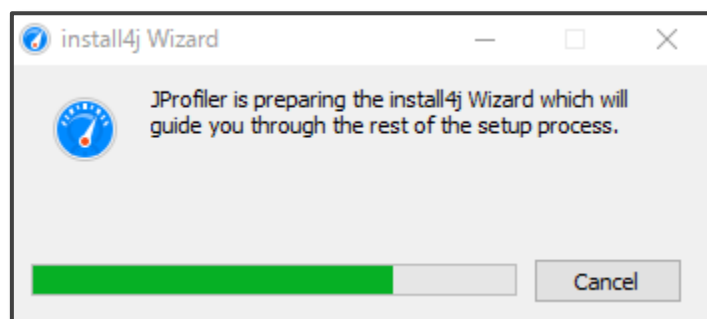
インストール手順

ダウンロード ページ (<https://www.xlsoft.com/jp/products/ejtechnologies/download.html>) から無償評価版を申し込むと、実際のダウンロード情報がお申し込み画面で入力された [Email] に自動的に送信されます。ダウンロード情報に従って、以下の画面が表示されたら、使用するマシンに適したリンクから JProfiler のダウンロードを開始します。

JProfiler

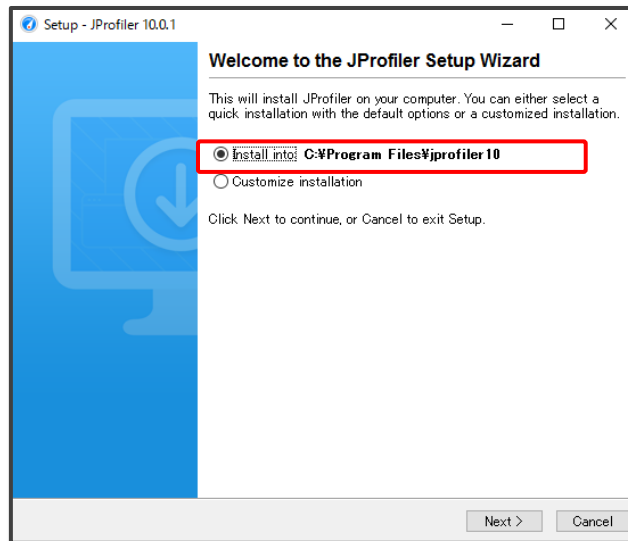
プラットフォーム	バージョン	ダウンロード
 64-BIT WINDOWS ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable with JRE (83 MB) ↓ ZIP Archive (114 MB)
 32-BIT WINDOWS ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable with JRE (81 MB) ↓ ZIP Archive (113 MB)
 MACOS ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ DMG Archive with JRE (134 MB)
 LINUX ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable (48 MB) ↓ RPM (84 MB) ↓ TAR.GZ Archive (84 MB)
 SOLARIS ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable (57 MB) ↓ TAR.GZ Archive (98 MB)
 AIX ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable (57 MB) ↓ TAR.GZ Archive (98 MB)
 FREEBSD ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable (57 MB) ↓ TAR.GZ Archive (98 MB)
 HP-UX ※評価版の期限は 10 日間です。	バージョン : 10.0 リリース日 : 2017/3/7	↓ Setup Executable (57 MB) ↓ TAR.GZ Archive (98 MB)

ダウンロード用のリンクをクリック後、**install4j** を使用したインストールの準備が開始されます (マルチプラットフォームの Java インストーラー ビルダー、**install4j** は、製品としても購入可能です)。



準備が終わると、**Welcome to the JProfiler Setup Wizard** ダイアログが表示されます。

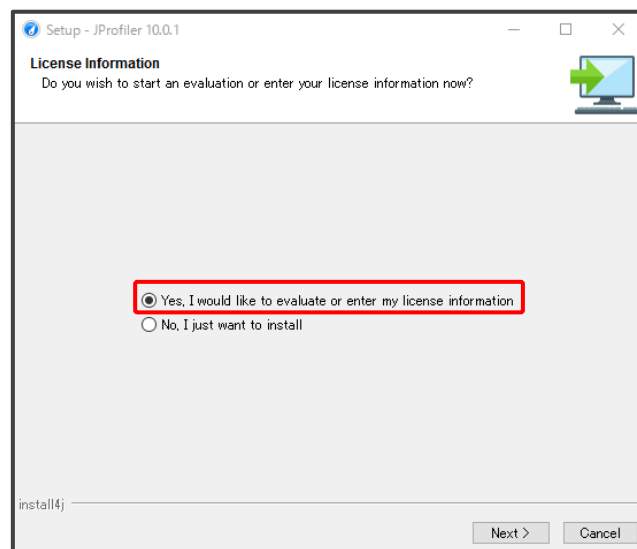
最初に、デフォルト設定によるクイック インストール (上部) またはカスタム インストール (下部) のどちらかを選択します。



インストール先を選択後、ライセンス契約に関する記載のある **[License Agreement]** ダイアログが表示されるので、**[I accept the agreement]** を選択します。

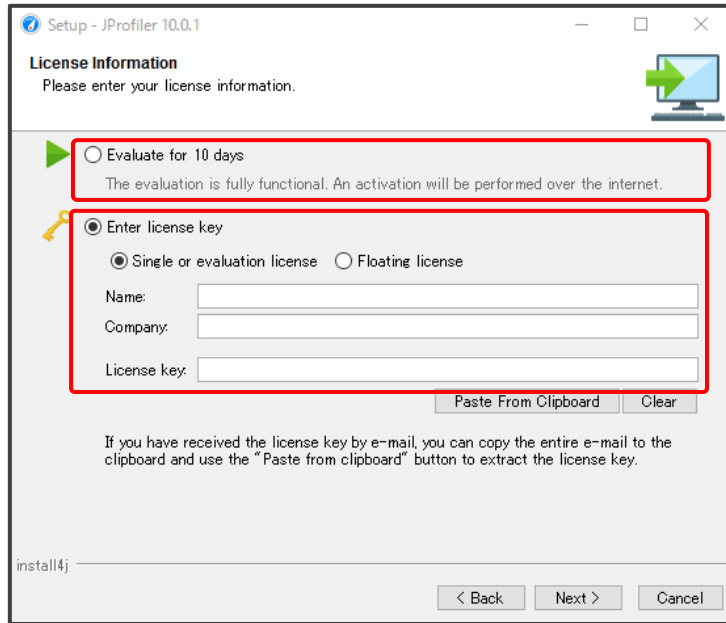
ライセンス契約に合意をすると、すぐにインストールが開始します。

インストール過程で、ライセンス情報を確認する **License Information** ダイアログが表示されます。JProfiler の無償評価版を使用する場合、または既にライセンスを購入済みの場合は、上部の **[Yes, I would like to evaluate or enter my license information]** にチェックを入れます。下部の **[No, I just want to install]** にチェックを入れて、ソフトウェアのインストールのみを行うこともできます。

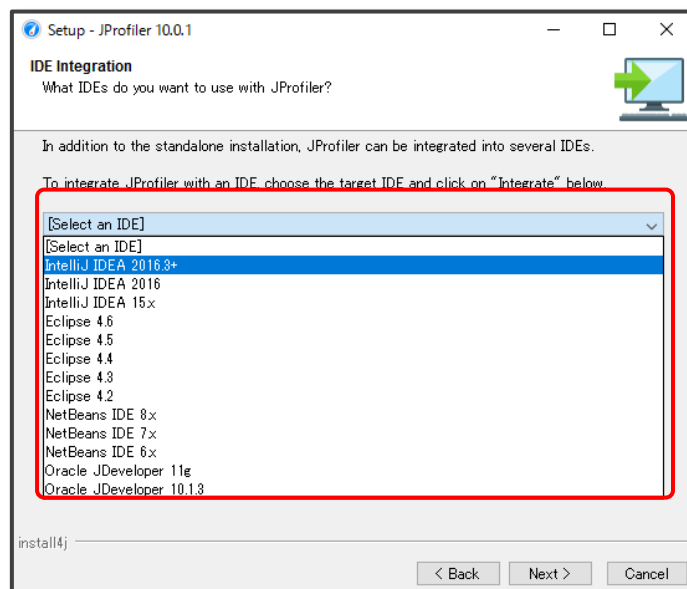


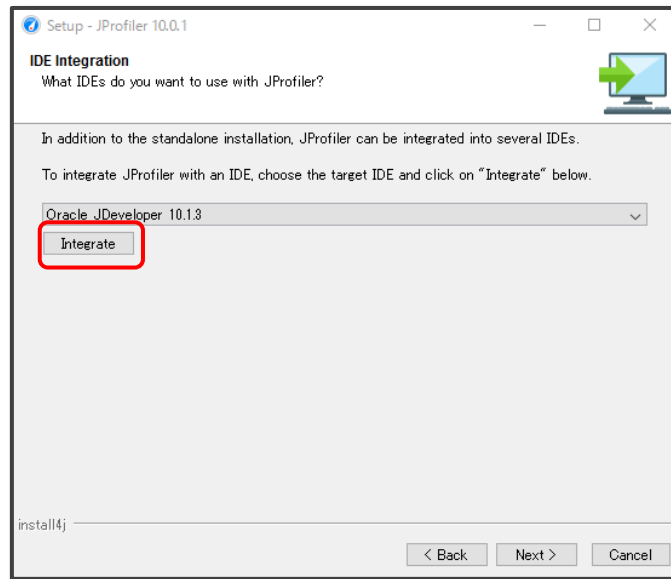
上記の手順で、無償評価版の使用開始またはライセンス キーの入力へ進むと、以下の画面が表示されます。無償評価版を使用希望の場合は、上部の **[Evaluate for 10 days]** にチェックを入れます。既にライセンスを購入済みの場合は、下部の **[Enter license key]** にチェックを入れて、ライセンス情報を入力します。

まず、購入済みのライセンスの種類を選択します。シングル ライセンスの場合は左の **[Single or evaluation license]**、フローティング ライセンスの場合は右の **[Floating license]** にチェックを入れます。ライセンスの種類を選択後、**[Name]** に名前、**[Company]** に会社名、そして **[License key]** にお持ちのライセンス キーを入力して、**[Next]** ボタンをクリックします。



次のダイアログでは、任意で JProfiler と統合する IDE を 1 つ選択することができます。ドロップダウン メニューから希望の IDE を選択して、**[Integrate]** をクリックします。

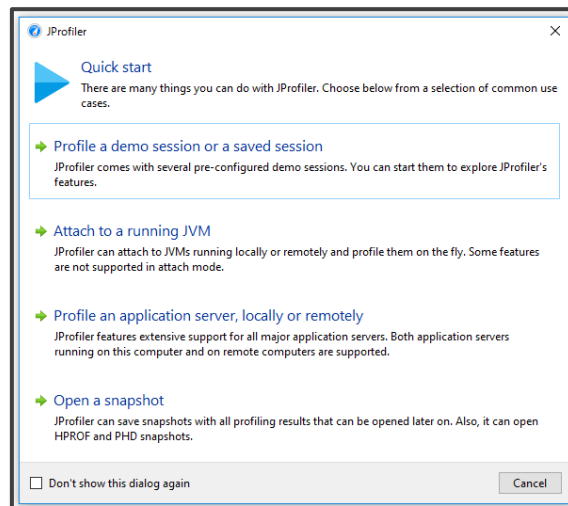




最後に、**Completing the JProfiler Setup Wizard** ダイアログが表示されたら、**[Finish]** ボタンをクリックするとインストールが完了します。

クイックスタート ダイアログ

JProfiler を起動すると、自動的に **Quick start** ダイアログが表示されます。



ダイアログには、簡単にアプリケーションの解析を始めるための 4 項目があります。

■ デモ セッションおよび既存のセッションの解析

詳細は [\[P.14\]](#) 「デモ セッションの実行」を参照してください。

■ 実行中の JVM へのアタッチ

JProfiler は、ローカルまたはリモートで実行中の JVM にアタッチして、その場で解析をすることができます。詳細は [\[P.11\]](#) JProfiler スタート センターの「クイック アタッチ」を参照してください。

■ アプリケーション サーバーのローカルまたはリモートでの解析

JProfiler は、ローカルおよびリモートで実行されるすべての主要なアプリケーション サーバーに対する幅広いサポートを提供します。詳細は [\[P.12\]](#) JProfiler スタート センターの「新しいセッション」を参照してください。

■ スナップショットを開く

JProfiler は、すべての解析結果のスナップショットを保存することができます。さらに、JProfiler は HPROF および PHD のスナップショットを開くこともできます。詳細は [\[P.13\]](#) JProfiler スタート センターの「スナップショットを開く」を参照してください。

※ダイアログ下部に表示される **[Don't show this dialog again]** の項目にチェックを入れることによって、次回からダイアログを非表示にすることができます。

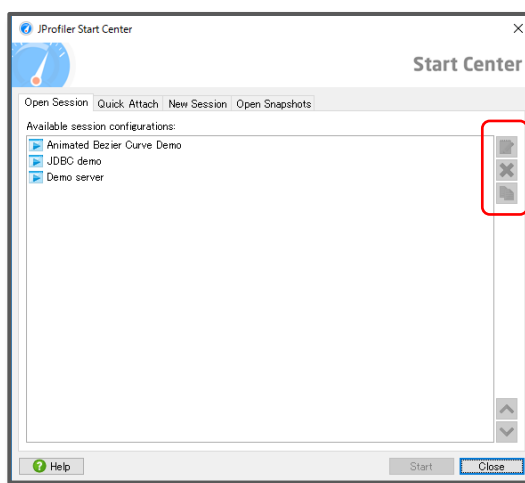
※**Quick start** ダイアログを呼び戻す場合、**[Shift]** キーおよび **[F1]** キーを同時に押してください。または、左上の **[Help]** のドロップダウン リストから、**Show Quickstart Dialog** を選択します。

JProfiler スタート センター

Quick start ダイアログでいずれかの項目を選択後、**Start Center** ダイアログが表示されます。

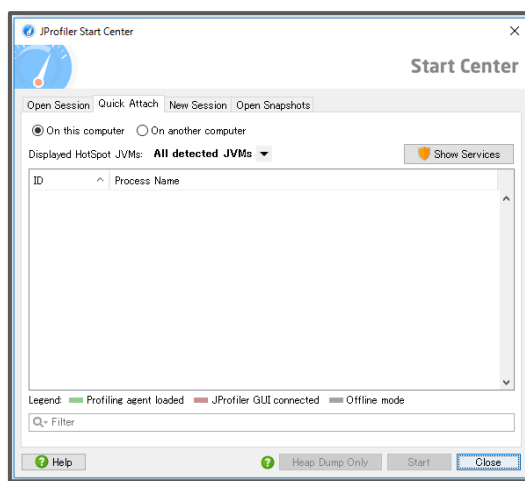
■ セッションを開く

[Open Session] タブは、設定済みのデモ セッションまたは自身が設定したすべてのセッションを含みます。任意のセッションをダブルクリックして、セッションを開始します。ダイアログの右側に表示されたアイコンを選択することで、セッションの削除またはコピーなどの編集をすることができます。デモ セッションの解析方法は、[\[P.14\]](#) を参照してください。



■ クイック アタッチ

[Quick Attach] タブは、すでに実行中の JVM を解析する際に使用します。セッションの終了後、保存することができます。



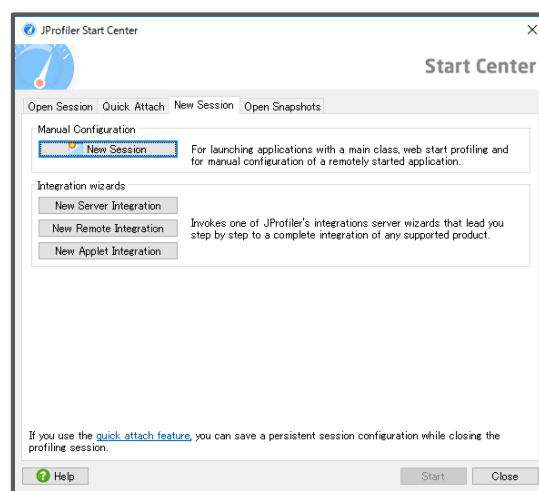
[on another computer] オプションは、リモート JVM の解析を準備するための [jpenable] と呼ばれるコマンドライン ツールの使用に役立ちます。

■ 新しいセッション

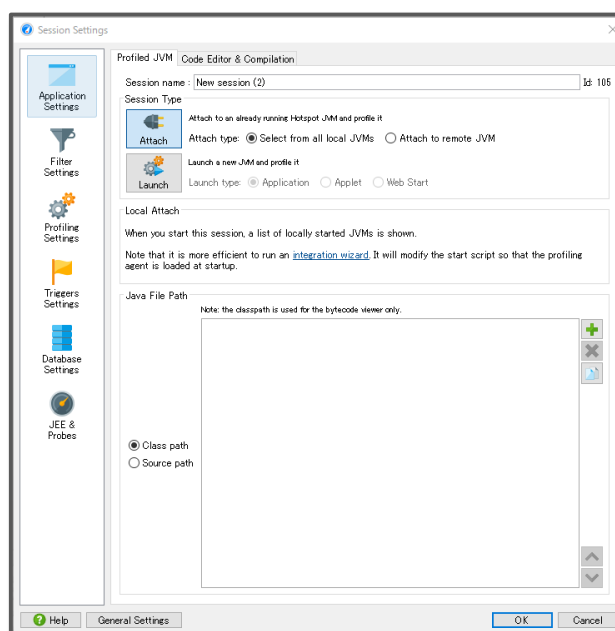
[New Session] タブでは、いくつかの方法で新しいセッションを作成することができます。

1. マニュアル設定

Manual Configuration 内の [New Session] ボタンを選択することで、新規のセッションを手動で開始できます。



Session Settings ダイアログが表示されるので、デフォルト設定に従う場合、[OK] をクリックします。セッションが開始します。

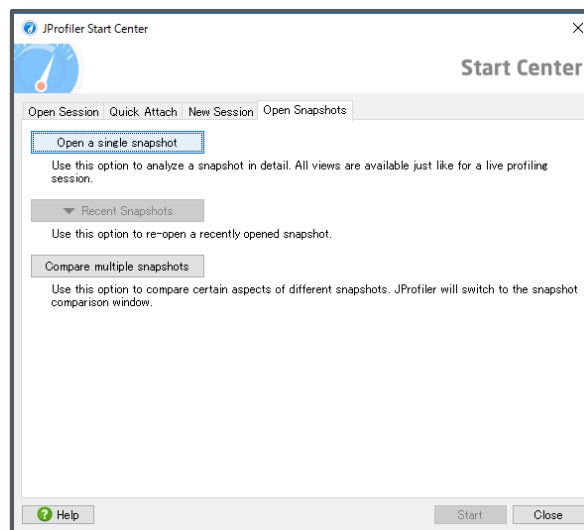



2. 統合ウィザードを介した設定

Integration wizards には、統合ウィザードを起動する **[New Server Integration]**、便利なショートカット機能の **[New Remote Integration]** および **[New Applet Integration]** を含む 3 つのボタンがあります。設定を終えると、セッションをすぐに開始するか、選択した新規のセッションが "open session" タブに表示されます。

■ スナップショットを開く

[Open Snapshots] タブでは、事前に保存されたスナップショットを開くことができます。**[Open a single snapshot]** ボタンをクリックして **.jps** ファイルを選択します。または、**[Compare multiple snapshots]** を選択して、スナップショットの比較を作成します。



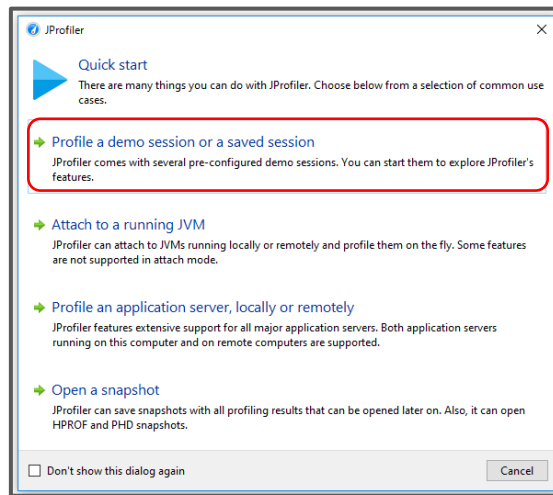
※セッションを開かずに **[Cancel]** ボタンをクリックして **Start center** を閉じると、すべての解析ビューの動作が停止します。**Start center** を再度開くには、メニューバーにある **[Session]** のドロップダウンリストから **Start center** を選択します。または、左上にある  を選択します。

※新しいセッションを別のウィンドウで開くには、メニューバーにある **[Session]** のドロップダウンリストから **[New Window]** を選択します。

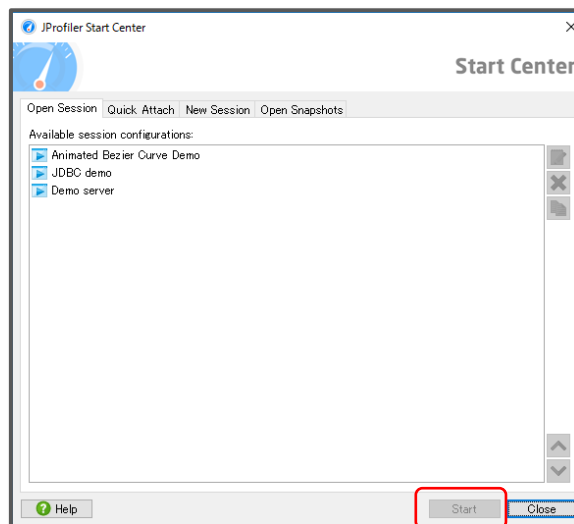
デモ セッションの実行

デモ セッションを実行することにより、JProfiler の主要な機能を理解することができます。

1. **Quick start** ダイアログの一番上に表示される **[Profile a demo session or a saved session]** を選択して、デモまたは既存のセッションの解析を開始します。

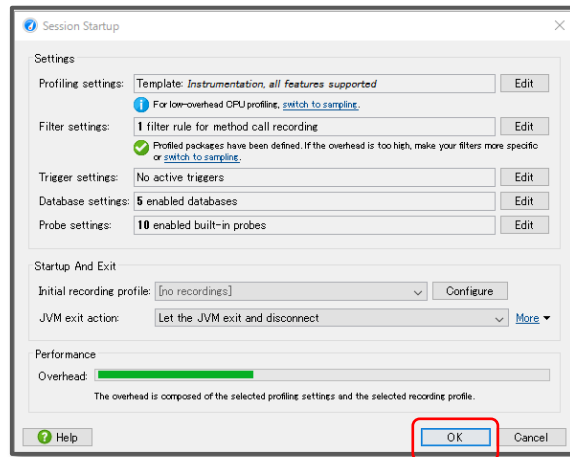


2. **Start Center** ダイアログが自動的に表示されます。3 つの異なるデモ セッションから 1 つを選択して、**[Start]** ボタンをクリックします。



3. デモ セッション選択後、**Session Startup** ダイアログが表示されます。デフォルト設定に従う場合、**[OK]** ボタンをクリックして、デモ セッションを開始します。

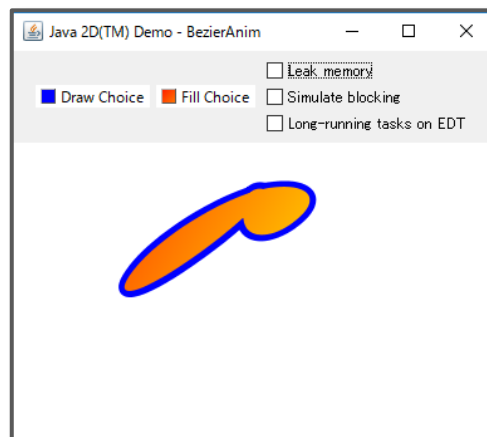
※デモ セッションの Java のソース コードは、"**{JProfiler installdirectory}/demo/**"> で見つけることができます。



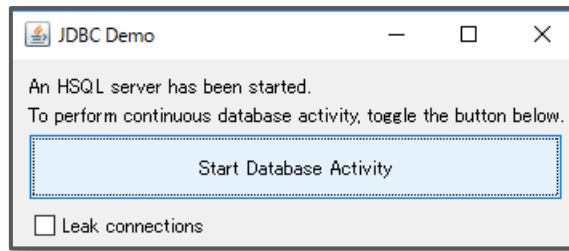
■ 各種デモ セッション

JProfiler には、3 つのデモ セッションが含まれています。

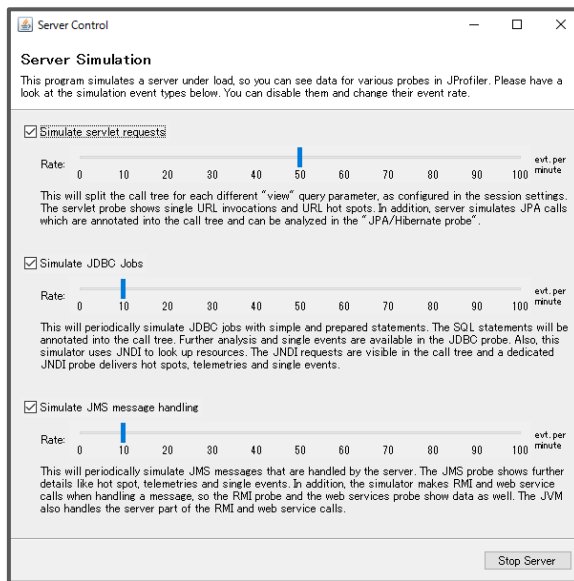
1. **Animated Bezier Curve Demo** では、メモリ リークの発生 (Leak Memory)、スレッドのブロッキング (Simulate Blocking)、および EDT 上で長時間実行するタスク (Long-running tasks on EDT) を再現します。



2. **JDBC demo** では、HSQL サーバーおよび HSQL データベースを使用します。[**Start Database Activity**] を選択して、HSQL データベースの活動を開始します。HSQL データベースの活動を停止したい場合は、[**Stop Database Activity**] を選択します。また、[**Leak connections**] を選択して、コネクション リークを再現することができます。



3. **Demo server** では、3 つのシミュレーション タイプ (Simulate servlet requests、Simulate JDBC Jobs、または Simulate JMS message handling) から、データのさまざまなプローブを確認することができます。



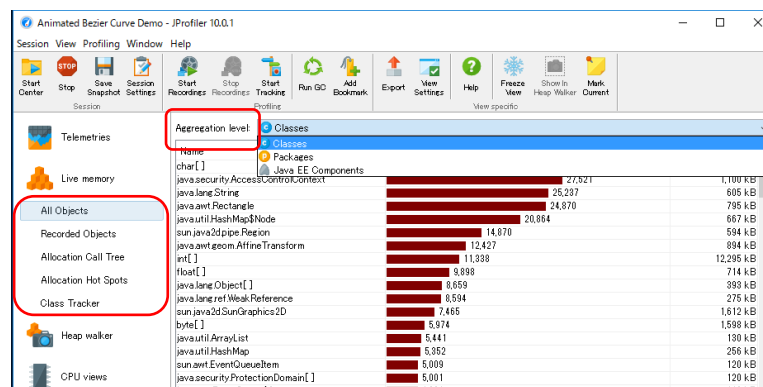
解析における各種機能

JProfiler の機能は、サイド バーにあるビュー セクションに見やすく表示されています。

■ Live memory (メモリ解析)

メモリ解析では、オブジェクトのモニタリング後、問題のスポットを検出します。メモリ解析を行う際、JProfiler のサイド バーから **[Live memory]** を選択します。**[Live memory]** 以下に、**All Objects**、**Recorded Objects**、**Allocation Call Tree**、および **Allocation Hot Spots**、**Class Tracker** を含む 5 つのビューが表示されます。


All Objects ビューは、ヒープ使用量を、インスタンス数と共に表示します。また、クラス、パッケージ、および Java EE コンポーネント別での表示も可能です。表示を変更する場合は、**[Aggregation level]** のドロップダウン メニューから、表示したいレベルを選択します。

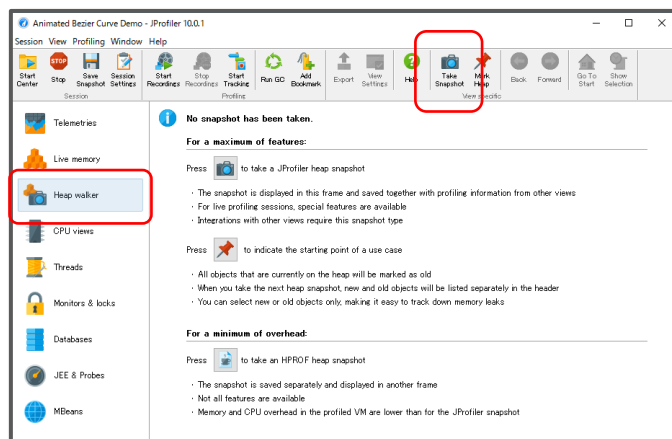



※特定の時間における、オブジェクト アロケーションを表示するには、**Recorded Objects** ビューを使用します。

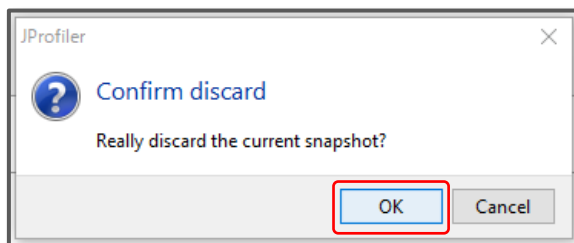
■ Heap walker (ヒープ ウォーカー)

ダイナミックなビューのメモリ解析とは異なり、ヒープ ウォーカーは、ヒープのスナップショットによって JProfiler 特有のドリルダウン機能を使用して、メモリ リークを検出します。わずかなステップで各インスタンスを確認し、オブジェクトを選択して分析することができます。

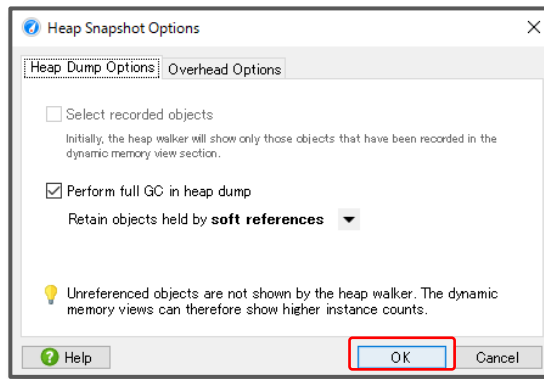
1. メモリ リークを検出するには、JProfiler のサイド バーから **[Heap walker]** を選択します。この時点では、スナップショットが撮られていません。ヒープのスナップショットを撮るには、 をクリックします。



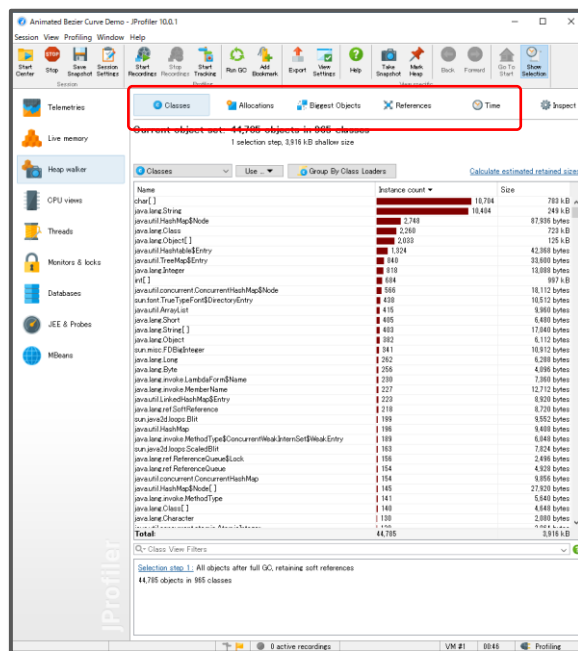
- ※既にスナップショットが存在する、またはスナップショットを撮り直す場合には、既存のスナップショットを削除する必要があります。再度スナップショットを撮るには、トップメニュー バーの  を選択し、**[OK]** ボタンをクリックします。



2. **Heap Snapshot Options** ダイアログが表示されます。デフォルト設定に従う場合は **[OK]** ボタンをクリックします。

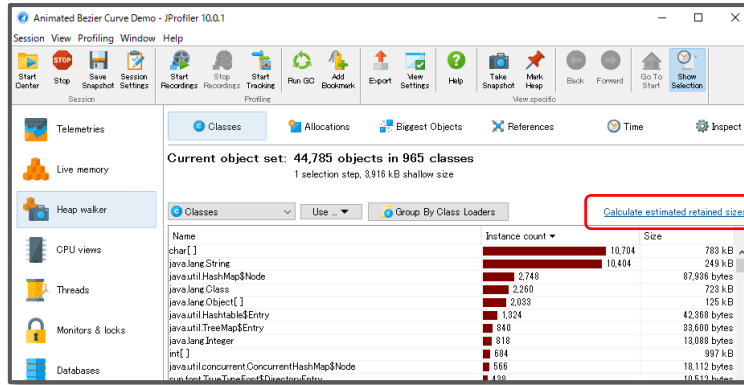


3. スナップショットを撮ると、すべてのクラスが自動的に表示されます。スナップショットを撮ったオブジェクトは、**Current object set** として表示されます。**Current objects set** の情報は、**Classes**、**Allocations**、**Biggest Objects**、**References**、および **Time** の 5 つの異なるビューから確認することができます。

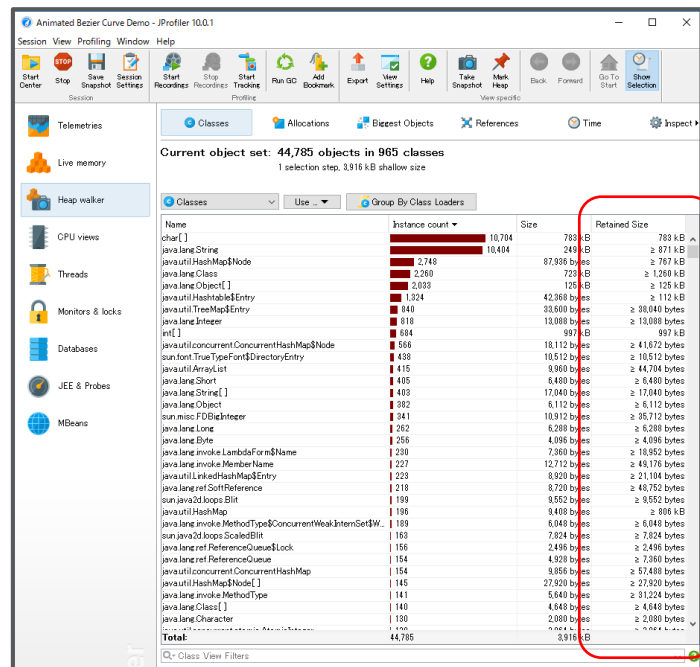


※この画面へ戻るには、JProfiler 左上にある **[View]** のドロップダウン リストから、**[Heap Walker Start View]** を選択します。

4. これらのオブジェクトをドリルダウンしていきます。各オブジェクトのインスタンス数は、**Current object set** 内に表示されています。すべてのオブジェクトのサイズを計算したい場合は、画面右上の **Calculate estimated retained sizes** を選択します。




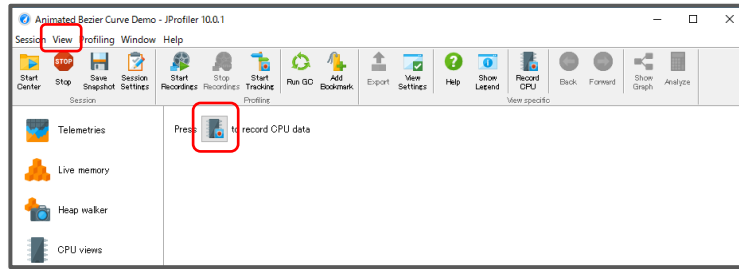
5. 計算終了後、**Calculate estimated retained sizes** は非表示となり、代わりに結果が表示されます。右端に **Retained Size** が追加され、ガベージコレクションによって解放されるメモリを表示します。




■ CPU views (CPU 解析)

CPU 解析は、CPU 処理時間がどこに費やされているのか検査し、パフォーマンスのボトルネックを特定します。JProfiler のサイドバーから **[CPU views]** を選択します。

1. CPU 解析を開始するには、 を選択して CPU データの記録を開始します。または、JProfiler のメニューバーにある **[View]** のドロップダウンリストから、**[Record CPU Data]** を選択します。



2. CPU データの記録を中止するには、ツール バーにある  を選択します。または、メニューバーにある **[view]** のドロップダウン リストから、**[Stop Recording CPU Data]** を選択します。

※CPU の記録過程は、サイド バーの **[CPU views]** から確認することができます。**[CPU views]** 以下には、**Call Tree**、**Hot Spots**、**Call Graph**、**Method Statistics**、**Complexity Analysis**、**Call Tracer**、および **JavaScript XHR** の 7 つの項目があります。

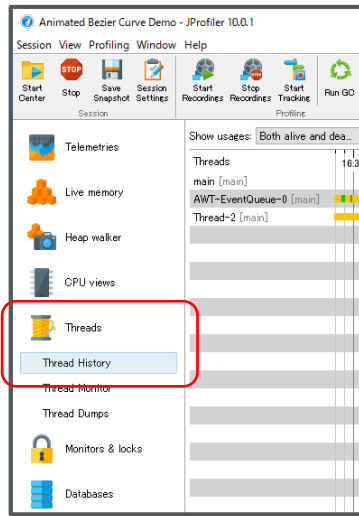
※CPU データの記録を新たに再開した場合、すべてのデータがリセットされます。

■ Threads (スレッド解析)

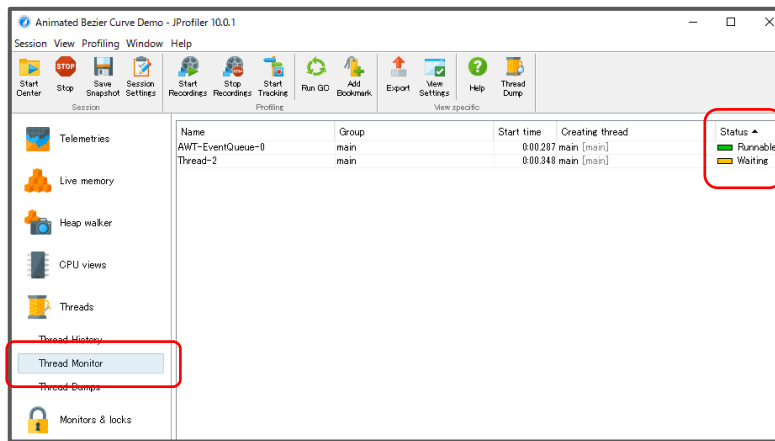
スレッド解析は、スレッドの動作を確認して、デッドロックを解決することができます。また、アプリケーションのモニター使用量に関する詳しい情報を提供します。

スレッド解析を開始するには、サイド バーの **[Threads]** を選択します。**[Threads]** には、**Thread History**、**Thread Monitor**、および **Thread Dumps** を含む 3 つの項目があります。

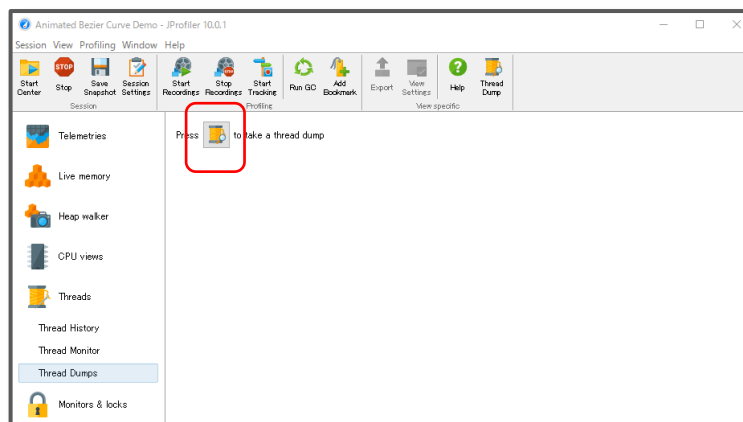
1. **Thread History** ビュー では、JVM 上にあるすべてのスレッドの状況に関する詳しい履歴情報を表示します。





2. **Thread Monitor** ビューでは、現在進行中のスレッドに関する動的な情報を表示します。






3. **Threads Dumps** ビューでは、アクティブなスレッドのスタック トレースを使用して、手動で入手したスレッド ダンプを表示します。

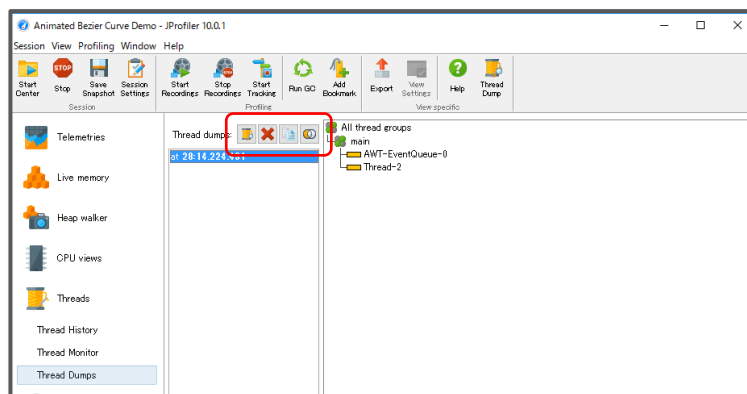


 をクリックして、スレッド ダンプを作成します。

新たにスレッド ダンプを作成するには、 を選択します。

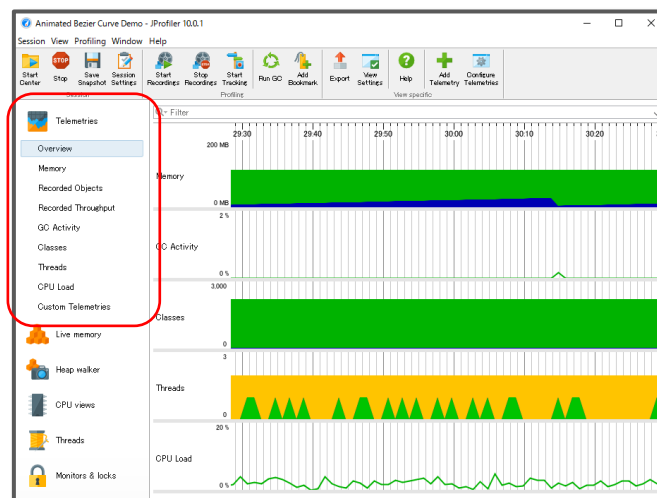
スレッド ダンプを削除するには、 を選択します。

スレッド ダンプをクリップボードにコピーするには、 を選択します。
複数のスレッド ダンプを作成して、それぞれで比較をするには、 を選択します。



■ Telemetries (VM テレメトリ情報)

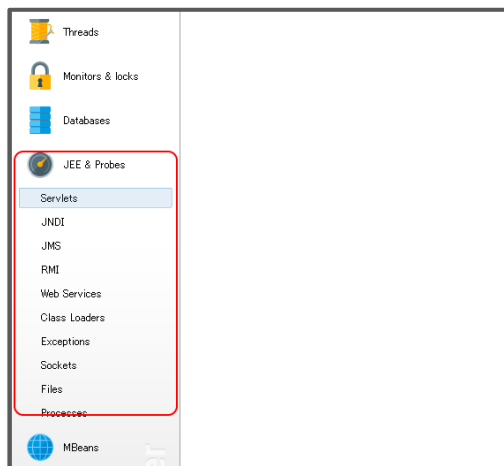
VM (仮想マシン) テレメトリ情報は、解析した JVM についての情報グラフによって確認することができます。VM テレメトリ情報を確認するには、サイド バーから **[Telemetries]** を選択します。



[Telemetries] には、最大限のヒープ サイズと使用領域、空き領域を表示する **Memory**、ヒープ上のオブジェクトの合計数を表示する **Recorded Objects**、ガベージ コレクトされたオブジェクト数を表示する **Recorded Throughput**、ガベージ コレクターの活動をパーセンテージで表示する **GC Activity**、JVM によって読み込まれたクラス数を表示する **Classes**、JVM 内に存在するスレッドの合計数を表示する **Threads**、および解析過程における CPU の読み込みをパーセンテージで表示する **CPU Load** を含む 7 つのビューがあります。

■ JEE & Probes (JEE およびプローブ)

ここでは、JVM の高水準のサブシステムである、JDBC コールやファイル I/O、またはカスタム プローブを使った独自のサブシステムを測定します。 を選択して、プローブの記録を開始します。




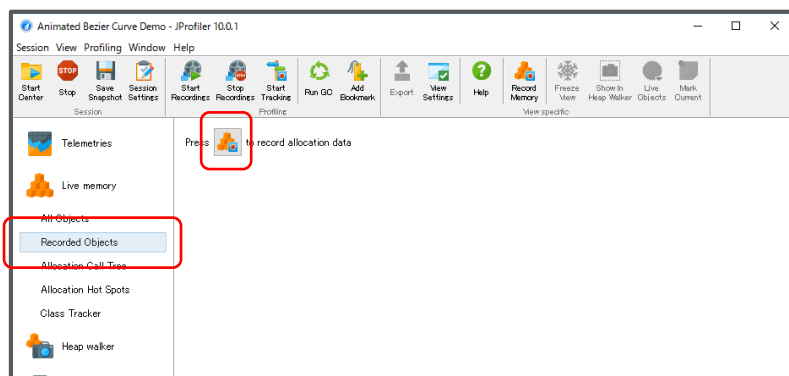
JEE & Probes 以下には、次のビューがあります。

- ◆ **Servlets** 検出されたリクエスト URL ごとにコールを分割します。
- ◆ **JNDI** コール ツリーに JNDI クエリ文字列をアノテートします。
- ◆ **JMS** コール ツリーに JMS メッセージの詳細をアノテートします。
- ◆ **RMI** コール ツリーに RMI サーバー コールをアノテートします。
- ◆ **Web Services** JAX-WS-R1、Axis 2、および Apache CXF からの Web サービス クライアント コールをホット スポット ビューで表示します。
- ◆ **Class Loaders** VM 内のすべてのクラス ローダーを表示します。
- ◆ **Exceptions** 例外クラス名をペイロードとして表示します。
- ◆ **Sockets** コール ツリーに関連のある `java.net.SocketAddress` オブジェクトの `toString()` 値をアノテートします。
- ◆ **Files** コール ツリーにファイル名をアノテートします。
- ◆ **Processes** コール ツリーに実行ファイルへの完全パスをアノテートします。

問題の解決

■ パフォーマンスの問題

問題の起こっているパフォーマンスを検出するには、JProfiler の CPU セクションを開きます。パフォーマンスに関する問題の多くは、一時的なオブジェクトが過度に生成されているために発生します。このような場合、サイド バーの **[Live memory]** 以下にある **Recorded Objects** ビューから、アロケーションを減らすべき箇所を検出することができます。 を選択して、アロケーション データを記録します。



また、I/O や他のサブシステムをプローブによって測定する場合、サイド バーにある **[JEE & Probes]** を選択します。このセクションでは、高水準の情報がどこの処理で多くの時間を費やしているかを表示します。

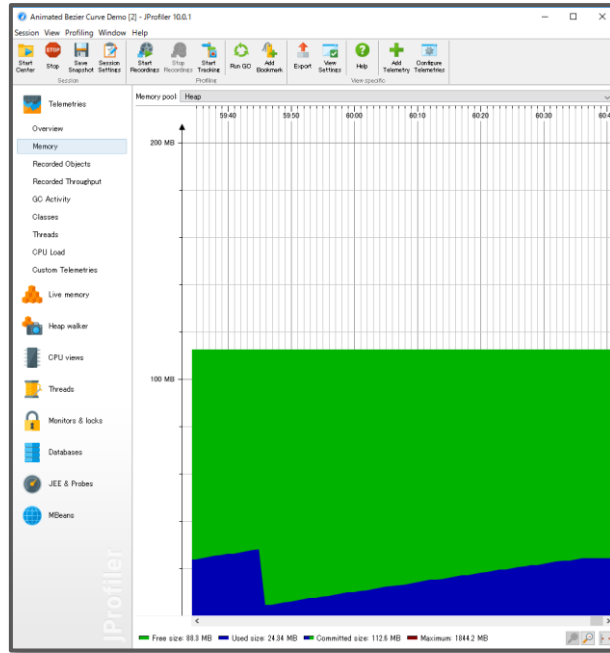
■ 過度のメモリ消費

アプリケーションが過度のメモリを使用している場合、サイド バーの **[Live memory]** ビュー セクションからメモリが消費されている箇所を確認することができます。また、**[Heap walker]** の **References** ビューにより、ヒープに残っている不必要なオブジェクトを検出することができます。

■ メモリ リーク

アプリケーションのメモリが時間の経過とともに増加している場合、メモリ リークの可能性が考えられます。

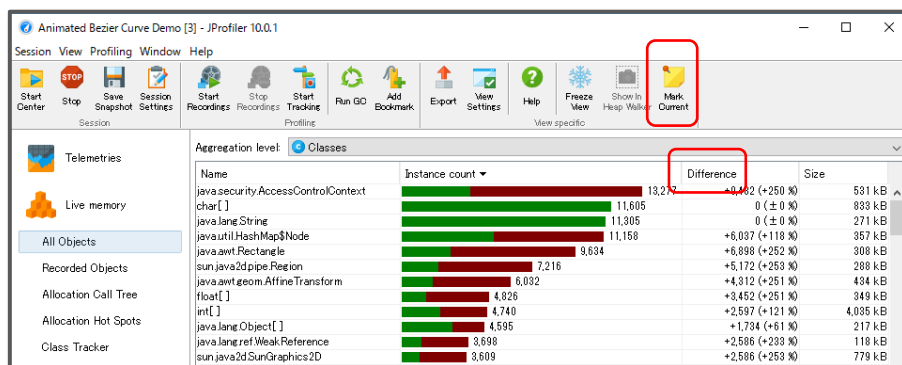
1. サイド バーの **[Telemetries]** 以下から、**Memory** ビューを使用して、メモリ リークを確認することができます。



2. メモリ リークは、特にアプリケーション サーバーにとっては致命的な問題となりかねません。そのため、**Memory** ビューでメモリ リークを確認した後、サイド バーの **[Live memory]** または **[Heap walker]** 機能を使用して、原因の追究を行います。

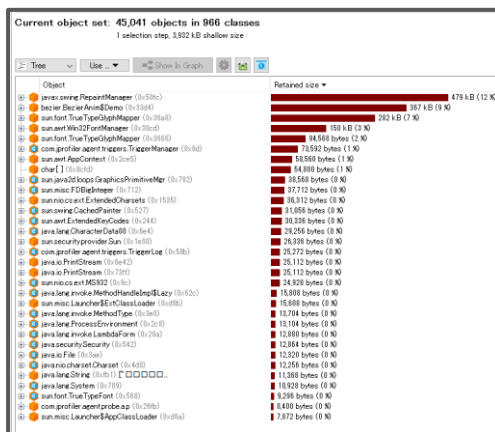
[Live memory] 機能を使用する場合には、サイド バーの **[Live memory]** 以下にある **All Objects** ビュー、または **Recorded Objects** ビューを選択します。

次に、画面右上の **[Mark Current]** を選択して、現在のメモリ量を記録します。記録したメモリ量は、緑色の棒グラフとして表示されます。記録時のメモリとの差異は、**Difference** 内で表示されます。記録を削除するには、いずれかのクラスを右クリックして、**[Remove Mark]** を選択します。

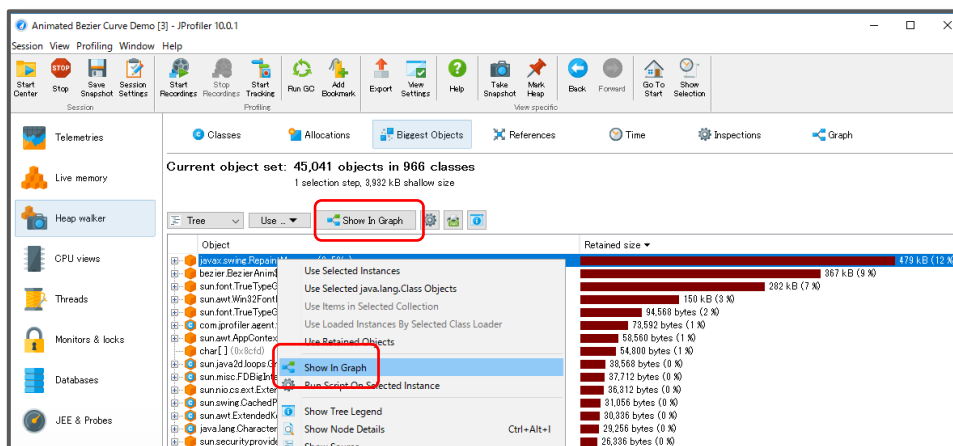


※これにより、特定の時間内のオブジェクト アロケーションの変化を確認することができます。

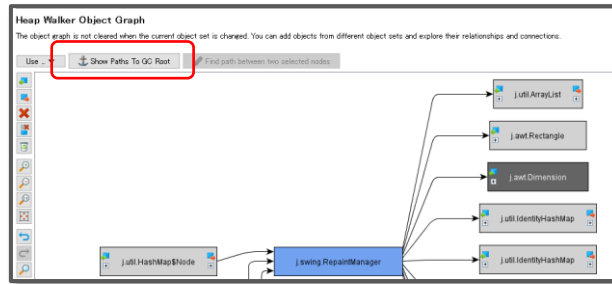
[Heap walker] 機能を使用する場合、サイドバーの [Heap walker] を選択して、スナップショットを撮ります。スナップショットを撮ったヒープを、さまざまな方法で分析することができます。**Biggest Objects** ビューは、メモリリークの原因を検出します。**Retained size** は、オブジェクトがヒープから削除された場合に解放されるメモリのサイズのことを指します。**Biggest Objects** ビューは、確保されたままで、メモリ量が最大であるオブジェクトから順に列挙します。このオブジェクトツリーによって、エラーのあるリファレンスヘッドリルダウンすることができます。



[Heap walker] の **Biggest Objects** ビューから特定のオブジェクトを分析する場合、オブジェクトを選択して右クリックし、[Show In Graph] を選択します。または、オブジェクトを選択し、[Show In Graph] ボタンをクリックします。



その後表示されるグラフ内で、特定のオブジェクトを選択することによって、[Show Paths To GC Root] ボタンが有効になります。このボタンをクリックすると、ガベージコレクターまでのルートを確認できます。



このようにして、メモリ リークの原因を検出することができます。

■ 検出の困難なバグ

見落とされがちでありながらたいへん有用な JProfiler の使用方法に、デバッグが挙げられます。多くのバグは、手作業や従来のデバッガーによって検出するのが非常に困難です。JProfiler の **Threads** ビューを使用することによって、競合状態やデッドロックを追跡するのが難しいマルチスレッドにおいてもバグを検出することができます。使用方法は、本ガイド 21 ページをご参照下さい。

■ 品質保証

アプリケーション開発プロセスにおいて定期的に解析を実行することは、問題が起こり得る箇所を推測するのに最適です。たとえテストケースでアプリケーションが問題ないと示された場合でも、パフォーマンスやメモリのボトルネックに対する理解を深めることで、プロジェクトが進行するにつれ、アプリケーションのデザインに柔軟性を持つことができます。これにより、実際のニーズに合わなかった場合でも、リエンジニアリングにかかるコストを軽減することができます。JProfiler の **[Telemtries]** セクションで表示される情報を利用することにより、アプリケーションの開発ステータスを確認することができます。また、解析のスナップショットの保存機能により、プロジェクトの進捗状況を記録できます。オフライン解析機能では、アプリケーション上で自動的に解析を行うことができます。

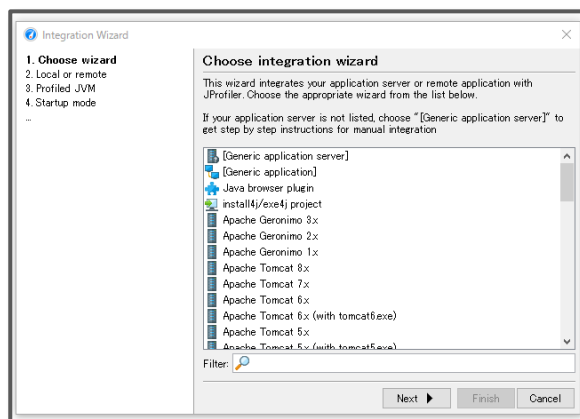
アプリケーション サーバーの統合

■ アプリケーション サーバーの統合手順

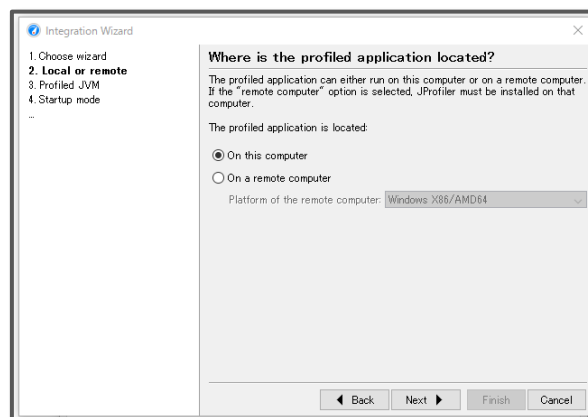
JProfiler のアプリケーション サーバーの統合ウィザードは、アプリケーション サーバーの解析を容易にします。アプリケーション統合は、以下のいずれかの方法で行います。

- I. **Start Center** の **[New session]** タブから、**New Server Integration** を選択します。
- II. メニューバーにある **[Session]** のドロップダウン リストから、**[Integration wizards]** 以下にある **New Server Integration** を選択します。

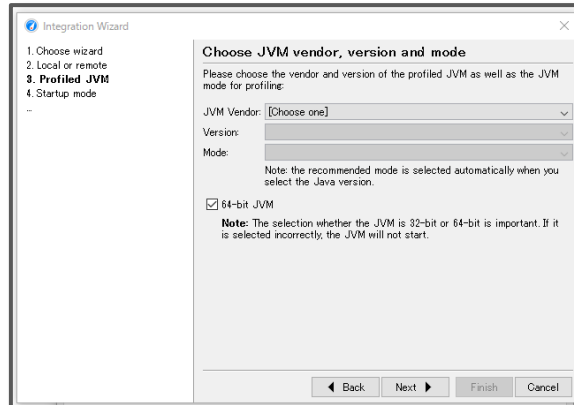
1. **Integration Wizards** ダイアログが表示されます。最初に、JProfiler と統合するアプリケーションサーバーを選択後、**[Next]** ボタンをクリックします。



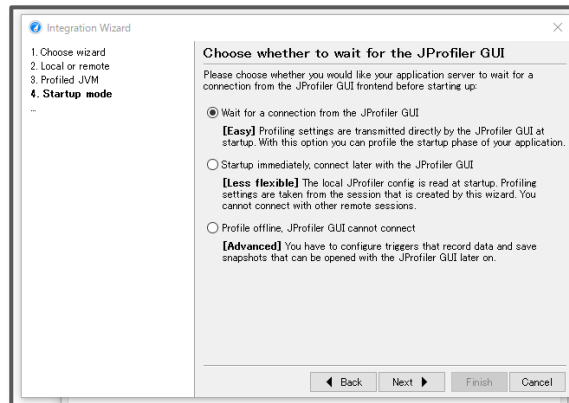
2. 2 つ目のステップでは、解析するアプリケーションまたはアプリケーション サーバーが、ローカルのマシン上 (“On this computer”) で実行中なのか、もしくはリモート マシン上 (“On a remote computer”) で実行中なのかを選択後、**[Next]** ボタンをクリックして次に進みます。



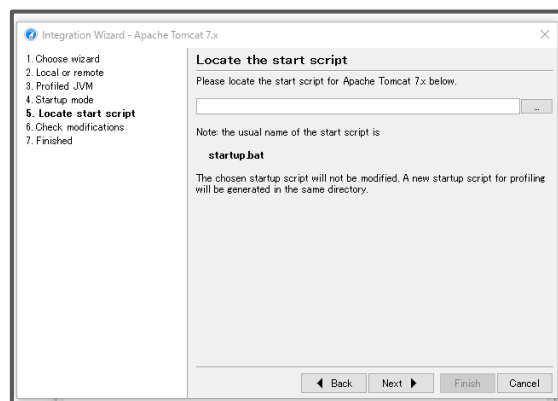
3. 解析する JVM のベンダーとバージョン、および JVM のモードを選択後、[Next] ボタンをクリックして次に進みます。



4. **Wait for a connection**、**Startup Immediately**、または **Profile offline** の 3 つの選択肢から任意のスタートアップ モードを選択し、[Next] ボタンをクリックします。



5. 新たに、**Locate the start script** ダイアログが表示されます。選択したアプリケーション サーバー（画像は Apache Tomcat 7x）のスタート スクリプトの場所を選択します。

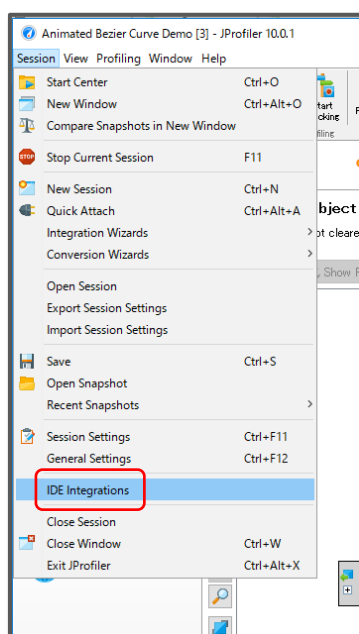


この後のステップは、選択したスタートアップ モードにより異なります。

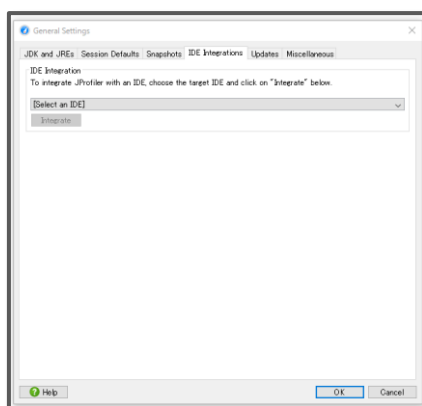
IDE 統合

■ IDE の統合手順

JProfiler は、多くの主要 IDE とシームレスな統合ができます。IDE 統合を行うには、メニューバーにある **[Session]** のドロップダウン リストから、**[IDE Integrations]** を選択します。



General Settings ダイアログが表示されたら、統合したい IDE を選択して、**[Integrate]** ボタンをクリックします。



IDE 統合設定が完了すると、クラスパス、メインクラス、作業ディレクトリ、使用する JVM または他のオプションを再び特定することなく、統合した IDE から JProfiler を立ち上げることができます。また、ソースコードナビゲーションは、IDE 上で実行されます。

JProfiler のライセンスについて

JProfiler は、有効なライセンスなしでは使用できません。

<https://www.xlsoft.com/jp/products/ejtechnologies/index.html> へアクセスして、10 日間有効な無償評価版のダウンロード、またはライセンスの購入をご検討ください。無償評価版のダウンロード後に、何らかの理由で JProfiler を評価できない場合には、xlsoftkk@xlsoft.com までお問い合わせください。以前のメジャーバージョン用に発行されたライセンスは、新しいバージョンでは使用できませんのでご了承ください。