

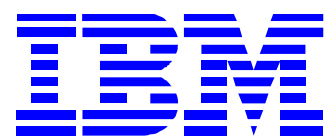
WebSphere Application Server v8.5

메모리 누수(Memory Leak) 감지 및 방지

(2012. 08.)

IBM SWG WebSphere CTP

이정운 과장(juwlee@kr.ibm.com)



0) 메모리 누수(Memory Leak) 감지 및 방지의 간략 소개

안녕하세요 freeman 입니다.

이번 강좌에서 언급드릴 내용은 IBM WAS v8.5 에서 강화된 기능인 메모리 누수(Memory Leak) 감지 및 방지 기능입니다.

애플리케이션을 개발하여 WAS 에서 구동 하다 보면 가장 많이 겪는 문제점 중의 하나가 바로 메모리 누수 입니다. 항상 원하는 것은 애플리케이션을 100% 무결하게 작성하는 것이지만 애플리케이션 개발자가 WAS 라던지 FrameWork 을 100% 이해하고 정말 잘 애플리케이션을 작성하는 경우가 많지 않기 때문에 잘못된 classloader 의 구조와 연관된다든지, 실수에 의해서 메모리 누수가 종종 운영환경에서 발생되고는 합니다.

IBM WAS v8.5 에서는 운영시에 애플리케이션 코드의 의심되는 패턴 감시를 통해서 탐 다운 패턴 기반의 메모리 누수 감지나 방지, 조치를 제공할 수 있으며 이를 통해 보다 높은 가용성을 제공하는 방안을 추가로 제공할 수 있습니다. 즉, 메모리 누수가 발생할 가능성이 있는 애플리케이션이라고 하여도 WAS 자체에서 모니터링 및 진단 액션을 수행하여 누수 가능성이 있는지 리포팅이나 heapdump 를 생성하여 해당 애플리케이션에 대한 수정을 지원하며 필요시 강제적으로 누수를 해제할 수도 있습니다.

이와 같이 누수 발견, 방지, 조치가 사용 가능으로 설정된 경우, IBM WAS 는 애플리케이션 및 모듈 활동을 모니터링하고 애플리케이션 또는 개별 모듈이 중지될 때 누수를 발견하고 수정하는 진단 조치를 수행하며 이 기능은 서버를 순환하지 않고도 빈번한 애플리케이션 재배포를 포함한 애플리케이션의 가동 시간을 늘리는 데 도움이 됩니다

이렇게 사전 조치 가능한 메모리 누수 정책은 다음과 같이 다양하게 설정할 수 있으며 이를 통해서 메모리 누수를 사전에 방지하기 위한 다양한 기능을 IBM WAS 에서 제공할 수 있습니다.

메모리 누수 정책 구성

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.express.iseries.doc/ae/ttrb_configmemleak.html?resultof=%22%6d%65%6d%6f%72%79%22%20%22%6d%65%6d%6f%72%69%22%20%22%6c%65%61%6b%22%20%22%64%65%74%65%63%74%69%6f%6e%22%20%22%64%65%74%65%63%74%22%20

```
[11/17/11 12:01:05:911 EST] 00000005 LeakDetection E CWMML0015E: The web application [WasSwat#WasSwatWeb.war] created a ThreadLocal with key of type [test.memleak.MyThreadLocal] (value [test.memleak.MyThreadLocal@216c691]) and a value of type [test.memleak.MyCounter] (value [test.memleak.MyCounter@21942ff]) but failed to remove it when the web application was stopped.
```

그럼 이제 실제 강좌를 통해서 해당 기능을 한번 살펴볼까요? ^^&

1) 메모리 누수(Memory Leak) 감지 및 방지

1. 메모리 누수 감지 및 방지를 위해서 필요한 설정 방식은 손쉽게 하단의 메모리 누수 정책 중 필요한 부분만 간단하게 JVM 옵션에 추가하면 됩니다. .

메모리 누수 정책 구성

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.express.iseries.doc/ae/ttrb_configmemleak.html?resultof=%22%6d%65%6d%6f%72%79%22%20%22%6d%65%6d%6f%72%69%22%20%22%6c%65%61%6b%22%20%22%64%65%74%65%63%74%69%6f%6e%22%20%22%64%65%74%65%63%74%22%20

표 1. com.ibm.ws.runtime.component.MemoryLeakConfig.detectAppCLLeaks

정보	값
이름	com.ibm.ws.runtime.component.MemoryLeakConfig.detectAppCLLeaks
설명	서버가 종료되거나 애플리케이션이 중지되면 WebSphere 애플리케이션 서버는 누수된 클래스 로더를 판별하여 메모리 누수를 디버그하는 데 도움이 되는 기타 추가 정보 및 경고를 발행합니다. 개선된 클래스 로더 누수 발견 PMR도 참조하십시오.
기본값	false

표 2. com.ibm.ws.runtime.component.MemoryLeakConfig.clearAppCLLeaks

정보	값
이름	com.ibm.ws.runtime.component.MemoryLeakConfig.clearAppCLLeaks
설명	예방 클래스 로더 누수 중개 및 수정을 사용하십시오. 이 특성이 true로 설정되면, WebSphere 애플리케이션 서버는 발견된 클래스 로더 누수를 치료하기 위해 애플리케이션을 대신하여 중재합니다.
기본값	false

표 3. com.ibm.ws.runtime.component.MemoryLeakConfig.preventJreMemoryLeaks

정보	값
이름	com.ibm.ws.runtime.component.MemoryLeakConfig.preventJreMemoryLeaks
설명	WebSphere 애플리케이션 서버를 사용하여 스레드 컨텍스트 클래스 로더에 싱글톤을 로드하는 JRE에서 발생하는 특정 클래스의 메모리 누수를 제거할 수 있습니다.
기본값	true

표 4. com.ibm.ws.runtime.component.MemoryLeakConfig.generateHeapDumps

정보	값
이름	com.ibm.ws.runtime.component.MemoryLeakConfig.generateHeapDumps
설명	true를 설정하면 메모리 누수가 발견되는 경우 heapdump가 작성됩니다.
기본값	true

참고적으로 해당 설정은 wsadmin 을 통해서 script 형식으로도 가능합니다.

```
C:\IBM\WebSphere85\WAppServer\Wprofiles\WAppSrv02\bin>wsadmin
WASX7209I: SOAP 커넥터를 사용하여 kr050578Node07 노드의 "server1" 프로세스에 연
결되었습니다. 프로세스 유형은 UnManagedProcess입니다.
WASX7029I: 도움말을 보려면, "$Help help"를 입력하십시오.
wsadmin>set leakConfig [$AdminControl completeObjectName "type=MemoryLeakConfig,
*"]
WebSphere:cell=kr050578Node03Cell,name=LeakConfig,type=MemoryLeakConfig,node=kr0
50578Node07,process=server1
wsadmin>$Help all $leakConfig
이름 : WebSphere:cell=kr050578Node03Cell,name=LeakConfig,type=MemoryLeakConfig,n
ode=kr050578Node07,process=server1
설명 : Information on the management interface of the MBean
클래스 이름: com.ibm.ws.runtime.component.MemoryLeakConfig
```

속성	유형	액세스
JvmThreadGroupNames	java.lang.String	RW
FilterPrefixes	java.lang.String	RW
RenewThreadPoolNames	java.lang.String	RW
DetectAppCLLeaks	boolean	RW
ClearAppCLLeaks	boolean	RW
MonitorSystemApps	boolean	RW
NoDumps	boolean	RW
GenerateHeapDumps	boolean	RW
GenerateSystemDumps	boolean	RW
ClearReferencesStatic	boolean	RW
ClearReferencesInterruptThreads	boolean	RW
ClearReferencesStopTimerThreads	boolean	RW
ClearReferencesHttpClientKeepAliveThread	boolean	RW
ClearReferencesThreadLocal	boolean	RW
CheckThreadLocalLeaks	boolean	RW
LeakSweeperDelay	int	RW
ThreadPoolRenewalDelayFactor	int	RW
PreventJreMemoryLeaks	boolean	RW
LeakConfiguration	java.lang.String	RO

2. 예를 들어 상단에서 언급한 옵션 중에 애플리케이션 클래스 로더 누수를 탐지하기 위한 com.ibm.ws.runtime.component.MemoryLeakConfig.detectAppCLLeaks 옵션을 하단과 같이 관리콘솔의 JVM 메뉴에서 사용자 정의 특성으로 정의하면 됩니다. (JVM 에 적용되는 옵션이므로 해당 WAS 의 재시작이 필요합니다.)

The screenshot shows the WebSphere Management Console interface. On the left is a navigation tree with categories like '시작' (Start), '안내된 활동' (Guided Action), '서버' (Server), '애플리케이션' (Application), '서비스' (Service), '자원' (Resource), '보안' (Security), '환경' (Environment), '시스템 관리' (System Management), '사용자 및 그룹' (User and Group), '모니터링 및 튜닝' (Monitoring and Tuning), and '문제점 해결' (Troubleshooting). The '환경' (Environment) category is selected, leading to the '애플리케이션 서버' (Application Server) page for 'server1'. The '프로세스 정의' (Process Definition) tab is active, showing the 'JVM (Java Virtual Machine)' configuration. Under '사용자 정의 특성' (User-defined Properties), a table lists various properties. The property 'com.ibm.ws.runtime.component.MemoryLeakConfig.detectAppCLLeaks' is highlighted with a red dashed box, showing its value as 'true'.

선택	이름	값	설명
<input type="checkbox"/>	com.ibm.security.jgss.debug	off	
<input type="checkbox"/>	com.ibm.security.krb5.Krb5Debug	off	
<input type="checkbox"/>	com.ibm.ws.management.event.pull_notification_timeout	120000	
<input checked="" type="checkbox"/>	com.ibm.ws.runtime.component.MemoryLeakConfig.detectAppCLLeaks	true	

3. 메모리 누수 감지 정책의 설정을 완료했으면 실제로 메모리 누수 감지를 테스트하기 위한 애플리케이션을 배포합니다.

엔터프라이즈 애플리케이션

설치된 애플리케이션을 관리하려면 이 페이지를 사용하십시오. 단일 애플리케이션을 여러 서버로 배치시킬 수 있습니다.

환경 설정

시작	중지	설치	설치 제거	업데이트	롤아웃 업데이트	파일 제거	내보내기	DDL 내보내기	파일 내보내기
<div> </div>									
선택	이름	애플리케이션 상태							
다음 자원을 관리할 수 있습니다.									
<input type="checkbox"/>	DefaultApplication								
<input type="checkbox"/>	LeakTestEAR								
<input type="checkbox"/>	ivtApp								
<input type="checkbox"/>	query								

4. 애플리케이션 배포가 끝났으면 메모리 누수 감지 테스트를 위해서 샘플 servlet 을 수행합니다.

ExampleFactory.java

http://localhost:9080/LeakTestWAR/LeakServlet

http://localhost:9080/LeakTestWAR/LeakServlet

Heap Size (KB): 107072 Max Heap Size (KB): 524288 WebContainer : 0 served this servlet 0 times

5. 연속적으로 해당 샘플 servlet 을 몇 번 수행하면 하단과 같이 Heap Size 가 증가되면서 메모리 누수가 발생하는 것을 확인 할 수 있습니다.

```

[12. 8. 17 16:16:01:416 KST] 0000005a SystemOut O Heap Size (KB): 241216
[12. 8. 17 16:16:01:416 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:02:412 KST] 0000005a SystemOut O Heap Size (KB): 241216
[12. 8. 17 16:16:02:412 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:03:352 KST] 0000005a SystemOut O Heap Size (KB): 241216
[12. 8. 17 16:16:03:353 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:04:954 KST] 0000005a SystemOut O Heap Size (KB): 241216
[12. 8. 17 16:16:05:201 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:07:642 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:07:642 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:09:479 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:09:479 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:10:931 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:10:931 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:11:823 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:11:823 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:12:819 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:12:820 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:13:668 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:13:668 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:14:877 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:14:877 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:15:654 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:15:655 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:16:16:683 KST] 0000005a SystemOut O Heap Size (KB): 266880
[12. 8. 17 16:16:16:683 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
    
```

6. 테스트 애플리케이션을 통해서 메모리 누수가 발생되면 해당 애플리케이션을 관리콘솔을 통해서 중지 작업을 수행합니다.

엔터프라이즈 애플리케이션

설치된 애플리케이션을 관리하려면 이 페이지를 사용하십시오. 단일 애플리케이션을 여러 서버로 배치시킬 수 있습니다.

환경 설정

시작	중지	설치	설치 제거	업데이트	롤아웃 업데이트	파일 제거	내보내기	DDL 내보내기
<div> <div>선택</div> <div>미름</div> </div> <div> <div>다음 자원을 관리할 수 있습니다.</div> <div>애플리케이션 상태</div> </div>								
<input type="checkbox"/>		DefaultApplication						
<input type="checkbox"/>		LeakTestWAR_war						
<input type="checkbox"/>		ivtApp						

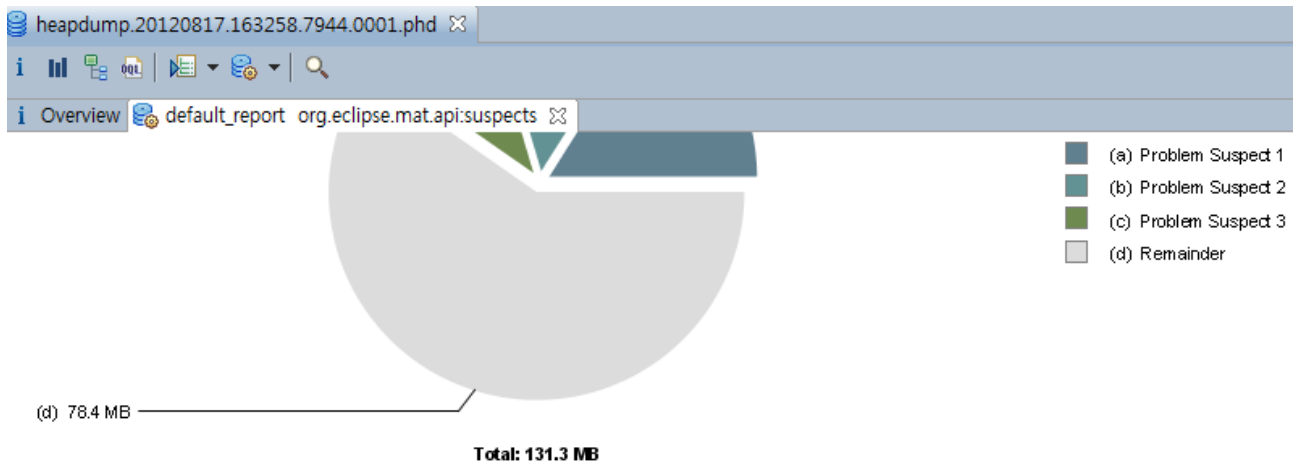
7. 애플리케이션 중지 후에 시간이 조금 흐르면 하단과 같이 애플리케이션 클래스 로더 누수가 검출되었다는 결과를 확인할 수 있습니다.

```
[12. 8. 17 16:16:28:904 KST] 0000005a SystemOut O Max Heap Size (KB): 524288
[12. 8. 17 16:32:47:981 KST] 0000005b AdminHelper A ADMIN1011I: LeakTestWAR_war 애플리케이션을 중지하려고 시도했습니다.
[12. 8. 17 16:32:47:982 KST] 0000005b CompositionUn A WSVR0192I: BLA WebSphere:blaname=LeakTestWAR_war에서
[12. 8. 17 16:32:47:984 KST] 0000005b ApplicationMg A WSVR0217I: LeakTestWAR_war 애플리케이션을 중지하는 중
[12. 8. 17 16:32:48:000 KST] 0000005b servlet I com.ibm.ws.webcontainer.servlet.ServletWrapper doDestr
[12. 8. 17 16:32:48:011 KST] 0000005b ApplicationMg A WSVR0220I: LeakTestWAR_war 애플리케이션이 중지됨
[12. 8. 17 16:32:49:479 KST] 0000005b CompositionUn A WSVR0193I: BLA WebSphere:blaname=LeakTestWAR_war에서
[12. 8. 17 16:32:58:025 KST] 00000055 MemoryLeakMgr W CWMML0020W: 애플리케이션 클래스 로더 누수: [
com.ibm.ws.classloader.CompoundClassLoader@3f4e6a1f[war:LeakTestWAR_war/LeakTestWAR_war]
Local ClassPath:
Parent: com.ibm.ws.classloader.CompoundClassLoader@57639aa9[app:LeakTestWAR_war]
Delegation Mode: PARENT_FIRST].
[12. 8. 17 16:32:58:025 KST] 00000036 MemoryLeakMgr W CWMML0020W: 애플리케이션 클래스 로더 누수: [
com.ibm.ws.classloader.CompoundClassLoader@57639aa9[app:LeakTestWAR_war]
Local ClassPath:
Parent: com.ibm.ws.classloader.ProtectionClassLoader@b97e6cb5
Delegation Mode: PARENT_FIRST].
[12. 8. 17 16:33:01:621 KST] 00000055 MemoryLeakMgr I CWMML0021I: 힙 덤프가 생성되었습니다.
[12. 8. 17 16:33:05:505 KST] 00000036 MemoryLeakMgr I CWMML0021I: 힙 덤프가 생성되었습니다.
[12. 8. 17 16:33:05:506 KST] 00000036 MemoryLeakMgr I CWMML0032I: 클래스 로더 메모리 누수 수준이 현재 차를 불가능합니다.
[12. 8. 17 16:33:05:512 KST] 00000055 MemoryLeakMgr I CWMML0032I: 클래스 로더 메모리 누수 수준이 현재 차를 불가능합니다.
```

8. 애플리케이션 클래스 로더 누수 검출이 되었다는 결과와 함께 해당 프로파일 폴더를 확인해보면 하단과 같이 Heap dump 가 생성된 것도 확인할 수 있습니다.

로컬 디스크 (C:) > IBM > WebSphere85 > AppServer > profiles > AppSrv02 > bin >				
포함	공유 대상	새 폴더		
이름	수정된 날짜	유형	크기	
heapdump.20120820.110845.7128.0003.phd	2012-08-20 오전...	PHD 파일	22,423KB	
heapdump.20120820.110506.7128.0002.phd	2012-08-20 오전...	PHD 파일	25,371KB	
heapdump.20120820.110502.7128.0001.phd	2012-08-20 오전...	PHD 파일	25,389KB	

참고: 해당 Heap dump 를 ISA 나 Memory Analyzer 와 같은 분석툴을 이용해서 분석해보면 하단과 같이 메모리 누수가 검출된 것을 확인할 수 있습니다.



▼ Problem Suspect 1

The class "**com.ibm.jwlee.servlet.LeakServlet**", loaded by "<system class loader>", occupies **22,020,336 (15.99%)** bytes. The memory is accumulated in one instance of "**com.ibm.jwlee.servlet.MyCounter[]**" loaded by "<system class loader>".

Keywords

com.ibm.jwlee.servlet.MyCounter[]
com.ibm.jwlee.servlet.LeakServlet

[Details »](#)

9) 참고 자료

1. 이 가이드는 IBM WAS v8.5 최초 사용자를 위한 기본 가이드입니다.
2. IBM WAS 자체에 아직 익숙하지 않으신 분들은 가급적 기본강좌인 '하나씩 쉽게 따라 해보는 IBM WAS v7' 강좌를 먼저 읽고 이 강좌를 읽으시는 것이 훨씬 이해에 됩니다.
(http://www.websphere.pe.kr/xe/?mid=was_info_re&page=3&document_srl=800)
3. 가급적 IBM WAS v8.5 InfoCenter 의 해당 카테고리를 한 번 읽어보고 난 후에 작업하시기 바랍니다.
4. InfoCenter – WebSphere Application Server v8.5
(<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>)
5. InfoCenter – Memory leaks in Java Platform, Enterprise Edition applications
(http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.nd.multiplatform.doc/ae/ctrb_memleakdetection.html?resultof=%22%6d%65%6d%6f%72%79%22%20%22%6d%65%6d%6f%72%69%22%20%22%6c%65%61%6b%22%20%22%64%65%74%65%63%74%69%6f%6e%22%20%22%64%65%74%65%63%74%22%20)