제 6 장 WebSphere Product Center 로거

WebSphere Product Center는 WebSphere Product Center 내의 문제점을 해결하는 데 사용할 수 있는 로그를 생성하는 사전에 구성된 파일을 제공합니다. 이 문서에서는 로깅 메커니즘 개요를 제공하며 로그 파일 설정 방법을 설명합니다.

WebSphere Product Center 서비스 로그 구성 파일

다음 파일은 전체 WebSphere Product Center 내의 다양한 서브시스템을 제어합니다. 생성된 로그 위치는 각 파일에 정의됩니다.

참고: 모든 경로는 $TOP에 상대적입니다.

/etc/logs/eventprocessor.log.xml

/etc/logs/scheduler.log.xml

/etc/logs/system.log.xml

/etc/logs/appsvr.log.xml

/etc/logs/workflowengine.log.xml

런타임 생성 로그

런타임 생성 로그는 오류를 볼 수 있으며, 문제점이 WebSphere Product Center 또는 내부 지원 하부 구조와 관련된 경우 문제점을 해결하는 데 도움을 줍니다.

WebSphere Product Center에서 생성된 로그 파일은 $TOP/logs/*.log에 저장됩니다.

로그 파일 구성

WebSphere Product Center 로그 파일의 등록 정보는 필요에 따라 편집될 수 있습니다(예: 위치, 최대 크기, 형식). 다음 절은 로그를 구성하고 WebSphere Product Center 로그 파일을 구성할 때 사용할 수 있는 값 목록을 제공하는 데 사용되는 요소에 대해 설명합니다.

위치 변경

참고: 파일 및 롤링 추가에만 적용됩니다.

생성된 로그 파일 위치를 변경하려면 지정된 로그 구성 파일 매개변수를

변경하십시오.

예를 들어, 다음과 같습니다.

<param name="File"   value="${TOP}/logs/webserver_db.log " />

파일 크기 변경

참고: 롤링 추가에만 적용됩니다.

로그 파일 크기는 출력의 최상의 순서의 출력 순환 및 제거를 시작하기 전에 지정된 기억장치 크기로 설정될 수 있습니다. 파일이 자르기를 시작하는 시기를 제어하려면 로그 파일 크기 매개변수 값을 변경하십시오.

예를 들어, 다음과 같습니다.

<param name="maxFileSize" value="10MB" />

파일 백업 옵션 변경

참고: 롤링 추가에만 적용됩니다.

로거는 로그 파일에 대해 지정된 수의 백업을 유지하도록 정의할 수 있습니다. 최대값에 도달하면 가장 오래된 파일이 버려집니다.

예를 들어, 다음과 같습니다.

<param name="maxBackupIndex" value="2" />

변환 패턴 변경

로그의 레이아웃 구성은 변환 패턴을 재정의하여 변경할 수 있습니다.

예를 들어, 다음과 같습니다.

<param name="ConversionPattern" value=
"%d [%t] %-5p %c (%F:%L) %x- %m%n"/>

변환 패턴은 C의 printf 함수의 변환 패턴에 밀접하게 관련되어 있습니다. 변환 패턴은 리터럴 텍스트와 *변환 지정자*라고 하는 형식 제어 표현식으로 구성되어 있습니다.

참고: 변환 패턴 내에서 리터럴 텍스트를 마음대로 삽입할 수 있습니다.

변환 지정자

각 변환 지정자는 퍼센트 부호 "%"로 시작되며 그 다음에 선택적으로 *형식 수정자* 및 *변환 문자가 옵니다*.

% (형식 수정자)(변환 문자)

예:

%-5p [%t]: %m%n

- 형식 수정자는 필드 너비, 채우기, 왼쪽 및 오른쪽 정렬 등을 제어합니다.
- 변환 문자는 데이터 유형(예: 카테고리, 우선순위, 날짜 및 스레드 이름)을 지정합니다.

기본적으로 관련 정보는 현상태대로 출력됩니다. 그러나 형식 수정자를 지원하면 최소 필드 너비, 최대 필드 너비 및 정렬을 변경할 수 있습니다.

선택적 형식 수정자는 퍼센트 부호와 변환 문자 사이에 위치합니다. 예에서 변환 지정자
%-5p는 로깅 이벤트 우선순위가 5자의 너비로 왼쪽으로 정렬되어야 한다는 것을 의미합니다.

첫 번째 선택적 형식 수정자는 마이너스(-) 문자인 *왼쪽 정렬 플래그*입니다. 그 다음에는 선택적 *최소 필드 너비* 수정자가 옵니다. 이는 출력할 최소 문자 수를 나타내는 10진 상수입니다. 데이터 항목에 더 적은 수의 문자를 요구하면 최소 너비에 도달할 때까지 왼쪽 또는 오른쪽으로 채워집니다.

기본값은 왼쪽(오른쪽 정렬)으로 채우는 것이지만 왼쪽 정렬 플래그로 오른쪽 채우기를 지정할 수 있습니다. 채우기 문자는 공백입니다. 데이터 항목이 최소 필드 너비보다 더 크면 필드는 데이터에 맞게 확장됩니다. 값은 자르지 못합니다.

이 작동은 *최대 필드 너비* 수정자를 사용하여 변경될 수 있으며, 이 수정자는 10진 상수가 뒤에 오는 점으로 지정되어 있습니다. 데이터 항목이 최대 필드보다 길면 여분의 문자는 끝이 아니라 데이터 항목의 *시작*부터 제거됩니다.

예를 들어, 최대 필드 너비가 8자이고 데이터 항목이 10자이면 데이터 항목의 처음 2자가 제거됩니다.

참고: 이 작동은 끝부터 절단되는 C에서의 printf 함수에서 벗어납니다.

다음 페이지에는 변환 지정자를 정의하기 위해 사용되는 값이 나와 있습니다.

형식 수정자

다음은 카테고리 변환 지정자에 대한 다양한 형식 수정자의 예입니다.

형식 수정	왼쪽 정	최소	최대	주석

자	렬		너비	
%20c	False	20	없음	카테고리 이름 길이가 20자보다 짧으면 왼쪽이 공백으로 채워집니다.
%-20c	True	20	없음	카테고리 이름 길이가 20자보다 길면 오른쪽이 공백으로 채워집니다.
%30c	NA	없음	30	카테고리 이름이 30자보다 길면 시작부터 잘립니다.
%20.30c	False	20	30	카테고리 이름이 20자보다 짧으면 왼쪽이 공백으로 채워집니다. 그러나 카테고리 이름이 30자보다 길면 시작부터 잘립니다.
%-20.30c	True	20	30	카테고리 이름이 20자보다 짧으면 오른쪽이 공백으로 채워집니다. 그러나 카테고리 이름이 30자보다 길면 시작부터 잘립니다.

변환 문자

다음은 인식되는 변환 문자 목록입니다.

변환 문자	결과
c	로깅 이벤트의 카테고리를 출력하는 데 사용됩니다. 정밀도 지정자는 선택적으로 카테고리 변환 *지정자* 다음에 올 수 있으며, 이 지정자는 대괄호로 묶인 10진 상수입니다. 정밀도 지정자가 제공되면 카테고리 이름의 가장 오른쪽 구성요소에 해당하는 숫자가 인쇄됩니다. 기본적으로 카테고리 이름은 전부 인쇄됩니다. 예를 들면, 카테고리 이름 "a.b.c"의 경우 패턴 %c{2}는 "b.c"를 출력합니다.
	로깅 이벤트의 날짜를 출력하는 데 사용됩니다. 중괄호 안에 들어 있는 날짜 형식 지정자는 날짜 변환 지정자 다음에 올 수 있습니다.

d	예: %d{HH:mm:ss,SSS} 또는 %d {dd MMM yyyy HH:mm:ss,SSS}. 날짜 형식 지정자가 제공되지 않으면 ISO8601 형식이 사용됩니다. 날짜 형식 지정자는 SimpleDateFormat의 시간 패턴 문자열과 동일한 구문을 허용합니다. 표준 JDK의 일부인 경우에도 SimpleDateFormat의 성능은 아주 부족합니다. 더 나은 결과를 위해서는 log4j 날짜 형식기를 사용할 것을 권장합니다. 이는 AbsoluteTimeDateFormat, DateTimeDateFormat 및 ISO8601DateFormat을 지정하기 위해 각각 "ABSOLUTE", "DATE" 및 "ISO8601" 문자열 중 하나를 사용하여 지정할 수 있습니다. 예: %d {ISO8601} 또는 %d{ABSOLUTE}. 이러한 전용 날짜 형식기는 SimpleDateFormat보다 훨씬 더 잘 수행됩니다.
m	로깅 이벤트와 연관된 WebSphere Product Center 제공 메시지를 출력하는 데 사용됩니다.
n	플랫폼 종속 행 분리자 문자 또는 문자 세트를 출력합니다. 이 변환 문자는 실질적으로 "\n" 또는 "\r\n"과 같은 이동할 수 없는 행 분리자 문자열 사용과 동일한 성능을 제공합니다. 따라서 행 분리자를 지정하는 데 선호하는 방법입니다.
p	로깅 이벤트의 우선순위를 출력하는 데 사용됩니다.
r	로깅 이벤트를 작성할 때까지 WebSphere Product Center 시작 이후에 경과되는 밀리초 수를 출력하는 데 사용됩니다.
t	로깅 이벤트를 생성한 스레드 이름을 출력하는 데 사용됩니다.
x	로깅 이벤트를 생성한 스레드와 연관된 NDC(nested diagnostic context)를 출력하는 데 사용됩니다.
%	%% 조합은 하나의 퍼센트 부호를 출력합니다.

WebSphere Product Center 로깅 설정 파일

다음 예는 WebSphere Product Center 로그 파일이 정의되는 방법을 증명합니다. 굵은체의 항목은 WebSphere Product Center 로그 파일의 구성을 설정

합니다.

```
<!-- basic ASYNC appender -->
<appender name="ASYNC" class="org.apache.log4j.AsyncAppender">
<appender-ref ref="DEFAULT"/>
</appender>



<!-- basic CONSOLE appender. This is the same as doing system.out-->
<appender name="STDOUT" class="org.apache.log4j.ConsoleAppender">
<layout class="org.apache.log4j.PatternLayout">
<param name="ConversionPattern" value=

"[%t] %-5p %c (%F:%L) %x- %m%n"/>
</layout>
</appender>

<!-- simple FILE appender. The file will be opened and if append is true  -->
<!--                    it will not be truncated                        -->
<appender name="DEFAULT" class="org.apache.log4j.FileAppender">
  <param name="File"   value="${TOP}/logs/tomcat_default.log " />
  <param name="Append" value="true" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value=

"%d [%t] %-5p %c (%F:%L) %x- %m%n"/>
   </layout>
</appender>

<!-- Rolling FILE appender. The file will be opened and if append is true  -->
<!--                    it will not be truncated                        -->
<!--                    maxFileSize: How big before you rotate         -->
<!--                    maxBackupIndex: How many backups do you keep?  -->
  <appender name="DB" class="org.apache.log4j.RollingFileAppender">
     <param name="File"   value="${TOP}/logs/tomcat_db.log " />
     <param name="Append" value="true" />
<param name="maxFileSize" value="10MB" />
<param name="maxBackupIndex" value="2" />
<layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value=

"%d [%t] %-5p %c (%F:%L) %x- %m%n"/>
 </layout>
</appender>

<!-- For the austin.db category, you want to have only a few logs kept hence -->
<!--  the rollingappender -->
<category name="austin.db" additivity=" false">
     <priority value="INFO" />
     <appender-ref ref="DB" />
</category>

<!-- ROOT CATEGORY -->
```

```
<!-- MUST ALWAYS BE LAST ENTRY AND HAVE AN APPENDER-->
<!-- If a logging event is not caught by any other logger it will be handled by
this-->
<!-- rule. -->
<root>
    <priority value="error"/>
    <appender-ref ref="DEFAULT"/>
</root>

</log4j:configuration>
```