



# Benutzerhandbuch

## Oracle WebLogic FastPack

## Inhalt:

1	Einführung .....	5
2	Metriken .....	6
2.1	JVM Metriken .....	6
2.1.1	Current CPU load .....	6
2.1.2	Used Memory .....	6
2.1.3	Max Memory .....	6
2.1.4	Loaded Classes .....	6
2.1.5	Thread Count.....	6
2.1.6	Memory Pool Sun Old Gen used .....	6
2.1.7	Specific GC Collection Time Sun PS Scavenge (minor) .....	6
2.1.8	Specific GC Collection Time Sun PS Mark Sweep (major) .....	6
2.1.9	Specific GC Collection Time Sun New Parallel Copy (minor) .....	6
2.1.10	Specific GC Collection Time Sun Concurrent Mark Sweep (major) .....	6
2.1.11	Specific GC Collection Time Sun Mark Sweep Compact (major).....	7
2.1.12	Specific GC Collection Time Sun Copy (minor) .....	7
2.2	Prozess & System Metriken .....	7
2.2.1	Processor Time.....	7
2.2.2	System Time.....	7
2.2.3	User Time .....	7
2.2.4	Process Count .....	7
2.2.5	Run Queue, current .....	7
2.2.6	Blocked.....	7
2.2.7	Memory Available .....	7
2.2.8	Free Swap .....	7
2.2.9	Free Memory .....	7
2.2.10	Used Memory.....	8
2.2.11	Temp Usage .....	8
2.2.12	Var Usage .....	8
2.2.13	Disk Reads/sec .....	8
2.2.14	Disk Writes/sec .....	8
2.2.15	Pages Input.....	8
2.2.16	Pages Ouput.....	8
2.2.17	Used Diskspace .....	8
2.2.18	Total Diskspace .....	8
2.2.19	Received Bytes/sec .....	8
2.2.20	Transmitted Bytes/sec .....	8
2.2.21	Received Packets Dropped.....	8
2.2.22	Transmitted Packets Dropped.....	9
2.2.23	Received Packets Error .....	9
2.2.24	Transmitted Packets Error .....	9
2.3	JMX Metriken .....	10
2.3.1	Open Sessions Count.....	10
2.3.2	Session High Count.....	10
2.3.3	Session Servlet Pool Size.....	10
2.3.4	EJB Cache Miss Count.....	10
2.3.5	EJB Cache Access Count.....	10
2.3.6	EJB Pool Timeout Total Count.....	11
2.3.7	EJB Transactions Committed Total Count.....	11
2.3.8	EJB Transactions Timed Out Total Count .....	11

2.3.9	EJB Transactions Rolled Back Total Count .....	11
2.3.10	JDBC Connection Pool Connection Delay Time .....	11
2.3.11	JDBC Connection Pool Active Connections High Count .....	11
2.3.12	JDBC Connection Pool Active Connections Current Count .....	11
2.3.13	Execute Queue Pending Request Current Count .....	12
2.3.14	Execute Queue Execute Thread CurrentIdle Count .....	12
2.3.15	Execute Queue Execute Thread TotalCount .....	12
2.3.16	Thread Pool Execute Thread Total Count .....	12
2.3.17	Thread Pool Throughput .....	12
2.3.18	Thread Pool Hogging Thread Count .....	12
2.3.19	Thread Pool Pending User Request Count .....	12
2.3.20	JTA Active Transactions Total Count .....	13
2.3.21	JTA Transaction Committed Total Count .....	13
2.3.22	JTA Transaction Abandoned Total Count .....	13
2.4	JMS Metriken .....	14
2.4.1	JMS Runtime Current Connections .....	14
2.4.2	JMS Runtime Connections High Count .....	14
2.4.3	JMS Producer Pending Messages .....	14
2.4.4	JMS Producer Pending Bytes .....	14
2.4.5	JMS Destination Pending Messages .....	14
2.4.6	JMS Destination Pending Bytes .....	14
2.4.7	JMS Consumer Pending Messages .....	15
2.4.8	JMS Consumer Pending Bytes .....	15
2.4.9	JMS Messages Sent Count .....	15
2.4.10	JMS Messages Received Count .....	15
2.5	Applikations Metriken .....	16
2.5.1	API Times .....	16
2.5.2	PurePath Duration (Count) .....	16
2.5.3	DB Count .....	16
2.5.4	PurePath Duration .....	16
2.5.5	PurePath Duration w/o Suspension .....	16
2.5.6	PurePath Response Time .....	16
2.5.7	DB Time .....	16
2.6	Plugin Metriken .....	17
2.6.1	System Monitor Plugin .....	17
2.6.2	Active Monitoring Plugin .....	17
3	Dashboards .....	18
3.1	Application Overview .....	19
3.2	Operating System .....	21
3.3	WebLogic Health .....	23
3.4	WebLogic JMS Overview .....	25
3.5	JVM .....	27
4	Inbetriebnahme .....	29
4.1	Download .....	29
4.2	Installation .....	29
4.3	Konfiguration .....	33
4.3.1	System Profil .....	33
4.3.2	Dashboards .....	38
5	Erweiterung des FastPacks: Zusätzliche Oracle WebLogic JMX Werte .....	40
5.1	Standard Messwerte .....	40
5.2	Benutzerdefinierte Messpunkte .....	41

5.2.1	Oracle WebLogic Server Administration Console: Metrik identifizieren ..	41
5.2.2	Metrik in dynaTrace erfassen.....	45
5.2.3	Visualisierung von Oracle WebLogic Metriken.....	51

# 1 Einführung

Das dynaTrace Oracle WebLogic FastPack ist in Zusammenarbeit mit dem BIT (Bundesamt für Informatik und Telekommunikation Schweiz) entstanden. Das FastPack stellt dabei die erste Phase des Lifecycle Integrationsprojektes dar. Diese erste Phase beinhaltet im Wesentlichen ein Basismonitoring der Oracle WebLogic Applicationserver Infrastruktur.

## 2 Metriken

In diesem Kapitel werden sämtliche für die Grundmodule (dynaTrace Oracle WebLogic FastPack) verwendeten Metriken näher beschrieben. Die hier genannten Metriken werden direkt in den in Kapitel 3 beschriebenen Dashboards angewendet.

### 2.1 JVM Metriken

Der dynaTrace Agent kann interne Metriken, welche den Status der JVM darstellen, liefern. Diese Metriken helfen, das Verhalten der JVM besser einzusehen.

#### 2.1.1 *Current CPU load*

Auslastung der CPU durch den JVM Prozess.  
Einheit: %

#### 2.1.2 *Used Memory*

Aktuelle Speicherbelegung der Java Virtual Machine.  
Einheit: kB

#### 2.1.3 *Max Memory*

Maximaler Arbeitsspeicher, welcher von der Java Virtual Machine verwendet werden kann.  
Einheit: kB

#### 2.1.4 *Loaded Classes*

Anzahl gegenwärtiger geladener Klassen innerhalb der Java Virtual Machine.  
Einheit: Anzahl

#### 2.1.5 *Thread Count*

Aktuelle Anzahl an Threads (Daemon und Non-Daemon) innerhalb der Java Virtual Machine.  
Einheit: Anzahl

#### 2.1.6 *Memory Pool Sun Old Gen used*

Aktuelle Speicherauslastung des Old Gen Spaces. Überleben junge Objekte die Garbage Collection im Nursery, werden diese in den Old Space verschoben. Somit sind ältere resp. Langlebige Objekte in diesem Bereich organisiert.  
Anmerkung: Diese Metrik ist erst ab JVM Version 1.5 oder höher verfügbar.  
Einheit: kB

#### 2.1.7 *Specific GC Collection Time Sun PS Scavenge (minor)*

Collection time dieses spezifischen Garbage Collectors innerhalb des letzten sample Intervalls (10 Sekunden).  
Einheit: ms

#### 2.1.8 *Specific GC Collection Time Sun PS Mark Sweep (major)*

Collection time dieses spezifischen Garbage Collectors innerhalb des letzten sample Intervalls (10 Sekunden).  
Einheit: ms

#### 2.1.9 *Specific GC Collection Time Sun New Parallel Copy (minor)*

Collection time dieses spezifischen Garbage Collectors innerhalb des letzten sample Intervalls (10 Sekunden).  
Einheit: ms

#### 2.1.10 *Specific GC Collection Time Sun Concurrent Mark Sweep (major)*

Collection time dieses spezifischen Garbage Collectors innerhalb des letzten sample Intervalls (10 Sekunden).  
Einheit: ms

### **2.1.11      *Specific GC Collection Time Sun Mark Sweep Compact (major)***

Collection time dieses spezifischen Garbage Collectors innerhalb des letzten sample Intervalls (10 Sekunden).

Einheit: ms

### **2.1.12      *Specific GC Collection Time Sun Copy (minor)***

Collection time dieses spezifischen Garbage Collectors innerhalb des letzten sample Intervalls (10 Sekunden).

Einheit: ms

## **2.2            Prozess & System Metriken**

Folgende Metriken werden vom dynaTrace Unix System-Monitor geliefert und beinhalten im Wesentlichen relevante Kennzahlen auf Betriebssystem-Ebene.

### **2.2.1            *Processor Time***

Aktuelle Prozessorauslastung (Summe aus [System Time](#) und [User Time](#))

Einheit: %

### **2.2.2            *System Time***

Prozessorauslastung verursacht durch Betriebssystem Komponenten (Kernel).

Einheit: %

### **2.2.3            *User Time***

Prozessorauslastung verursacht durch Benutzer Komponenten (Non-Kernel, Benutzer Prozesse, etc).

Einheit: %

### **2.2.4            *Process Count***

Anzahl aktiver Prozesse welche über den „ps“ Befehl sichtbar sind.

Einheit: Anzahl

### **2.2.5            *Run Queue, current***

Anzahl Prozesse, welche auf Run-Time warten.

Einheit: Anzahl

### **2.2.6            *Blocked***

Anzahl Prozesse, welche sich im non-interruptible sleep Zustand befinden.

Einheit: Anzahl

### **2.2.7            *Memory Available***

Prozentsatz des Gesamtspeichers, welcher direkt oder indirekt (durch Aufräumen von Caches und Buffers) verfügbar ist.

Einheit: %

### **2.2.8            *Free Swap***

Aktuell nicht belegter Swap Space (Auslagerungsdatei / Partition).

Einheit: MB

### **2.2.9            *Free Memory***

Aktuell nicht belegter und somit verfügbarer Arbeitsspeicher.

Einheit: kB

### **2.2.10      *Used Memory***

Aktuell belegter Arbeitsspeicher.  
Einheit: kB

### **2.2.11      *Temp Usage***

Prozentsatz des belegten Disk Space für das unter /tmp gemountete Filesystem.  
Einheit: %

### **2.2.12      *Var Usage***

Prozentsatz des belegten Disk Space für das unter /var gemountete Filesystem.  
Einheit: %

### **2.2.13      *Disk Reads/sec***

Leseoperationen der Harddisk.  
Einheit: Bytes pro Sekunde

### **2.2.14      *Disk Writes/sec***

Schreiboperationen der Harddisk.  
Einheit: Bytes pro Sekunde

### **2.2.15      *Pages Input***

Pages Input entspricht der Rate, zu welcher Pages von der Harddisk gelesen werden und daher zu Hard Page Faults führen. Abhängig von den Eigenschaften der Harddisk können hohe Werte darauf hinweisen, dass das System mit zu wenig physischem Memory ausgestattet ist, was zu intensivem Paging führen und somit die Gesamtperformance des System beeinträchtigen kann.  
Einheit: kB

### **2.2.16      *Pages Output***

Pages Output entspricht der Rate, zu welcher Pages auf die Disk geschrieben werden. Diese Operationen generieren keine Hard Page Faults, allerdings kann eine hohe Aktivität indirekt darauf hinweisen, dass zu wenig physisches Memory vorhanden ist und das Betriebssystem somit diese Daten auslagern muss. Ist genügend physisches Memory vorhanden, wird der Virtual Memory Manager weniger dazu gezwungen die Daten auszulagern und die Metrik „Pages Output“ ist entsprechend klein.  
Einheit: kB

### **2.2.17      *Used Diskspace***

Gesamt verwendeter Disk Space.  
Einheit: kB

### **2.2.18      *Total Diskspace***

Gesamt Disk Kapazität.  
Einheit: kB

### **2.2.19      *Received Bytes/sec***

Anzahl vom Netzwerkadapter empfangener Bytes.  
Einheit: Bytes

### **2.2.20      *Transmitted Bytes/sec***

Anzahl vom Netzwerkadapter gesendeter Bytes.  
Einheit: Bytes

### **2.2.21      *Received Packets Dropped***

Anzahl verworfener Netzwerkpakete.  
Einheit: Anzahl.



### **2.2.22      *Transmitted Packets Dropped***

Anzahl verworfener Netzwerkpakete.  
Einheit: Anzahl.

### **2.2.23      *Received Packets Error***

Anzahl fehlerhaft erhaltener Netzwerkpakete.  
Einheit: Anzahl.

### **2.2.24      *Transmitted Packets Error***

Anzahl fehlerhaft gesendeter Pakete.  
Einheit: Anzahl.

## 2.3 JMX Metriken

Java Management Extensions (JMX) ist eine vom Java Community Process (JSR-3) entwickelte Spezifikation zur Verwaltung und Überwachung von Java-Anwendungen. Die einzelnen Komponenten werden mit MBeans verwaltet, überwacht und über die JMX Schnittstelle abgefragt.

Oracle WebLogic bietet eine Vielzahl an MBeans, über welche diverse Performance Metriken verfügbar sind. Die vollständige Liste der verfügbaren MBeans für den Oracle WebLogic Application Server ist unter folgendem Link verfügbar:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/core/index.html](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/core/index.html)

dynaTrace stellt standardmässig vorkonfigurierte WebLogic MBean als Measures zur Verfügung. Weitere JMX Werte werden durch das Plugin „WebLogic Extended“, welches Bestandteil des WebLogic FastPacks ist, hinzugefügt.

Die in diesem FastPack verwendeten JMX Metriken können jederzeit durch beliebige Oracle WebLogic JMX Measures erweitert werden (siehe Kapitel 5 „Erweiterung des FastPacks“).

Die Folgenden MBeans werden vom Oracle WebLogic FastPack verwendet:

### 2.3.1 *Open Sessions Count*

Liefert die Anzahl gegenwärtig offener Session

Einheit: Anzahl

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/WebAppComponentRuntimeMBean.html?skipReload=true#OpenSessionsCurrentCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/WebAppComponentRuntimeMBean.html?skipReload=true#OpenSessionsCurrentCount)

### 2.3.2 *Session High Count*

Liefert die „Hochwassermarkte“ (Maximal Wert) der Sessions seit der Aktivierung des Servers.

Einheit: Anzahl

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/WebAppComponentRuntimeMBean.html?skipReload=true#OpenSessionsHighCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/WebAppComponentRuntimeMBean.html?skipReload=true#OpenSessionsHighCount)

### 2.3.3 *Session Servlet Pool Size*

Liefert die Single Threaded Servlet Pool Size, so wie diese im weblogic.xml konfiguriert wurde.

Einheit: Anzahl

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/WebAppComponentRuntimeMBean.html?skipReload=true#SingleThreadedServletPoolSize](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/WebAppComponentRuntimeMBean.html?skipReload=true#SingleThreadedServletPoolSize)

### 2.3.4 *EJB Cache Miss Count*

Repräsentiert die Gesamtanzahl an fehlgeschlagenen Zugriffsversuchen auf ein Bean im Cache.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/EJBCacheRuntimeMBean.html?skipReload=true#CacheMissCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/EJBCacheRuntimeMBean.html?skipReload=true#CacheMissCount)

### 2.3.5 *EJB Cache Access Count*

Repräsentiert die Gesamtanzahl an Zugriffsversuchen auf ein Bean im Cache.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/EJBCacheRuntimeMBean.html?skipReload=true#CacheAccessCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/EJBCacheRuntimeMBean.html?skipReload=true#CacheAccessCount)

### **2.3.6 EJB Pool Timeout Total Count**

Liefert die Gesamtanzahl an Threads, welche in einen Timeout gelaufen sind, während diese auf eine verfügbare Bean Instanz aus dem Pool gewartet haben.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/EJBPoolRuntimeMBean.html?skipReload=true#TimeoutTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/EJBPoolRuntimeMBean.html?skipReload=true#TimeoutTotalCount)

### **2.3.7 EJB Transactions Committed Total Count**

Anzahl bestätigter EJB Transaktionen.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/EJBTransactionRuntimeMBean.html?skipReload=true#TransactionsCommittedTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/EJBTransactionRuntimeMBean.html?skipReload=true#TransactionsCommittedTotalCount)

### **2.3.8 EJB Transactions Timed Out Total Count**

Anzahl EJB Transaktionen welche durch eine zeitliche Überschreitung zurückgesetzt wurden.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/EJBTransactionRuntimeMBean.html?skipReload=true#TransactionsTimedOutTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/EJBTransactionRuntimeMBean.html?skipReload=true#TransactionsTimedOutTotalCount)

### **2.3.9 EJB Transactions Rolled Back Total Count**

Anzahl EJB Transaktionen welche zurückgesetzt wurden.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/EJBTransactionRuntimeMBean.html?skipReload=true#TransactionsRolledBackTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/EJBTransactionRuntimeMBean.html?skipReload=true#TransactionsRolledBackTotalCount)

### **2.3.10 JDBC Connection Pool Connection Delay Time**

Durchschnittliche Zeit, welche zum Aufbau einer Verbindung zum Datenbankserver benötigt wird.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JDBCConnectionPoolRuntimeMBean.html?skipReload=true#ConnectionDelayTime](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JDBCConnectionPoolRuntimeMBean.html?skipReload=true#ConnectionDelayTime)

### **2.3.11 JDBC Connection Pool Active Connections High Count**

Höchste Anzahl an Verbindungen, welche seit dem Erstellen des Connection Pools gleichzeitig genutzt wurden.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JDBCConnectionPoolRuntimeMBean.html?skipReload=true#ActiveConnectionsHighCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JDBCConnectionPoolRuntimeMBean.html?skipReload=true#ActiveConnectionsHighCount)

### **2.3.12 JDBC Connection Pool Active Connections Current Count**

Anzahl JDBC Connections welche derzeit durch die Applikationen verwendet werden.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JDBCConnectionPoolRuntimeMBean.html?skipReload=true#ActiveConnectionsCurrentCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JDBCConnectionPoolRuntimeMBean.html?skipReload=true#ActiveConnectionsCurrentCount)

### **2.3.13      *Execute Queue Pending Request Current Count***

Anzahl wartender Requests in der Execute Queue.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ExecuteQueueRuntimeMBean.html?skipReload=true#PendingRequestCurrentCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ExecuteQueueRuntimeMBean.html?skipReload=true#PendingRequestCurrentCount)

### **2.3.14      *Execute Queue Execute Thread CurrentIdle Count***

Anzahl an Threds welche der Execute Queue zugewiesen sind und sich zurzeit im „idle“ Status befinden.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ExecuteQueueRuntimeMBean.html?skipReload=true#ExecuteThreadCurrentIdleCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ExecuteQueueRuntimeMBean.html?skipReload=true#ExecuteThreadCurrentIdleCount)

### **2.3.15      *Execute Queue Execute Thread TotalCount***

Gesamtanzahl Execute Threads welche der Execute Queue zugewiesen sind.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ExecuteQueueRuntimeMBean.html?skipReload=true#ExecuteThreadTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ExecuteQueueRuntimeMBean.html?skipReload=true#ExecuteThreadTotalCount)

### **2.3.16      *Thread Pool Execute Thread Total Count***

Gesamtanzahl an Threads im Thread Pool.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#ExecuteThreadTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#ExecuteThreadTotalCount)

### **2.3.17      *Thread Pool Throughput***

Durchschnittliche Anzahl an Requests, welche pro Sekunde abgearbeitet werden.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#Throughput](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#Throughput)

### **2.3.18      *Thread Pool Hogging Thread Count***

Liefert die Anzahl der Threads, welche zurzeit an einen Request gebunden sind. Diese Threads werden als „stuck“ deklariert sofern sie nicht während des konfigurierten Timeouts beendet werden.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#HoggingThreadCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#HoggingThreadCount)

### **2.3.19      *Thread Pool Pending User Request Count***

Anzahl der noch nicht bearbeiteten User Requests in der Priority Queue. Die Priority Queue beinhaltet Requests von internen Subsystemen und User Requests. Diese Metrik liefert jedoch nur die Anzahl aller noch nicht bearbeiteten User Requests.

Quelle: dynaTrace 3.5 built-in

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#PendingUserRequestCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/ThreadPoolRuntimeMBean.html?skipReload=true#PendingUserRequestCount)

### **2.3.20      *JTA Active Transactions Total Count***

Anzahl der aktiven Transaktionen auf dem Server.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JTARuntimeMBean.html?skipReload=true#ActiveTransactionsTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JTARuntimeMBean.html?skipReload=true#ActiveTransactionsTotalCount)

### **2.3.21      *JTA Transaction Committed Total Count***

Anzahl bestätigter Transaktionen auf dem Server.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JTARuntimeMBean.html?skipReload=true#TransactionCommittedTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JTARuntimeMBean.html?skipReload=true#TransactionCommittedTotalCount)

### **2.3.22      *JTA Transaction Abandoned Total Count***

Anzahl abgebrochener Transaktionen auf dem Server.

Quelle: WebLogic Extended Plugin

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JTARuntimeMBean.html?skipReload=true#TransactionAbandonedTotalCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JTARuntimeMBean.html?skipReload=true#TransactionAbandonedTotalCount)

## 2.4 JMS Metriken

Die folgenden Metriken liefern Werte zur Performance Analyse des Java Message Service. Sämtliche Metriken werden vom WebLogic Extended Plugin zur Verfügung gestellt.

### 2.4.1 JMS Runtime Current Connections

Aktuelle Anzahl an Verbindungen zum WebLogic Server.

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSRuntimeMBean.html?skipReload=true#ConnectionsCurrentCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSRuntimeMBean.html?skipReload=true#ConnectionsCurrentCount)

### 2.4.2 JMS Runtime Connections High Count

Höchste bisher gemessene Anzahl an gleichzeitigen JMS Verbindungen zum WebLogic Server.

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSRuntimeMBean.html?skipReload=true#ConnectionsHighCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSRuntimeMBean.html?skipReload=true#ConnectionsHighCount)

### 2.4.3 JMS Producer Pending Messages

Anzahl noch nicht verarbeiteter Messages (uncommitted und unacknowledged) von diesem Producer.

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSProducerRuntimeMBean.html?skipReload=true#MessagesPendingCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSProducerRuntimeMBean.html?skipReload=true#MessagesPendingCount)

### 2.4.4 JMS Producer Pending Bytes

Anzahl noch nicht verarbeiteter Bytes (uncommitted und unacknowledged) von diesem Producer.

Einheit: Byte

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSProducerRuntimeMBean.html?skipReload=true#BytesPendingCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSProducerRuntimeMBean.html?skipReload=true#BytesPendingCount)

### 2.4.5 JMS Destination Pending Messages

Anzahl nicht verarbeiteter Messages in der Destination.

Die Anzahl noch nicht verarbeiteter Messages wird zusätzlich zu der Gesamtanzahl an Messages gesehen.

Als „pending“ wird eine Message bezeichnet welche innerhalb einer Transaction zwar verschickt, allerdings noch nicht committed wurde. Auch bereits empfangene Messages, welche noch nicht den Status „committed“ oder „acknowledged“ haben, werden als „pending“ bezeichnet.

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSDestinationRuntimeMBean.html?skipReload=true#MessagesPendingCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSDestinationRuntimeMBean.html?skipReload=true#MessagesPendingCount)

### 2.4.6 JMS Destination Pending Bytes

Anzahl noch nicht verarbeiteter Bytes, welche in der Destination noch gehalten werden. Die Anzahl der „pending“ Bytes wird zusätzlich zu der Gesamtanzahl von „Current Number of Bytes“ gesehen.

Einheit: Byte

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSDestinationRuntimeMBean.html?skipReload=true#BytesPendingCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSDestinationRuntimeMBean.html?skipReload=true#BytesPendingCount)

#### **2.4.7        *JMS Consumer Pending Messages***

Anzahl noch nicht verarbeiteter Messages (uncommitted und unacknowledged) von diesem Consumer.

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSConsumerRuntimeMBean.html?skipReload=true#MessagesPendingCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSConsumerRuntimeMBean.html?skipReload=true#MessagesPendingCount)

#### **2.4.8        *JMS Consumer Pending Bytes***

Anzahl noch nicht verarbeiteter Bytes (uncommitted und unacknowledged) von diesem Consumer.

Einheit: Byte

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSConsumerRuntimeMBean.html?skipReload=true#BytesPendingCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSConsumerRuntimeMBean.html?skipReload=true#BytesPendingCount)

#### **2.4.9        *JMS Messages Sent Count***

Anzahl verschickter JMS Nachrichten

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSSessionRuntimeMBean.html?skipReload=true#MessagesSentCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSSessionRuntimeMBean.html?skipReload=true#MessagesSentCount)

#### **2.4.10       *JMS Messages Received Count***

Anzahl empfangener JMS Nachrichten.

Einheit: Anzahl

Referenz:

[http://download.oracle.com/docs/cd/E14571\\_01/apirefs.1111/e13951/mbeans/JMSSessionRuntimeMBean.html?skipReload=true#MessagesReceivedCount](http://download.oracle.com/docs/cd/E14571_01/apirefs.1111/e13951/mbeans/JMSSessionRuntimeMBean.html?skipReload=true#MessagesReceivedCount)

## **2.5            Applikations Metriken**

Bei den in diesem Kapitel beschriebenen Metriken handelt es sich um Messpunkte, welche durch die dynaTrace ByteCode Instrumentierung in Echtzeit aus dem jeweils ausgeführten Applikationscode ausgelesen werden. Die Datenquelle dieser Messpunkte ist somit immer ein dynaTrace Agent.

### **2.5.1            *API Times***

Das Layer Breakdown Chart ermöglicht es, einen Überblick über die Total Execution Times oder CPU Times der Applikation zu bekommen. In der Grafik werden wird die Aufteilung der unterschiedlichen Layer (API) visualisiert.

Einheit: ms

Referenz: <https://community.dynatrace.com/community/display/DOCDT35/Layer+Breakdown>

### **2.5.2            *PurePath Duration (Count)***

Anzahl Transaktionen (z.B. Web Requests).

Einheit: Count

### **2.5.3            *DB Count***

Anzahl ausgeführter "Execute" Datenbank Statements ohne Berücksichtigung von „Prepare“ Statements.

Einheit: Count

### **2.5.4            *PurePath Duration***

Misst die Zeit, von Begin (Entry Event) bis Ende (Exit Event) einer Transaktion.

Einheit: ms

### **2.5.5            *PurePath Duration w/o Suspension***

Misst die Zeit, von Begin (Entry Event) bis Ende (Exit Event) einer Transaktion ohne Runtime Suspensions (z.B Garbage Collections).

Einheit: ms

### **2.5.6            *PurePath Response Time***

Misst die Gesamtzeit, von Begin bis Ende des ersten Knotens des PurePaths.

Einheit: ms

### **2.5.7            *DB Time***

Repräsentiert die Gesamtzeit, welche vom Datenbank Layer (Connection Komponente + Datenbank Management System) für benötigt wurde.

Einheit: ms



## 2.6 Plugin Metriken

In diesem Kapitel werden Metriken beschrieben, welche durch den Einsatz von weiteren Plugins im Referenz System Profil verfügbar sind.

### 2.6.1 System Monitor Plugin

Das dynaTrace Unix System Monitor Plugin stellt Metriken zur System Performance zur Verfügung. Dazu gehören Kennzahlen wie Memory, CPU und beispielsweise Disk Auslastung. Auch Informationen über Prozesse und Netzwerk Werte werden von diesem Plugin geliefert. Diese Metriken werden auf Dashboards übersichtlich dargestellt. Dies erlaubt es, Probleme zu korrelieren und die Ursache einzugrenzen.

Metriken welche im Kapitel 2 (Prozess & System Metriken) beschrieben werden, stammen von diesem Plugin.

Die Einstellungen dieses Plugins werden im Kapitel 4.3.1.2 (Operating System Monitoring Konfiguration) genauer beschrieben.

### 2.6.2 Active Monitoring Plugin

Mit dynaTrace können verschiedene Plugins für das Active Monitoring verwendet werden. Das dynaTrace URL Monitor Plugin zum Beispiel überwacht die Verfügbarkeit einer einzelnen URL, z.B. einer Login Seite. Dies ist noch kein echtes Active Monitoring, gibt aber einen ersten Hinweis auf die Verfügbarkeit der Applikation. Echtes Active Monitoring kann mit dynaTrace implementiert werden, ist jedoch ausserhalb des Scopes dieser Beschreibung.

Es gibt zudem noch mehrere Arten von Active Monitoring Plugins. Einige Beispiele:

- URL Monitoring
- Web Transaction Monitoring (WTM)
- Generic Command Execution (z.B. "Ping")
- ...

Der Unterschied zwischen dem URL Monitoring und dem Web Transaction Monitoring ist, dass das URL Monitoring nur die Verfügbarkeit einer einzigen URL zulässt und als Ergebnis u.a. die Zeit und den http Response Code (z.B. 200) zurückgibt.

Das Web Transaction Monitoring erlaubt es, eine ganze Serie von Anfragen, z.B. Login -> Search -> Logout zu definieren und retourniert als Ergebnis u.a. die Zeit und ob der gesamte Workflow funktioniert hat (Ja/Nein resp. 1/0).

Das dynaTrace WebLogic FastPack verwendet das komplexere Web Transaction Monitoring Plugin.

Die Konfiguration wird im Kapitel 4.3.1.3 (Synthetic Transaction Monitor) näher beschrieben.

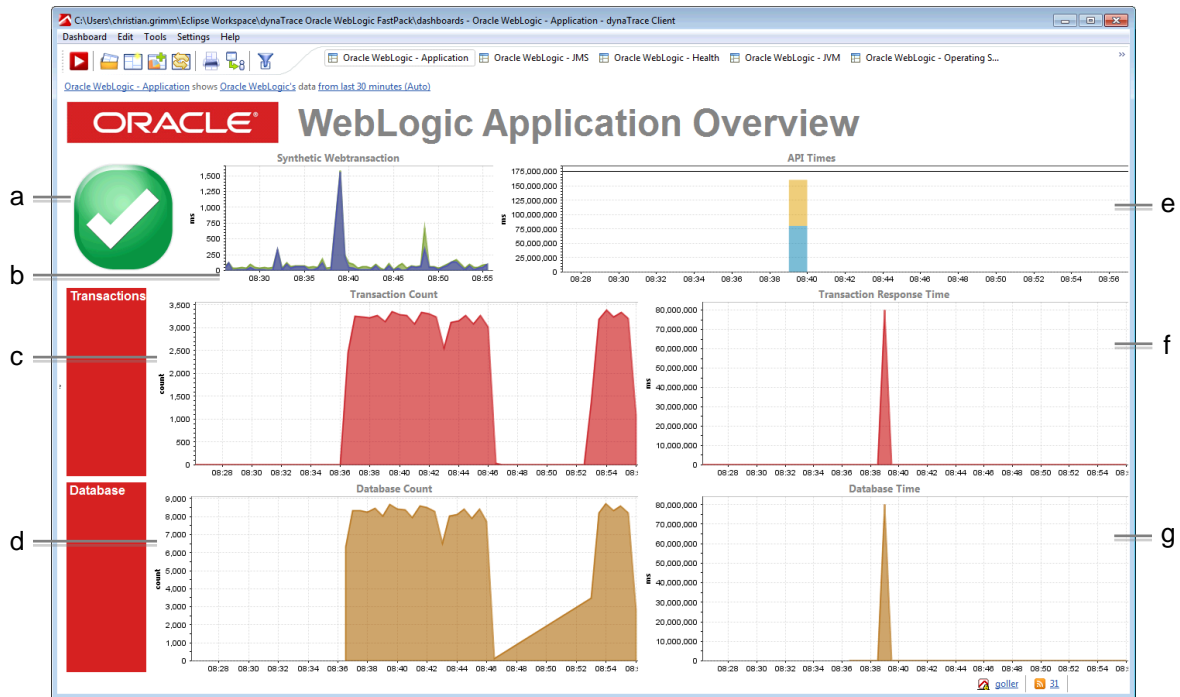
### 3 Dashboards

Für das WebLogic Basis Monitoring wurden 5 Referenz Dashboards definiert, welche in Form eines dynaTrace Fastpacks für Oracle WebLogic Monitoring zusammengefasst wurden.

- Oracle WebLogic – Application Overview
- Oracle WebLogic – Operating System
- Oracle WebLogic – WebLogic Health
- Oracle WebLogic – JMS
- Oracle WebLogic – JVM

### 3.1 Application Overview

Das Application Overview Dashboard ermöglicht den Überblick über die aktuell ausgeführten Transaktionen. Sowohl applikatorische als auch Datenbank orientierte Transaktionen werden visualisiert. Dies ermöglicht einen Einblick über die Anzahl der zurzeit ausgeführten Transaktionen und deren Antwortzeiten. Zusätzlich wird mittels Aktiv-Monitoring geprüft, ob elementare Bereiche des Systems für die Endanwender verfügbar sind.



Datenquellen:

- dynaTrace Agents
- Web Transaction Monitoring Plugin (Konfiguration siehe Kapitel 4.4.1) oder URL Monitoring Plugin (Konfiguration siehe Kapitel 4.3.1.3)

Folgende Metriken sind auf diesem Dashboard verfügbar:

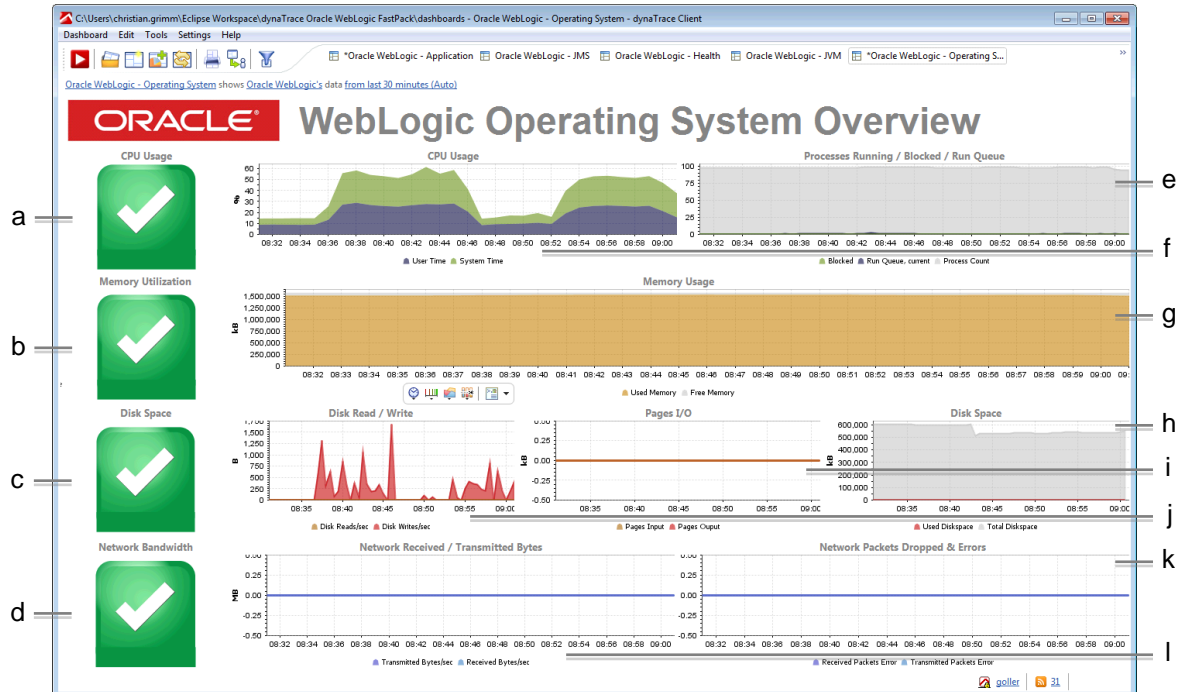
- a) Status des Active Monitorings  
(muss manuell konfiguriert werden. Siehe Kapitel 4.3.1.3 Synthetic Transaction Monitor)
- b) Response Zeit des Active Monitorings  
(muss manuell konfiguriert werden. Siehe Kapitel 4.3.1.3 Synthetic Transaction Monitor)
- c) Transaction Count  
Anzahl PurePaths pro Intervall (siehe [PurePath Count](#))
- d) Transaction Response Time  
Durchschnittliche Antwortzeit aller PurePaths pro Intervall (siehe [PurePath Response Time](#))
- e) Layer Breakdown  
API Breakdown. Zeigt die Zeiten der einzelnen Layers/APIs an (siehe [API Times](#))
- f) Database Count  
Anzahl SQL Statements pro Intervall (siehe [DB Count](#))

- g) Database Time  
Durchschnittliche Antwortzeit aller SQL Statements pro Intervall (siehe [DB Time](#))

Den beiden Dashlets (a,b) für das Active Monitoring müssen jeweils manuell für die gewünschten Metriken und Thresholds konfiguriert werden. Die Ampel steht auf „grün“ solange „Transaction Success“ vom Web Transaction Monitor geliefert wird.

## 3.2 Operating System

Das Operating System Dashboard wurde erstellt, um den Gesamtzustand eines Hosts abbilden zu können. Das Dashboard enthält vor allem hardware-nahe Werte auf Betriebssystem Ebene. Das im dynaTrace FastPack enthaltene Dashboard kann derzeit ausschliesslich mit UNIX Betriebssystemen verwendet werden.



Datenquelle:

- Unix Monitor Plugin (Konfiguration siehe Kapitel 4.4.1)

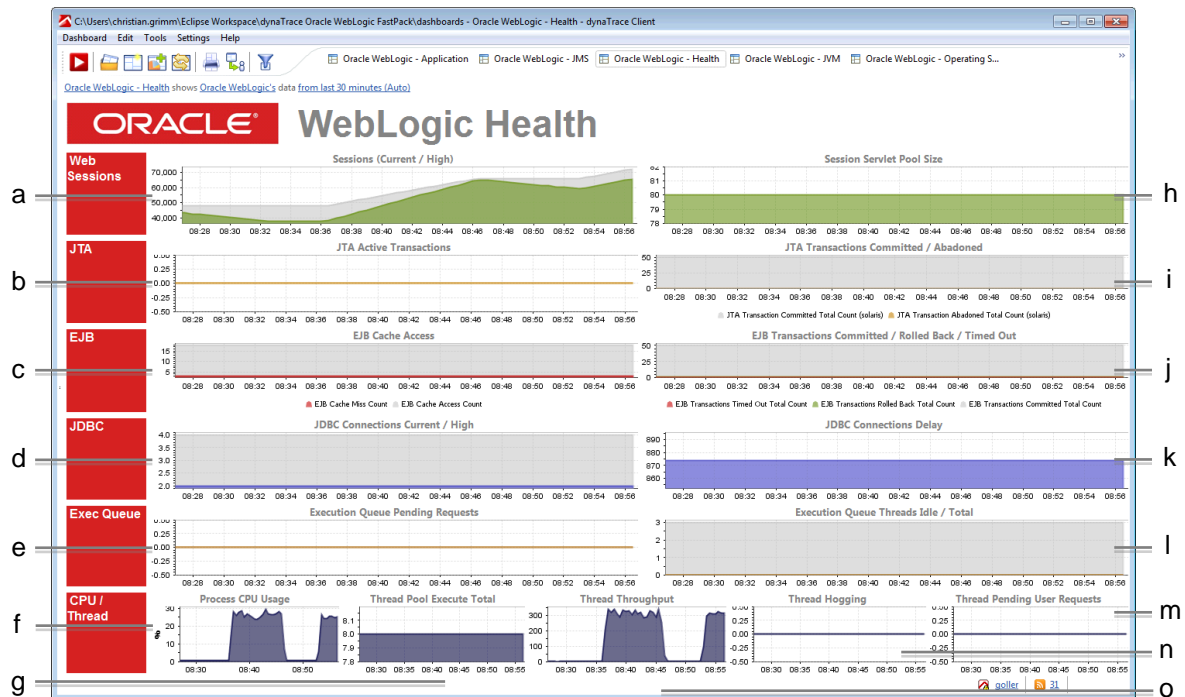
Folgende Metriken sind auf diesem Dashboard verfügbar:

- CPU Status Traffic Light
  - Bei über 80% Warning und bei über 90% Severe (siehe [Processor Time](#))
- Memory Utilization Traffic Light.
  - Bei weniger als 10% freiem Memory Warning und bei weniger als 5% Severe (siehe [Memory Available](#))
  - Bei weniger als 200MB verfügbarem Swap Space Warning und bei weniger als 100MB Severe (siehe [Free Swap](#))
- Disk Space Traffic Light
  - Bei mehr als 80% Auslastung des „var“ Spaces „Warning“, bei mehr als 90% „Severe“ (siehe [Var Usage](#))
  - Bei mehr als 80% Auslastung des „temp“ Spaces „Warning“, bei mehr als 90% „Severe“ (siehe [Temp Usage](#))
- Network Bandwidth Traffic Light
  - Mehr als 10MB/sec ist „Warning“ und bei mehr als 12MB/sec „Severe“. Dies gilt sowohl für inbound als auch outbound Traffic (siehe [Received Bytes/sec](#) und [Transmitted Bytes/sec](#))

- e) Processes
  - Anzahl (Total) laufende Prozesse (siehe [Process Count](#)).
  - Anzahl Prozess in „non-interruptible sleep“ (siehe [Blocked](#))
  - Anzahl Prozesse, welche auf Run-Time warten (siehe [Run Queue, current](#))
- f) CPU Usage
  - Auslastung der CPU nur durch User-Prozesse, keine System Prozesse (siehe [User Time](#))
  - Auslastung der CPU nur durch Prozesse des Systems (siehe [System Time](#))
- g) Memory Usage
  - Aktuell verwendeter Speicher (siehe [Used Memory](#))
  - Verfügbarer Speicher (siehe [Free Memory](#))
- h) Disk Space
  - Aktuelle Diskauslastung (siehe [Used Diskspace](#))
  - Disk Gesamtkapazität (siehe [Total Diskspace](#))
- i) Pages I/O. Beinhaltet folgende Metriken:
  - Paged/Swapped in (siehe [Pages Input](#))
  - Paged/Swapped out (siehe [Pages Output](#))
- j) Disk Read / Write
  - Harddisk Aktivität beim Lesen von Daten (siehe [Disk Reads/sec](#))
  - Harddisk Aktivität beim Schreiben von Daten (siehe [Disk Writes/sec](#))
- k) Network Packets Dropped & Errors
  - Anzahl Netzwerk Pakete, welche beim Empfangenen verworfen wurden (siehe [Received Packets Dropped](#))
  - Anzahl Netzwerk Pakete welche beim Senden verworfen wurden (siehe [Transmitted Packets Dropped](#))
  - Anzahl fehlerhaft empfangener Netzwerk Pakete (siehe [Received Packets Error](#))
  - Anzahl fehlerhaft gesendeter Netzwerk Pakete (siehe [Transmitted Packets Error](#))
- l) Network Received / Transmitted Bytes
  - Netzwerk Traffic, empfangene Bytes pro Sekunde (siehe [Received Bytes/sec](#))
  - Netzwerk Traffic, gesendete Bytes pro Sekunde (siehe [Transmitted Bytes/sec](#))

### 3.3 WebLogic Health

Das WebLogic Health Dashboard beinhaltet interne Metriken über den Zustand des Oracle WebLogic Application Servers.



Datenquellen:

- dynaTrace Agents
- WebLogic Extended Plugin (Bestandteil des WebLogic FastPacks)

Folgende Metriken sind auf diesem Dashboard verfügbar:

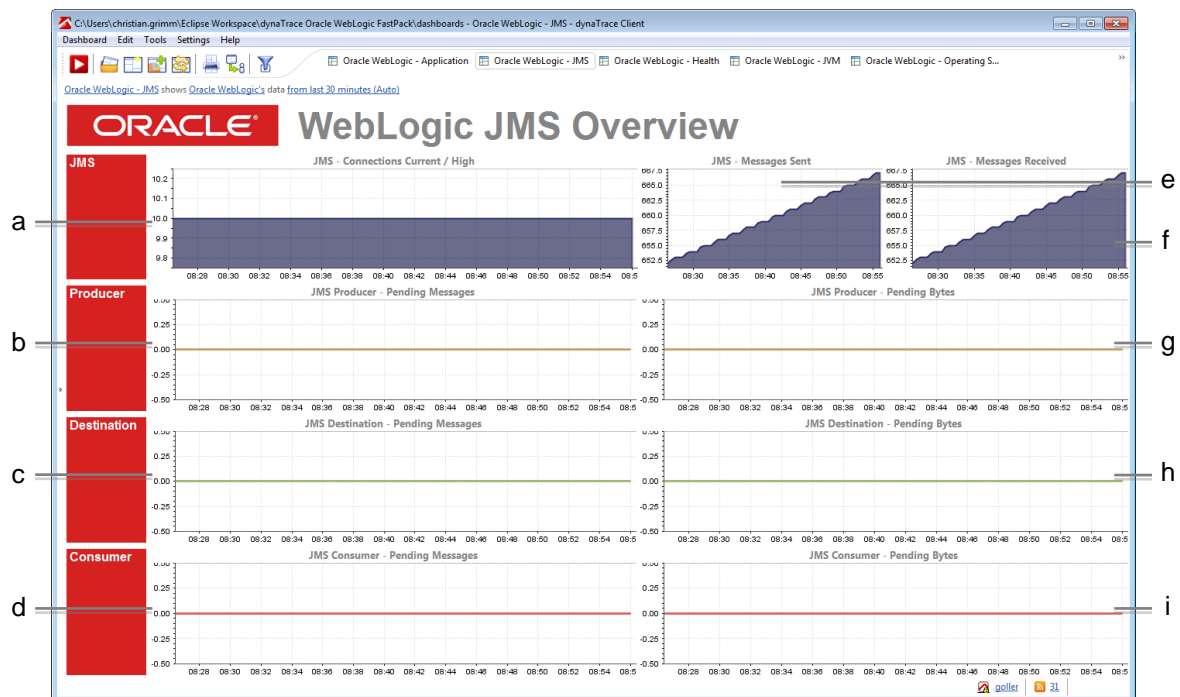
- Sessions (Current / High)
  - Anzahl der gegenwärtigen offenen Sessions (siehe [Open Sessions Count](#))
  - „Hochwasser Marke“ für die gesamte Anzahl offener Sessions (siehe [Session High Count](#))
- JTA Active Transactions
  - Anzahl aktiver Transaktionen (siehe [JTA Active Transactions Total Count](#))
- EJB Access
  - Anzahl Versuche, auf ein Bean im Cache zuzugreifen (siehe [EJB Cache Access Count](#))
  - Anzahl misslungener Versuche, um auf ein Bean im Cache zuzugreifen (siehe [EJB Cache Miss Count](#))
- JDBC Connections Current / High
  - Höchste Anzahl gleichzeitiger Verbindungen seit dem Start des Connection Pools (siehe [JDBC Connection Pool Active Connections High Count](#))
  - Anzahl der Verbindungen, die die Applikation im Moment hält (siehe [JDBC Connection Pool Active Connections Current Count](#))
- Execution Queue Pending Requests
  - Anzahl Objekte welche in der Execution Queue auf die Abarbeitung warten (siehe [Execute Queue Pending Request Current Count](#))

- f) Process CPU Usage
  - Aktuelle CPU Auslastung durch den WebLogic Prozess
- g) Thread Pool Execute Total
  - Anzahl aller Threads im Thread Pool (siehe [Thread Pool Execute Thread Total Count](#))
- h) Session Servlet Pool Size
  - Zeigt die aktuell konfigurierte Session Pool Size an, welche im "weblogic.xml" definiert ist (siehe [Session Servlet Pool Size](#))
- i) JTA Transactions Committed / Abandoned
  - Gesamtanzahl an „Committed“ Transaktionen (siehe [JTA Transaction Committed Total Count](#))
  - Gesamtanzahl an „Abandoned“ Transaktionen (siehe [JTA Transaction Abandoned Total Count](#))
- j) EJB Transactions Committed / Rolled Back / Timed Out
  - Total Anzahl bestätigter EJB Transaktionen (siehe [EJB Transactions Committed Total Count](#))
  - Total Anzahl EJB Transaktionen welche zurückgesetzt wurden (siehe [EJB Transactions Rolled Back Total Count](#))
  - Total Anzahl EJB Transaktionen welche durch eine zeitliche Überschreitung zurückgesetzt wurden (siehe [EJB Transactions Timed Out Total Count](#))
- k) JDBC Connection Delay
  - Durchschnittliche Zeit, um eine physische Verbindung zur Datenbank aufzubauen (siehe [JDBC Connection Pool Connection Delay Time](#))
- l) Execution Queue Threads Idle / Total
  - Anzahl Threads in der Execute Queue, welche sich im „idle“ Zustand befinden. (siehe [Execute Queue Execute Thread CurrentIdle Count](#))
  - Total Anzahl an Executions Threads, welche sich in der Execute Queue befinden (siehe [Execute Queue Execute Thread TotalCount](#))
- m) Thread Pending User Requests
  - Durchschnittliche Anzahl noch nicht abgeschlossener Anfragen pro Sekunde (siehe [Thread Pool Pending User Request Count](#))
- n) Thread Hogging
  - Anzahl aktueller Threads, welche durch Requests gebunden sind. Diese Threads können unter Umständen definitiv hängen bleiben (siehe [Thread Pool Hogging Thread Count](#))
- o) Thread Throughput
  - Durchschnittliche Anzahl abgeschlossener Anfragen pro Sekunde (siehe [Thread Pool Throughput](#))



### 3.4 WebLogic JMS Overview

Dieses Dashboard stellt detaillierte Metriken des Message Queue Subsystems des Oracle WebLogic Application Servers dar.



Datenquellen:

- dynaTrace Agents
- WebLogic Extended Plugin (Bestandteil des WebLogic FastPacks)

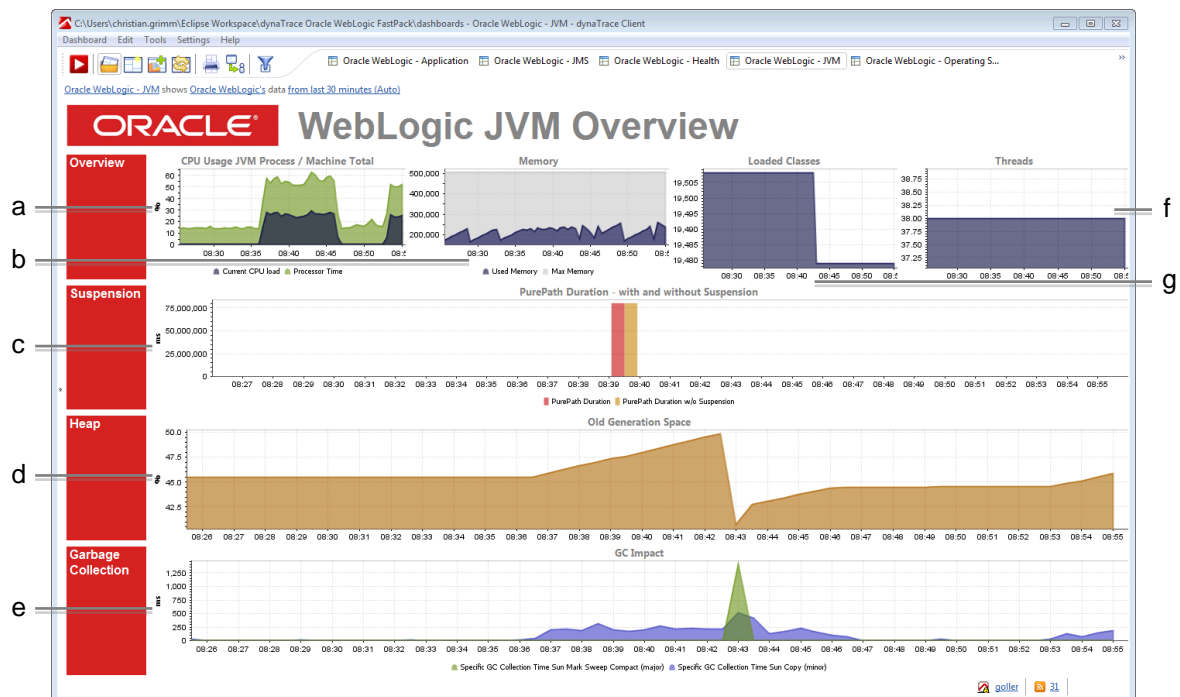
Folgende Metriken sind auf diesem Dashboard verfügbar:

- a) JMS – Connections Current / High
  - Anzahl aktiver Verbindungen (siehe [JMS Runtime Current Connections](#))
  - Hochwassermarke für max. Anzahl Verbindungen (siehe [JMS Runtime Connections High Count](#))
- b) JMS Producer – Pending Messages
  - JMS Producer Pending Messages (siehe [JMS Producer Pending Messages](#))
- c) JMS Destination – Pending Messages
  - JMS Destination Pending Messages (siehe [JMS Destination Pending Messages](#))
- d) JMS Consumer – Pending Messages
  - JMS Consumer Pending Messages (siehe [JMS Consumer Pending Messages](#))
- e) JMS Messages Sent
  - Anzahl verschickter JMS Nachrichten (siehe [JMS Messages Sent Count](#))
- f) JMS Messages Received
  - Anzahl empfangener JMS Nachrichten (siehe [JMS Messages Received Count](#))
- g) JMS Producer – Pending Bytes
  - JMS Producer Pending Bytes – (siehe [JMS Producer Pending Bytes](#))

- h) JMS Destination – Pending Bytes
  - JMS Destination Pending Bytes (siehe [JMS Destination Pending Bytes](#))
- i) JMS Consumer – Pending Bytes
  - JMS Consumer Pending Bytes (siehe [JMS Consumer Pending Bytes](#))

### 3.5 JVM

Das JVM Details Dashboard beinhaltet interne Werte über den Zustand der Java Virtual Machine. Dieses Dashboard ist derzeit auf die Sun JVM ausgerichtet.



Datenquellen:

- dynaTrace Agent
- UNIX Monitor Plugin

Folgende Metriken sind auf diesem Dashboard verfügbar:

- CPU**
  - Aktuelle Prozessorauslastung durch die JVM in % (siehe [Current CPU load](#))
  - Aktuelle Prozessorauslastung Total (System)
- Memory**
  - Aktuelle Speicherbelegung (siehe [Used Memory](#))
  - Maximal verfügbarer Arbeitsspeicher (siehe [Max Memory](#))
- PurePath Duration – with and without Suspension**
  - PurePath Duration Total (siehe [PurePath Duration](#))
  - PurePath Duration ohne Runtime Suspensions (siehe [PurePath Duration w/o Suspension](#))
- Old Generation Space**
  - Auslastung des Old Generation Space in kB (siehe [Memory Pool Sun Old Gen used](#))
- GC Impact**
  - GC Zeit Sun PS Scavenge, Minor (siehe [Specific GC Collection Time Sun PS Scavenge \(minor\)](#))
  - GC Zeit Sun PS Mark Sweep, Major (siehe [Specific GC Collection Time Sun PS Mark Sweep \(major\)](#))
  - GC Zeit Sun New Parallel Copy, Minor (siehe [Specific GC Collection Time Sun New Parallel Copy \(minor\)](#))

- GC Zeit Sun Concurrent Mark Sweep, Major (siehe [Specific GC Collection Time Sun Concurrent Mark Sweep \(major\)](#))
  - GC Zeit Sun Mark Sweep Compact, Major (siehe [Specific GC Collection Time Sun Mark Sweep Compact \(major\)](#))
  - GC Zeit Sun Copy, Major (siehe [Specific GC Collection Time Sun Copy \(minor\)](#))
- f) Threads
- Anzahl laufende Threads, Deamon und Non-Deamon (siehe [Thread Count](#))
- g) Classes
- Anzahl aktuell geladener Klassen innerhalb der Java Virtual Machine (siehe [Loaded Classes](#))

## 4 Inbetriebnahme

Im Folgenden wird beschrieben, wie das dynaTrace Oracle WebLogic FastPack installiert und konfiguriert wird. Zudem wird aufgezeigt, welche Schritte notwendig sind, um das FastPack an Umgebungen mit mehreren parallelen Oracle WebLogic Instanzen anzupassen.

Die Inbetriebnahme des dynaTrace FastPacks besteht im Wesentlichen aus drei Schritten:



- Download
- Installation
- Konfiguration

### 4.1 Download

Unter folgender URL kann das dynaTrace Oracle WebLogic Extended Metrics Plugin sowie das dynaTrace Oracle WebLogic FastPack heruntergeladen werden:

<https://community.dynatrace.com/community/display/DL/Oracle+WebLogic+FastPack>

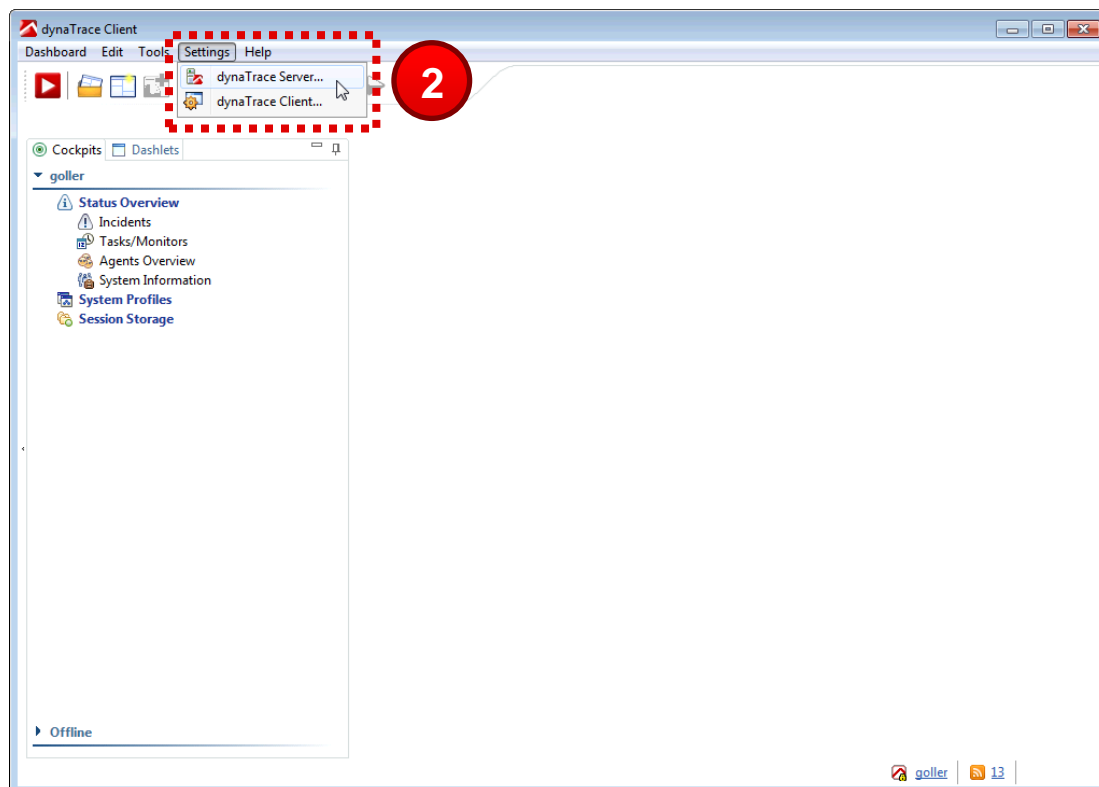
Die heruntergeladenen Files sollten wie folgt aussehen:

Name	Type
 com.dynatrace.weblogic.extended_1.0.0.0.jar	Executable Jar File
 dynaTrace Oracle WebLogic FastPack.dtp	dynaTrace Plugin

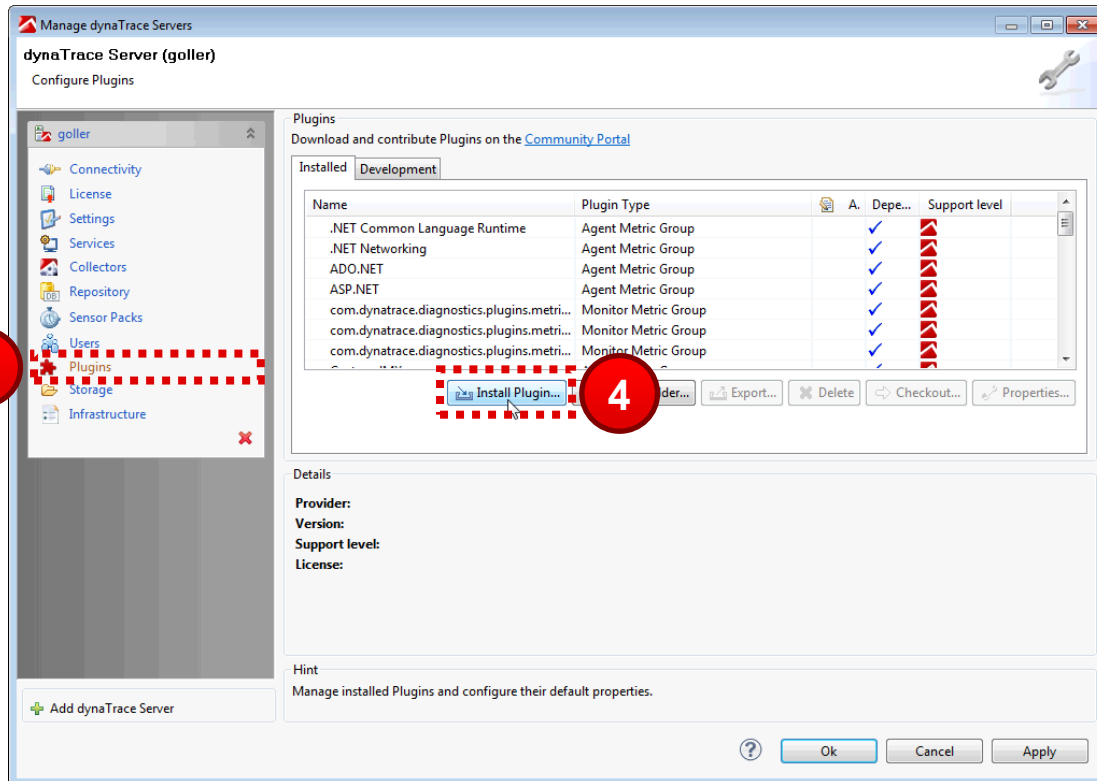
1

### 4.2 Installation

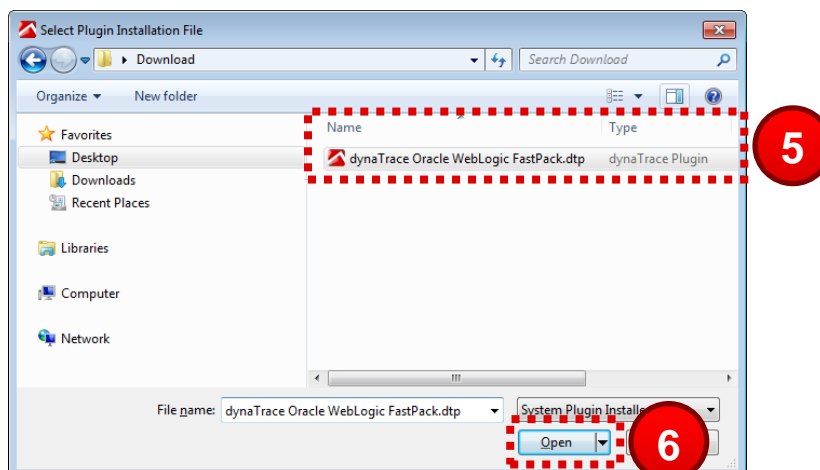
Zur Installation des FastPacks wird der dynaTrace Client gestartet. Im Menü unter „Settings“ können anschliessend die dynaTrace Server Einstellungen aufgerufen werden.



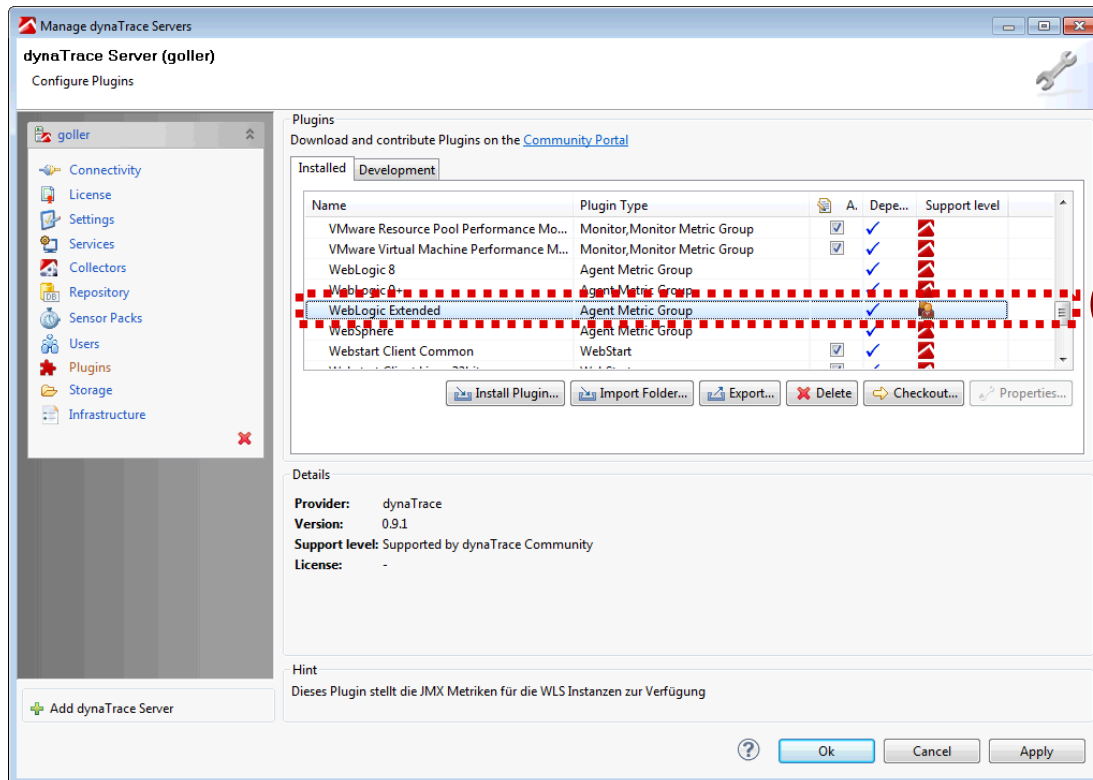
In den Server Einstellungen unter „Plugins“ können nun zuerst das dynaTrace Oracle WebLogic Extended Metrics Plugin und anschließend das FastPacks installiert werden. Dazu den Button „Install Plugin...“ betätigen und die heruntergeladene Datei auswählen:



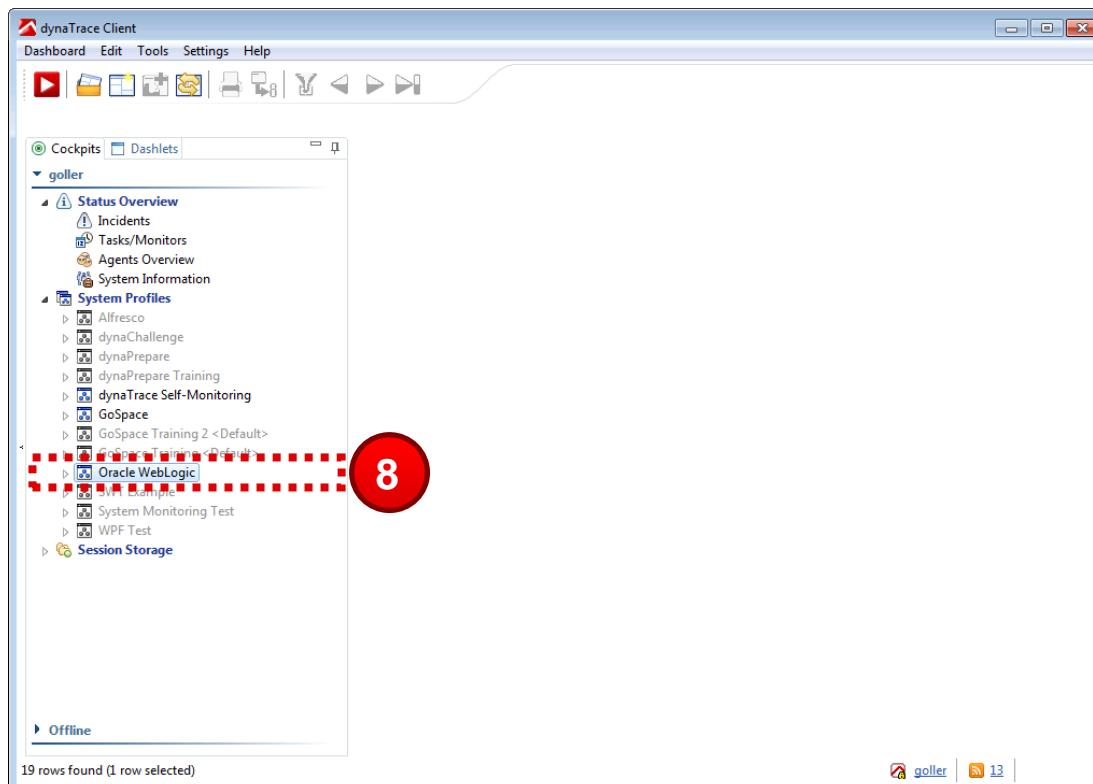
Die heruntergeladene Datei auswählen und mit „Open“ bestätigen:  
Wichtig: Unbedingt zuerst das dynaTrace Oracle WebLogic Extended Metrics Plugin installieren und erst danach das dynaTrace Oracle WebLogic FastPack.



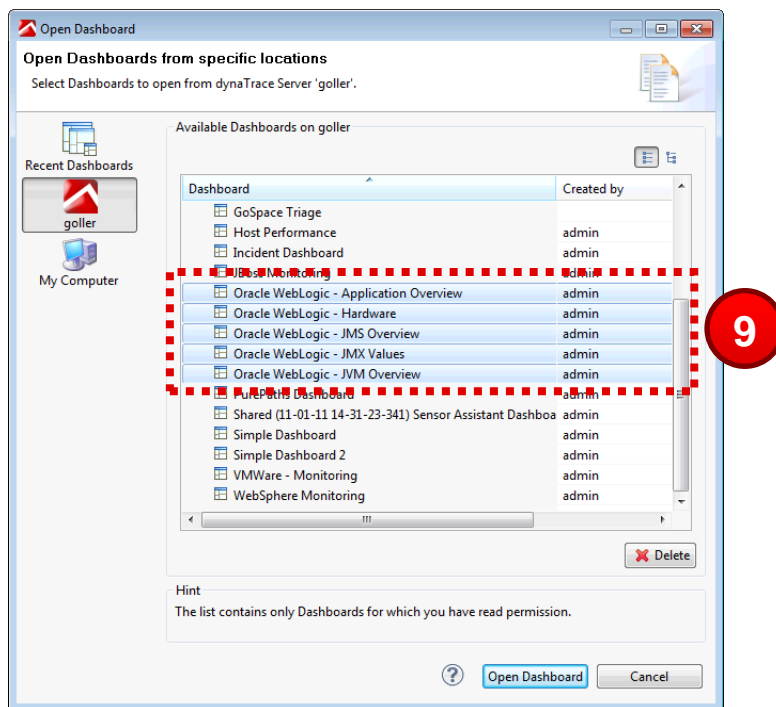
Danach überprüfen, ob die Installation des Plugins „WebLogic Extended“ ordnungsgemäss durchgeführt werden konnte (In der Spalte „Deployed“ sollte das „Check“ Symbol angezeigt werden):



Zusätzlich ist das Oracle WebLogic Referenz Profil nun unter „Cockpits“ im Hauptfenster des dynaTrace Clients verfügbar:



Auch die fünf Oracle WebLogic Refrenz Dashboards wurden mit dem Plugin automatisch installiert:





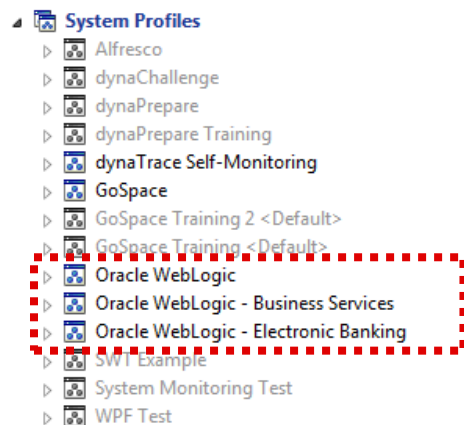
## 4.3 Konfiguration

Der nächste Schritt der Inbetriebnahme des Oracle WebLogic FastPacks besteht darin, das Referenz System Profil („Oracle WebLogic“) und die mitgelieferten Dashboards auf die vorhandene Umgebung anzupassen.

### 4.3.1 System Profil

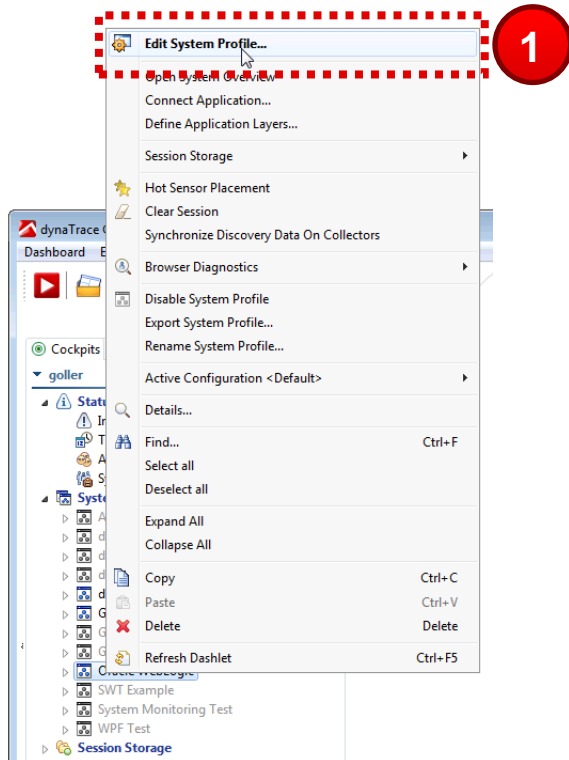
Das im FastPack enthaltene System Profil „Oracle WebLogic“ kann direkt verwendet werden. Werden mehrere verschiedene Oracle WebLogic Instanzen betrieben, bietet sich optional die Möglichkeit an, dass Referenz System Profil für die jeweilige Instanz zu kopieren.

Beispiel:

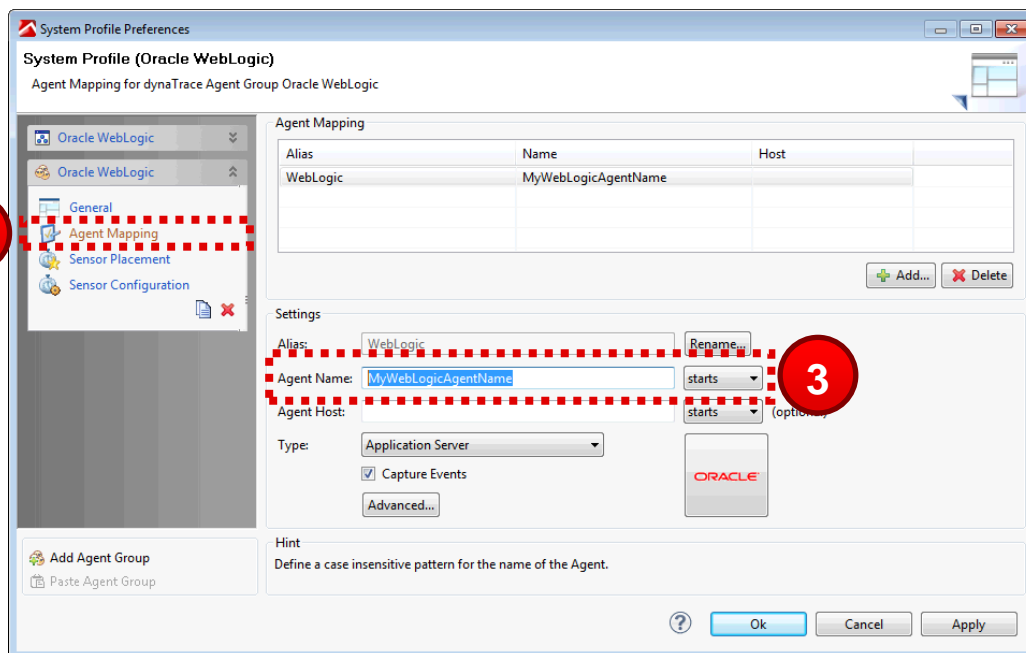


### 4.3.1.1 AgentGroup Konfiguration

Um das System Profil mit Daten aus dem Oracle WebLogic Application Server zu versorgen, ist eine Anpassung der Agent Group Konfiguration im entsprechenden System Profil notwendig. Wichtig dabei ist vor allem, dass der Name des dynaTrace Agents mit der Definition im AgentGroup Mapping übereinstimmt.



Im System Profil muss unter der Kategorie „Agent Mapping“ der Agent Name entsprechend angepasst werden:

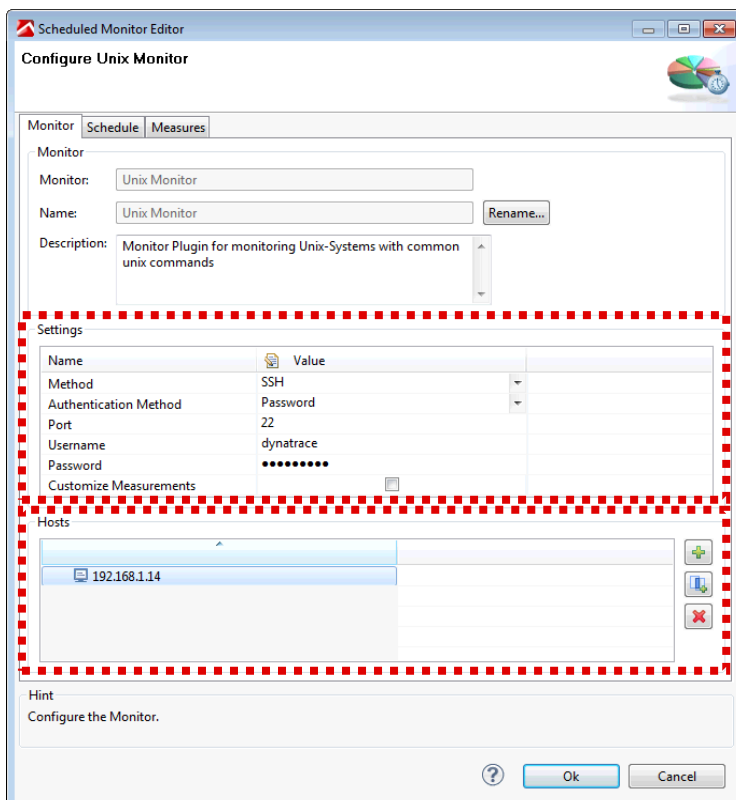
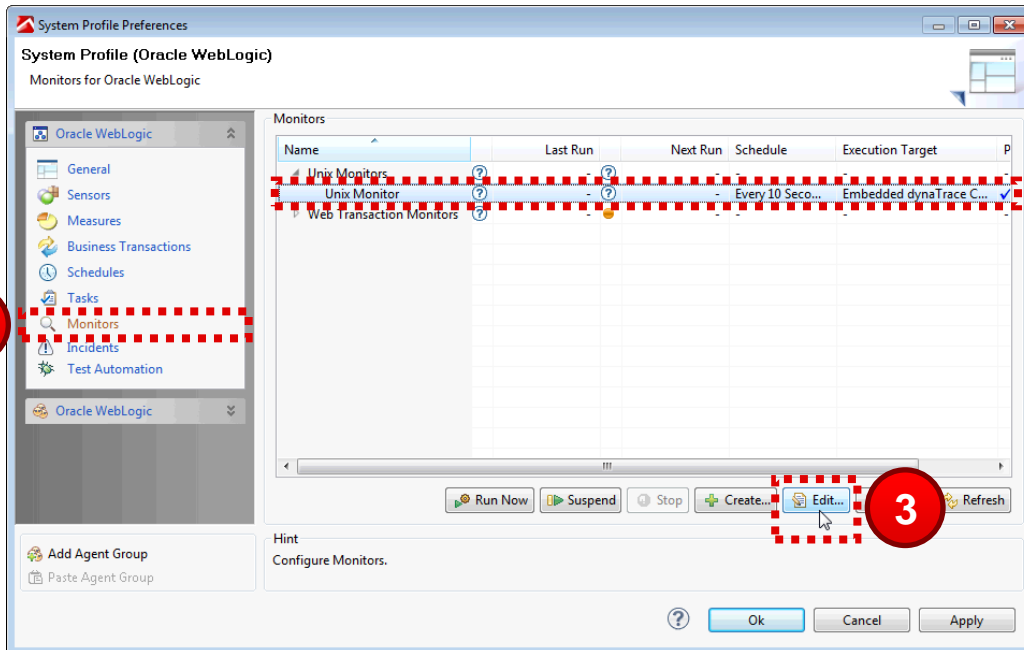


### 4.3.1.2 Operating System Monitoring Konfiguration

Das Referenz System Profil beinhaltet auch eine Monitoring Komponente für das Betriebssystem, auf welchem der Oracle WebLogic Application Server betrieben wird. Zurzeit werden mit dieser Version des FastPacks ausschliesslich UNIX Systeme unterstützt.

Weitere Informationen über das UNIX System Monitoring Plugin sind unter folgendem Link verfügbar:  
<https://community.dynatrace.com/community/display/DOCDT35/Unix+System+Monitoring>

In den Eigenschaften des System Profils können unter der Kategorie „Monitors“ die aktiven Monitoring Plugins verwaltet werden. Über die „Edit...“ Schaltfläche können danach die Einstellungen des Unix Monitoring Plugins angepasst werden:



Connection Settings:  
Diese Einstellungen definieren die grundlegenden Verbindungseigenschaften

Hosts:  
In der Host Liste, kann der Oracle WebLogic Server eingetragen werden.

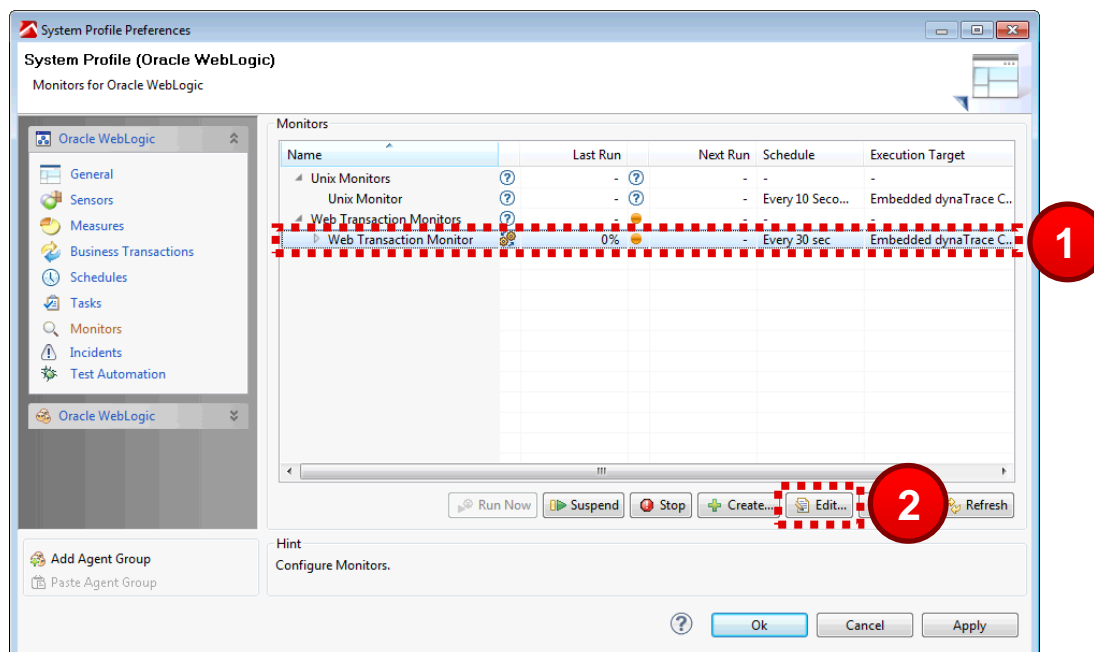
### 4.3.1.3 Synthetic Transaction Monitor

Im Refrenz Profil ist ein vorkonfiguriertes Monitoring Plugin aktiv, welches synthetische Web Transaktionen erzeugt und deren Performance (Response Time) misst. Gedacht ist, dass hier eine synthetische Transaktion definiert wird, welcher den Verfügbarkeitsstatus des Systems aus der End User Perspektive repräsentiert. Ein Beispiel hierfür könnte sein, den Verfügbarkeitsstatus der Login Seite mittels einem URL Check zu überprüfen. Auch komplexere Abläufe können mit diesem Plugin umgesetzt werden.

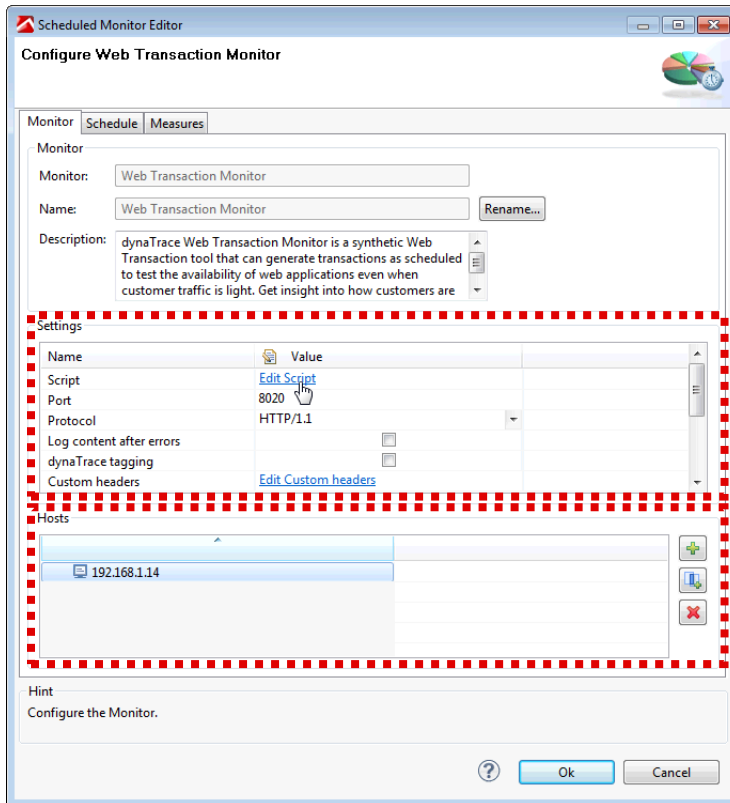
Weitere Informationen über das Web Transaction Monitoring Plugin sind unter folgendem Link verfügbar:

<https://community.dynatrace.com/community/display/DOCDT35/Web+Transaction+Monitor>

In den Eigenschaften des System Profils können unter der Kategorie „Monitors“ die aktiven Monitoring Plugins verwaltet werden. Über die „Edit...“ Schaltfläche können danach die Einstellungen des Web Transaction Monitoring Plugins angepasst werden:



In den Eigenschaften des Plugins kann nun definiert werden, welches Script für die synthetische Transaktion ausgeführt werden soll.  
Zudem wird festgelegt, auf welche Hosts die synthetischen Transaktionen angewendet wird:



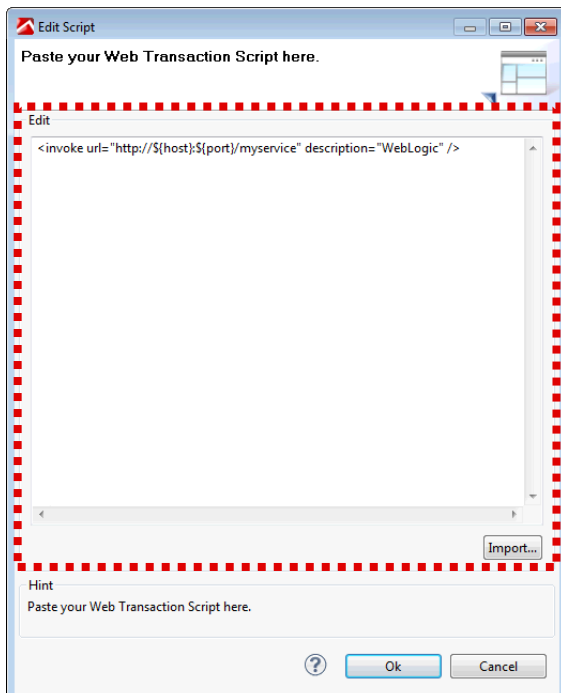
3

Connection Settings:  
Diese Einstellungen  
definieren die grundlegenden  
Verbindungseigenschaften

4

Hosts:  
In der Host Liste kann der  
Oracle WebLogic Server  
eingetragen werden.

Über den „Edit Script“ Link im Settings Panel kann der Aufbau der synthetischen Transaktion definiert werden. Als Standard wird eine einfache URL aufgerufen:



5

Edit:  
Script für die synthetische  
Web Transaktion.

### 4.3.2 Dashboards

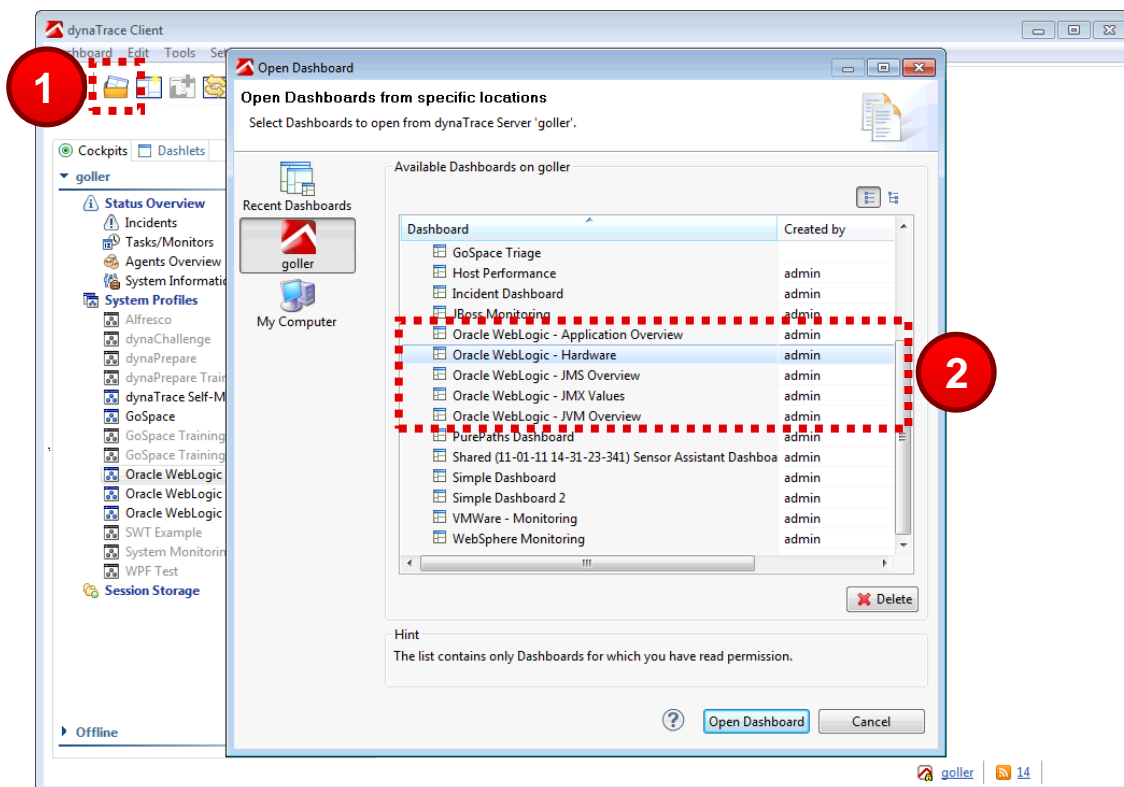
Zum Umfang des dynaTrace Oracle WebLogic FastPacks gehören fünf Referenz Dashboards:

- Oracle WebLogic – Application Overview
- Oracle WebLogic – Operating System
- Oracle WebLogic – WebLogic Health
- Oracle WebLogic – JMS
- Oracle WebLogic – JVM

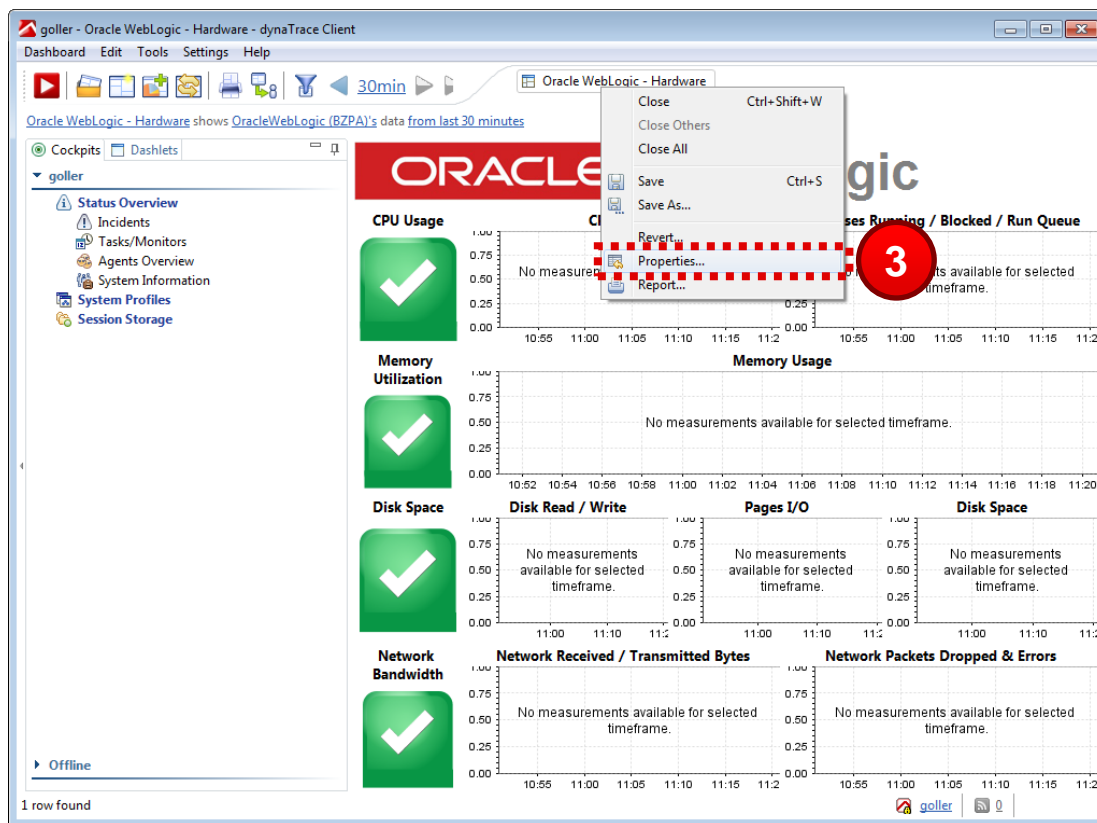
Damit diese Dashboards korrekt arbeiten, müssen sie einem System Profil zugeordnet werden. Standardmässig beziehen die Referenz Dashboards ihre Werte aus dem Referenz Profil. Wird in einer komplexen Umgebung mit mehreren Oracle WebLogic Application Servern gearbeitet, empfehlen wir, die Dashboards analog zu den System Profilen zu halten und entsprechende Kopien für die unterschiedlichen Systeme bzw. Oracle WebLogic Application Server Instanzen anzulegen.

Folgende Schritte sind für sämtliche Referenz Dashboards auszuführen, falls nicht direkt mit dem Referenz Profil selbst gearbeitet wird, sondern unterschiedliche WebLogic Application Server Instanzen verwendet werden.

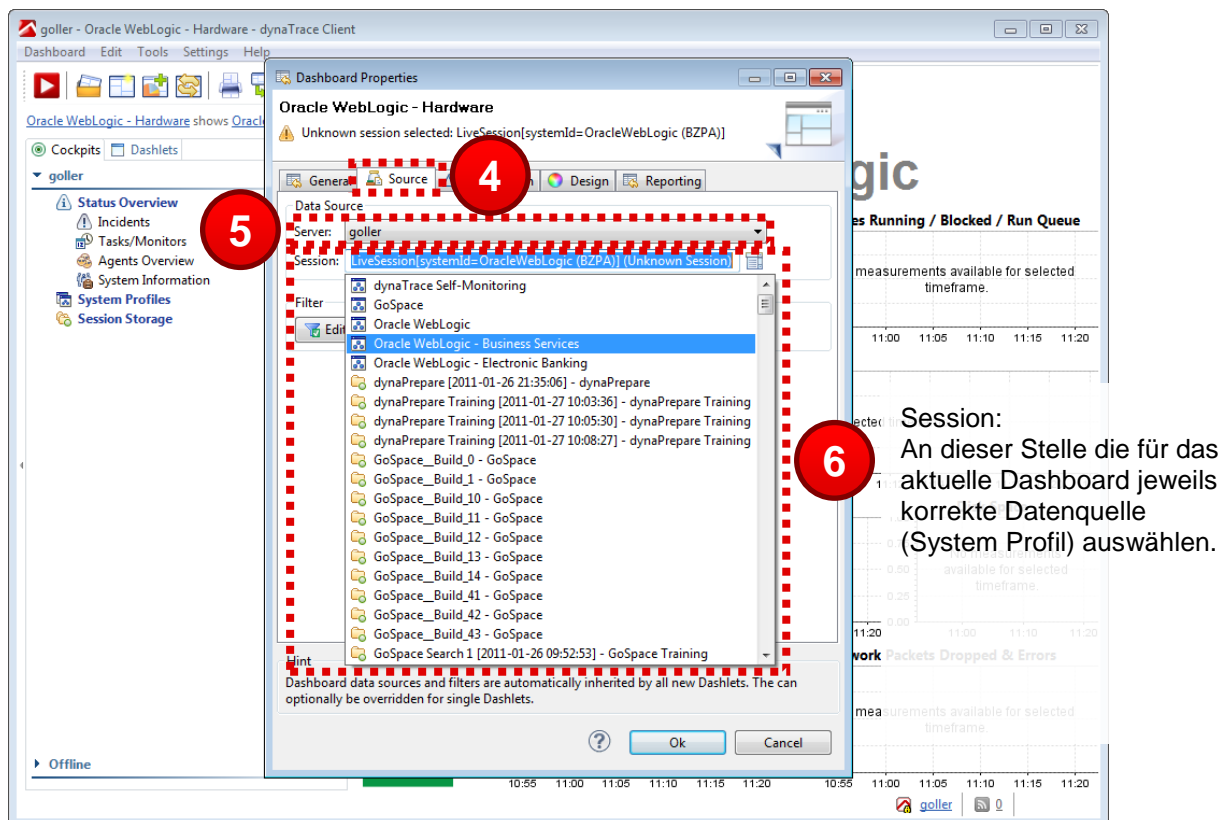
Referenz Dashboard öffnen:



Über das Kontext Menü können die Eigenschaften des jeweiligen Dashboards aufgerufen werden:



Danach die Einstellungen bezüglich der Datenquelle an das jeweilige SystemProfil anpassen:



## 5 Erweiterung des FastPacks: Zusätzliche Oracle WebLogic JMX Werte

Dieses Kapitel soll die Möglichkeiten aufzeigen, wie zusätzliche Metriken aus dem Oracle WebLogic Application Server in dynaTrace angezeigt und ausgewertet werden können.

Grundsätzlich bietet der Oracle WebLogic Application Server eine Vielzahl unterschiedlicher Messwerte, welche via JMX (MBeans) abgefragt werden können. Einige dieser Messwerte können ohne zusätzlichen Konfigurationsaufwand in dynaTrace angezeigt werden. Diese Messwerte und deren Anwendung werden in Kapitel 5.1 (Standard Messwerte) behandelt.

In Kapitel 5.2 (Benutzerdefinierte Messpunkte) wird auf den Vorgang eingegangen, wie Messwerte aus dem Oracle WebLogic Application Server ausgelesen werden können, welche nicht zu den vorkonfigurierten Defaults und dem FastPack von dynaTrace gehören.

### 5.1 Standard Messwerte

Zur dynaTrace Installation gehören bereits vorkonfigurierte Messpunkte für den Oracle WebLogic Application Server. Die zurückgelieferten Messwerte sind jeweils als „Total“ (über den ganzen Application Server) zu verstehen, da diese noch nicht auf die Ebene einzelner Instanzen (z.B. DataSources) aufgeteilt sind. Sollen die Messpunkte weiter spezifiziert werden, z.B. um einen Wert einer einzelnen Instanz eines Objekts abzugreifen, empfehlen wir eigene Messpunkte (siehe Kapitel 5.2.2) anzulegen.

Folgende Standard-Messwerte gehören zum Installationsumfang von dynaTrace:

WebLogic 8:

- EJB Pool Access Total Count WebLogic
- EJB Pool Beans In Use Current Count WebLogic
- EJB Pool Miss Total Count WebLogic
- EJB Pool Pooled Beans Current Count WebLogic
- EJB Pool Timeout Total Count WebLogic
- Execute Queue Execute Thread CurrentIdle Count WebLogic
- Execute Queue Execute Thread TotalCount WebLogic
- Execute Queue Pending Request Current Count WebLogic
- Execute Queue Serviced Request Total Count WebLogic
- JDBC Connection Pool Active Connections Average Count WebLogic
- JDBC Connection Pool Active Connections Current Count WebLogic
- JDBC Connection Pool Active Connections High Count WebLogic
- JDBC Connection Pool Connection Delay Time WebLogic
- Thread Pool Execute Thread Idle Count WebLogic
- Thread Pool Execute Thread Total Count WebLogic
- Thread Pool Hogging Thread Count WebLogic
- Thread Pool Pending User Request
- Thread Pool Throughput WebLogic

WebLogic 9+:

- EJB Pool Access Total Count WebLogic
- EJB Pool Beans In Use Current Count WebLogic
- EJB Pool Miss Total Count WebLogic
- EJB Pool Pooled Beans Current Count WebLogic
- EJB Pool Timeout Total Count WebLogic
- Execute Queue Execute Thread CurrentIdle Count WebLogic
- Execute Queue Execute Thread TotalCount WebLogic
- Execute Queue Pending Request Current Count WebLogic
- Execute Queue Serviced Request Total Count WebLogic
- JDBC Connection Pool Active Connections Average Count WebLogic
- JDBC Connection Pool Active Connections Current Count WebLogic



- JDBC Connection Pool Active Connections High Count WebLogic
- JDBC Connection Pool Connection Delay Time WebLogic
- Thread Pool Execute Thread Idle Count WebLogic
- Thread Pool Execute Thread Total Count WebLogic
- Thread Pool Hogging Thread Count WebLogic
- Thread Pool Pending User Request
- Thread Pool Throughput WebLogic

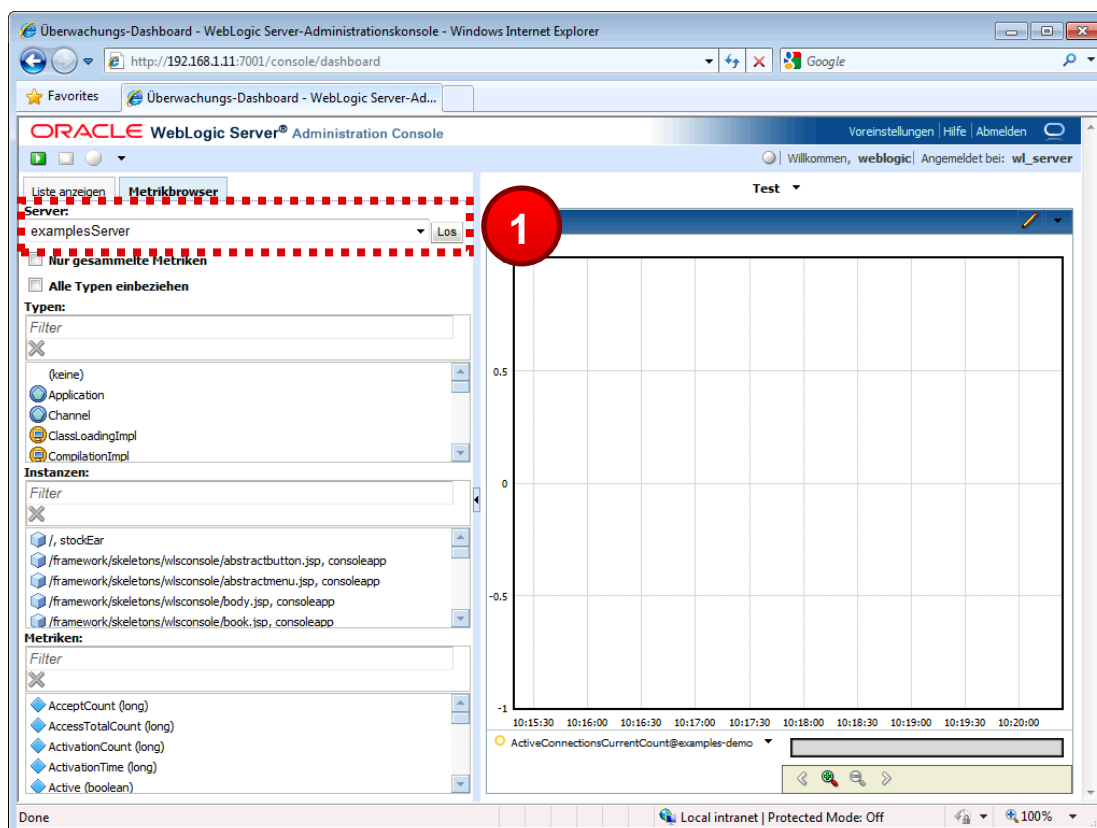
## 5.2 Benutzerdefinierte Messpunkte

Im Folgenden wird anhand eines praktischen Beispiels aufgezeigt, welche Schritte benötigt werden, um einen benutzerspezifischen Messpunkt in dynaTrace zu erstellen. Das vorgestellte Verfahren kann auf beliebige Metriken des Oracle WebLogic Application Servers angewendet werden.

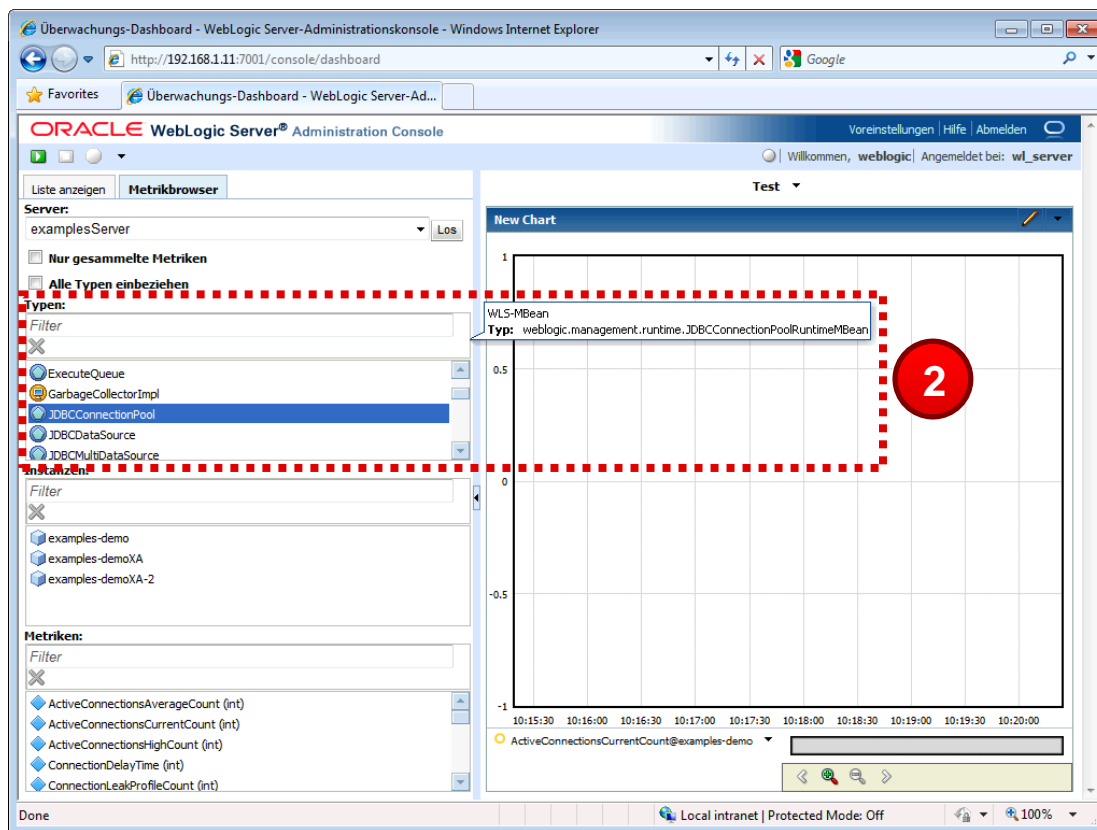
### 5.2.1 Oracle WebLogic Server Administration Console: Metrik identifizieren

In einem ersten Schritt wird die gewünschte Metrik in der Oracle WebLogic Server Administration Console ausfindig gemacht und deren Wert kurz mittels Chart überprüft. Ziel dabei ist es, die gewünschte Metrik isolieren und identifizieren zu können. Folgende Schritte zeigen auf, wie dies gemacht werden kann:

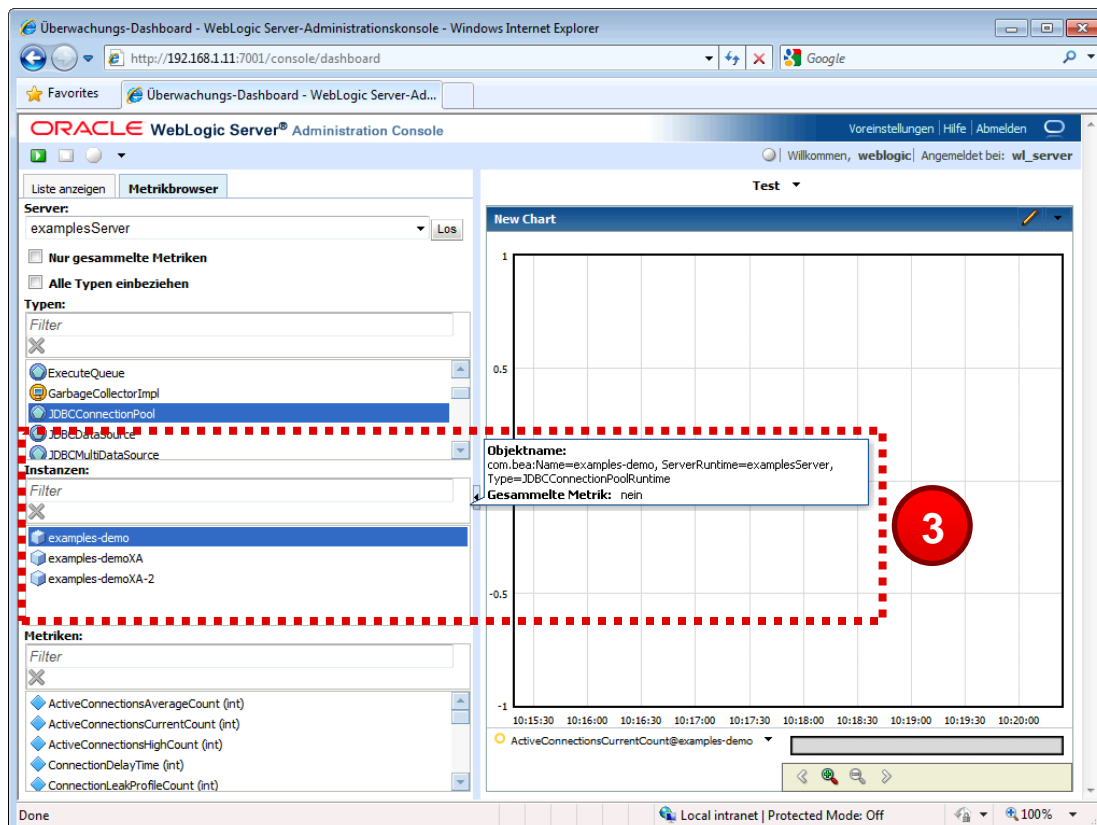
Server wählen (Beispiel: “examplesServer”):



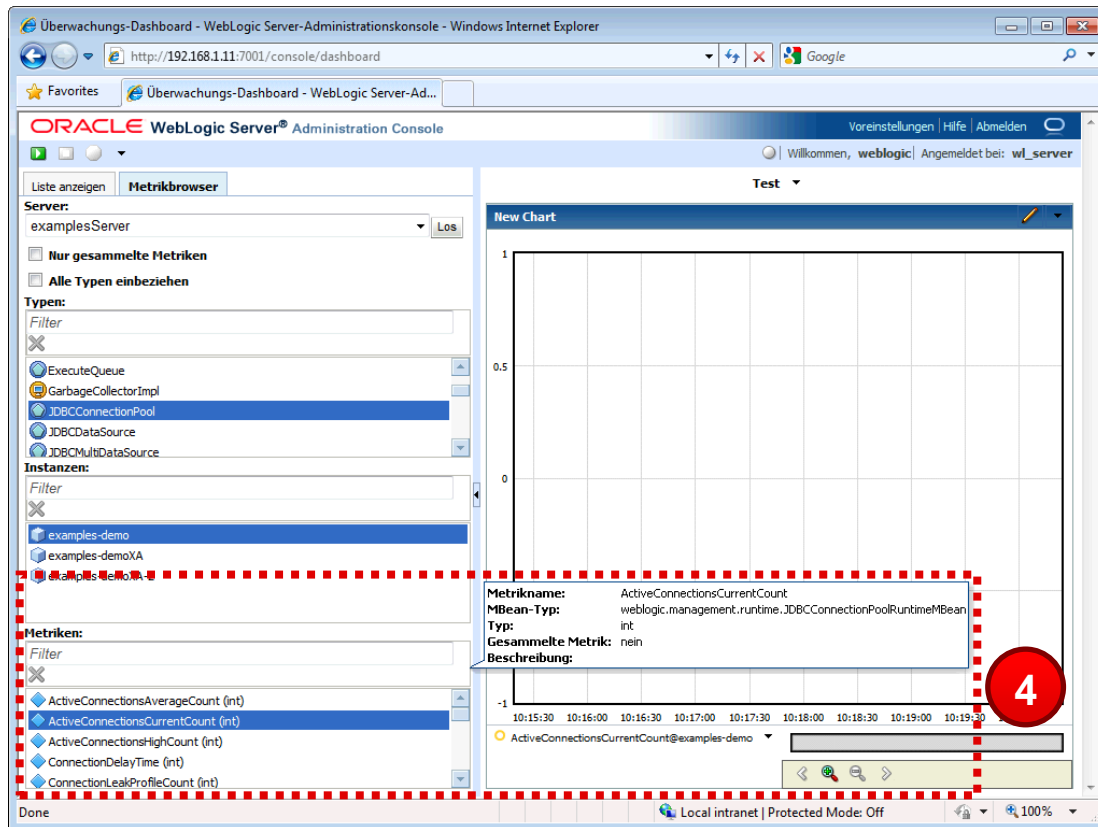
Typ wählen (Beispiel: "JDBCConnectionPool"):



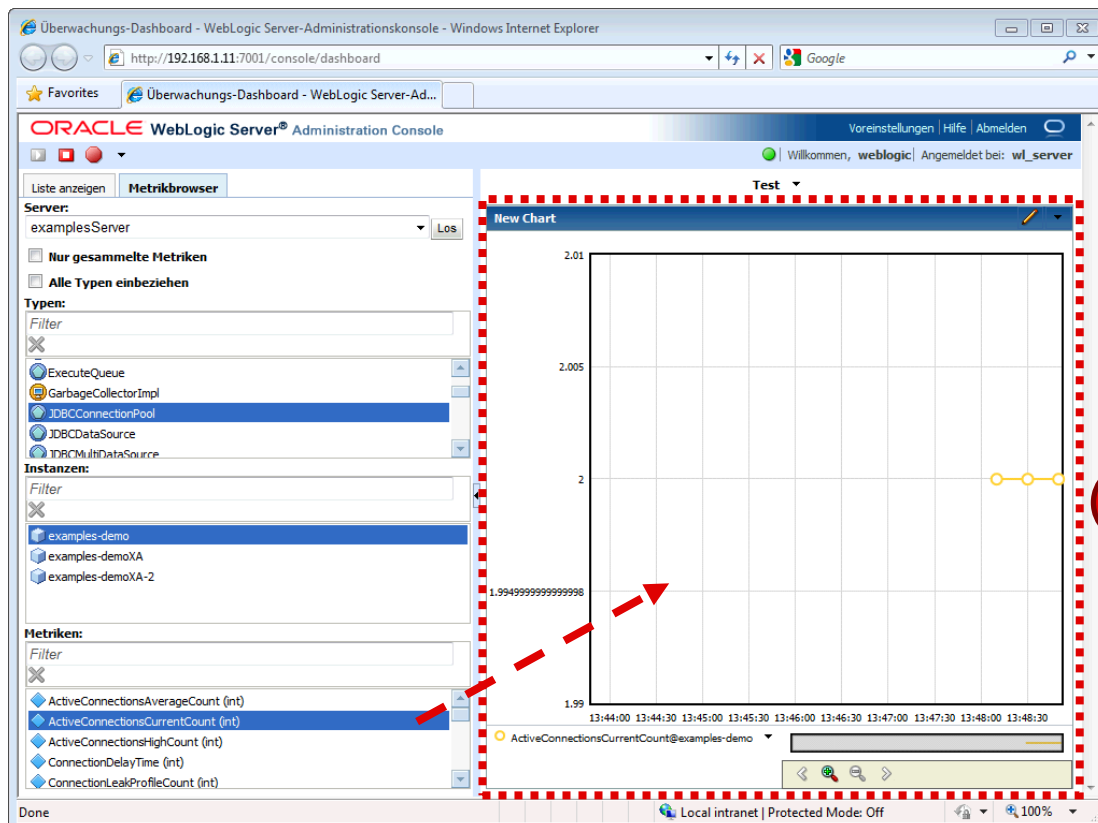
Instanz wählen (Beispiel: "examples demo"):



Metrik wählen (Beispiel: "ActiveConnectionsCurrentCount"):



Wert der Metrik überprüfen. Dazu die jeweilige Metrik mittels Drag & Drop dem Chart hinzufügen:



Wurde die gewünschte Metrik dem Chart zugewiesen, kann im ToolTip deren vollwertiger JMX Pfad eingesehen werden.

Beispiel der relevanten Werte aus untenstehender Grafik:

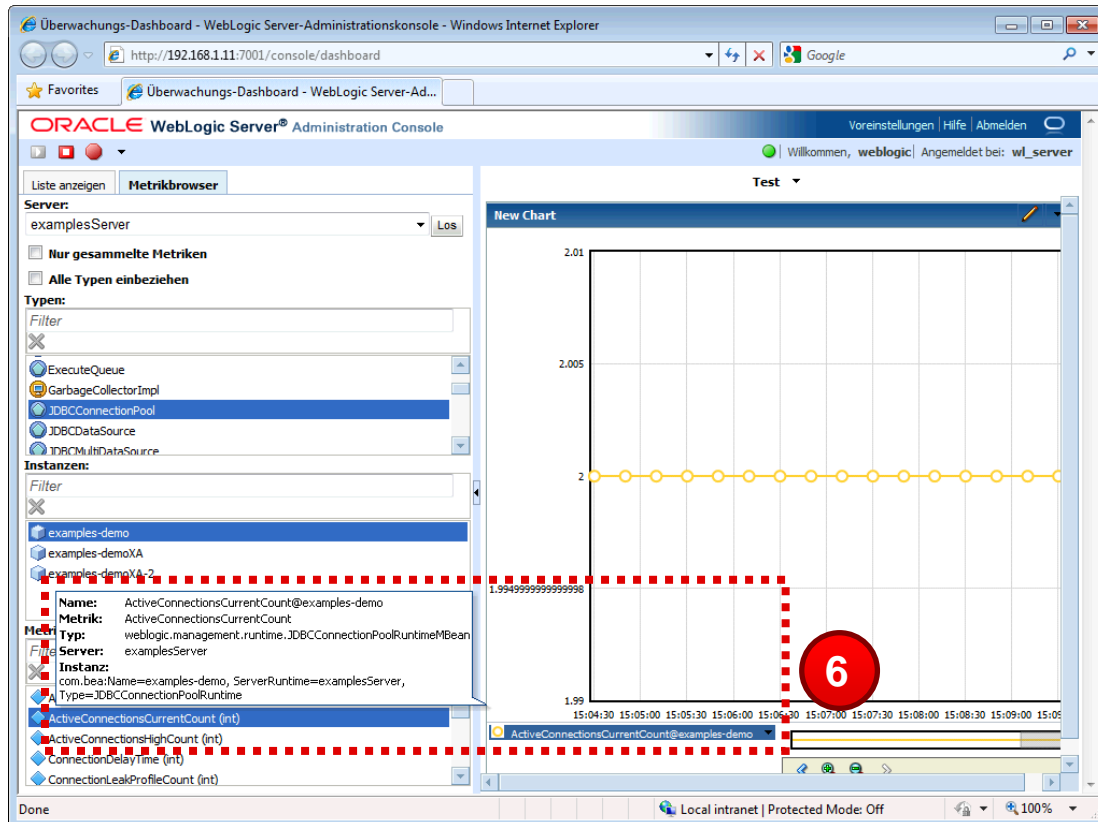
Metrik: [ActiveConnectionsCurrentCount](#)

Typ: [weblogic.management.runtime.JDBCConnectionPoolRuntimeMBean](#)

Server: [examplesServer](#)

Instanz: [examples-demo](#)

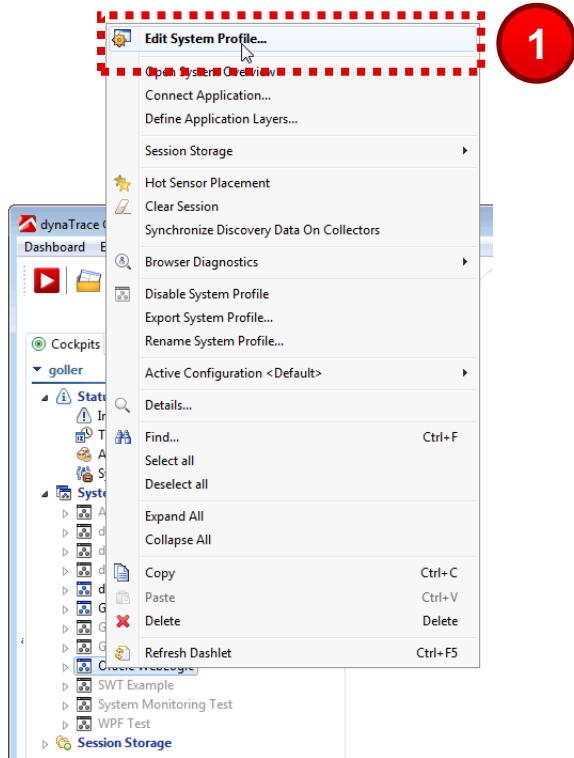
Diese vier Werte werden im anschliessenden Kapitel 5.2.2 (Metrik in dynaTrace erfassen) wieder verwendet.



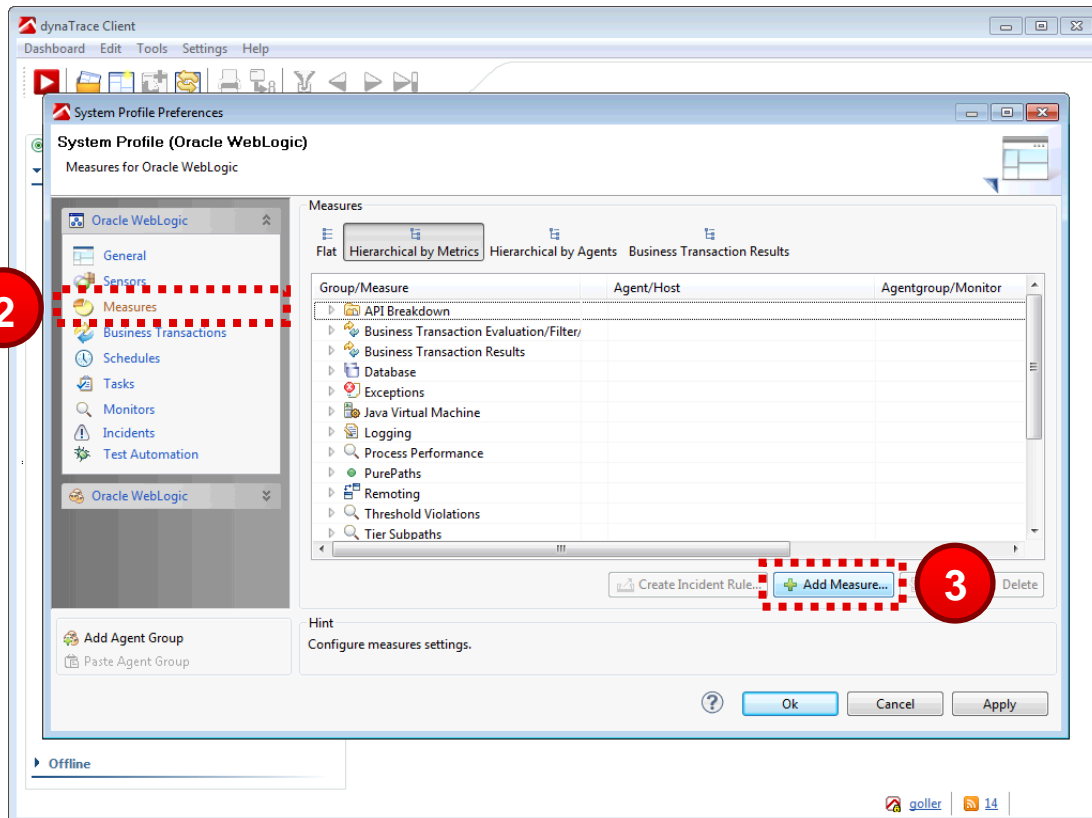
## 5.2.2 Metrik in dynaTrace erfassen

Aus den zusammengetragenen Werten des vorherigen Abschnitts, kann nun der Pfad bestimmt werden, in welchem die Metrik innerhalb der MBean Baumstruktur zu finden ist. Mittels folgenden Schritten ist es möglich, die vorhin innerhalb der Oracle Management Console identifizierte Metrik in dynaTrace abzubilden.

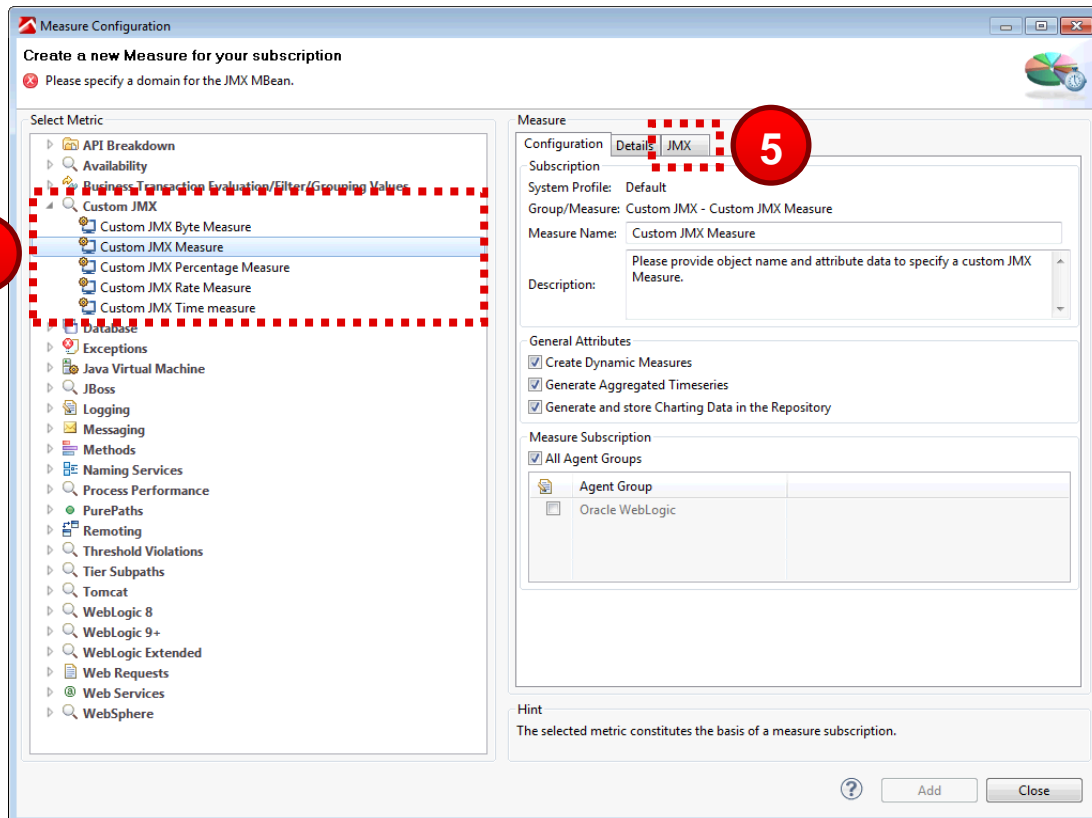
Als Erstes das entsprechende System Profil bearbeiten:



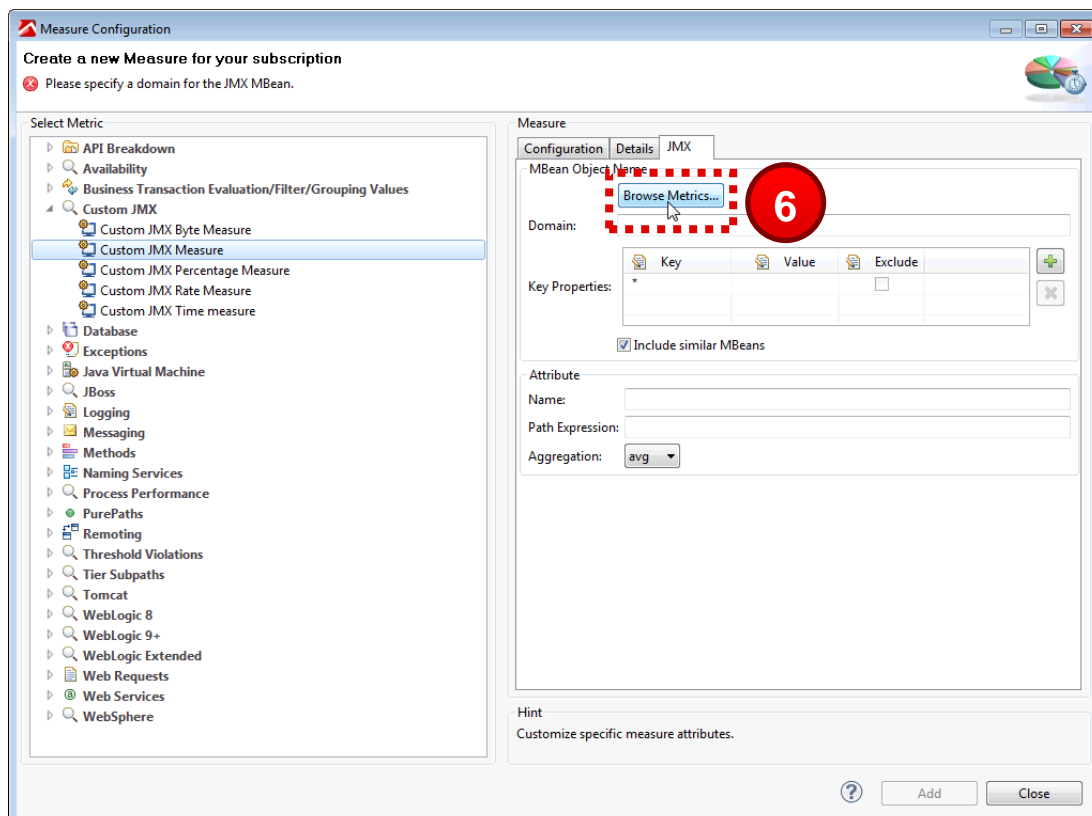
Danach unter der Kategorie „Measures“ mittels der „Add Measure...“ Schaltfläche ein neues Measure hinzufügen:



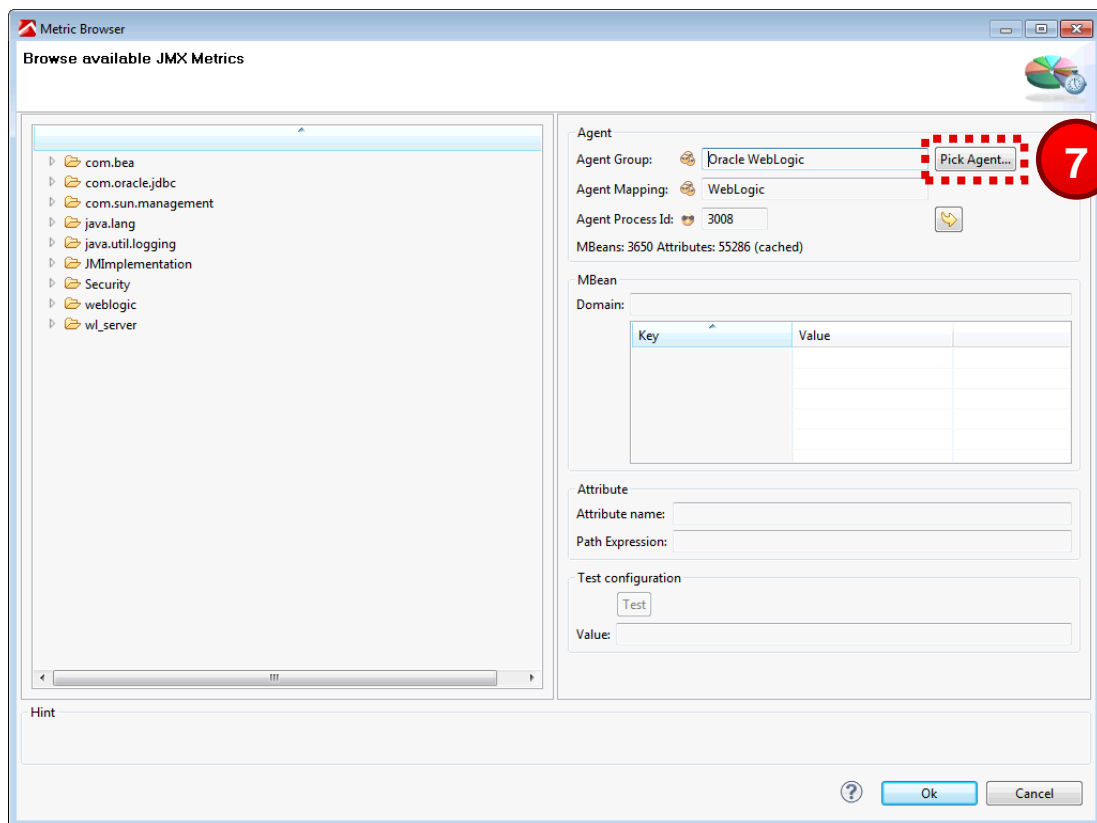
Im Konfigurationsdialog auf der rechten Seite unter „Custom JMX“ das Measure selektieren, welches mit dem Datentyp des JMX Wertes übereinstimmt:



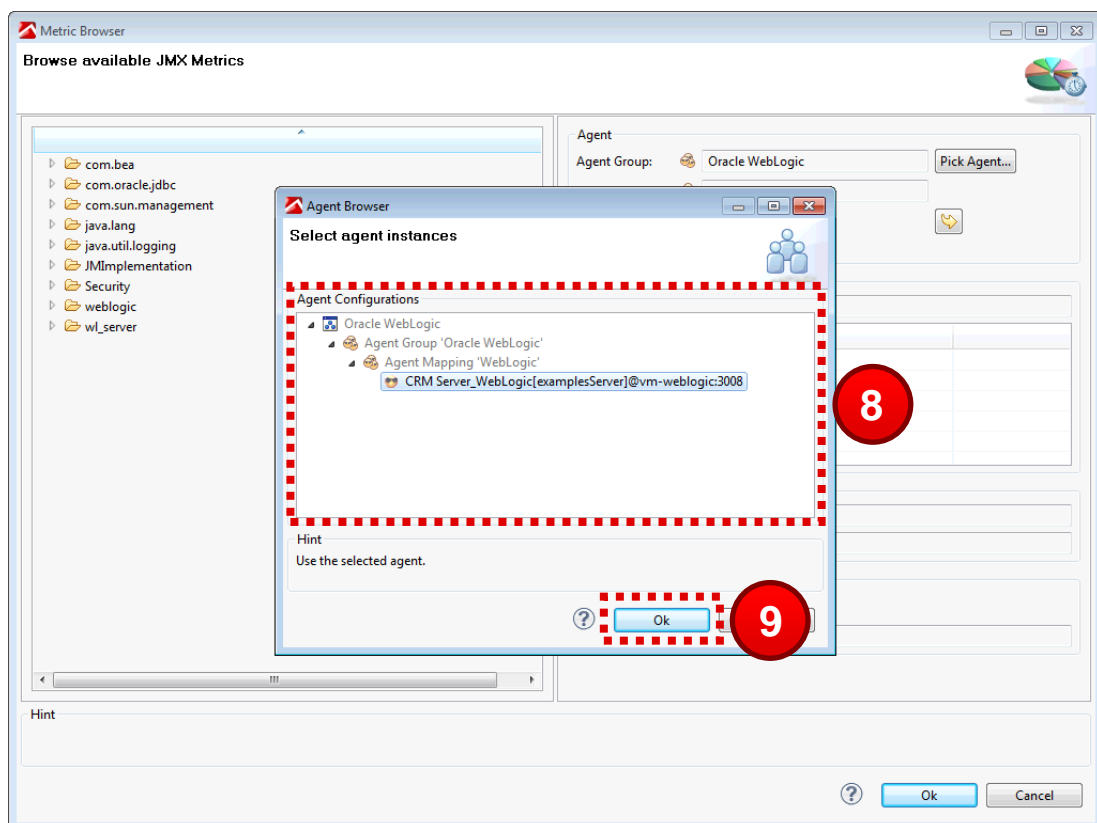
Über die „Browse Metrics...“ Schaltfläche den dynaTrace Metric Browser öffnen:



Als Nächstes muss der zum System Profil gehörende dynaTrace Agent gewählt werden:



Agent selektieren:



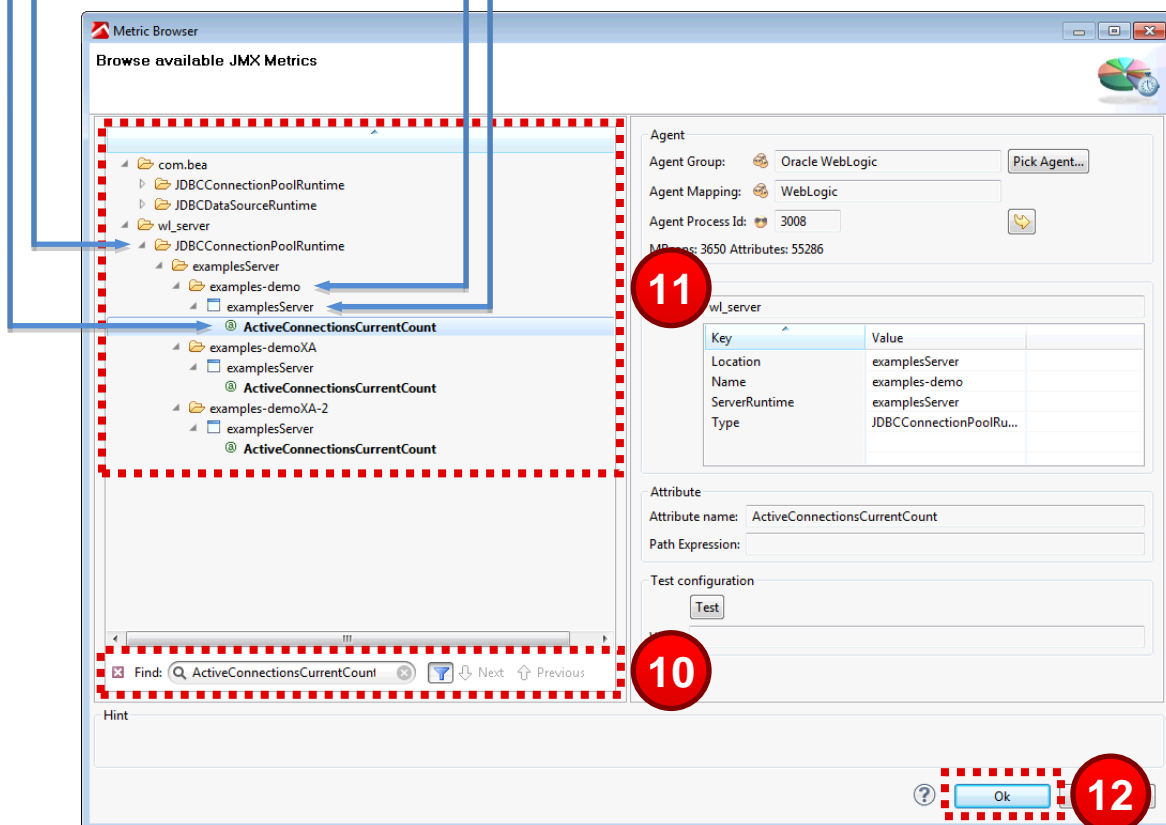


Auf der linken Seite werden nun in der Baumstruktur sämtliche verfügbare MBeans aufgelistet. Über das Suchfenster unten links kann zudem auch nach beliebigen Werten (z.B. [ActiveConnectionsCurrentCount](#)) gesucht werden.

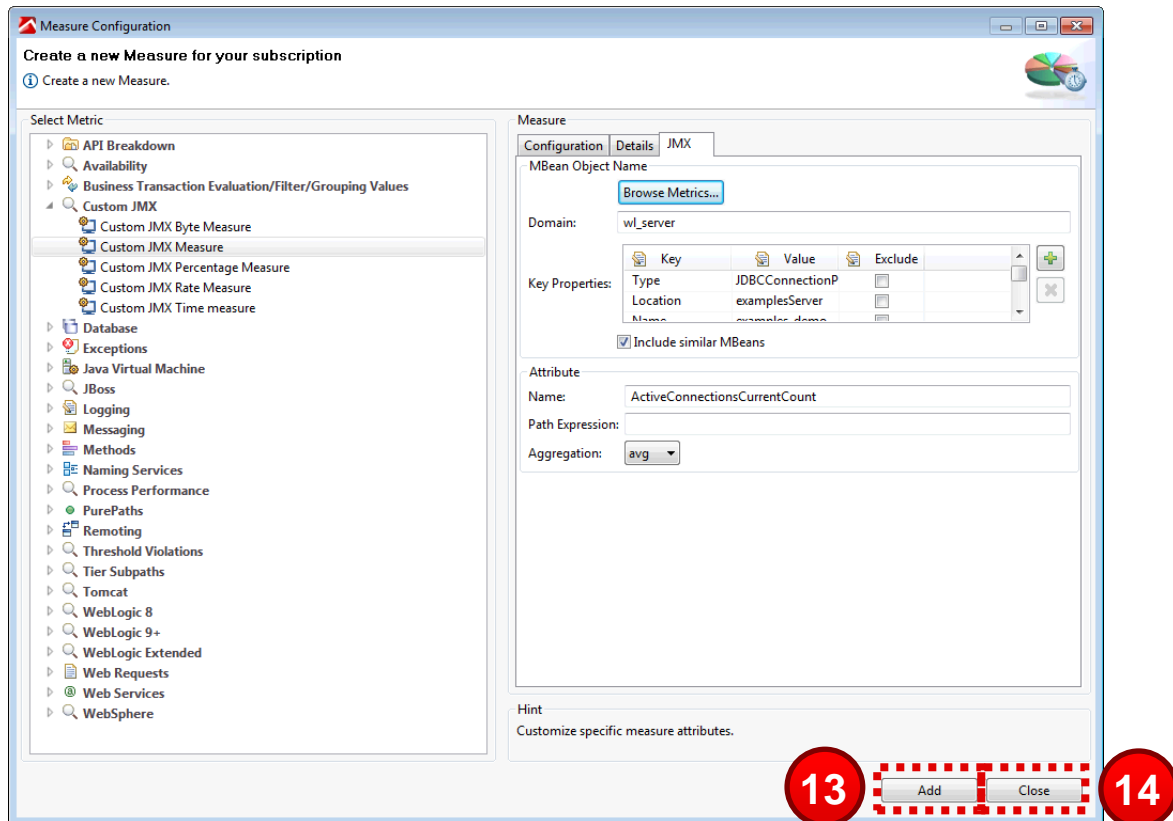
An dieser Stelle können die gesammelten Werte aus der Oracle WebLogic Console verwendet werden um die korrekte Metrik zu finden.

Beispiel:

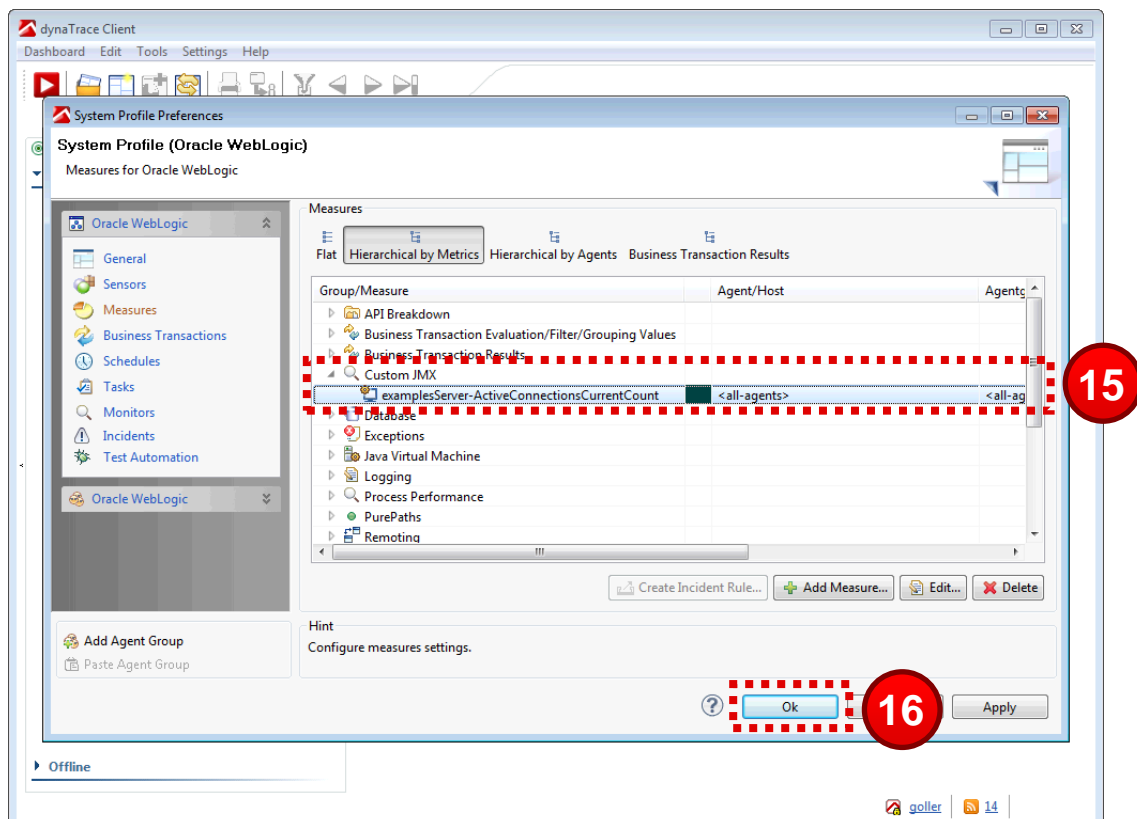
Metrik: [ActiveConnectionsCurrentCount](#)  
Typ: [weblogic.management.runtime.JDBCConnectionPoolRuntimeMBean](#)  
Server: [examplesServer](#)  
Instanz: [examples-demo](#)



Im Measure Configuration Dialog nochmals kurz die ausgefüllten Werte überprüfen und das neue Measure mittels Klick auf „Add“ erstellen. Danach den Dialog mittels „Close“ schließen:



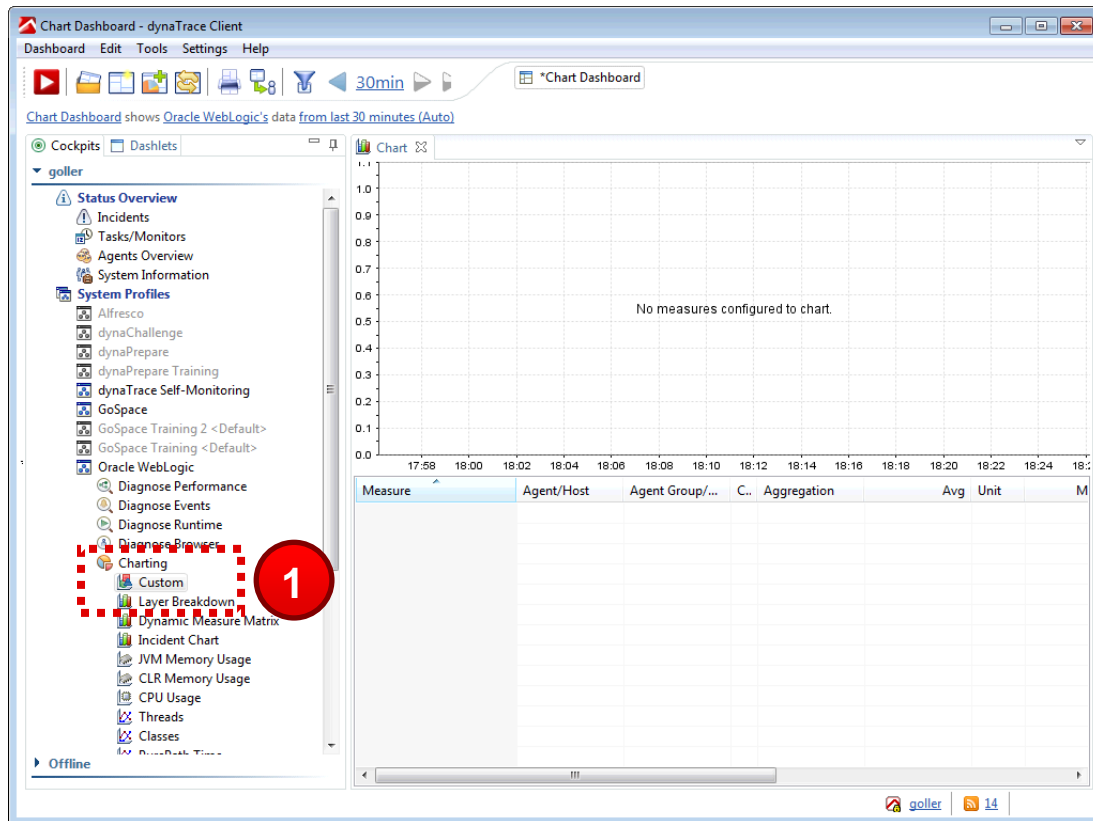
Das neu angelegte Measure ist nun unter der Gruppe „Custom JMX“ ersichtlich:



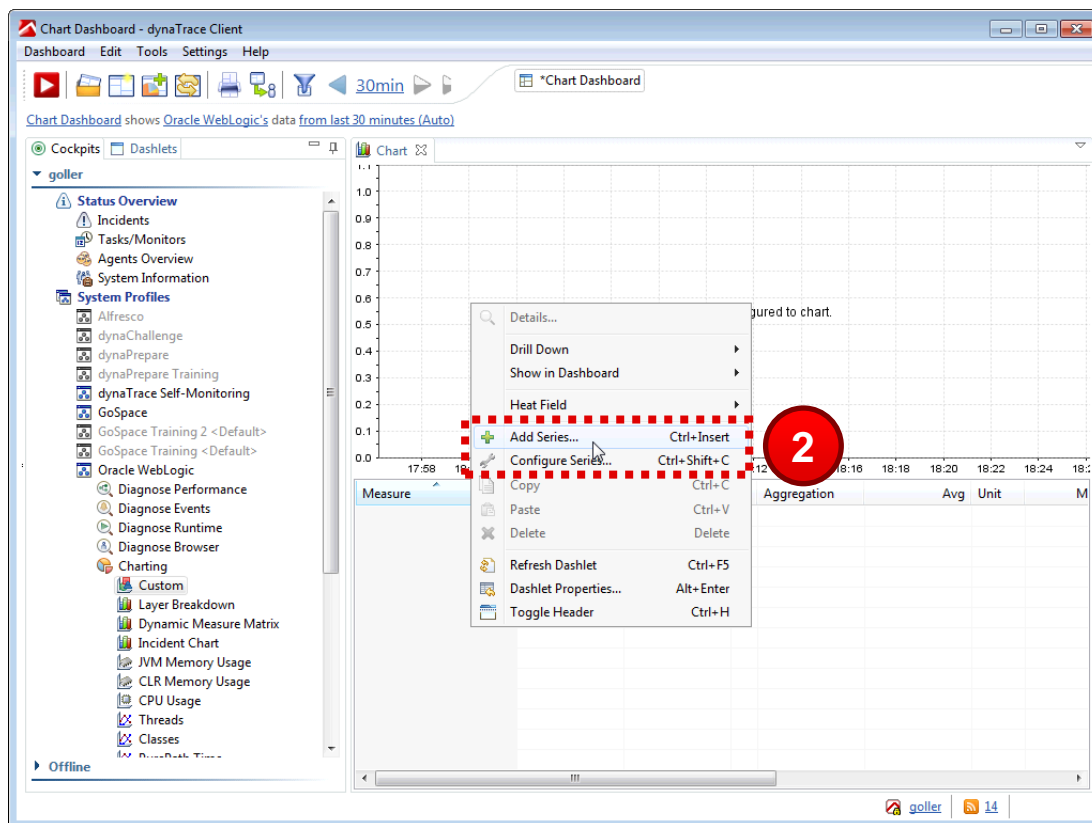
### 5.2.3 Visualisierung von Oracle WebLogic Metriken

Das im vorherigen Kapitel angelegte Measure kann mit wenig Aufwand auf beliebigen Dashboards visualisiert werden. Im folgenden Beispiel wird dazu ein leeres Dashboard verwendet, allerdings steht an dieser Stelle die Option offen, auch die zum dynaTrace WebLogic FastPack gehörenden Referenz Dashboards beliebig zu erweitern.

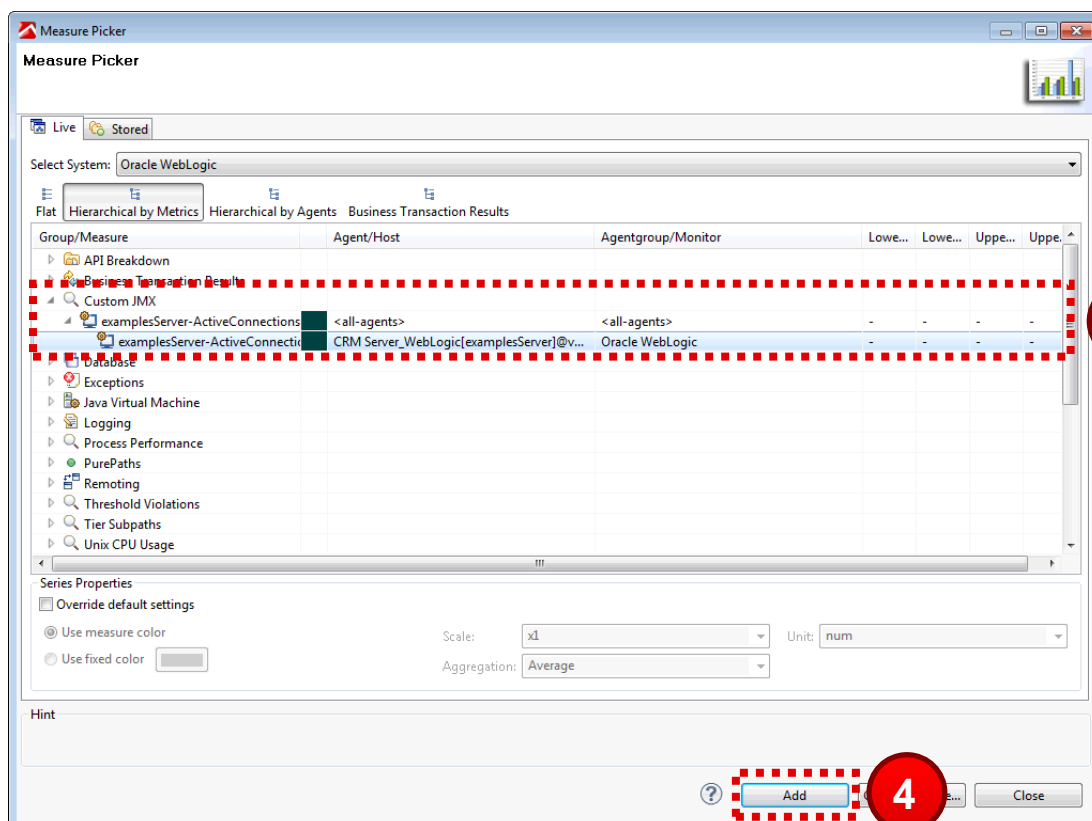
In einem ersten Schritt wird mittels Doppelklick ein neues Custom Chart dem Dashboard hinzugefügt:



Anschließend kann mittels Kontext Menü eine neue „Series“ hinzugefügt werden:



Im Measure Picker Dialog ist das im vorherigen Kapitel angelegte Measure nun unter der Gruppe „Custom JMX“ verfügbar und kann selektiert werden:



Nach dem Hinzufügen, ist die Oracle WebLogic Metrik im dynaTrace Dashboard ersichtlich:

