

# Utilisation de CA APM Trucs et astuces

**Saad Ouchkir**

Sr. Consultant, Technical Sales

agility  
made possible™



# Agenda

- Amélioration de l'instrumentation (Java EE / .NET)
- Analyse de la consommation mémoire
- Couverture des environnements non Java EE / .Net
- Fonctionnalités de l'Enterprise Manager

# Légende pour les fonctionnalités listées dans ce document

## Fonctionnalités Standards

Testées et supportées par le processus de support standard de CA Technologies

## Offre Global Delivery

Fonctionnalités additionnelles et payantes supportées par CA Technologie

## Field Pack

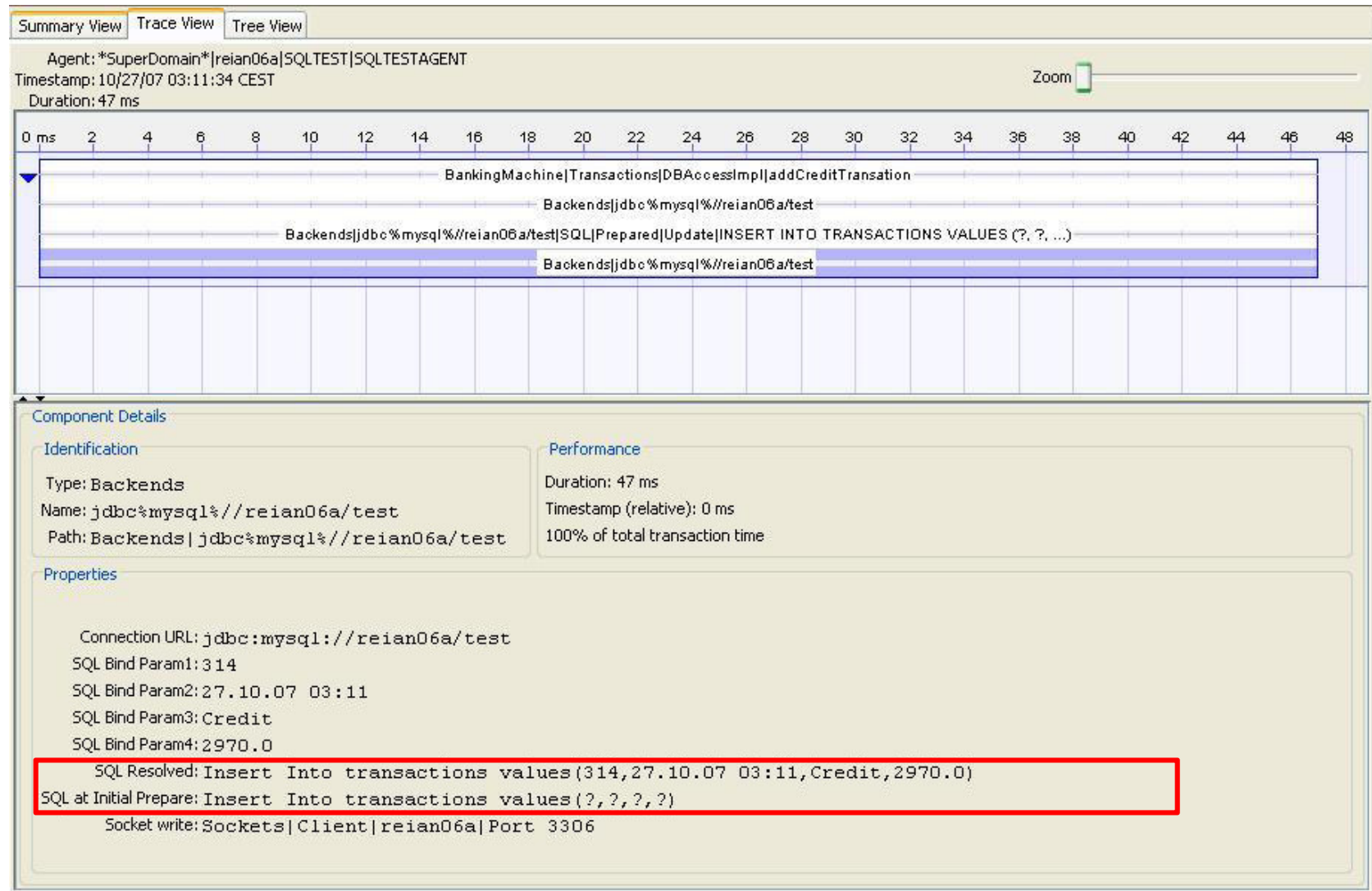
Fonctions réalisées dans le cadre d'une implémentation CA et dont le support doit être pris en compte au travers d'une prestation CA Services

# Amélioration de l'instrumentation (Java EE / .NET)

- Ajoute les paramètres de connexion SQL aux Transaction Traces
- Etend les métriques SQL standards
- Idéal pour les environnements de QA où il est important de voir les arguments des requêtes SQL
- Doit être utilisé uniquement pour des besoins de diagnostic

# SQL Agent With Params

Offre  
Global Delivery



- PowerPack qui inclut une instrumentation spécialisée pour Microsoft SharePoint
- S'appuie sur l'agent .Net
- Calcule les temps de réponse pour les Web Parts, Services etc.

# Microsoft SharePoint

Fonctionnalité  
Standard

## SharePoint Overview



Overview

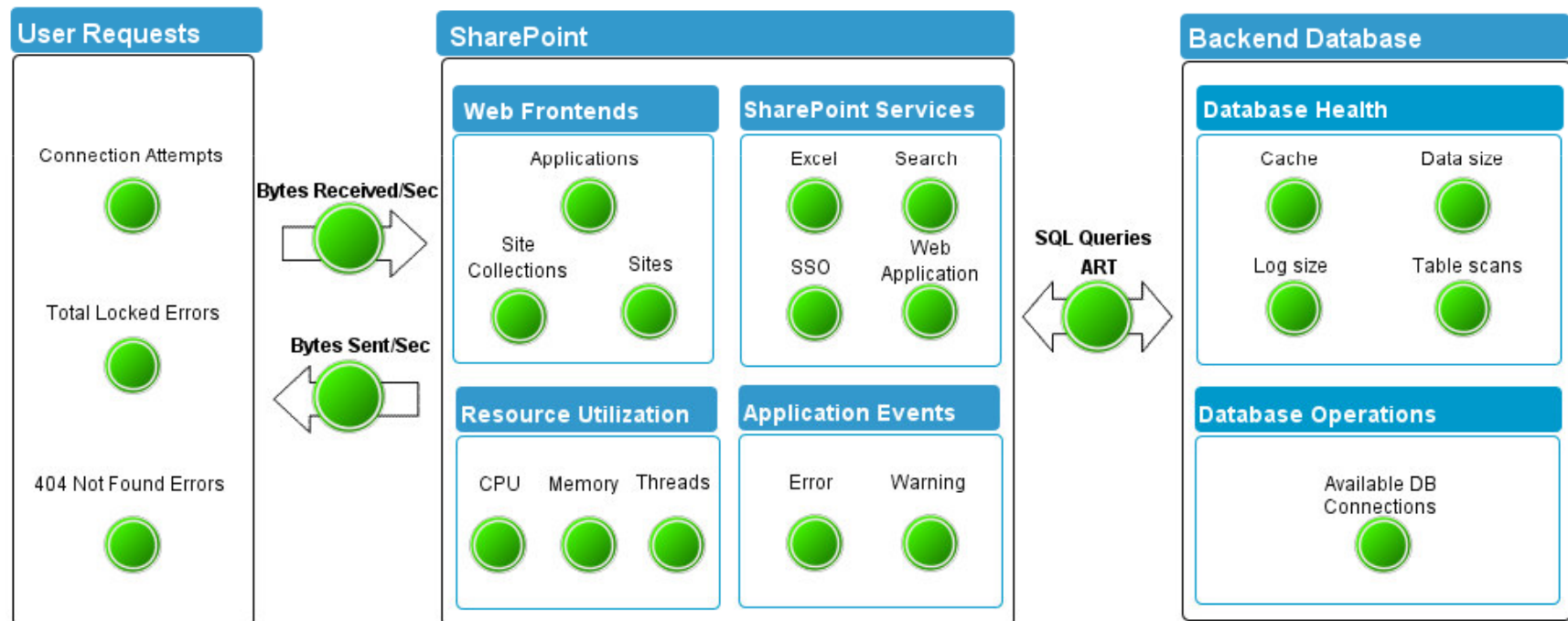
IIS Performance

Web Frontends

SharePoint Services

Database Operations

Backend





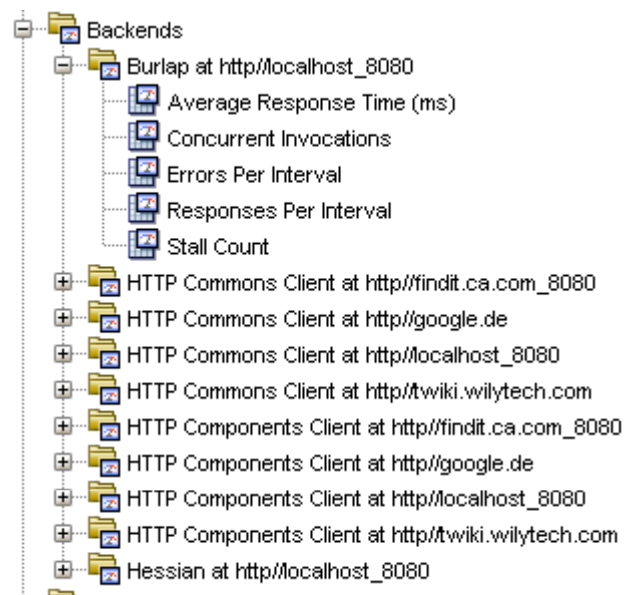
- 2 extensions de l'agent permettant un nommage intelligent des métriques
- Permet d'utiliser le résultat d'appel aux méthodes des objets passés en paramètre ou retournés par la méthode instrumentée
  - Exemple: Différencier les temps de réponse de la méthode « paint » de la classe « TextPane » en fonction de la police utilisée

```
TraceOneMethodWithParametersOfClass:TextPane paint BlamePointTracer_SGNF "Render | Text  
Data | {{data.getInvocationParameterAsObject(0).getFont().getFontName()}}"
```

- Extension de l'agent Introscope qui apporte une instrumentation des communications HTTP utilisant l'une des APIs client ci-dessous :
  - Caucho
    - Hessian Web Service Protocol [<http://hessian.caucho.com/>]
    - ☐ Burlap Web Service Protocol [<http://www.caucho.com/resin-3.0/protocols/burlap.xtp>]
  - Apache Commons HTTP Client [<http://hc.apache.org/>]
  - Apache Components HTTP Client [<http://hc.apache.org/>]
  - Spring HTTP Clients

# HTTP Communication Extension

Field Pack

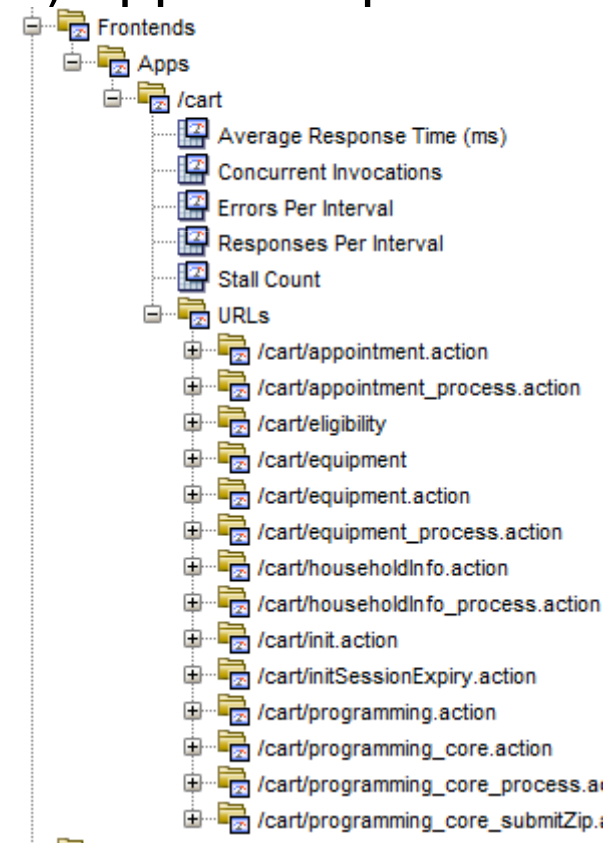


## Aperçu du nœud Backends dans l'investigateur

Component	R/T ▾	Concurr...	Errors	Respon...	Stalls
HTTP Commons Client at http/google.de	83979	1	0	4	1
HTTP Components Client at http/Mocalhost_8080	43641	2	0	10	1
HTTP Components Client at http/google.de	20992	1	0	4	0
HTTP Commons Client at http/Mocalhost_8080	4642	1	0	8	0
Hessian at http/Mocalhost_8080	4475	6	0	17	0
Burlap at http/Mocalhost_8080	4125	1	0	8	0
HTTP Components Client at http/twiki.wilytech.com	1535	1	0	4	0
HTTP Commons Client at http/twiki.wilytech.com	1144	1	0	4	0
HTTP Components Client at http/findit.ca.com_8080	90	1	0	4	0
HTTP Commons Client at http/findit.ca.com_8080	86	0	0	4	0

## Typeview montrant les Backends dans l'investigateur

- Extension de l'agent permettant d'instrumenter Struts2
- Permet d'identifier les use cases (actions) appelées par les utilisateurs
- Trace la méthode « execute » de la classe «org.apache.struts2.impl.StrutsActionProxy» en extrayant
  - Namespace
  - Nom de l'action
  - Méthode de l'action



- Un tracer léger qui ne génère pas de métriques, et se contente de peupler des Transaction Traces quand elles sont générées.
- Apporte une plus de visibilité sans polluer la SmartStor

## Exemple:

**SetFlag: DiagTracing**  
**TurnOn: DiagTracing**

**IdentifyMatchingClassesAs: com.company.\* DiagTracing**  
**TraceComplexMethodsIfFlagged: DiagTracing FullTTDetailsTracer "{packageandclassname}:{method}"**

# Analyse de la consommation mémoire

- LeakHunter est une extension de l'agent qui traque la taille des Collections Java
- Sa valeur ajoutée par rapport à de simple Heap Dumps :
  - Montre le graphe d'appel et identifie l'endroit où s'est fait l'allocation des objets
  - Montre graphiquement l'évolution de la taille des Collections
- Désactivé par défaut, doit être activé en modifiant le fichier IntroscopeAgent.profile

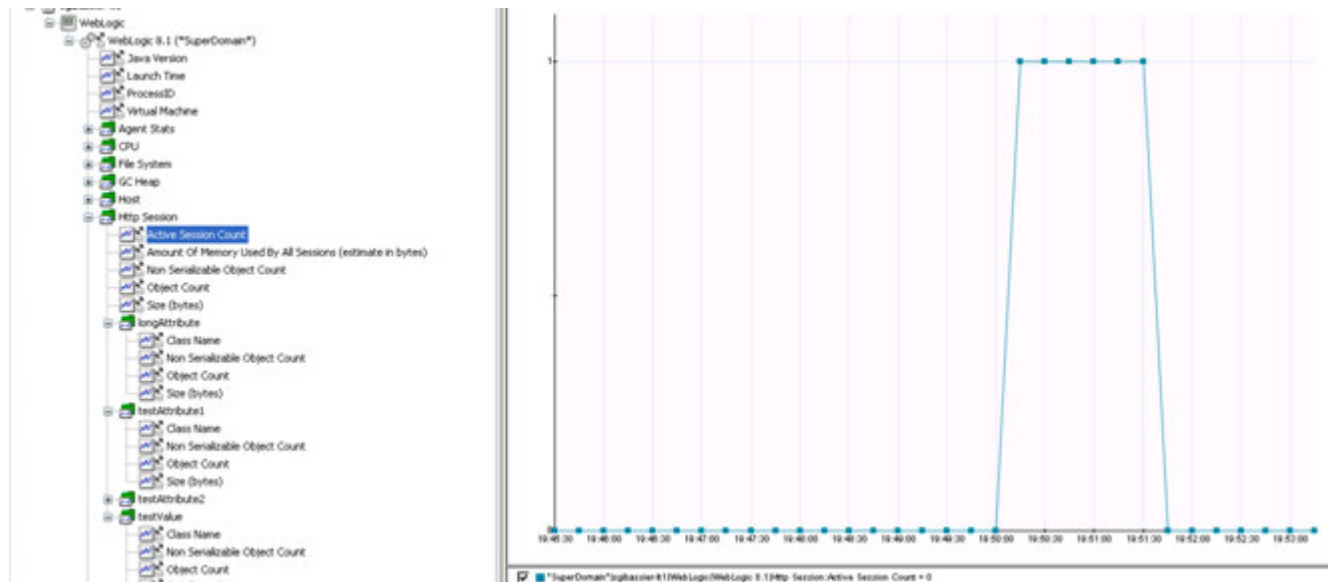
**Documentation: CA Wily Introscope Java Agent Guide (Chapter 12)**

- Un tracer standard qui permet de compter le nombre d'instance d'une classe.
- L'évolution de cette valeur s'affiche sous la forme d'une courbe dans l'Investigateur
  - Exemple : Compter le nombre d'instances de toutes les classes du package com.company

```
IdentifyMatchingClassesAs: com.company.* InstanceCounts
```



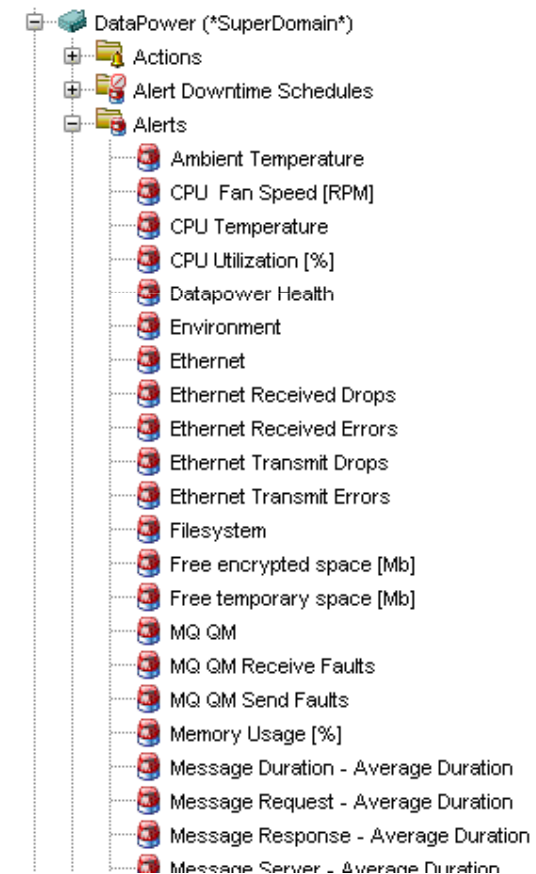
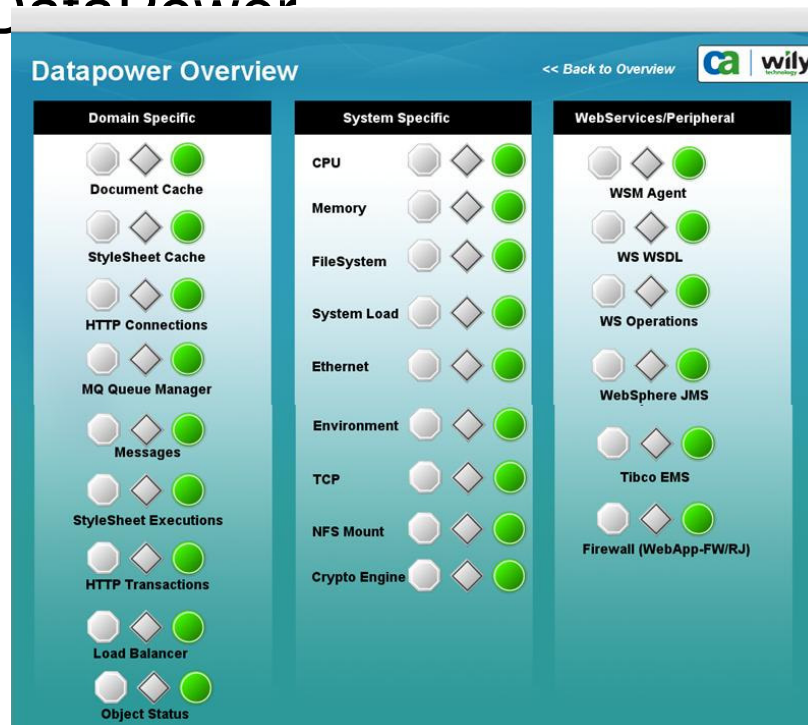
- Compte le nombre de Sessions HTTP actives : le compteur est incrémentée/décrémentée automatiquement en fonction des ouvertures/fermetures de sessions
- Permet de corréler la performance de l'application avec



# Couverture des environnements non Java EE / .Net

— Fournit une visibilité sur l'état de santé des boîtiers

Datapower



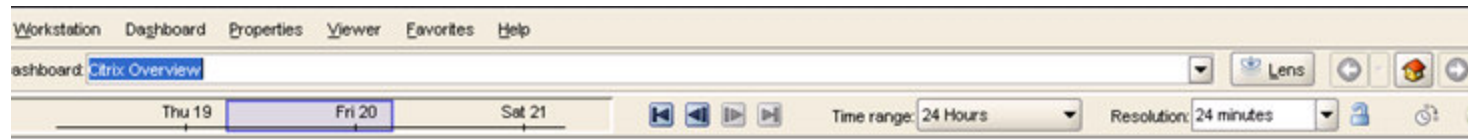
- Un EPA Plugin qui poll des MIBs SNMP et intègre les métriques collectées dans Introscope.
- Utilise une syntaxe « pseudo-pbd »

Exemple:

```
1.3.6.1.4.1.2725.1.5.1.1.3.1 IntCounter {enterprises.2725.1.4.1} | {enterprises.2725.1.1.1} | Connection Pools | {1.3.6.1.4.1.2725.1.5.1.1.2.1}:Minimum Connections
```

- Collecte les métriques Windows PerfMon pour les composants les plus communs :
  - CPU
  - Memoire
  - SQL Server
  - IIS
  - Etc.
- EPA qui tourne dans un processus séparé
- Peut collecter les métriques PerfMon à distance, éliminant ainsi le besoin d'installer un agent sur la machine

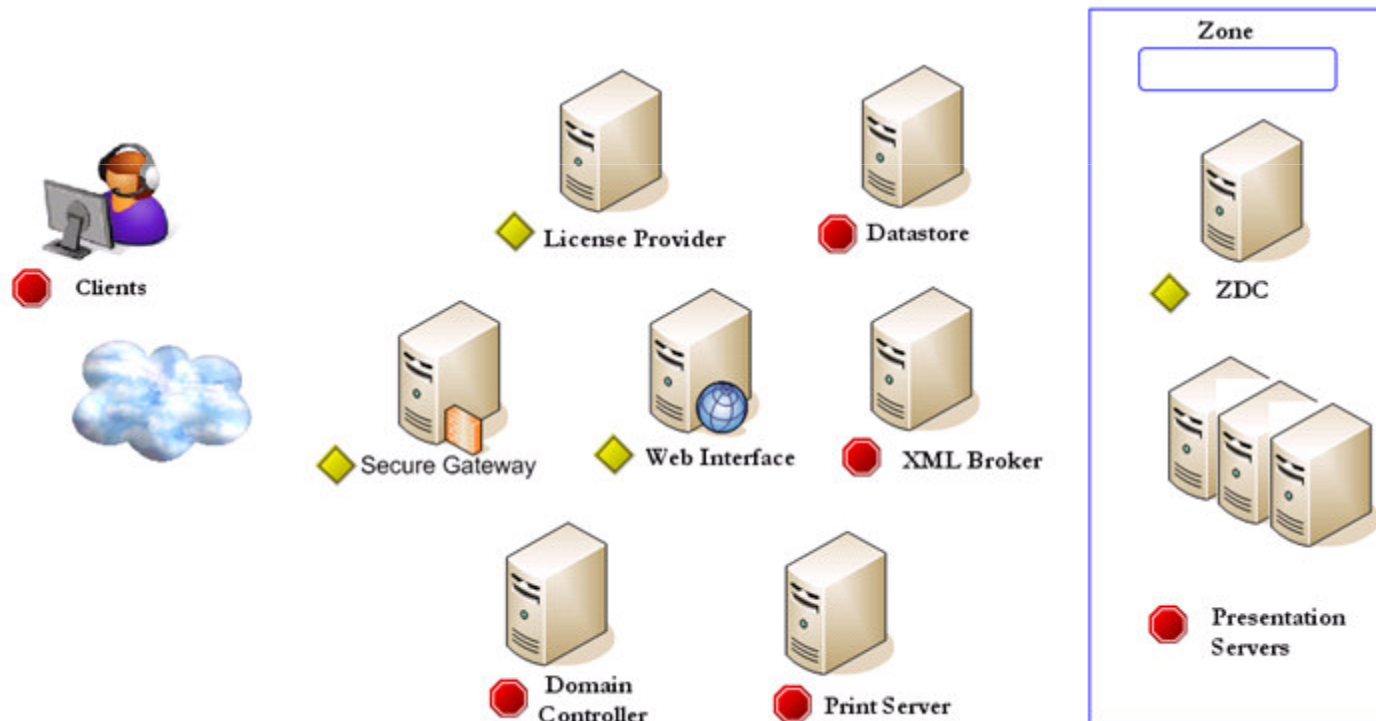
- Un EPA qui collecte les métriques en provenance de CITRIX Presentation Server et permet de :
  - Superviser des utilisateurs par session ou adresse IP
  - Comprendre la consommation des ressources serveurs consommées par utilisateur et application
  - Identifier les processus les plus consommateurs en terme de ressources
  - Mesurer la latence entre les utilisateurs distants et les serveurs CITRIX
- Présente l'information collectée sous forme de tableaux de bord



Introscope



Overview



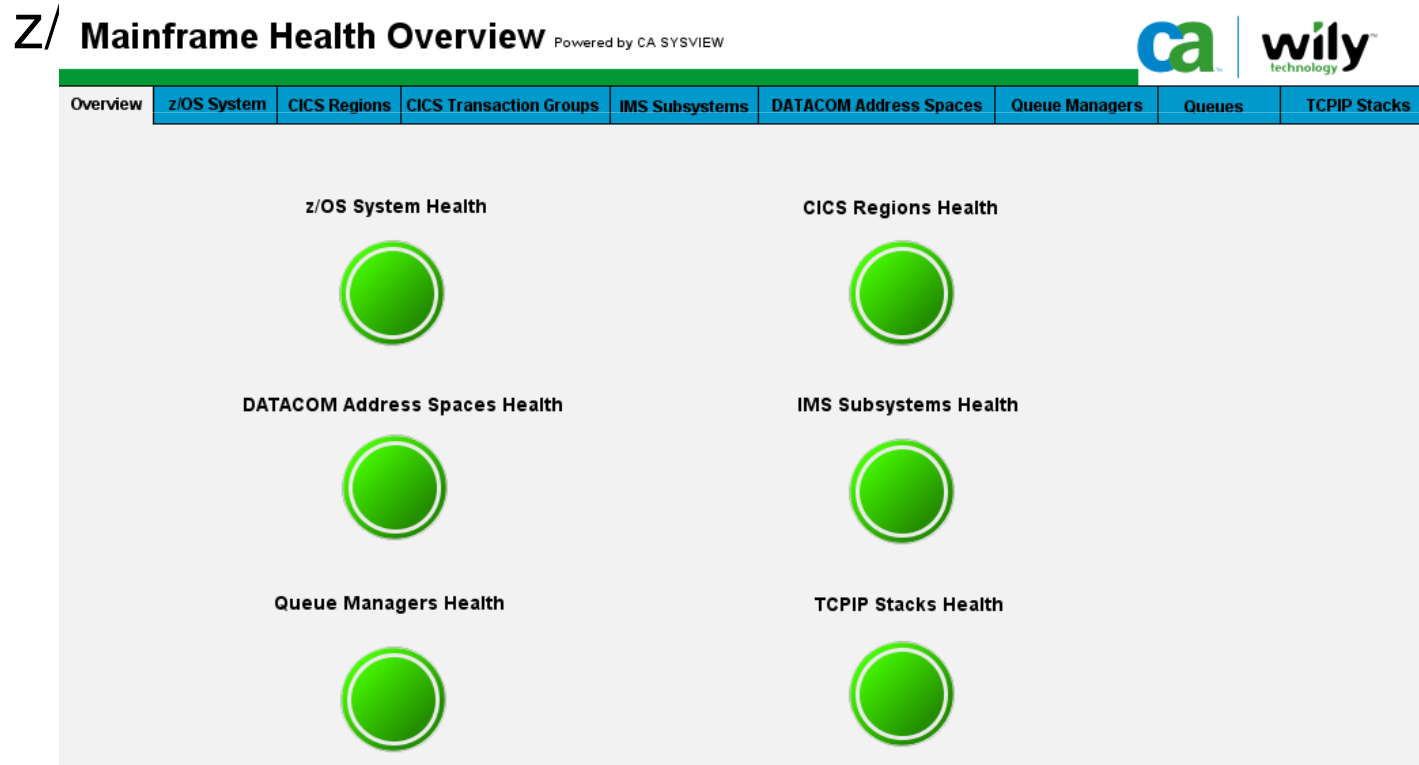
- Se connecte à distance sur les serveurs Web
- Permet de détecter des problèmes affectant les serveurs web tels que :
  - Indisponibilité du serveur Web
  - Mauvais temps de réponse du serveur Web
  - Problèmes de Domain Name System (DNS)
  - Problèmes de Load balancing
  - Configuration du serveur web (worker processes, threads, cache etc.)
- Permet de superviser :
  - Apache web server, Serveurs web basés sur Apache, Microsoft IIS, Sun ONE



- Se connecte à distance sur la base Oracle et offre les fonctions suivantes
  - Supervision du statut de la base
  - Tableaux de bord fournis en standard
  - Remonte des indicateurs tels que la taille des table space, le nombre de full table scan, ... etc.
  - Exécute des requêtes SQL sur les vues V\$
- Extensible pour exécuter toutes requêtes SQL et intégrer le résultat dans Introscopie

- Détecte les changements dans l'environnement de l'application
  - Code Java
  - File Systems
  - Tables de la base de données
- Corrèle les changements avec les dégradations de performance (Affiche des petits points en bas des graphes pour signaler un changement détecté)

- Suit les performances des transaction exécutées dans le CICS
- Apporte une visibilité sur les métriques mainframe et

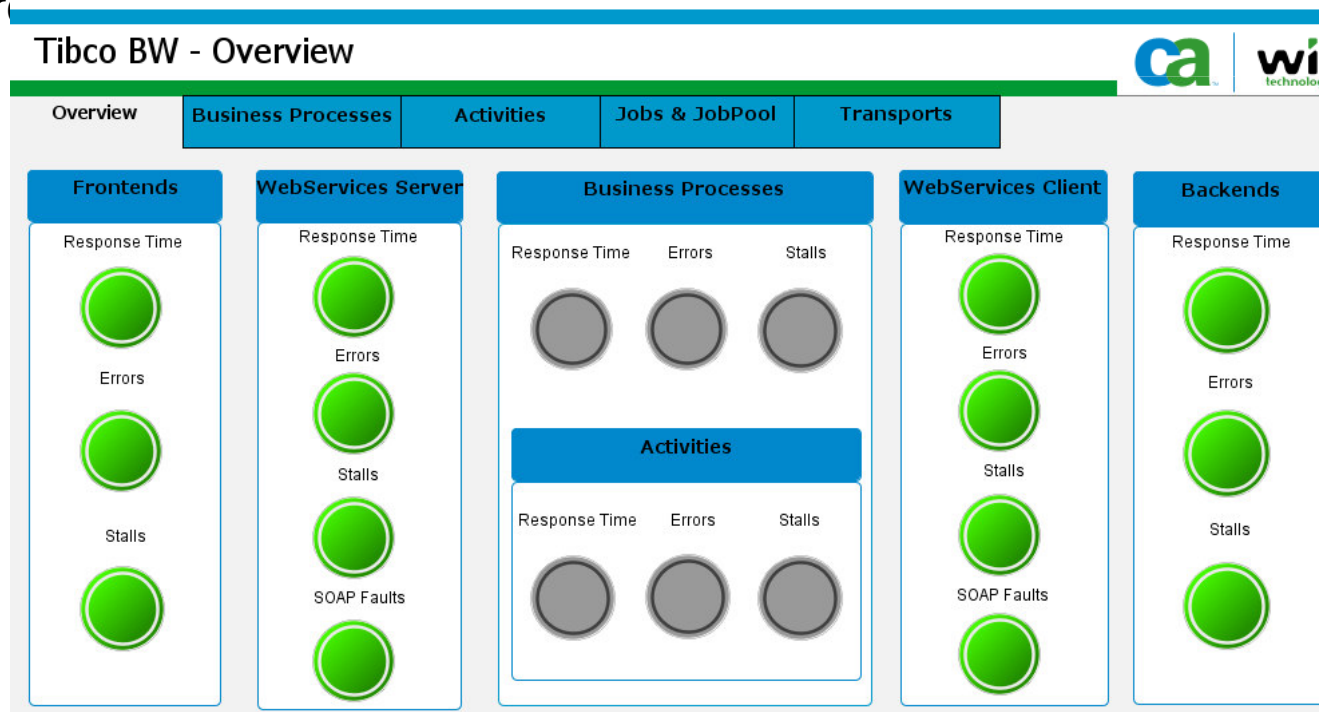


- Permettent la supervision de plateformes spécifiques SOA :
  - Oracle Service Bus (OSB)
  - TIBCO BusinessWorks
  - TIBCO Enterprise Message Service
  - webMethods Broker
  - webMethods Integration
  - WebSphere Process Server (WPS)
  - WebSphere Enterprise Service Bus (WESB)
  - WebSphere Message Queue (WebSphere MQ)
  - WebSphere Message Broker (WebSphere MB).

# SOA Performance Management & Power WebSphere MQ

Fonctionnalité  
Standard

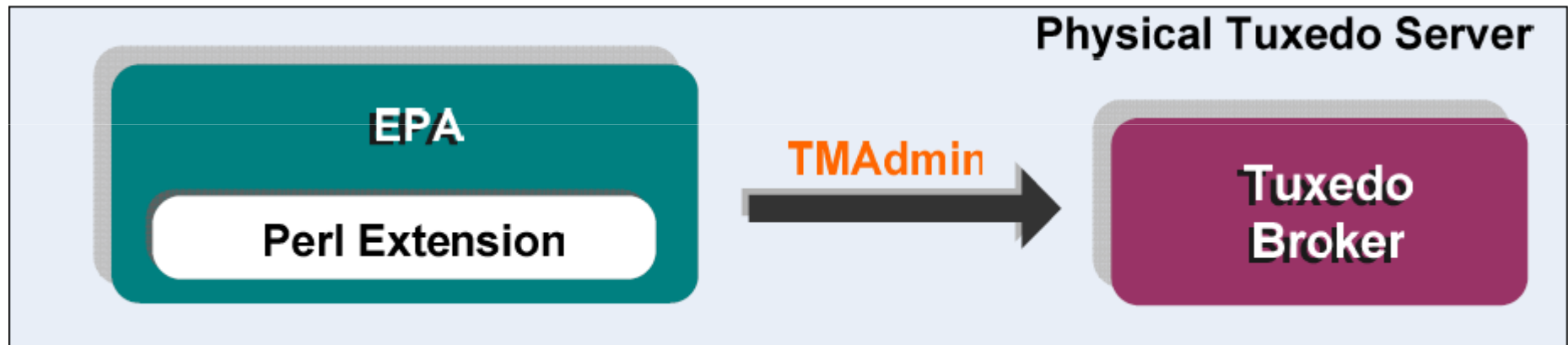
- Permet le suivi d'une transaction unitaire à l'intérieur des middlewares transactionnels
- Facilite la visibilité de bout en bout
- Offre également une supervision spécifique pour les bus de services / brokers



- Instrumentation spécifique pour PeopleSoft
- Visibilité sur les appels Tuxedo exécutés depuis PeopleSoft



- Supervision de l'état de Tuxedo Broker
  - Etat des files
  - Performance des services Tuxedo

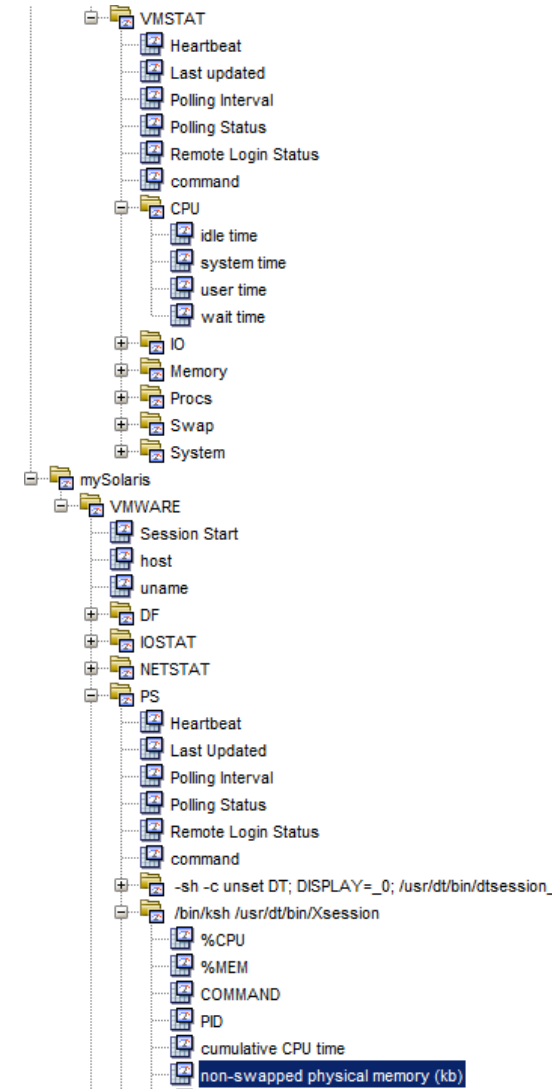


- Instrumentation des APIs Jolt et WTC qui permettent d'appeler les services Tuxedo depuis un programme Java

# Remote Unix Monitoring Agent

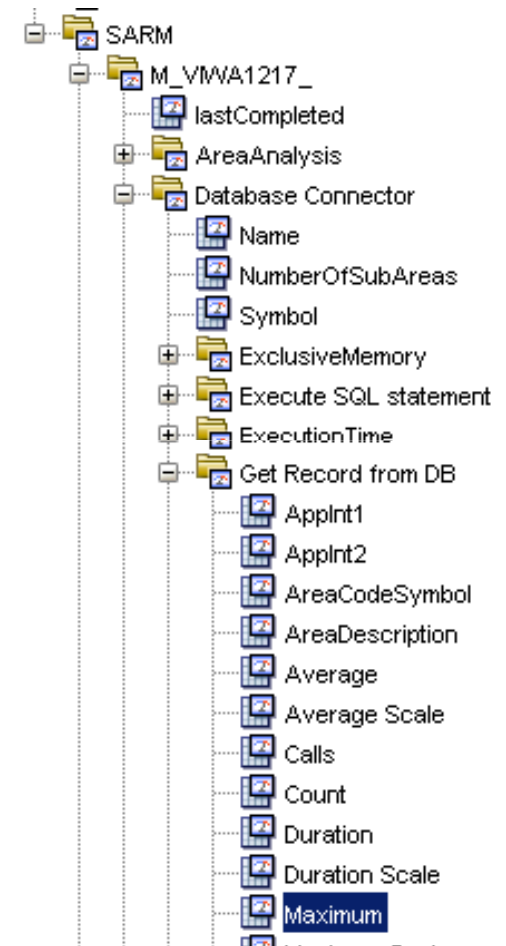
Field Pack

- EPA qui exécute à distance via SSH des commandes telles que :
  - VMStat
  - Netstat
  - Ou tout autre script
- Permet de collecter des indicateurs Système dans Introscope





- Une supervision complète de Siebel comprenant :
  - Mesure du ressenti des utilisateurs
  - Instrumentation des modules Java
  - Récupération des données SARM pour les modules non Java qui sont présentées en tant que métriques Introscope



# Enterprise Manager

# Baseline Comparison Type View

Field Pack

— Permet de comparer la performance entre deux plages horaires

General Traces Search Metric Count Comparison													
EARLIER TIME RANGE							LATER TIME RANGE						
SuperDomain\MARDA32-D820\LogiSphere\Banking							SuperDomain\MARDA32-D820\LogiSphere\Banking						
6/12/09 9:05 AM 20 Minutes							6/12/09 9:30 AM 20 Minutes						
>> GO <<							<< EXPORT >>						
metric	start value	end value	% value	start min	end min	% min	start max	end max	% max	start count	end count	% count	
JDBC:Average Query Time (ms)	573.0	249.0	130.1204...	107.0	110.0	-2.72727...	573.0	574.0	-0.17421...	11	7	57	
JDBC:Average Update Time (ms)	767.0	864.0	-11.2268...	388.0	319.0	21.63009...	897.0	864.0	3.819444...	11	7	57	
JDBC:Concurrent Invocations	9.0	8.0	12.5	2.0	2.0	0.0	9.0	8.0	12.5	11	7	57	
JDBC:Queries Per Interval	18.0	16.0	12.5	4.0	4.0	0.0	19.0	16.0	18.75	11	7	57	
JDBC:Queries Per Second	17.0	19.0	-10.5263...	4.0	5.0	-20.0	18.0	19.0	-5.26315...	11	7	57	
JDBC:Updates Per Interval	7.0	2.0	250.0	1.0	2.0	-50.0	7.0	7.0	0.0	11	7	57	
JDBC:Updates Per Second	7.0	6.0	16.66666...	1.0	1.0	0.0	7.0	7.0	0.0	11	7	57	
JDBC\Called ApplicationServer\JDBC Connection Pool\bankAppPool Allocated JDBC Connection...	4.0	3.0	33.33333...	0.0	0.0	NaN	4.0	3.0	33.33333...	11	7	57	
JDBC\Called ApplicationServer\JDBC Connection Pool\bankAppPool Available JDBC Connection...	9.0	9.0	0.0	5.0	6.0	-16.6666...	9.0	9.0	0.0	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, ADDRESS1, ADDRESS2, CITY, ST, ZIP FRO...	162.0	153.0	5.882352...	100.0	107.0	-6.54205...	194.0	195.0	-0.51282...	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, ADDRESS1, ADDRESS2, CITY, ST, ZIP FRO...	274.0	278.0	-1.43884...	200.0	210.0	-4.76190...	299.0	286.0	4.545454...	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, ADDRESS1, ADDRESS2, CITY, ST, ZIP FRO...	17.0	16.0	6.25	4.0	6.0	-33.3333...	17.0	16.0	6.25	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, CCN FROM ACCOUNTS WHERE ACCTID =?:...	253.0	127.0	99.21259...	102.0	125.0	-18.4000...	282.0	262.0	7.633587...	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, CCN FROM ACCOUNTS WHERE ACCTID =?:...	272.0	249.0	9.236947...	202.0	205.0	-1.46341...	297.0	266.0	11.65413...	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, CCN FROM ACCOUNTS WHERE ACCTID =?:...	28.0	21.0	33.33333...	20.0	21.0	-4.76190...	29.0	29.0	0.0	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, CCN, COMMENT1, COMMENT2 FROM ACCO...	180.0	145.0	249.236947791164653	102.0	102.0	-0.98039...	180.0	189.0	-4.76190...	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, CCN, COMMENT1, COMMENT2 FROM ACCO...	286.0	266.0	0.0	207.0	200.0	3.5	296.0	286.0	3.496503...	11	7	57	
JDBC[SQLDynamic]Query[SELECT ACCTID, NAME, CCN, COMMENT1, COMMENT2 FROM ACCO...	18.0	6.0	200.0	5.0	5.0	0.0	18.0	19.0	-5.26315...	11	7	57	
JDBC[SQLDynamic]Query[SELECT PORTFOLIO FROM ACCOUNTS WHERE NAME = *:Average ...	189.0	176.0	7.386363...	100.0	101.0	-0.99009...	189.0	193.0	-2.07253...	11	7	57	
JDBC[SQLDynamic]Query[SELECT PORTFOLIO FROM ACCOUNTS WHERE NAME = *:Average ...	290.0	220.0	31.81818...	222.0	202.0	9.900990...	299.0	289.0	3.460207...	11	7	57	
JDBC[SQLDynamic]Query[SELECT PORTFOLIO FROM ACCOUNTS WHERE NAME = *:Average ...	17.0	19.0	-10.5263...	4.0	13.0	-69.2307...	18.0	19.0	-5.26315...	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET CCN WHERE ACCTID =?:Average Result ...	273.0	293.0	-6.82593...	204.0	232.0	-12.0689...	293.0	293.0	0.0	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET CCN WHERE ACCTID =?:Average Update ...	192.0	195.0	-1.53846...	154.0	151.0	1.986754...	197.0	195.0	1.025641...	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET CCN WHERE ACCTID =?:Updates Per Sec...	7.0	7.0	0.0	1.0	1.0	0.0	7.0	7.0	0.0	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET COMMENT1 WHERE ACCTID =?:Average ...	291.0	240.0	21.25	108.0	101.0	6.930693...	291.0	295.0	-1.35593...	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET COMMENT1 WHERE ACCTID =?:Average ...	194.0	83.0	133.7349...	63.0	57.0	10.52631...	194.0	171.0	13.45029...	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET COMMENT1 WHERE ACCTID =?:Updates ...	6.0	5.0	20.0	2.0	2.0	0.0	7.0	7.0	0.0	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET PORTFOLIO WHERE ACCTID =?:Average ...	268.0	232.0	15.51724...	208.0	203.0	2.463054...	289.0	293.0	-1.36518...	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET PORTFOLIO WHERE ACCTID =?:Average ...	398.0	218.0	82.56880...	50.0	102.0	-50.9803...	398.0	305.0	30.49180...	11	7	57	
JDBC[SQLDynamic]Update[UPDATE ACCOUNTS SET PORTFOLIO WHERE ACCTID =?:Updates ...	5.0	6.0	-16.6666...	1.0	1.0	0.0	6.0	6.0	0.0	11	7	57	

— Voulez vous voir cette colonne renseignée

Overview	Traces	Errors	Search	Metric Count	Portal	SOA Dependency Map
Timestamp	Duration (ms)	Description	UserID	Total Duration (ms)		
Wed Dec 09 00:04:23 CET 2...	172	/wps/myportal/ut/p/c5/04_SB8K8xL...		172		
Wed Dec 09 00:04:23 CET 2...	375	/wps/myportal/ut/p/c5/04_SB8K8xL...		375		
Wed Dec 09 00:04:23 CET 2...	1157	/wps/myportal/ut/p/c5/fYzLDolwFA...		1157		
Wed Dec 09 00:04:23 CET 2...	32	/wsrp/WSRPBaseService_v2		32		
Wed Dec 09 00:04:23 CET 2...	5047	/wps/myportal/ut/p/c5/fYzLDolwFA...		5047		
Wed Dec 09 00:04:23 CET 2...	4078	/wps/myportal/ut/p/c5/fYzJDolwFE...		4078		

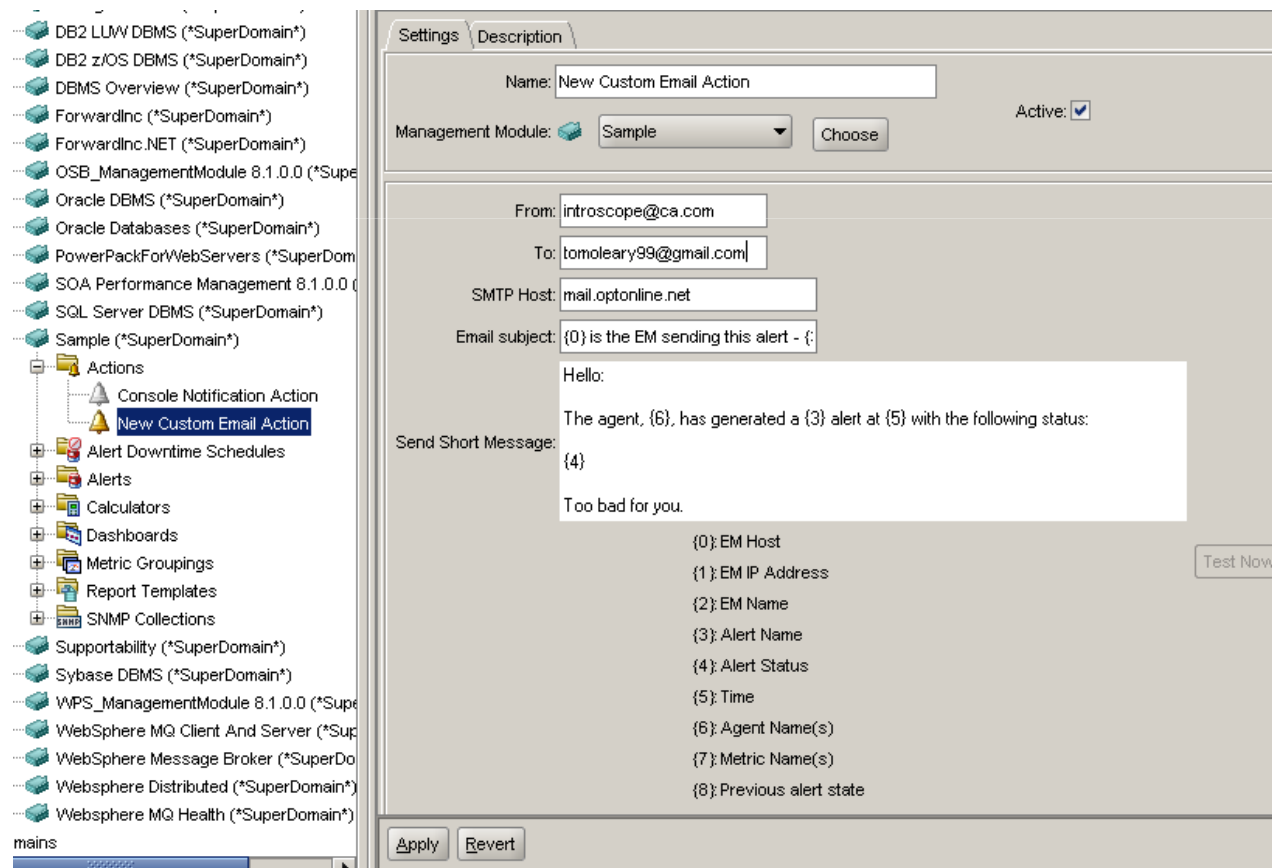
— Mais votre application est plus complexe que ce que Introscope sait remonter nativement.

— Avec UserIdAccessor, Ce n'est plus un problème !

Exemple:

```
TraceOneMethodWithParametersIfFlagged: HTTPServlet service UserIdAccessorTracer  
"{{0}.getSession().getAttribute('patientBean').getEmail()}"
```

- Action qui permet la personnalisation de l'objet et le corps des e-mails de notification :



# Persistenttransactiontracer

- Contrôler la génération des Transaction Traces depuis un fichier XML.
- Il suffit de placer le fichier PersistentTransactionTracerConfiguration.xml dans le répertoire config de l'EM et d'y définir les conditions de génération des traces

```
- <TraceSession agentExpression=".*" enabled="true">  
  <!-- trace all transactions lasting longer than 5 sec -->  
  <ThresholdFilter thresholdInMillis="5000" />  
</TraceSession>
```

## En synthèse

- Des extensions qui permettent d'améliorer l'instrumentation (Java EE / .NET)
- Fonctions d'analyse de la consommation mémoire et de recherche de fuites mémoire
- Couverture des environnements non Java EE / .Net
- Étendre le fonctionnement de l'Enterprise Manager

merci