

다중 플랫폼용 MQSeries Adapter Kernel



# 빠른 시작

버전 1 릴리스 1



다중 플랫폼용 MQSeries Adapter Kernel



# 빠른 시작

버전 1 릴리스 1

주!

이 정보 및 이 정보가 지원하는 제품을 사용하기 전에 139 페이지의 『주의사항』에 나와 있는 정보를 읽어 보십시오.

#### 제6판(2001년 4월)

이 개정판은 새로운 개정판에서 달리 언급하지 않는 한, 다중 플랫폼용 MQSeries Adapter Kernel(제품 번호 5648-D75)의 버전 1, 릴리스 1, 수정 레벨 1 및 모든 후속 릴리스와 수정판에 적용됩니다.

IBM은 귀하의 의견을 환영합니다. 전자 우편을 통해 [idrcf@hursley.ibm.com](mailto:idrcf@hursley.ibm.com)으로 이 책에 대한 의견을 보내실 수 있습니다.

IBM은 귀하가 제공한 정보를 적절한 방식으로 사용하거나 배포할 수 있으며, 귀하는 이에 대해 책임을 지거나 사용에 제한을 받지 않습니다.

# 목차

그림 . . . . .	v	OS/400 설치 필수조건 . . . . .	38
표 . . . . .	vii	리모트 AWT 사용. . . . .	39
<b>MQSeries Adapter Kernel 빠른 시작을 시</b>		접속된 클라이언트 사용 . . . . .	40
<b>작합니다</b> . . . . .	ix	커널 구성요소 . . . . .	41
이 책의 독자. . . . .	ix	<b>제3장 커널 설치</b> . . . . .	45
관련 정보. . . . .	ix	설치 준비. . . . .	46
규칙. . . . .	xi	커널 설치. . . . .	47
변경사항 요약. . . . .	xiii	설치 후 작업 완료. . . . .	50
<b>제1장 MQSeries Adapter Offering에 대한</b>		설치 검증. . . . .	53
<b>정보</b> . . . . .	1	확인 절차. . . . .	55
빌드 시간 및 런타임 . . . . .	2	일반적인 확인 문제점. . . . .	56
커널에 대한 정보. . . . .	4	선택적 확인 . . . . .	58
커널 작업 방법 . . . . .	10	자동 설치 사용 . . . . .	59
커널 런타임 구성요소. . . . .	10	커널 업그레이드. . . . .	61
메시지 및 메시지 형식 . . . . .	13	커널 제거. . . . .	62
라우팅 및 전달 . . . . .	14	<b>제4장 커널 사용</b> . . . . .	65
런타임 플로우 . . . . .	15	프로덕션 준비 . . . . .	65
커널의 소스측 . . . . .	16	커널 구성. . . . .	67
커널의 대상측 . . . . .	21	구성 개요. . . . .	67
트랜잭션 기능 . . . . .	30	시작 및 구성시 사용되는 파일. . . . .	73
추적 . . . . .	31	설정 파일. . . . .	73
WebSphere Business Integrator 및		구성 파일. . . . .	74
WebSphere Application Server와 함께		MQSeries 및 MQSeries Integrator 구성	101
MQSeries Adapter Kernel 사용 . . . . .	31	성능 권장사항 . . . . .	102
JMS 리스너 . . . . .	31	커널 시작 . . . . .	102
자국어 지원 . . . . .	33	커널 정지 . . . . .	104
<b>제2장 커널 설치 계획</b> . . . . .	35	커널 유지보수 . . . . .	105
하드웨어 . . . . .	35	문제점 진단. . . . .	105
소프트웨어 . . . . .	36	버전 번호 . . . . .	106
		예외 메시지. . . . .	106
		추적 메시지. . . . .	107
		유틸리티. . . . .	108

MQSeries 큐 작성 . . . . .	108	MQSeries 메시지 설명자 헤더 . . . . .	123
<b>제5장 MQSeries Adapter Kernel API 사</b>		MQSeries Integrator가 없는 MQSeries . . .	124
<b>용</b> . . . . .	109	MQSeries Integrator 버전 1 헤더 . . . . .	125
<b>제6장 추가 정보 확보</b> . . . . .	111	MQSeries Integrator 버전 2 헤더 . . . . .	126
인터넷에서 사용 가능 . . . . .	111	<b>부록D. 구성 파일 샘플</b> . . . . .	129
참조서 . . . . .	111	최소 구성 파일 샘플 . . . . .	134
<b>부록A. 통신 모드</b> . . . . .	113	<b>부록E. 설정 파일 샘플</b> . . . . .	137
JMS 오브젝트 저장영역 사용 . . . . .	116	<b>주의사항</b> . . . . .	139
<b>부록B. 검증 구성</b> . . . . .	119	<b>상표</b> . . . . .	141
<b>부록C. 메시지 헤더</b> . . . . .	121	<b>용어집</b> . . . . .	143
MQSeries Adapter Kernel 메시지 설명자		<b>색인</b> . . . . .	149
헤더 . . . . .	121		

---

## 그림

1. MQSeries Adapter Offering 개요	6	5. 데이터 변환	71
2. 메시지 정렬, 송신, 라우트 및 추적 -- 개요	16	6. 데이터 플로우	71
3. 단순한 구성에서 데이터 플로우에 의해 연결된 응용프로그램	68	7. 구성과 관련된 데이터 플로우	72
4. 단순한 구성에서 다른 통신 전송에 의해 연결된 응용프로그램	70	8. 구성 파일의 상위 레벨 구조	77





---

## 표

1. 이 책에서 사용되는 규칙 . . . . .	xi	7. 공통 구성: JMS를 통해 메시지 송신	93
2. 공통 구성: MQSeries 서버에서 다른 MQSeries 서버로 메시지 송신 . . . . .	89	8. 공통 구성: JMS를 통한 메시지 수신	93
3. 공통 구성: 리모트 큐 관리자를 통해 MQSeries 서버에서 MQSeries 서버로 메시지 송신 . . . . .	90	9. 통신 모드 및 지원 Java 클래스	114
4. 공통 구성: 호스트 서버를 사용하고 있는 MQSeries 클라이언트에서 MQSeries 서버로 메시지 송신 . . . . .	90	10. 통신 모드 및 형식기 인터페이스	115
5. 공통 구성: 메시지를 수신하는 MQSeries 서버 . . . . .	91	11. 형식기 인터페이스, 형식기 클래스 이름 및 목적 . . . . .	115
6. 공통 구성: 메시지를 수신하는 호스트 서버를 사용하고 있는 MQSeries 클라이언트 . . . . .	92	12. LMS 클래스 및 트랜잭션 지원	115
		13. MQSeries Adapter Kernel 헤더	121
		14. MQSeries 헤더 . . . . .	123
		15. MQSeries Integrator 버전 1 헤더 -- RFH1. . . . .	125
		16. MQSeries Integrator 버전 2 헤더 -- RFH2. . . . .	127



---

# MQSeries Adapter Kernel 빠른 시작을 시작합니다

이 책에서는 MQSeries Adapter Kernel을 설명하고 이 제품의 설치 및 사용 방법을 설명합니다.

커널을 사용하려면 다음과 같은 일반적인 단계를 수행하십시오.

1. 1 페이지의 『제1장 MQSeries Adapter Offering에 대한 정보』 내용을 읽어 보십시오.
2. 설치를 준비하십시오. 세부사항은 46 페이지의 『설치 준비』를 참조하십시오.
3. 커널을 설치하십시오. 세부사항은 47 페이지의 『커널 설치』를 참조하십시오.
4. 설치를 검증하십시오. 세부사항은 53 페이지의 『설치 검증』을 참조하십시오.
5. 커널을 구성하십시오. 세부사항은 67 페이지의 『커널 구성』을 참조하십시오.
6. 원하면 커널에 대해 작업할 선택적 소프트웨어를 구성하십시오. 세부사항은 101 페이지의 『MQSeries 및 MQSeries Integrator 구성』을 참조하십시오.
7. MQSeries Adapter Builder를 사용하여 어댑터를 빌드하고 테스트한 다음 전개하십시오. 세부사항은 MQSeries Adapter Builder 문서를 참조하십시오.
8. 커널을 시작하십시오. 세부사항은 102 페이지의 『커널 시작』을 참조하십시오.

이 책을 사용하려면 필수조건과 선택적인 제품에 대해서도 알아야 합니다. 35 페이지의 『제2장 커널 설치 계획』을 참조하십시오. 또한 111 페이지의 『참조서』를 참조하십시오.

---

## 이 책의 독자

이 책은 MQSeries Adapter Kernel을 계획, 설치 및 사용해야 하는 사용자를 위한 것입니다.

---

## 관련 정보

자세한 정보는 다음을 참조하십시오.

- `readme.txt` 파일. 이 파일에는 보통 이 책이 완성된 후에 사용할 수 있게 된 정보가 들어 있습니다. 설치하기 전에 `readme.txt` 파일은 제품 CD-ROM의 루트 디렉토리에 있습니다. 설치 후 `readme.txt` 파일은 MQSeries Adapter Kernel 설치 루트 디렉토리에 있습니다.
- *Problem Determination Guide*, 문서 번호 GC34-5897에서는 MQSeries Adapter Kernel에 대한 특정 문제점을 해결하기 위한 추적을 포함한 도구를 설명합니다. *Problem Determination Guide*는 제품과 함께 설치한 MQSeries Adapter Kernel 정보 센터에서 사용 가능합니다.
- MQSeries Adapter Kernel 정보 센터에 제공되는 온라인 응용프로그램 프로그래밍 인터페이스(API) 문서. 이 정보는 커널이 기능하는 방식을 이해하고 문제점을 진단하는 보조 수단으로 제공됩니다. 109 페이지의 『제5장 MQSeries Adapter Kernel API 사용』을 참조하십시오.
- 책과 도움말 시스템을 포함한 MQSeries Adapter Builder 정보.
- MQSeries 제품군 웹 사이트([www.ibm.com/software/ts/mqseries/](http://www.ibm.com/software/ts/mqseries/)).  
이 웹 사이트의 링크를 따라가면 다음을 할 수 있습니다.
  - MQSeries Adapter Offering을 비롯한 MQSeries 제품군에 대한 최신 정보를 얻을 수 있습니다.
  - 이 책의 최신 개정판을 포함하여 HTML 및 PDF 형식의 MQSeries 서적에 액세스할 수 있습니다.
  - MQSeries 지원팩을 다운로드할 수 있습니다.

# 규칙

MQSeries Adapter Kernel 문서는 다음과 같은 인쇄 및 입력 규칙을 사용합니다.

표 1. 이 책에서 사용되는 규칙

규칙	의미
굵은체	명령 이름을 나타냅니다. 그래픽 사용자 인터페이스(GUI)를 참조할 때 메뉴, 메뉴 항목, 레이블 및 단추를 나타냅니다.
모노스페이스	명령 프롬프트에 입력해야 하는 텍스트와 파일, 이름, 경로와 같이 반복해서 사용해야 하는 값 그리고 기능, 클래스 및 메소드와 같은 프로그래밍 언어 요소를 나타냅니다. 모노스페이스는 화면 텍스트와 코드 예를 나타냅니다.
기울임체	사용자가 제공해야 하는 변수 값을 나타냅니다(예를 들어, <i>fileName</i> 에 대한 파일 이름을 제공합니다). 기울임체는 책의 제목과 강조표시를 나타내기도 합니다.
%	루트 특권이 필요없는 명령에 대한 UNIX 명령셸 프롬프트를 나타냅니다.
#	루트 특권이 필요한 명령에 대한 UNIX 명령셸 프롬프트를 나타냅니다.
C:\>	Windows 시스템에서 명령 프롬프트를 나타냅니다.
>	메뉴를 설명하는 데 사용한 경우 일련의 메뉴 선택사항을 표시합니다. 예를 들어, “파일 > 신규”는 “파일 메뉴에서 신규 명령을 누르십시오”를 의미합니다.
명령 입력	명령을 “입력”하거나 “발행”하도록 지시한 경우 명령을 입력하고 Return을 누르십시오. 예를 들어 지시사항 “ls 명령을 입력하십시오”는 ls를 명령 프롬프트에 입력하고 Return을 누르라는 의미입니다.
[ ]	구문 설명에서 선택적 항목을 묶습니다.
{ }	구문 설명에서 항목을 선택해야 하는 목록을 묶습니다.
	구문 설명에서 중괄호({ })로 묶인 선택사항 목록에서 항목들을 분리합니다.
...	구문 설명의 밑줄임표는 이전 항목을 한번 이상 반복할 수 있음을 나타냅니다. 예제의 밑줄임표는 예제에서 정보가 생략되었음을 나타냅니다.

주: Epic이라는 용어가 커널 소프트웨어와 이 책에서 일부 값 및 이름에 나타납니다. MQSeries Adapter Offering의 경우 이 용어 자체는 어떠한 의미도 없습니다.



---

## 변경사항 요약

제6판(현재 개정판)에는 제5판과는 다른 다음과 같은 추가사항 및 변경사항이 포함되어 있습니다.

- 여러 변경 사항을 반영하기 위한 런타임 플로우에 대한 설명 갱신. 15 페이지의 『런타임 플로우』 부분을 참조하십시오.
- MQSeries Adapter Kernel을 WebSphere Business Integrator와 함께 사용하는 것에 대한 정보. 세부사항은 31 페이지의 『WebSphere Business Integrator 및 WebSphere Application Server와 함께 MQSeries Adapter Kernel 사용』을 참조하십시오.
- 다른 종류의 어댑터와 함께 제공되는 자국어 지원 레벨에 대한 정보. 세부사항은 33 페이지의 『자국어 지원』을 참조하십시오.
- 설치 지시사항의 명료화. 47 페이지의 『커널 설치』 부분을 참조하십시오.
- 자동 설치에 대한 정보. 세부사항은 59 페이지의 『자동 설치 사용』을 참조하십시오.
- 커널 구성에 도움이 되는 구성에 대한 개념적인 개요. 세부사항은 67 페이지의 『구성 개요』를 참조하십시오.
- 새로운 헤더 값에 대한 정보. 세부사항은 121 페이지의 『MQSeries Adapter Kernel 메시지 설명자 헤더』를 참조하십시오.

제5판에는 제4판에 대한 다음과 같은 추가 및 변경 사항이 포함됩니다.

- Windows 2000, OS/400, HP-UX 및 Solaris 플랫폼에서의 커널 사용 방법. 이 플랫폼들에 대한 지원은 MQSeries Adapter Kernel 버전 1.1에서 새로운 내용입니다. 커널은 이전에 Windows NT 및 AIX에서만 사용 가능했습니다.
- MQSeries Adapter Kernel 버전 1.1을 반영하기 위한 모든 설치 지시사항 갱신.
- aqmconfig.xml 파일을 사용하여 MQSeries Adapter Kernel을 구성하는 데 대한 정보. 커널은 이전에 aqmconfig.properties 파일로 구성되었습니다. 세부사항은 74 페이지의 『구성 파일』을 참조하십시오.

- 새로운 MQ 및 JMS(Java Message Service) 통신 모드에 대한 정보. 세부사항은 113 페이지의 『부록A. 통신 모드』를 참조하십시오.
- 추적 정보는 이 문서에서 새로운 *Problem Determination Guide* 문서로 이동되었습니다. 자세한 정보는 *Problem Determination Guide*를 참조하십시오.



---

## 제1장 MQSeries Adapter Offering에 대한 정보

IBM MQSeries Adapter Kernel은 MQSeries Adapter Offering이라는 응용프로그램 통합 제품 세트의 일부입니다. MQSeries Adapter Offering은 MQSeries 메시징 및 기타 메시지 전달 서비스와 함께 사용자가 비즈니스 프로세스의 포인트-투-포인트 통합 관리 비용, 위험 및 복잡성을 줄일 수 있도록 해줍니다.

포인트-투-포인트 통합에서 각 응용프로그램은 다른 응용프로그램과 개별적으로 통합합니다. 각 인터페이스는 서로 다르며, 다양한 인터페이스가 있습니다. 일반적으로 한 응용프로그램이 변경되면 많은 인터페이스가 변경되어야 합니다. 응용프로그램의 수가 증가함에 따라 포인트-투-포인트 통합 비용이 급격히 증가합니다. 일반적으로 각각의 새로운 응용프로그램을 통합하는 것이 이전 응용프로그램을 통합하는 것보다 많은 작업을 필요로 합니다.

MQSeries Adapter Offering을 통해 포인트-투-포인트 통합 사용에서 일대다(*one-to-any*) 통합 사용으로 이주할 수 있습니다. 일대다 통합은 다음을 포함하는 많은 장점이 있습니다.

- 모든 응용프로그램이 하나의 공통 인터페이스를 사용할 수 있습니다.
- 메시지 형식의 소스 응용프로그램 데이터는 하나 이상의 대상 응용프로그램으로 라우트됩니다.
- 일반적으로 한 응용프로그램의 변경사항은 해당하는 하나의 인터페이스에만 영향을 미칩니다.
- 응용프로그램에 공통적인 공통 인터페이스를 사용하면(예를 들어, XML(extensible markup language)과 같은 산업 표준) 비용면에서 좀 더 효과적일 수 있습니다. 적은 노력으로 더 많은 응용프로그램을 지원할 수 있습니다.
- 응용프로그램의 수가 증가하면 일대다 통합 비용 효과도 향상됩니다. 새로운 응용프로그램 각각을 추가할 경우 다른 모든 응용프로그램의 인터페이스를 크게 변경할 필요는 없습니다.
- 통합 작업을 자동화할 수 있으며 템플릿에 기초할 수 있습니다.

MQSeries Adapter Offering은 응용프로그램이나 비즈니스 프로세스를 전혀 변경하지 않고도 전개할 수 있습니다. 일반적으로 모든 통합 작업은 MQSeries Adapter Offering에서 수행되므로 사용자 정의 코드를 작성할 필요성이 줄어듭니다.

MQSeries Adapter Offering에서 하나의 응용프로그램으로(부터)의 인터페이스는 어댑터가 제공합니다. 모든 응용프로그램은 응용프로그램 환경과 메시징 환경 사이에 인터페이스를 제공하기 위해 최소 한 개 이상의 어댑터가 필요합니다. 각 어댑터는 응용프로그램 및 메시지 유형에 고유합니다.

MQSeries Adapter Kernel은 선택적으로 중개 및 메시지 전송을 위해 MQSeries Integrator로 전개될 수 있습니다. MQSeries Adapter Offering은 IBM 및 타사에서 제공하는 서비스 제공으로 보충이 가능합니다.

어댑터 사용 예에는 다음이 포함됩니다.

- 판매 주문 추가.
- 고객 레코드 동기화.
- 재고 레코드 동기화.
- 항목 동기화.
- 판매 주문 동기화.

---

## 빌드 시간 및 런타임

MQSeries Adapter Offering은 두 가지 기본적인 구성요소 즉, 어댑터 빌더(빌더라고도 함)와 어댑터 커널(커널이라고도 함)로 구성됩니다. 이 절에서는 어댑터 오퍼링으로 빌드되고 실행된 어댑터는 물론 이러한 구성요소에 대해서도 설명합니다.

### 어댑터

응용프로그램으로(부터)의 인터페이스를 제공하는 소프트웨어. 어댑터는 MQSeries Adapter Builder를 사용하여 빌드됩니다. 일반적으로 각 어댑터는 응용프로그램으로 송신되거나 수신된 한 가지 메시지 유형에 고유하게 빌드됩니다. 어댑터 자체는 MQSeries Adapter Offering의 일부가 아닙니다.

어댑터는 공유 라이브러리로 컴파일되는 C 또는 Java 소스 코드로 구성됩니다. 어댑터 및 MQSeries Adapter Kernel이 함께 실행되는 경우 MQSeries Adapter Offering의 런타임 기능을 수행합니다.

MQSeries Adapter Builder로 모델화되는 방법에 따라, 어댑터는 제어 플로우, 데이터 플로우, 순차 탐색, 결정 및 반복을 포함한 조건부 분기, 데이터 입력, 데이터 컨텍스트 저장, 데이터 요소 변환, 트랜잭션 제어, 논리 조작 및 사용자 정의 코드와 같은 다양한 기능을 포함할 수 있습니다.

어댑터는 재사용할 수 있습니다.

어댑터에는 기본적으로 다음 두 가지 유형이 있습니다.

- 데이터를 송신하는 응용프로그램에 대한 소스 어댑터
- 데이터를 수신하는 응용프로그램에 대한 대상 어댑터

한 가지 유형의 메시지를 하나의 응용프로그램에서 두번째 응용프로그램으로 송신하는 경우 하나의 소스 어댑터와 하나의 대상 어댑터가 필요합니다. 두번째 응용프로그램이 한 가지 유형의 메시지를 첫번째 응용프로그램으로 보내야 하는 경우 또 다른 소스 어댑터 및 대상 어댑터가 필요합니다. 따라서 한 가지 유형의 메시지를 첫번째 응용프로그램에서 두번째 응용프로그램으로 송신한 다음 또 다른 유형의 메시지를 다시 두번째 응용프로그램에서 첫번째 응용프로그램으로 송신하려면 일반적으로 네 개의 어댑터가 전개됩니다.

각 메시지 유형에 대해 개별적인 어댑터가 필요합니다.

세번째 어댑터 유형인 Java 서비스 세션 bean 어댑터는 커널의 타겟층에서 IBM WebSphere Application Server와 엔터프라이즈 bean으로 사용할 때 사용됩니다. Sun Microsystems EJB(Enterprise JavaBeans) 스펙의 WebSphere Application Server 구현을 통해 Java 서비스 세션 bean 어댑터 및 기타 엔터프라이즈 bean 사용이 가능해졌습니다. 자세한 내용은 31 페이지의 『WebSphere Business Integrator 및 WebSphere Application Server와 함께 MQSeries Adapter Kernel 사용』 및 MQSeries Adapter Builder 문서를 참조하십시오.

## **MQSeries Adapter Builder**

가상적으로 응용프로그램에 대한 어댑터를 빌드할 수 있게 해주는 그래픽

사용자 인터페이스(GUI). 사용자 인터페이스는 MQSeries Integrator의 사용자 인터페이스와 유사합니다. 자세한 정보는 MQSeries Adapter Builder Information Center를 참조하십시오.

## MQSeries Adapter Kernel

응용프로그램 프로그래밍 인터페이스(API), C 및 Java로 된 여러 실행 가능 프로그램, 여러 구성 파일의 세트. 커널은 어댑터의 전개 및 실행을 가능하게 합니다. 커널은 직접 어댑터를 지원하는 것 외에도 단순한 메시지 라우팅과 같은 관련 기능을 수행합니다. 또한 메시지 작성, 트랜잭션 제어, 추적 및 MQSeries나 기타 메시징 소프트웨어와의 인터페이스 등과 같은 인프라스트럭처 서비스도 제공합니다.

커널은 소스 어댑터 또는 대상 어댑터가 실행되는 각 컴퓨터에 설치됩니다.

MQSeries Adapter Offering을 사용하면 비즈니스 프로세스와 각 응용프로그램을 미들웨어, 메시지 세부사항 및 기타 응용프로그램과 분리할 수 있습니다. 메시징에 대한 공통 인터페이스는 기존의 응용프로그램이나 비즈니스 프로세스를 변경하지 않고 새로운 응용프로그램을 추가할 수 있게 해줍니다.

MQSeries Adapter Kernel은 두 개의 티어로 전개할 수 있습니다. 한 티어는 런타임의 소스측이고 다른 티어는 런타임의 대상측입니다. 2 티어 전개는 효율적인 조작과 낮은 관리 오버헤드를 제공합니다. 라우팅 및 전달용 세번째 티어는 런타임의 소스측과 타겟측 둘 사이에 있을 필요가 없습니다. 그러나 선택적으로 복잡한 라우팅, 데이터 변환 및 데이터 조정과 같은 중개를 수행하기 위해 MQSeries Integrator를 추가할 수 있습니다.

지정된 곳 외에 이 문서의 나머지 부분은 MQSeries Adapter Kernel에만 국한됩니다. MQSeries Adapter Builder에 대한 자세한 정보는 해당 제품의 정보 센터를 참조하십시오.

## 커널에 대한 정보

간단히 말하면 런타임 즉, 사용자가 빌드한 커널과 어댑터는 다음과 같은 목적을 갖습니다.

1. 데이터를 소스 응용프로그램에서 대상 응용프로그램으로 전송.

2. MQSeries 또는 다른 메시징 소프트웨어를 사용함으로써 커널을 통해 라우트 되는 응용프로그램 공통 형식의 메시지로 소스 응용프로그램 데이터 변환.
3. 메시지를 대상 응용프로그램으로 라우트.
4. 데이터를 대상 응용프로그램으로 가져오는 방법 판별.
5. 커널을 통해 라우트된 메시지 형식에서 어댑터를 통해 라우트되는 대상 응용 프로그램 형식으로 데이터 변환.

이 절에서는 커널의 상위 레벨 기능을 다룹니다. 이러한 기능은 15 페이지의 『런타임 플로우』에서 보다 자세히 설명합니다.

커널에는 두 가지 측면이 있습니다.

- 소스측은 소스 응용프로그램으로부터 메시지가 수신될 때 시작되고 메시지가 메시지 큐에 놓일 때 끝납니다.
- 대상측은 메시지 큐로부터 메시지가 수신될 때 시작되고 메시지가 대상 응용프로그램으로 송신될 때 끝납니다.

각각은 일반적으로 다른 컴퓨터에 상주하지만, 둘 다 같은 컴퓨터에 있을 수도 있습니다.

6 페이지의 그림1을 참조하십시오. 다음 순서로 설명합니다.

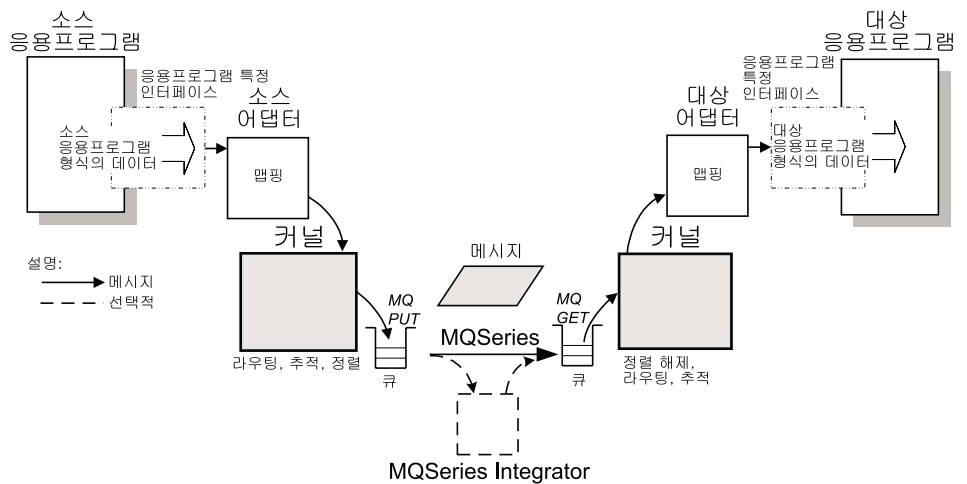
### 커널의 소스측

1. 커널의 소스측에서 소스 응용프로그램은 응용프로그램 특정 인터페이스를 사용하여 소스 응용프로그램 형식으로 데이터를 MQSeries Adapter Builder에 빌드된 소스 어댑터로 송신합니다. 각 메시지 유형에 대해 예를 들어, 『판매 주문 추가』 또는 『고객 레코드 동기화』에 대해 다른 소스 어댑터가 필요합니다. 응용프로그램 특정 인터페이스는 반드시 MQSeries Adapter Offering 외부에서 개발해야 합니다. 응용프로그램 특정 인터페이스의 정확한 특성은 소스 응용프로그램 또는 대상 응용프로그램의 특성에 따라 달라집니다. API 호출 및 사용자 엑시트, 파일 읽기 및 쓰기, 데이터베이스 트리거, 메시지 큐 등이 예입니다.

소스 어댑터가 소스 응용프로그램 프로세스 중 실행됨에 유의하십시오. 소스 어댑터가 작동하려면 소스 어댑터를 포함하는 디먼이나 서버가 실행되고 있어야 합니다.

2. 소스 어댑터는 빌드 방법에 따라 그 기능을 수행합니다. 일반적인 기능은 데이터 요소의 변환 즉, 본문 데이터에 대하여 소스 응용프로그램 형식에서 통합 메시징 형식으로 요소를 맵핑하는 것입니다. 본문 데이터와 제어 값을 나타내는 추가 메타 데이터는 커널 메시지 홀더 오브젝트에 들어갑니다.
3. 소스 어댑터가 원시 어댑터를 사용하여 메시지 홀더 오브젝트를 커널로 전달할 때 커널은 메시지 홀더 오브젝트의 제어 값(메시지 제어 값)을 사용하여 메시지 홀더 오브젝트를 통신 메시지 형식으로 정렬하고 통신 메시지를 라우팅하는 것을 제어합니다.  
메시지에 특정 메시지 제어 값이 없는 경우 커널은 디폴트 값 또는 구성 파일에서 확보한 메시지 제어 값을 사용할 수 있습니다. 메시지 제어 값 정의에 대한 내용은 17 페이지의 『메시지 제어 값』을 참조하십시오.
4. 커널은 메시지 정렬, 간단한 라우팅 및 선택적으로 추적과 같은 기능을 수행합니다. 13 페이지의 『메시지 및 메시지 형식』, 14 페이지의 『라우팅 및 전달』 및 31 페이지의 『추적』을 참조하십시오.

그림 1. MQSeries Adapter Offering 개요



### 커널의 소스측에서 대상측으로 전달

5. 커널은 원시 어댑터를 사용하여 메시지를 적절한 메시지 큐에 놓습니다.

소스측에서 사용되는 두 가지 송신 메소드가 있습니다.

- `sendMsg`는 메시지를 송신하고 즉시 리턴합니다. `sendMsg` 메소드는 `begin`, `commit` 및 `rollback` 메소드를 사용하여 메시지를 트랜잭션적으로 송신할 수도 있습니다. 즉, 다른 조작이 성공적으로 완료될 경우 메시지를 송신할 수 있습니다. 자세한 정보는 30 페이지의 『트랜잭션 기능』을 참조하십시오.
- `sendRequestResponse`는 메시지를 송신하고 응답을 기다립니다. `sendRequestResponse` 메소드는 트랜잭션적으로 발행할 수 없습니다. 세번째 메소드 `sendResponse`는 송신자가 응답을 요청할 때 커널의 대상측에서 사용됩니다.

MQSeries 또는 기타 메시징 소프트웨어는 메시지를 전송합니다. 9 페이지의 『MQSeries 또는 기타 메시징 소프트웨어의 역할』 부분을 참조하십시오. MQSeries Adapter Offering을 지원하려면 메시징 소프트웨어가 미리 구성되어 있어야 합니다.

선택적으로 MQSeries Integrator가 대상으로 커널에 구성된 경우, MQSeries Integrator는 중개 기능을 수행할 수 있습니다. 9 페이지의 『MQSeries Integrator 역할』을 참조하십시오. 마지막 목적지인 메시지 큐가 MQSeries Integrator 규칙이나 메시지 플로우에 따라 구성된 경우, MQSeries Integrator는 메시지를 메시지 큐로 송신합니다.

메시지는 적절한 메시지 큐에 도착합니다.

### 커널의 대상측

6. 커널의 대상측에 런타임과 대상 응용프로그램 사이의 인터페이스를 위한 두 가지 가능한 전달 모델이 있습니다.

- 가장 일반적인 모델은 *PUSH*로서, 커널은 대상 응용프로그램으로의 메시지 전달 시작 및 관리를 담당합니다. *PUSH* 모델은 일반적으로 MQSeries Adapter Offering을 지원하기 위해 대상 응용프로그램을 변경할 필요가 없습니다.
- *PULL* 모델에서 대상 응용프로그램은 메시지 수신 관리를 담당합니다. *PULL* 모델의 경우 MQSeries Adapter Offering을 지원하기 위해 대상

응용프로그램을 변경해야 합니다. 대상 응용프로그램은 대상 응용프로그램에 대한 커널 인터페이스를 관리해야 합니다.

PUSH 모델의 경우 대상측에서 메시지를 가져와 전달하기 전에 먼저 사용자가 커널 프로세스를 시작해야 합니다. 102 페이지의 『커널 시작』을 참조하십시오.

PUSH 모델에서 커널은 메시지 큐에서 메시지를 가져옵니다. 추적이 사용 가능하다면 추적을 수행합니다. 적절한 대상 어댑터를 선택하여 계속해서 메시지를 라우트합니다. 일반적으로 메시지 유형 각각에 대해 다른 대상 어댑터가 필요합니다.

7. 커널은 메시지를 적절한 대상 어댑터로 전달합니다. 대상 어댑터는 그 안에 빌드된 기능을 수행합니다. 일반적인 기능은 통합 메시징 형식 요소에서 대상 응용프로그램 형식의 요소로 맵핑하는 것입니다.

대상 어댑터는 MQSeries Adapter Kernel 어댑터 디먼이나 WebSphere Application Server에 의해 호스팅될 수 있습니다. 후자에 대한 논의는 31 페이지의 『WebSphere Business Integrator 및 WebSphere Application Server 와 함께 MQSeries Adapter Kernel 사용』을 참조하십시오.

8. 대상 어댑터는 MQSeries Adapter Offering 외부에서 개발된 응용프로그램 특정 인터페이스를 사용하여 대상 응용프로그램 형식으로 데이터를 대상 응용프로그램으로 송신합니다.
9. 대상 어댑터가 메시지를 전달할 때 그 메시지는 메시지 큐로부터 확약됩니다. 그러면 메시지가 큐에서 제거됩니다.
10. 소스 어댑터가 수신확인을 요청하기 위해 메시지 제어 값을 설정한 경우, 커널은 sendResponse 메소드를 사용하여 메시지 전달 수신확인 또는 대상 어댑터 출력을 소스 어댑터로 전달합니다.
11. 오류가 발생하면 커널이 원래 메시지를 오류 큐에 놓습니다. 커널이 원래 메시지를 오류 큐에 놓을 수 없으면 확약이 발생하지 않습니다.



## **MQSeries 또는 기타 메시징 소프트웨어의 역할**

MQSeries Adapter Offering 통신 메시지는 메시지 큐를 통해 전송됩니다. 메시지 큐는 MQSeries 또는 JMS(Java Message Service)와 같은 메시징 소프트웨어에 의해 제공됩니다. MQSeries Adapter Offering이 전송하는 메시지는 다음과 같은 유형의 큐를 사용합니다.

- MQSeries Adapter Offering 용어로 리시브 큐. 메시지를 수신하기 위한 기본 입력 큐로 사용됩니다. 대상 응용프로그램마다 여러 개의 리시브 큐가 있을 수 있습니다.
- MQSeries Adapter Offering 용어로 오류 큐. 리시브 큐에서 가져온 메시지를 처리할 수 없을 때 사용됩니다.
- 옵션으로 리플라이 큐. sendRequestResponse 메소드와 함께 사용됩니다.

MQSeries Adapter Offering은 다음 메시지 유형과 같은 특정 MQSeries 기능을 사용합니다.

- 데이터그램, sendMsg 메소드에 사용됩니다.
- 요청, sendRequestResponse 메소드에 사용됩니다.
- 응답, sendRequestResponse 메소드 및 sendResponse 메소드에 사용됩니다.

MQSeries는 선택적으로 응용프로그램 특정 인터페이스로서 역할을 할 수 있습니다.

검증된 MQSeries 및 MQSeries Adapter Offering 목록에 대해서는 119 페이지의 『부록B. 검증 구성』을 참조하십시오. 지원되는 MQSeries 및 기타 소프트웨어 버전 목록에 대해서는 36 페이지의 『소프트웨어』를 참조하십시오.

## **MQSeries Integrator 역할**

MQSeries Integrator는 선택적으로 MQSeries Adapter Kernel로 전개될 수 있습니다. 중개를 위해 여러가지 잠재적 요구사항을 충족시키는 데 사용할 수 있습니다.

- 복합 라우팅, 즉, 메시지 헤더 또는 메시지 본문의 내용에 기초한 라우팅. 라우팅은 메시지 본문 변경 내용에 따라 동적으로 변경될 수 있습니다. 복합 라우팅 및 단순 라우팅에 대해서는 14 페이지의 『라우팅 및 전달』을 참조하십시오.
- 데이터 변환, 즉, 다른 문서 형식으로 변경.

- 데이터 조정, 즉, 메시지 본문 내용 변경. 예를 들어 소스 응용프로그램이 값 each를 필드에 제공했지만 대상 응용프로그램이 필드 값을 ea로 예상한 경우, 데이터 조정은 제공된 값을 예상 값으로 바꿉니다.

MQSeries Integrator를 사용하여 사용자의 사이트에서 대부분의 라우팅을 수행할 수 있습니다. 또한 MQSeries Adapter Kernel의 라우팅 기능을 덜 사용할 수도 있습니다.

MQSeries Integrator 및 MQSeries Adapter Offering의 검증된 구성 목록을 보려면 119 페이지의 『부록B. 검증 구성』을 참조하십시오. 지원되는 MQSeries Integrator 및 기타 소프트웨어 버전 목록에 대해서는 36 페이지의 『소프트웨어』를 참조하십시오.

---

## 커널 작업 방법

다음 항목이 이 절에서 설명됩니다.

- 『커널 런타임 구성요소』
- 13 페이지의 『메시지 및 메시지 형식』
- 14 페이지의 『라우팅 및 전달』
- 15 페이지의 『런타임 플로우』

---

## 커널 런타임 구성요소

사용자가 빌드한 어댑터, 개발한 사용자 정의 코드 및 MQSeries Adapter Kernel이 함께 실행되는 경우, MQSeries Adapter Offering 기능을 제공합니다.

커널 런타임의 주요 구성요소는 다음과 같습니다.

### 소스 어댑터

응용프로그램의 데이터를 통합 메시지 형식(본문 데이터)으로 변환하기 위해 특정 응용프로그램용으로 빌드된 소프트웨어(일반적으로 MQSeries Adapter Builder를 사용하여). 소스 어댑터는 일반적으로 응용프로그램 프로세스 내에서 또는 별도의 프로세스로서 소스 응용프로그램과 같은 기계에서 실행됩니다. 파일, C 구조 및 Java 오브젝트가 소스 데이터의 예입

니다. 통합 메시징 형식의 예는 일반적으로 OAG 또는 RosettaNet과 같은 업계 표준을 따르는 XML입니다.

### 메시지 홀더

통합 메시지와 커널이 사용하는 기타 제어 데이터를 캡슐화하는 데 커널이 사용하는 메타 데이터용 컨테이너. 메타 데이터의 예는 소스 및 대상 응용프로그램의 응용프로그램 ID(논리 ID), 메시지 범주(예를 들어, OAG), 메시지 유형(예를 들어, "Purchase Order") 및 송신 또는 수신되는 통신 메시지(본문 데이터)입니다.

### 원시 어댑터

메시지 홀더 오브젝트를 송신하고 수신하는 데 사용되는 소프트웨어. 메시지를 송신하는 경우, 원시 어댑터는 단순한 데이터 라우팅과 하나 이상의 통신 전송 메커니즘을 지원하는 기능을 제공합니다. 단순한 데이터 라우팅은 메시지 범주나 메시지 유형과 같은 메시지 홀더 오브젝트의 메타 데이터를 기초로 합니다. 메시지는 비동기식으로 또는 동기식으로 송신될 수 있습니다. 기본 통신 전송 메커니즘이 트랜잭션 메시징을 지원하는 경우, 메시지는 1 단계 트랜잭션 제어 하에 송신될 수 있습니다. 트랜잭션 지원은 사용되는 전송 메커니즘 기능으로 제한됩니다. 메시지 홀더 오브젝트는 전송 메커니즘이 사용하는 통신 메시지 형식으로 정렬됩니다. 통신 메시지가 수신되면 원시 어댑터가 메시지를 다시 메시지 홀더 오브젝트로 정렬합니다.

### 어댑터 디먼

어댑터 작업 프로그램을 인스턴스화하는 프로세스. 시작 후 어댑터 디먼은 활동 상태로 남습니다. 각 대상 응용프로그램의 경우, 응용프로그램 리시브 큐 각각에 대해 하나의 어댑터 디먼이 있을 수 있습니다.

### 어댑터 작업 프로그램

각 메시지를 적절한 대상 어댑터로 전달하는 프로세스. 각 작업 프로그램은 하나의 원시 어댑터를 관리합니다. 어댑터 디먼은 작업 프로그램을 작성하고 시작합니다.

여러 작업 프로그램을 두는 목적은 대상 어댑터로의 멀티스레드 메시지 전달을 가능하게 하는 것입니다. 각 작업 프로그램은 원시 어댑터와 함께 하

나의 스레드를 처리할 수 있습니다. 하나의 작업 프로그램만 있는 경우, 대상 어댑터로 메시지를 전달한 다음 대상 응용프로그램으로 전달하는 것은 단일 스레드입니다.

원시 어댑터 관리 외에도 작업 프로그램은 다음 작업도 수행합니다.

- 추적이 사용 가능하면 추적 클라이언트를 인스턴스화합니다.
- 각 대상 응용프로그램에 적절한 로그인 클래스를 인스턴스화합니다.
- 메시지의 본문 유형 및 본문 범주를 기초로 대상 어댑터를 선택합니다.
- 메시지를 선택된 대상 어댑터로 송신합니다.
- 약속을 수행할 수 없는 경우, 롤백을 수행하고 해당 어댑터 디먼하에서 다른 모든 작업 프로그램의 플래그를 설정하며 원시 어댑터를 종료합니다. 그러면 메시지에 문제가 있다는 신호를 보냅니다. 모든 작업 프로그램을 종료하면 다른 작업 프로그램이 같은 결과로 같은 문제점 메시지를 다시 처리하지 않습니다.
- 다른 작업 프로그램이 종료를 위해 설정한 플래그를 인식하는 경우 그 자체와 원시 어댑터를 종료합니다.

### 대상 어댑터

데이터를 통합 메시징 형식(본문 데이터)에서 대상 응용프로그램이 필요로 하는 데이터 유형으로 변환하기 위해 특정 응용프로그램용으로 빌드된(일반적으로 MQSeries Adapter Builder를 사용하여) 소프트웨어. 대상 어댑터는 메시지를 전달하기 위해 대상 응용프로그램에서 필요한 API를 호출합니다. 대상 어댑터는 응용프로그램 또는 응용프로그램 클라이언트와 같은 기계에서 실행됩니다.

### Java 서비스 세션 bean 어댑터

WebSphere Application Server와 같은 EJB 서버에서 호스팅되는 Java 언어 EJB 어댑터 유형.

### 구성 구성요소

큐 이름과 같은 오브젝트로 논리 ID를 해석하는 데 사용되는 데이터. 구성 데이터는 파일이나 WebSphere Business Integrator 제품의 LDAP 구조에 지정할 수 있습니다. 데이터는 다음과 같은 커널 구성을 제어합니다.

- 메시지 정렬 및 라우팅

- 설치 확인
- 통신 모드
- 추적

완전한 구성 파일 설명에 대해서는 74 페이지의 『구성 파일』을 참조하십시오. 커널과 함께 작업할 제품 구성에 대한 정보는 WebSphere Business Integrator 문서를 참조하십시오.

### 추적 구성요소

추적 메시지를 기록하는 소프트웨어. 대부분의 커널 구성요소는 추적 구성요소를 사용합니다. 추적 개요에 대해서는 31 페이지의 『추적』을, 추적 세부사항에 대해서는 *Problem Determination Guide*를 참조하십시오.

---

## 메시지 및 메시지 형식

MQSeries 및 MQSeries Adapter Offering에서 메시지는 한 프로그램에서 송신하는 데이터 콜렉션이며, 다른 프로그램을 위한 것입니다. 메시지 형식은 언제나 특정 시간 메시지 플로우의 메시지 위치에 따라 달라집니다. MQSeries Adapter Kernel은 다음과 같이 세 가지 유형의 메시지를 지정합니다.

- 통합 메시지 -- 대상 응용프로그램으로 송신하기 위해 XML과 같은 다른 형식으로 변환된 소스 응용프로그램의 데이터로 구성된 메시지. 통합 메시지는 메시지 본문 데이터로 메시지 홀더 오브젝트에 삽입됩니다. XML은 데이터 표현을 위한 표준입니다. XML 형식은 문서 유형 정의(DTD)로 정의됩니다. DTD는 문서(이 경우는 메시지 본문)의 정식 정의를 포함하는 하나 이상의 파일입니다. 메시지 본문은 응용프로그램 공통 형식인 것이 좋습니다. 메시지 본문의 형식이 독점적이거나 달리 특수화될 수도 있지만 이러한 유형의 형식은 권장하지 않습니다.

비즈니스 오브젝트 문서(BOD)는 MQSeries Adapter Offering이 통합 메시지에 메시지 본문을 정의하는 데 사용할 수 있습니다. BOD는 한 조직 내에서 또는 조직들 간에 이동하는 표준 비즈니스 프로세스의 표현입니다. 『구매 주문 추가』, 『제품 사용가능성 표시』 및 『판매 주문 추가』 등이 예입니다. BOD는 개방형 응용프로그램 그룹(OAG)에 의해 XML로 정의됩니다. BOD 사용을 권장하기는 하지만 필수적인 것은 아닙니다.

- **메시지 홀더 오브젝트** -- MQSeries Adapter Kernel에 고유한 제어 값을 나타내는 통합 메시지 및 추가 헤더 메타 데이터를 포함하는 오브젝트. 소스 어댑터는 메시지 홀더 오브젝트를 작성하고 적절한 제어 정보를 설정하며, 송신될 통합 메시지가 있는 경우 본문 데이터를 설정합니다. 대상 어댑터는 메시지 홀더 오브젝트를 수신하고 본문 데이터를 가져오며 본문 데이터를 대상 응용프로그램 특정 데이터로 변환합니다. 소스 어댑터 및 대상 어댑터는 MQSeries Adapter Builder를 사용하여 작성됩니다.
- **통신 메시지** -- 사용 중인 통신 전송 고유의 메시징 형식으로 변환된 메시지 홀더 오브젝트와 통신 전송 고유 정보. 일부 통신 전송은 두 가지 이상의 메시징 형식을 지원합니다. 일반적으로 통신 메시지와 결합된 커널 헤더 메타 데이터 값은 통신 전송에 의해 응용프로그램 데이터로 간주됩니다. 더 자세한 정보는 113 페이지의 『부록A. 통신 모드』를 참조하십시오. MQSeries 전송 예는 MQSeries 고유 메시지 헤더와 정렬된 메시지 홀더 오브젝트로 구성됩니다. 고유 MQSeries 형식으로 다음이 포함됩니다.
  - MQSeries가 추가되는 MQSeries 메시지 헤더
  - MQSeries Integrator가 사용되는 경우, 버전 고유 메시지 헤더
    - MQSeries Integrator 버전 1.1이 사용되는 경우, MQSeries Integrator 버전 1 메시지 헤더
    - MQSeries Integrator 버전 2가 사용되는 경우, MQSeries Integrator 버전 2 메시지 헤더
  - 제어 값을 나타내는 커널 고유 헤더 메타 데이터
  - 통합 메시지(본문 데이터)

MQSeries Adapter Offering 메시지 헤더에 사용된 관련 필드와 그 설명 목록에 대해서는 121 페이지의 『부록C. 메시지 헤더』를 참조하십시오.

---

## 라우팅 및 전달

커널은 소스 어댑터의 각 메시지를 라우트하여 적절한 대상 어댑터로 전달합니다. 라우팅은 다음 두 단계로 수행됩니다.

1. 커널의 소스측은 메시지를 적절한 메시지 큐에 놓습니다.

2. 커널의 대상측은 메시지 큐에서 메시지를 가져오고 적절한 대상 어댑터를 호출합니다.

라우팅은 여러 가지 요소로 결정됩니다.

- 메시지 큐. 가장 기본적인 레벨에서 MQSeries Adapter Offering 라우팅을 지원하도록 메시지 큐를 반드시 구성해야 합니다.
- 메시지의 메시지 제어 값. 소스 논리 ID, 목적지 논리 ID, 응답 대상 논리 ID, 본문 범주, 본문 유형, 트랜잭션 ID, 메시지 ID, 요청된 수신확인 및 시간 소인 등이 포함됩니다. 세부사항은 17 페이지의 『메시지 제어 값』을 참조하십시오. 메시지의 목적지 논리 ID는 커널의 구성 파일을 대체할 수 있습니다. 라우팅은 각 메시지 헤더의 이러한 메시지 제어 값 변경에 따라 동적으로 변경될 수 있습니다. 그러나 메시지 본문 데이터(통합 메시지) 내용은 라우팅을 결정할 수 없습니다.
- 커널 구성 파일의 메시지 제어 값. 파일은 목적지 논리 ID, 큐 이름 및 연관된 대상 어댑터를 지정할 수 있습니다. 이 파일을 편집하여 구성을 결정하고 수정하십시오. 자세한 정보는 74 페이지의 『구성 파일』을 참조하십시오.
- 선택적으로 MQSeries Integrator를 사용하여 복합 라우팅을 포함한 메시지를 중개할 수 있습니다. 라우팅은 메시지 본문 변경 내용에 따라 동적으로 변경될 수 있습니다. 9 페이지의 『MQSeries Integrator 역할』을 참조하십시오. 대조적으로 MQSeries Adapter Offering은 단순한 라우팅만을 수행할 수 있습니다. 단순 라우팅은 메시지의 메시지 제어 값과 구성 파일의 연관된 메시지 제어 값의 조합을 기초로 합니다. 메시지 본문의 내용과는 관계가 없습니다.

메시지 전달을 수신확인하도록 커널에 요청할 수 있습니다. 이는 응용프로그램 레벨의 수신확인입니다.

---

## 런타임 플로우

이 절에서는 런타임 플로우 즉, 커널이 일반적인 프로덕션 환경에서 메시지를 송신, 라우트, 추적 및 전달하는 방법에 대하여 자세히 설명합니다. 런타임 플로우 다이어그램에 대해서는 16 페이지의 그림2를 참조하십시오.

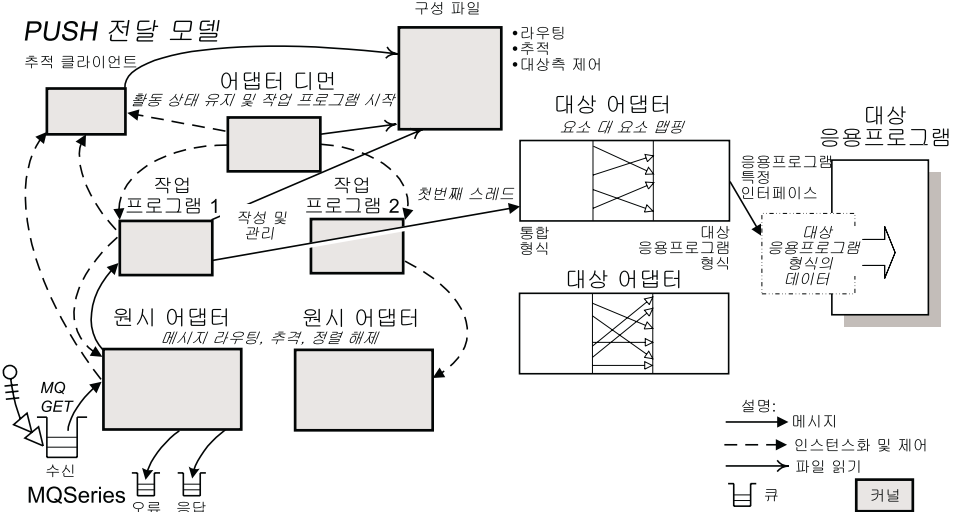
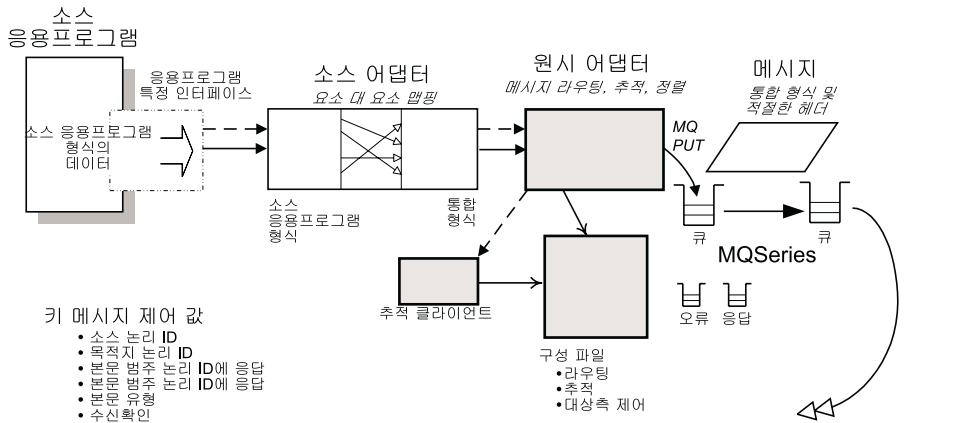


그림 2. 메시지 정렬, 송신, 라우트 및 추적 -- 개요

### 커널의 소스측

이 절에서는 커널의 소스측에서의 런타임 플로우를 설명합니다. 즉, 데이터가 소스 응용프로그램에서 소스 어댑터를 통해 통신 전송으로 이동하는 방식을 설명합니다. 21 페이지의 『커널의 대상측』에서는 통신 전송에서 데이터가 대상으로 이동하는 방식에 대하여 설명합니다.



1. 소스 어댑터는 응용프로그램 특정 인터페이스를 사용하여 메시지를 소스 응용프로그램으로부터 확보합니다. 일반적으로 응용프로그램 특정 인터페이스에서 소스 어댑터를 호출합니다.
2. 소스 어댑터는 MQSeries Adapter Builder로 빌드된 기능을 수행합니다. 일반적으로 소스 응용프로그램 형식의 데이터를 응용프로그램 공통 통합 형식 (메시지 본문에 대해서)으로 변환합니다.

기능의 일부로서 소스 어댑터는 메시지 제어 값을 MQSeries Adapter Kernel 헤더에 놓고 이러한 값을 사용하여 메시지를 봉합합니다. 처음 다섯 개의 메시지 제어 값은 정렬과 라우팅을 결정하고 마지막 값은 수신확인을 결정합니다.

### 메시지 제어 값

#### 소스 논리 ID

소스 응용프로그램의 논리 ID. 항상 메시지에 필요합니다.

#### 목적지 논리 ID

대상 응용프로그램의 논리 ID. 메시지에 없는 경우 구성 파일의 디폴트 값이 대신 사용됩니다. 구성 파일에서 메시지에 없는 값 대신 여러 목적지 논리 ID를 사용할 수 있습니다.

#### 응답 대상 논리 ID

응답하는 응용프로그램의 논리 ID는 응답이 요청될 경우 송신됩니다. 디폴트는 메시지의 소스 논리 ID입니다.

#### 본문 범주

OAG나 RosettaNet과 같은 메시지 응용프로그램 유형을 나타냅니다. 항상 메시지에 필요합니다.

#### 본문 유형

『판매 주문 추가』 또는 『재고 동기화』와 같은 특정 메시지 용도를 나타냅니다. 항상 메시지에 필요합니다.

#### 수신확인 요청

소스 응용프로그램이 응답을 요청하는지 여부를 판별합니다. 응답은 다음과 같은 형식일 수 있습니다.

- 대상 응용프로그램으로부터의 데이터 응답
- OAG BOD 확인 메시지

주: BOD 확인 메시지는 OAG에 의해 사전정의됩니다. 그 본문 범주는 OAG이고 본문 유형은 CONFIRM\_BOD\_003입니다. 데이터를 포함할 수도 있습니다.

이러한 응답은 응용프로그램 레벨 수신확인입니다.

커널이 sendRequestResponse 메소드를 사용하여 메시지를 송신하는 경우 sendRequestResponse 메소드로 수신되는 첫번째 응답만 사용됩니다. 원래 메시지가 여러 목적지로 전송되고 응답을 요청하는 경우(권장사항 아님) 첫번째 응답만 다시 소스 응용프로그램으로 송신됩니다.

디폴트는 수신확인이 아니므로 응답이 요청되거나 송신되지 않습니다.

3. 소스 어댑터는 원시 어댑터를 초기화하고 전달합니다.
  - 소스 어댑터가 실행 중인 응용프로그램의 논리 ID.
  - 메시지 제어 값과 메시지 본문 데이터를 포함하는 메시지 홀더 오브젝트.
4. 원시 어댑터는 구성 파일을 찾아 해당 소스 논리 ID의 추적이 사용 가능한지를 판별합니다. 추적이 사용 가능하면 원시 어댑터가 추적 클라이언트를 인스턴스화합니다.
5. 추적 클라이언트는 구성 파일을 찾아 다른 값을 사용하고 확보할 추적 레벨을 판별합니다. 추적 클라이언트는 추적 레벨을 사용하여 추적 메시지를 필터합니다. 추적 개요는 31 페이지의 『추적』을 참조하고 추적 세부사항은 *Problem Determination Guide*를 참조하십시오.
6. 원시 어댑터는 목적지 논리 ID를 메시지 홀더 오브젝트에서 찾습니다. 목적지 논리 ID가 있는 경우 사용됩니다.
  - 목적지 논리 ID가 없는 경우, 원시 어댑터가 소스 논리 ID, 본문 범주 및 본문 유형을 기초로 구성 파일에서 디폴트 목적지 논리 ID를 찾습니다.
  - 소스 논리 ID를 기초로 원시 어댑터는 다음과 같은 순서로 구성 파일에서 본문 범주 및 본문 유형 값에 대한 다단계 찾기를 수행합니다.
    - a. 특정 본문 범주 및 본문 유형 값에 대하여
    - b. 특정 본문 범주 값 및 디폴트 본문 유형 값에 대하여
    - c. 디폴트 본문 범주 값 및 특정 본문 유형 값에 대하여

d. 디폴트 본문 범주 및 본문 유형 값에 대하여

주: 커널은 구성 파일에서 값을 찾을 때마다 이러한 다단계 찾기를 사용합니다.

7. 이전 단계에서 판별된 각 목적지 논리 ID에 대하여 원시 어댑터는 목적지 논리 ID, 본문 범주 및 본문 유형을 기초로 통신 모드를 찾습니다. 다음 통신 모드가 지원됩니다.

<b>MQPP</b>	커널이 MQSeries 기본 서비스를 사용하여 메시지를 전송합니다.
<b>MQRFH1</b>	커널이 MQSeries를 사용하여 메시지를 전송하고 MQSeries Integrator 버전 1.1을 사용하여 메시지를 중개합니다.
<b>MQRFH2</b>	커널이 MQSeries를 사용하여 메시지를 전송하고 MQSeries Integrator 버전 2를 사용하여 메시지를 중개합니다.
<b>MQBD</b>	커널이 MQSeries 기본 서비스를 사용하여 메시지를 전송하지만 본문 데이터만을 송신하고 수신합니다.
<b>MQ</b>	커널이 MQSeries를 사용하여 메시지를 전송합니다.
<b>JMS</b>	커널은 JMS(Java Message Service)를 사용하여 메시지를 전송합니다.
<b>FILE</b>	커널이 메시지를 파일에 놓고 파일에서 가져옵니다. 이 모드는 진단 목적으로만 제공됩니다.

각 통신 모드에서 메시지 구조는 다릅니다. 13 페이지의 『메시지 및 메시지 형식』 부분을 참조하십시오. 통신 모드에 대한 자세한 정보는 113 페이지의 『부록A. 통신 모드』를 참조하십시오.

주: MQSeries Integrator가 사용된 경우 MQSeries Integrator가 메시지를 송신하는 최종 목적지에서는 MQSeries Integrator가 메시지를 수신하기 위해 사용하는 것과 동일한 통신 모드를 사용해야 합니다.

8. 통신 모드를 기초로 원시 어댑터는 그 자체 내의 서브클래스를 인스턴스화하여 메시지를 처리합니다. 서브클래스는 논리 메시지 전달 서비스라고 합니다. 각 통신 모드는 다른 논리 메시지 전달 서비스 서브클래스를 가집니다.

원시 어댑터가 목적지 논리 ID, 본문 범주 및 본문 유형을 논리 메시지 전달 서비스로 전달합니다.

9. 논리 메시지 전달 서비스 서브클래스는 메시지를 송신하는 데 필요한 매개변수를 찾습니다. 예를 들어 통신 모드가 MQPP인 경우 매개변수에는 리시브, 리플라이 및 오류 큐의 형식과 이름이 포함됩니다. 전달된 목적지 논리 ID, 본문 범주 및 본문 유형을 기초로 논리 메시지 전달 서비스는 구성 파일에서 다단계 찾기를 수행합니다.
  - a. 특정 본문 범주 및 본문 유형 값에 대하여
  - b. 특정 본문 범주 값 및 디폴트 본문 유형 값에 대하여
  - c. 디폴트 본문 범주 값 및 특정 본문 유형 값에 대하여
  - d. 디폴트 본문 범주 및 본문 유형 값에 대하여

이 경우 논리 메시지 전달 서비스는 메시지를 라우트하고 정렬해야 하는 모든 정보를 가집니다.

10. 논리 메시지 전달 서비스는 다음 작업을 수행합니다.
  - 통신 모드와 형식에 맞게 메시지를 정렬합니다. 각 통신 모드는 형식이 달리 지정되지 않는 한 디폴트 형식을 사용합니다. 예를 들어 통신 모드가 MQRFH2인 경우 논리 메시지 전달 서비스는 적절한 헤더를 작성하고 MQSeries를 사용하여 전송할 메시지의 구조를 설정하고 MQSeries Integrator 버전 2를 사용하여 중개합니다.
  - 메시지를 송신합니다. 예를 들어 통신 모드가 MQRFH2인 경우 메시지를 적절한 MQSeries 메시지 큐에 놓습니다.
11. 메시지를 송신하는 데 사용할 수 있는 두 가지 메소드가 있습니다.
  - 원시 어댑터가 sendMsg 메소드를 사용하여 메시지를 송신하는 경우 원시 어댑터는 응답을 기다리지 않습니다.
  - 원시 어댑터가 sendRequestResponse 메소드를 사용하여 메시지를 송신하는 경우, 논리 메시지 전달 서비스가 응답을 기다립니다. 논리 메시지 전달 서비스를 사용하여 원시 어댑터는 구성 파일에 설정된 수신 시간 종료 기간 동안 리플라이 큐를 모니터링합니다.  
수신 시간 종료 기간은 소스 응용프로그램 ID, 본문 범주 및 본문 유형을 기초로 합니다.

- 수신확인이 수신되면 원시 어댑터가 메시지를 리턴합니다.
  - 수신확인이 수신 시간 종료 기간 내에 수신되지 않으면 원시 어댑터가 메시지를 리턴하지 않습니다.
12. MQSeries 또는 기타 메시징 소프트웨어가 그 구성 방법에 따라 메시지를 전송합니다. 선택적으로 MQSeries Integrator는 중개 서비스를 수행합니다. 9 페이지의 『MQSeries Integrator 역할』을 참조하십시오.
  13. 소스 어댑터가 더 이상 원시 어댑터를 필요로 하지 않으면 원시 어댑터를 닫고 자원을 해제합니다.

## 커널의 대상측

이 절에서는 독립형 MQSeries Adapter Kernel을 사용하여 대상측에서 메시지를 수신하고 처리하는 것에 대하여 설명하고 WebSphere Application Server와 함께 커널을 사용하는 고급 수준의 설명을 제공합니다. 커널 대상측에서 JMS, WebSphere Business Integrator의 JMS Listener 구성요소 및 WebSphere Application Server와 함께 커널을 사용하는 것에 대한 설명은 31 페이지의 『WebSphere Business Integrator 및 WebSphere Application Server와 함께 MQSeries Adapter Kernel 사용』을 참조하십시오. 이 절에서는 커널이 대상 응용프로그램으로의 메시지 전달을 시작하고 관리하는 것을 책임지는 PUSH 전달 모델에 대하여 설명합니다. 모델에 대한 간단한 설명은 145 페이지의 『전달 모델』을 참조하십시오.

### 어댑터 작업 프로그램 개요

이 절에서는 MQSeries Adapter Kernel 어댑터 작업 프로그램의 구조와 작동에 대하여 설명합니다. MQSeries Adapter Kernel 구조에 대한 하나의 가정은 대상 응용프로그램이 다른 응용프로그램과의 통합 데이터 플로우에 적극적으로 참여하지 않는다는 것입니다. 즉 이 응용프로그램은 일반적으로 메시지를 처리하기 위해 적극적으로 폴링하지 않습니다. 이러한 경우 메시지 데이터는 대상 응용프로그램에 넣어져야 합니다. 어댑터 작업 프로그램은 서비스 인터페이스 유형 범위를 선택 및 호출하여 메시지 데이터를 응용프로그램이나 다른 서비스로 넣습니다.

MQSeries Adapter Kernel은 독립형 디먼(어댑터 디먼)이나 Enterprise JavaBeans 응용프로그램 서버(현재는 IBM WebSphere Application Server 확장판)에서 실행되는 어댑터 작업 프로그램을 호스팅할 수 있습니다. 메시지는 사용되는 대상 환

경 유형에 따라 서로 다른 방법으로 어댑터 작업 프로그램에 도달합니다. 독립형 어댑터 디먼이 사용되는 경우 이 디먼은 메시지를 수신하기 위해 원시 어댑터를 사용하는 하나 이상의 독립형 어댑터 작업 프로그램을 호스팅합니다. EJB 서버가 사용되는 경우 JMS 리스너 구성요소는 메시지를 수신하여 작업 프로그램 메시지 bean 으로 전달합니다.

사용된 대상 환경에 관계없이 어댑터 작업 프로그램이 메시지를 수신한 후 이 메시지를 적절한 대상 어댑터로 전달합니다. 그리고나서 대상 어댑터는 메시지를 대상 응용프로그램으로 전달하는 데 필요한 작업을 수행합니다. 특정 대상 응용프로그램에 대한 작업을 하기 위해 대상 어댑터가 작성됩니다. 어댑터 디먼, 응용프로그램 서버, 독립형 어댑터 작업 프로그램 및 작업 프로그램 메시지 bean은 주어진 소스 또는 대상 응용프로그램에 고유하지 않습니다.

어댑터 작업 프로그램은 EAB(Enterprise Access Builder) 명령 어댑터와 EJB 서비스 세션 bean의 두 가지 유형의 대상 어댑터 인터페이스를 핸들링합니다. 각 어댑터 유형에는 적절한 환경을 설정하고, 어댑터에 필요한 추가 구성 정보에 액세스하며, 어댑터 조작에 필요한 기타 하위 레벨 작업을 수행하는 핸들러가 포함됩니다. 사용된 핸들러는 구성 파일에 나열된 어댑터 유형에 따라 다릅니다. 두 가지 유형의 핸들러는 다음과 같은 추가 작업을 수행합니다.

- EAB 핸들러는 대상 어댑터에 연결 정보를 제공하는 데 사용되는 로그온 클래스를 확보하고 IBM CCF(Common Connector Framework) 런타임을 초기화합니다. 로그온 클래스는 대상 응용프로그램의 논리 ID로 전달되어 응용프로그램에 고유한 로그온 정보를 확보하는 데 사용됩니다.
- EJB 핸들러는 JNDI(Java Naming and Directory Interface) 연결을 확보한 후 서비스 세션 bean의 리모트 인터페이스와, 서비스 세션 bean에 액세스하는 데 필요한 기타 정보를 확보합니다.

독립형 어댑터 디먼 내에서의 어댑터 작업 프로그램의 기본 프로세스 플로우는 다음과 같습니다.

1. 시동시 어댑터 디먼은 커널의 구성 파일에서 제공되는 정보에 따라 하나 이상의 독립형 어댑터 작업 프로그램을 인스턴스화합니다. 응용프로그램의 논리 ID 및 선택적 본문 범주와 본문 유형 값이 어댑터 디먼으로 전달됩니다. 추가 구성 값을 확보하는 데 본문 범주 및 본문 유형 값이 사용됩니다.

2. 각 독립형 어댑터 작업 프로그램은 다음 작업을 수행합니다.
  - a. 어댑터 작업 프로그램은 원시 어댑터를 인스턴스화하고 메시지 수신을 시작합니다. 각 메시지는 트랜잭션 제어 하에서 수신되고 메시지 홀더 오브젝트로서 어댑터 작업 프로그램으로 리턴됩니다.
  - b. 수신된 각 메시지에 대하여 어댑터 작업 프로그램은 구성 파일로부터 메시지를 처리하기 위해 대상 어댑터 명령 유형을 검색하고 해당 명령 유형에 대한 적절한 핸들러를 확보합니다.
  - c. 핸들러는 구성 파일에서 대상 어댑터 인스턴스를 인스턴스화하는 데 필요한 추가 정보를 확보합니다. 대상 어댑터를 인스턴스화하고 메시지를 전달합니다.
  - d. 메시지가 성공적으로 처리되면(즉 예외, 오류 또는 불량 리턴 데이터가 없는 경우) 메시지는 수신 메시지 큐로부터 확약됩니다. 메시지가 성공적으로 처리되지 않으면 오류 큐로 들어갑니다. 메시지가 성공적으로 처리되지 못하고 오류 큐로 들어가지도 못한 경우 메시지는 롤백되고 모든 작업 프로그램은 종료합니다.

WebSphere Application Server 내에서 어댑터 작업 프로그램의 기본 프로세스 플로우의 다음과 같습니다.

1. WebSphere Application Server 확장판의 EJB 서버와 함께 작동되는 JMS 리스너 프로세스는 JMS 메시지를 수신합니다. 그리고 나서 메시지를 처리하기 위해 작업 프로그램 메시지 bean을 확보합니다. 응용프로그램의 논리 ID 및 선택적 본문 범주 및 본문 유형 값은 작업 프로그램 메시지 bean 환경의 일부입니다. 추가 구성 값을 확보하는 데 본문 범주 및 본문 유형 값이 사용됩니다.
2. 각 작업 프로그램 메시지 bean은 다음 작업을 수행합니다.
  - a. 작업 프로그램 메시지 bean은 원시 어댑터를 인스턴스화하고 원시 어댑터에서 receiveMsg 메소드를 사용하여 이를 JMS 메시지로 전달합니다. 원시 어댑터는 JMS 메시지를 메시지 오브젝트로 변환하고 메시지 홀더 오브젝트를 리턴합니다.
  - b. 각 수신된 메시지 홀더 오브젝트에 대하여 어댑터 작업 프로그램은 구성 파일에서 메시지 홀더 오브젝트를 처리하기 위한 대상 어댑터 명령 유형을 검색하고 해당 명령 유형에 적절한 핸들러를 확보합니다.

- c. 핸들러는 구성 파일에서 대상 어댑터 인스턴스를 인스턴스화하는 데 필요한 추가 정보를 확보합니다. 대상 어댑터를 인스턴스화하고 메시지 홀더 오브젝트를 대상 어댑터로 전달합니다.
- d. 메시지 홀더 오브젝트가 성공적으로 처리되면(즉, 예외, 오류 또는 불량 리턴 데이터가 없는 경우) 메시지는 수신 메시지 큐로부터 예약됩니다. 메시지 홀더 오브젝트가 성공적으로 처리되지 않으면 오류 큐로 들어갑니다. 메시지가 성공적으로 처리되지 못하고 오류 큐로 들어가지도 못한 경우 메시지는 톨백되고 모든 작업 프로그램은 종료합니다.

어댑터 디먼 및 독립형 어댑터 작업 프로그램을 사용하는 대상측 런타임 플로우는 다음과 같습니다.

1. 각 대상 응용프로그램의 리시브 큐에 대해 하나의 어댑터 디먼이 있습니다. 어댑터 디먼이 시작됩니다.

시동시 응용프로그램 ID 역할을 하는 이름이 부여됩니다. 일반적으로 각 어댑터 디먼 이름은 목적지 논리 ID 즉, 대상 응용프로그램의 논리 ID를 기초로 합니다. 예를 들어 어댑터 디먼이 그 목적지 논리 ID가 ABC인 대상 응용프로그램을 서비스하는 중이라면 어댑터 디먼 이름은 ABCdaemon입니다.

시동시 어댑터 디먼으로 전달할 수 있는 기타 매개변수에는 본문 범주와 본문 유형이 포함됩니다. 원시 어댑터는 나중에 이들을 사용하여 통신 모드 및 수신 메시지에 대한 리시브 큐를 판별합니다.

어댑터 디먼 시작에 대한 지시사항은 102 페이지의 『커널 시작』을 참조하십시오.

2. 시작할 때 어댑터 디먼은 구성 파일을 찾아서 해당 어댑터 디먼 이름에 대하여 추적이 사용 가능한지 여부를 판별합니다. 추적이 사용 가능하면 어댑터 디먼이 추적 클라이언트를 인스턴스화합니다.

추적에 대한 세부사항은 *Problem Determination Guide*를 참조하십시오.

3. 시작할 때 어댑터 디먼은 첫번째 작업 프로그램을 인스턴스화하고 이 작업 프로그램으로 어댑터 디먼의 이름, 메시지 본문 범주 및 본문 유형을 전달합니다.

4. 첫번째 작업 프로그램은 구성 파일을 찾아서 해당 어댑터 디먼 이름에 대하여 추적이 사용 가능한지 여부를 판별합니다. 추적이 사용 가능하다면 첫 번째 작



업 프로그램이 추적 클라이언트를 인스턴스화하고 추적 클라이언트는 구성 파일을 봄으로써 추적 레벨을 판별합니다. 올바른 추적 레벨 목록은 *Problem Determination Guide*를 참조하십시오.

5. 첫번째 작업 프로그램은 어댑터 디먼의 응용프로그램 ID에 기초하여 인스턴스화하고 시작할 최소 작업 프로그램 수를 나타내는 값을 구성 파일에서 찾아봅니다.

첫번째 작업 프로그램은 또한 종속 응용프로그램 ID도 찾습니다. 종속 응용프로그램 ID는 작업 프로그램이 서비스하는 응용프로그램의 이름입니다. 이 ID는 나중에 원시 어댑터로 전달됩니다.

6. 어댑터 디먼은 최소 작업 프로그램 수에 대해 첫번째 작업 프로그램을 조회합니다.
7. 어댑터 디먼은 첫번째 작업 프로그램을 시작한 다음 최소 작업 프로그램 수를 시작합니다.

여러 작업 프로그램을 두는 목적은 대상 어댑터로의 멀티스레드 메시지를 전달을 가능하게 하는 것입니다. 각 작업 프로그램은 원시 어댑터와 함께 하나의 스레드를 처리할 수 있습니다. 하나의 작업 프로그램만 있는 경우, 대상 어댑터로 메시지를 전달한 다음 대상 응용프로그램으로 전달하는 것은 단일 스레드입니다.

AIX 시스템에서 두 개의 스케줄 정책을 스레드에 사용할 수 있습니다: 프로세스 기본 스케줄 및 시스템 기본 스케줄. 프로세스에 기초한 스케줄(디폴트)에서 모든 사용자 스레드는 운영 체제(OS) 커널 스레드 풀에 맵핑되며 가상 프로세서 풀에서 실행됩니다. 시스템 기본 스케줄에서 각 사용자는 단일 OS 커널 스레드에 맵되고 단일 가상 프로세서에서 실행됩니다. AIX에서 C 실행 가능 파일로부터 호출되는 C 소스 어댑터를 사용 중이라면 시스템 기본 스케줄을 반드시 사용해야 합니다. AIX에서의 스레드 스케줄 설정에 대해서는 52 페이지의 6 단계를 참조하십시오.

Windows systems, HP-UX, Solaris 및 OS/400에서는 프로세스 기본 스케줄만 지원됩니다.

다른 작업 프로그램은 첫번째 작업 프로그램이 수행하는 다음 단계도 수행합니다.

8. 각 작업 프로그램은 연관된 원시 어댑터를 인스턴스화합니다. 각 작업 프로그램과 연관된 하나의 원시 어댑터가 있습니다. 종속 응용프로그램 ID, 본문 범

주 및 본문 유형이 원시 어댑터에 전달됩니다. 원시 어댑터는 이러한 세 값을 사용하여 통신 모드를 판별하고 논리 메시지 전달 서비스를 사용하여 수신되는 메시지에 대한 형식과 리시브 큐를 판별합니다. 이 프로세스는 송신 메시지에 사용되는 프로세스와 유사합니다.

9. 원시 어댑터는 협약 제어하에서 리시브 큐로부터 통신 메시지를 가져와 메시지 홀더 오브젝트로 변환합니다. 원시 커널 헤더에 대한 것을 제외한 통신 전송에 고유한 헤더를 모두 제거합니다.
10. 원시 어댑터는 메시지 홀더 오브젝트를 작업 프로그램으로 전달하는데, 작업 프로그램은 본문 범주, 본문 유형 및 요청된 수신확인 값을 메시지의 원시 커널 헤더로부터 읽습니다.

종속 응용프로그램 ID, 본문 범주 및 본문 유형을 기초로 하여 작업 프로그램은 구성 파일에서 다음과 같은 순서로 호출할 대상 명령 유형에 대한 단단계 찾기를 수행합니다.

- a. 특정 본문 범주 및 본문 유형 값에 대하여
- b. 특정 본문 범주 값 및 디폴트 본문 유형 값에 대하여
- c. 디폴트 본문 범주 값 및 특정 본문 유형 값에 대하여
- d. 디폴트 본문 범주 및 본문 유형 값에 대하여

대상 명령 유형에 기초하여 작업 프로그램은 적절한 대상 어댑터 유형 핸들러와 특정 어댑터 유형을 처리하는 Java 클래스를 판별합니다. 특정 대상 어댑터를 인스턴스화합니다.

11. 어댑터 유형 핸들러에는 EAB 명령 대상 어댑터 핸들러와 대상 어댑터로 작동하는 EJB 서비스 세션 bean 핸들러의 두 종류가 있습니다. 서로 다른 두 종류의 어댑터 핸들러는 다음과 같이 작동합니다.

주: 대상 어댑터로 작동하는 EJB 서비스 세션 bean 핸들러는 Windows NT 플랫폼에서 WebSphere Application Server와 함께 실행되는 WebSphere Business Integrator에서만 지원됩니다.

- EAB 명령 대상 어댑터 핸들러가 호출되면 CCF(Common Connector Framework) 환경을 초기화하고 구성 파일에서 확보한 이름의 로그온 클래스를 설정하며 구성 파일에서 확보한 이름의 EAB 대상 어댑터를 호출합니다.

- EJB 서비스 세션 bean 대상 어댑터는 적절한 구성 정보를 확보하기 위하여 WebSphere Business Integrator 및 WebSphere Application Server 와 상호작용하고 EJB 서비스 세션 bean을 호출해야 합니다. 커널 대상측에서 JMS, WebSphere Business Integrator의 JMS Listener 구성요소 및 WebSphere Application Server와 함께 커널을 사용하는 것에 대한 설명은 31 페이지의 『WebSphere Business Integrator 및 WebSphere Application Server와 함께 MQSeries Adapter Kernel 사용』을 참조하십시오.

12. 각 어댑터 유형에는 다음과 같은 서로 다른 인터페이스와 필수 지원 클래스가 있습니다.

- EAB 대상 어댑터 명령에는 세 가지 호출 메소드가 있습니다. 이 메소드는 다음과 같은 순서로 실행됩니다.
  - a. 메시지 입력 설정 메소드는 대상 어댑터로 처리할 메시지를 설정합니다.
  - b. 실행 메소드는 메시지 입력 설정 메소드를 사용하여 대상 어댑터에 놓은 메시지를 처리한 다음 기다립니다.
    - 1) 대상 어댑터는 MQSeries Adapter Builder를 사용하여 빌드된 기능을 수행합니다. 일반적으로 통합 메시지의 데이터를 대상 응용프로그램 형식으로 변환합니다. 요소를 요소에 맵합니다.
    - 2) 대상 어댑터는 응용프로그램 특정 인터페이스를 사용하여 메시지를 대상 응용프로그램으로 송신합니다.
    - 3) 대상 응용프로그램의 특성에 따라 대상 응용프로그램이 응답을 다시 대상 어댑터로 송신하거나 송신하지 않습니다.
  - c. 메시지 출력 가져오기(*get*) 메소드는 대상 어댑터로부터 응답을 가져옵니다. 응답은 대상 응용프로그램이 메시지를 수신했음을 나타냅니다. 데이터도 포함할 수 있습니다.
- EJB 서비스 세션 bean은 하나의 메소드를 호출하는데, 이 메소드는 TerminalDataContainer 오브젝트를 필요로 합니다. 이 메소드에 의해 리턴된 데이터는 응답 데이터로 간주되고 TerminalDataContainer 유형의 오브젝트여야 합니다.

13. 대상 어댑터 명령이 예외를 전달하거나 BOD 응답 확약(오류를 나타낼 수 있음)을 갖지 않는 경우 작업 프로그램은 원시 어댑터를 사용하여 리시브 큐로부터 수신된 메시지를 확약합니다.
14. 수신확인이 요청되면 작업 프로그램이 원시 어댑터에 대한 `sendResponse` 메소드를 호출합니다.
  - 대상 어댑터가 응답을 작성했다면 원래 메시지의 응답 대상 논리 ID를 응답 메시지의 목적지 논리 ID 필드에 넣습니다.
  - 대상 어댑터가 응답을 작성하지 않은 경우, 작업 프로그램이 완료 상태를 포함하는 BOD 응답 확약 메시지를 작성합니다.
    - 오류가 없는 경우 완료 상태는 성공적입니다.
    - 오류가 있는 경우 완료 상태는 오류 조건으로 설정됩니다.
15. 응답이 송신됩니다.
  - a. 작업 프로그램은 작성된 응답 메시지를 원시 어댑터로 송신합니다.
  - b. 원시 어댑터는 응답 메시지를 리플라이 큐에 놓습니다.
  - c. 원시 어댑터는 수신된 원시 메시지에 따라 응답 메시지를 송신합니다.
    - MQSeries 요청 메시지인 경우 원시 어댑터가 MQSeries 요청 메시지로부터 응답 큐 정보를 가져옵니다. 이 큐 정보는 메시지의 목적지 논리 ID를 대체합니다.
    - MQSeries 요청 메시지가 아닌 경우 원시 어댑터가 `sendMsg` 메소드를 사용하여 응답을 송신합니다.
16. 예외 또는 오류 상태의 BOD 확인 응답 메시지의 경우 작업 프로그램은 어댑터 디먼과 같은 디렉토리에 있는 `EpicSystemExceptionFilennnnnnnnn.log` 예외 파일에 예외 메시지를 기록하는데, 여기서 `nnnnnnnnn`은 로그 파일 번호입니다. 또한 WebSphere Business Integrator 클래스가 설치되어 있는 경우, 이 클래스는 예외를 WebSphere Business Integrator Solution Management 구성요소로 송신합니다. 106 페이지의 『예외 메시지』를 참조하십시오.
17. 예외 또는 오류 상태인 BOD 응답 확약 메시지의 경우, 작업 프로그램은 원시 어댑터로 하여금 원래 메시지를 오류 큐에 놓도록 지시합니다. 오류 큐의 이름은 원래 메시지의 종속 논리 ID, 본문 범주 및 본문 유형에 기초하여 구성 파일로부터 가져옵니다.

종속 응용프로그램 ID, 본문 범주 및 본문 유형을 기초로 하여 작업 프로그램은 구성 파일에서 다음과 같은 순서로 다단계 찾기를 수행합니다.

- a. 특정 본문 범주 및 본문 유형 값에 대하여
  - b. 특정 본문 범주 값 및 디폴트 본문 유형 값에 대하여
  - c. 디폴트 본문 범주 값 및 특정 본문 유형 값에 대하여
  - d. 디폴트 본문 범주 및 본문 유형 값에 대하여
- 원시 어댑터가 오류 메시지를 오류 큐에 놓을 수 있다면 원시 어댑터는 리시브 큐로부터 메시지를 확약하도록 지시받습니다.
  - 원시 어댑터가 오류 메시지를 오류 큐에 놓을 수 있다면 다음 사항이 발생합니다.
    - a. 작업 프로그램은 원시 어댑터가 확약하지 않고 롤백하도록 지시합니다.
    - b. 작업 프로그램이 그 어댑터 디먼 하의 모든 작업 프로그램을 종료하도록 지시하는 플래그를 설정합니다. 이것은 메시지에 문제가 있다는 것을 말합니다. 모든 작업 프로그램을 종료하면 다른 작업 프로그램이 같은 결과를 가져올 이 문제점 메시지를 다시 처리하지 않습니다.
    - c. 메모리 부족 오류가 발생하면, 작업 프로그램이 현재 메시지 처리를 완료할 때 중단하도록 플래그를 설정한다는 점을 제외하고는 다른 모든 예외와 같은 방법으로 예외가 처리됩니다. 그러면 다른 작업 프로그램이 더 많은 메모리를 사용할 수 있게 됩니다.
18. 원시 어댑터가 작업 프로그램에게 작업이 완료되었음을 알리는 경우, 작업 프로그램은 다음 두 플래그를 점검합니다.
- 이 작업 프로그램의 정지 여부. 이는 Java 메모리 부족 조건으로 인해 야기될 수 있습니다.
  - 모든 작업 프로그램의 정지 여부.
19. 플래그가 설정되면 작업 프로그램이 정지합니다. 플래그가 설정되지 않으면 작업 프로그램이 다음 메시지를 처리합니다. 작업 프로그램은 원시 어댑터가 메시지를 수신하도록 요청합니다.
20. 응답 메시지가 리플라이 큐에 놓이거나 오류 메시지가 오류 큐에 있는 경우 다음 사항이 발생합니다.

- a. MQSeries 또는 다른 메시징 소프트웨어는 이를 다시 커널의 소스측에 전달합니다.
- b. 소스 어댑터가 원시 어댑터의 `sendRequestResponse` 메소드를 호출한 경우, 커널은 리플라이 큐로부터 메시지를 검색하여 소스 어댑터로 리턴합니다. 소스 어댑터가 `sendMsg` 메소드를 호출한 경우 커널은 메시지를 소스 응용프로그램 리시브 큐에 놓습니다.

## 트랜잭션 기능

트랜잭션은 분리할 수 없는 작업 단위로서 실행해야 하는 조작 세트입니다. 트랜잭션을 구성하는 모든 조작이 성공적이라면 트랜잭션은 **확약됩니다**. 즉 모든 조작이 수행됩니다. 트랜잭션을 구성하는 하나 이상의 조작이 실패할 경우 트랜잭션은 **롤백됩니다**. 즉 어떤 조작도 수행되지 않습니다. 소스 어댑터는 MQSeries Adapter Kernel의 트랜잭션 기능을 사용함으로써 일련의 조작을 하나의 단위로 수행할 수 있으며, 트랜잭션이 확약될 경우 모든 조작은 성공적으로 수행되고 트랜잭션이 롤백될 경우 어떤 조작도 발생하지 않습니다.

트랜잭션 기능은 MQSeries Adapter Builder를 사용하거나 `begin`, `rollback` 및 `commit` 메소드를 커널의 Java API에서 `EpicNativeAdapter` 클래스에 사용하여 어댑터로 빌드할 수 있습니다. 트랜잭션 메소드가 올바르게 않은 컨텍스트에서 호출되면(예를 들어, 먼저 `begin` 메소드를 호출하지 않고 `commit` 메소드를 호출하거나 다른 트랜잭션 범위 내에서 `begin` 메소드를 호출) 커널이 호출을 무시하고 추적 경고를 발행합니다. API 사용에 대해서는 109 페이지의 『제5장 MQSeries Adapter Kernel API 사용』을 참조하십시오.

### 제한사항

다음과 같은 제한사항이 커널의 트랜잭션 기능과 연관됩니다.

- 트랜잭션은 `sendRequestResponse` 메소드로 지원되지 않습니다.
- 중첩 트랜잭션(즉, 다른 트랜잭션 내에서 호출되는 트랜잭션)은 지원되지 않습니다.
- 모든 통신 모드가 트랜잭션을 지원하지는 않습니다. 자세한 내용은 113 페이지의 『부록A. 통신 모드』를 참조하십시오.

## 추적

추적 메시지는 커널의 특정 지점에서 메시지 처리 상태를 포함합니다. 추적 메시지를 사용하여 커널 또는 어댑터에 대한 문제점을 진단하는 데 도움을 줄 수 있습니다. MQSeries Adapter Kernel *Problem Determination Guide*에서는 커널 추적 사용에 대해 설명합니다.

---

## WebSphere Business Integrator 및 WebSphere Application Server와 함께 MQSeries Adapter Kernel 사용

이 절에서는 MQSeries Adapter Kernel을 WebSphere Business Integrator 및 WebSphere Application Server 제품과 함께 사용하는 것에 대하여 설명합니다. 추가 세부사항은 WebSphere Business Integrator 문서를 참조하십시오.

### JMS 리스너

WebSphere Business Integrator에서는 MQSeries Adapter Kernel 및 WebSphere Application Server 확장판과 함께 대상 응용프로그램으로 메시지를 전달하는 또 다른 방법을 제공하는 JMS 리스너라는 구성요소를 제공합니다. JMS 리스너는 WebSphere Application Server의 EJB(Enterprise JavaBeans) 서버 내에서 실행됩니다. 이 절에서는 JMS 리스너의 기능에 대한 개요를 제공합니다. WebSphere Business Integrator 및 JMS 리스너 구성을 비롯하여 추가 세부사항은 WebSphere Business Integrator 문서를 참조하십시오. JMS 리스너를 대상으로 인식하도록 MQSeries Adapter Kernel을 구성하는 것에 대한 정보는 67 페이지의 『커널 구성』을 참조하십시오. JMS 리스너를 대상으로 사용하는 것은 어댑터 디먼으로 메시지를 송신하는 것과 동일합니다.

JMS 리스너를 사용하려면 먼저 MQSeries Adapter Kernel 작업 프로그램 메시지 bean 및 커널의 대상측에 대한 Java 서비스 세션 bean 어댑터 또는 EAB 어댑터 중 하나를 전개해야 합니다. MQSeries Adapter Builder를 사용하여 이러한 작업을 수행하십시오. WebSphere Business Integrator 환경에서 WebSphere Application Server 내의 커널 조작용 JMS Listner가 어댑터 작업 프로그램 대신 메시지를 수신하고 적절한 어댑터 작업 프로그램을 호출한다는 점을 제외하고

는 독립형 어댑터 디먼과 그 조적이 유사합니다. 독립형 MQSeries Adapter Kernel 환경에서 어댑터 디먼은 어댑터 작업 프로그램을 시작하고 어댑터 작업 프로그램은 차례로 메시지를 직접 수신합니다.

MQSeries Adapter Kernel이 JMS Listener와 함께 기능할 때 이벤트 순서는 다음과 같습니다.

1. JMS 리스너는 JMS 큐를 모니터링하면서 JMS 메시지 오브젝트를 EJB 클라이언트나 EJB가 아닌 응용프로그램 중 어느 하나로부터 수신합니다.
2. JMS 리스너는 작업 프로그램 메시지 *bean*을 인스턴스화하고 메시지 오브젝트를 작업 프로그램 메시지 *bean*으로 전달합니다. 작업 프로그램 메시지 *bean*은 특정 클라이언트와 연관된 임시 데이터를 캡슐화하는 엔터프라이즈 *bean*의 한 유형인 세션 *bean*의 인스턴스입니다.
3. 작업 프로그램 메시지 *bean*은 JMS 메시지 오브젝트를 MQSeries Adapter Kernel 메시지 홀더 오브젝트로 변환합니다.
4. 메시지 헤더 값에 기초하여 커널은 EAB 어댑터나 EJB 어댑터 중 하나를 호출합니다. 호출될 어댑터 유형이 EAB 어댑터인 경우 데이터 플로는 독립형의 경우에서와 같습니다. 호출될 어댑터 유형이 EJB 어댑터인 경우 EJB 핸들러가 호출되고 다음 작업을 수행합니다.
  - 호출할 올바른 서비스 세션 *bean*(홈 인터페이스), 적절한 호출 메소드 및 TerminalDataContainer 오브젝트에 대한 메소드의 입력 매개변수 유형을 판별합니다.
  - Mapper 클래스를 사용하여 서비스 세션 *bean*에 대한 적절한 TerminalDataContainer 데이터 구조로 메시지 홀더 오브젝트에 있는 응용프로그램 데이터를 변환합니다. TerminalDataContainer 오브젝트에는 메시지 홀더 오브젝트의 메타 데이터와 응용프로그램 오브젝트가 들어 있습니다. 많은 경우에 있어 응용프로그램 오브젝트는 메시지 홀더 오브젝트의 본문 데이터 XML 문서 문자열입니다.
  - 서비스 세션 *bean*을 호출하여 TerminalDataContainer 오브젝트를 서비스 세션 *bean*의 적절한 메소드로 전달합니다. Java 서비스 어댑터의 일부인 서비스 세션 *bean*은 메시지의 대상입니다.



5. 응답이 요청된 경우 작업 프로그램 메시지 bean은 응답 TerminalDataContainer 오브젝트를 메시지 홀더 오브젝트로 변환하고 그 응답을 원시 어댑터를 사용하여 송신합니다.
6. 오류가 발생하면 작업 프로그램 메시지 bean은 메시지 홀더 오브젝트를 원시 어댑터를 사용하여 오류 큐에 넣습니다.

---

## 자국어 지원

MQSeries Adapter Kernel에서는 Java 어댑터가 사용될 때 자국어 지원을 제공합니다. C 어댑터에 대해서는 자국어 지원이 제공되지 않습니다.



---

## 제2장 커널 설치 계획

이 장에서는 MQSeries Adapter Kernel의 구성요소와 필수조건을 나열합니다.

최신 정보에 대해서는 다음 MQSeries 제품군 웹 사이트를 참조하십시오.

[www.ibm.com/software/ts/mqseries/](http://www.ibm.com/software/ts/mqseries/)

IBM은 여기에 표시된 정보를 갱신할 권리를 확보하고 있습니다. 지원되는 소프트웨어의 레벨과 관련된 최신 정보는 다음을 참조하십시오.

[www.ibm.com/software/ts/mqseries/platforms/supported.html](http://www.ibm.com/software/ts/mqseries/platforms/supported.html)

---

## 하드웨어

MQSeries Adapter Kernel은 다음 하드웨어에서 실행됩니다.

- Windows NT 4.0, Service Pack 5 이상 또는 Windows 2000, Service Pack 1을 실행 중인 IBM PC 또는 호환 시스템.
- AIX 버전 4.3.2 또는 4.3.3을 실행 중인 IBM RS/6000 시스템.
- HP-UX 버전 11.0을 실행 중인 HP Series 9000 시스템.
- Solaris 버전 8을 실행 중인 Sun SPARC 또는 UltraSPARC 시스템.
- OS/400 버전 4.4 또는 4.5를 실행 중인 IBM AS/400 또는 iSeries 시스템.

주: OS/400에 MQSeries Adapter Kernel을 설치할 경우 Windows 시스템이 AS/400 시스템과 인터페이스해야 합니다. 세부사항은 38 페이지의 『OS/400 설치 필수조건』을 참조하십시오.

MQSeries Adapter Kernel은 제품 코드 및 데이터용으로 최소 25MB의 디스크 공간이 필요합니다.

어댑터를 보유하는 데 사용 가능한 디스크 공간이 충분한지 확인하십시오. 그 크기는 데이터 구조의 크기, 맵핑의 복잡도 및 사용된 사용자 정의 코드에 따라 달라집니다. 다음은 Windows 시스템에서 일부 다른 어댑터 크기의 예입니다. 사용

자 사이트의 어댑터는 추가 디스크 공간을 필요로 할 수 있습니다. 각 예는 어댑터 소스, 컴파일된 어댑터 코드, API 소스 및 컴파일된 API 코드를 MB 또는 KB로 나타냅니다.

- 판매 주문 추가용 소스 어댑터: 1.89MB
- 고객 레코드 동기화용 대상 어댑터: 389KB
- 채고 레코드 동기화용 대상 어댑터: 161KB
- 항목 동기화용 대상 어댑터: 249KB
- 판매 주문 동기화용 대상 어댑터: 579KB

또한 커널과 어댑터용 작업 공간으로 최소 20MB를 허용하십시오. 작업 공간 요구사항은 큐의 수와 크기 및 추적 파일의 크기와 같은 요소에 따라 다양할 수 있습니다.

---

## 소프트웨어

이 절에서는 MQSeries Adapter Kernel과 함께 사용하도록 지원된 소프트웨어를 나열합니다. 지원되는 레벨이 표시됩니다. 119 페이지의 『부록B. 검증 구성』을 참조하십시오. 개발 시스템에는 C 컴파일러가 필요하지만 프로덕션 시스템에는 필요 없다는 사실에 유의하십시오. 여기에 나열된 C 컴파일러는 정상적으로 MQSeries Adapter Kernel과 함께 테스트된 것입니다. 다른 C 컴파일러가 커널과 함께 올바르게 작동할 수도 있으나 공식적으로는 지원되지 않습니다.

Windows 시스템:

- Microsoft Windows NT 버전 4.0, Service Pack 5 이상 또는 Microsoft Windows 2000, Service Pack 1. Microsoft Windows의 버전과 Service Pack을 판별하려면 Windows 탐색기를 열고 도움말 > **Windows** 정보를 누르십시오.
- Microsoft Visual C++ 6.0 컴파일러.
- SupportPac MA88을 갖춘 MQSeries 버전 5.2.
- IBM JDK(Java Development Kit) 버전 1.2.2 또는 1.3.

주: Windows NT 및 Windows 2000은 현재 MQSeries Adapter Kernel이 JDK 버전 1.3을 지원하는 유일한 플랫폼입니다.

#### AIX:

- AIX 운영 체제 버전 4.3.2 또는 4.3.3.
- AIX용 IBM C Set++ 버전 3.1.3.
- SupportPac MA88을 갖춘 MQSeries 버전 5.2.
- Java Development Kit 버전 1.2.2. JDK 1.3은 지원되지 않습니다.
- X Window 시스템(X11R5 이상). 설치 시 필요하지만 런타임 시에는 필요 없음.

#### HP-UX:

- HP-UX 운영 체제 버전 11.0.
- HP-UX C/ANSI C 컴파일러. 세부사항은 readme.txt 파일을 참조하십시오.
- SupportPac MA88을 갖춘 MQSeries 버전 5.2.
- Java Development Kit 버전 1.2.2. JDK 1.3은 지원되지 않습니다.
- X Window 시스템(X11R5 이상). 설치 시 필요하지만 런타임 시에는 필요 없음.

#### Solaris:

- Solaris 운영 환경 버전 8.
- Sun Workshop 컴파일러 C/C++. 세부사항은 readme.txt 파일을 참조하십시오.
- SupportPac MA88을 갖춘 MQSeries 버전 5.2.
- Java Development Kit 버전 1.2.2. JDK 1.3은 지원되지 않습니다.
- X Window 시스템(X11R5 이상). 설치 시 필요하지만 런타임 시에는 필요 없음.

#### OS/400:

- 다음 프로그램을 포함한 OS/400 운영 체제 버전 4.4 또는 4.5.
  - Java Toolkit 및 Java Developer Kit 버전 1.2.2. JDK 1.3은 지원되지 않습니다. Java Toolkit 및 Java Developer Kit은 사용 허가된 프로그램 번호 5769-JV1로 제공됩니다. AS/400 시스템에 MQSeries Adapter Kernel

을 설치하는 데 필요한 Java Developer Kit 버전에 대해서는 『OS/400 설치 필수조건』을 참조하십시오.

- 사용 허가된 프로그램 번호 5769-SS1, 옵션 12로 제공되는 호스트 서버 옵션.
  - 사용 허가된 프로그램 번호 5769-SS1, 옵션 30으로 제공되는 Qshell Interpreter.
  - 사용 허가된 프로그램 번호 5769-TC1으로 제공되는 TCP/IP.
  - 사용 허가된 프로그램 번호 5769-CX2로 제공되는 AS/400용 통합 언어 환경 C.
- SupportPac MA88을 갖춘 MQSeries 버전 5.2.

OS/400에 MQSeries Adapter Kernel을 설치하기 위한 추가 요구사항에 대해서는 『OS/400 설치 필수조건』을 참조하십시오.

다음 제품이 MQSeries Adapter Kernel과 함께 지원됩니다.

- SupportPac MA88을 갖춘 MQSeries 버전 5.2.

주: MQSeries가 사용되지 않는 경우 JMS(Java Message Service) 구현과 같은 다른 메시징 소프트웨어 제품을 사용해야 합니다.

- MQSeries Integrator 버전 1.1
- MQSeries Integrator 버전 2

검증된 MQSeries Adapter Kernel, MQSeries 및 MQSeries Integrator 구성 목록에 대해서는 119 페이지의 『부록B. 검증 구성』을 참조하십시오.

---

## OS/400 설치 필수조건

이 절에서는 MQSeries Adapter Kernel을 AS/400이나 iSeries 시스템에 설치하기 위한 필수조건을 설명합니다. MQSeries Adapter Kernel을 AS/400 시스템에 설치하는 방법에 대한 세부사항은 48 페이지의 3 단계를 참조하십시오. AS/400 터미널은 고유하게 Java 그래픽을 지원하지 않으므로 커널의 Java GUI 설치 프로그램을 실행하려면 Windows 시스템과 같은 그래픽 사용 가능 워크스테이션이 필요합니다. 워크스테이션은 다음 방법 중 하나로 AS/400 시스템과 인터페이스할 수 있습니다.

- 모든 그래픽이 AS/400 시스템에서 처리되어 워크스테이션에 표시되는 리모트 AWT를 통해. 이는 『리모트 AWT 사용』에서 더 자세히 설명됩니다.
- 워크스테이션이 그래픽을 처리하고 표시하는 접속된 클라이언트로서. 이는 40 페이지의 『접속된 클라이언트 사용』에서 더 자세히 설명됩니다.

이 절에서는 Windows 시스템이 그래픽 사용 가능 워크스테이션으로 사용 중이라고 가정합니다.

## 리모트 AWT 사용

리모트 AWT가 사용되는 경우, Java 그래픽 처리가 AS/400에서 이루어지고 그래픽이 AS/400 시스템에 접속된 클라이언트 워크스테이션에 표시됩니다. 이 절에서는 리모트 AWT를 사용하여 AS/400 시스템에 MQSeries Adapter Kernel을 설치하기 위해 필요한 요구사항을 설명합니다.

다음 프로그램이 OS/400에 반드시 설치되어야 합니다.

- Java Toolkit 및 Java Developer Kit 버전 1.2.2. Java Toolkit 및 Java Developer Kit은 사용 허가된 프로그램 번호 5769-JV1로 제공됩니다. OS/400의 리모트 AWT 기능은 Java Developer Kit에서 제공됩니다.
- 사용 허가된 프로그램 번호 5769-TC1으로 제공되는 TCP/IP. TCP/IP에 대한 자세한 정보는 [www.ibm.com/servers/eserver/series/library/](http://www.ibm.com/servers/eserver/series/library/)의 AS/400 라이브러리에서 *AS/400TCP/IP Fastpath Setup Information* 및 *AS/400 TCP/IP Configuration* 문서를 참조하십시오.

다음은 워크스테이션 요구사항입니다.

- Windows 95, Windows 98, Windows NT 또는 Windows 2000을 실행 중인 IBM PC 시스템(또는 호환 가능).
- AS/400 시스템으로의 TCP/IP 연결.
- JDK 1.2.2 이상.

리모트 AWT를 설정하고 시작하려면 다음 단계를 수행하십시오.

1. JDK 1.2.2 이상이 워크스테이션에 설치되었는지 확인하십시오.
2. TCP/IP 연결이 AS/400 시스템과 워크스테이션 사이에 있는지 확인하십시오.

3. RAWTGui.jar 파일을 AS/400 시스템의 /QIBM/ProdData/Java400/jdk12 디렉토리에서 워크스테이션의 디렉토리로 복사하십시오.
4. 워크스테이션에서 RAWTGui.jar 파일을 복사한 디렉토리로 변경하고 다음 명령을 입력하여 리모트 AWT를 시작하십시오.

```
java -jar RAWTGui.jar
```

주: AS/400 시스템에서 Java 그래픽 처리시 자원 집중 특성으로 인해 리모트 AWT를 사용하면 접속 클라이언트를 사용하여 MQSeries Adapter Kernel을 설치하는 것보다 시간이 더 오래 걸릴 수 있습니다.

리모트 AWT에 대한 자세한 정보는 <http://www.ibm.com/servers/eserver/series/library/>의 AS/400 라이브러리를 참조하십시오.

## 접속된 클라이언트 사용

접속된 클라이언트를 사용하여 MQSeries Adapter Kernel을 AS/400 시스템에 설치하는 경우, Java 그래픽 처리는 AS/400 시스템이 아닌 클라이언트 워크스테이션에서 이루어집니다. 이 절에서는 접속된 클라이언트를 사용하여 AS/400 시스템에 MQSeries Adapter Kernel을 설치하기 위해 필요한 요구사항을 설명합니다.

다음 프로그램이 OS/400에 반드시 설치되어야 합니다.

- Java Toolkit 및 Java Developer Kit 버전 1.2.2. Java Toolkit 및 Java Developer Kit은 사용 허가된 프로그램 번호 5769-JV1로 제공됩니다.
- 사용 허가된 프로그램 번호 5769-SS1, 옵션 12로 제공되는 호스트 서버 옵션.
- 사용 허가된 프로그램 번호 5769-TC1으로 제공되는 TCP/IP.

다음은 워크스테이션 요구사항입니다.

- Windows NT 4.0, Service Pack 5 또는 Windows 2000, Service Pack 1을 실행 중인 IBM PC 시스템(또는 호환 가능).
- AS/400 시스템으로의 TCP/IP 연결.
- JDK 1.2.2 이상.



---

## 커널 구성요소

설치 후 MQSeries Adapter Kernel은 루트 디렉토리에 있습니다. 루트 디렉토리는 다른 디렉토리를 포함할 수 있는 서브디렉토리를 포함합니다. 루트와 그의 서브디렉토리가 설치 및 구성 관련 파일 요약과 함께 아래에 나열되어 있습니다.

**root** 디폴트 이름은 Windows 시스템에서 C:\Program Files\MQAK, AIX에서 /usr/lpp/mqak, HP-UX에서 /MQAK, Solaris에서 /opt/MQAK, OS/400에서 /QIBM/ProdData/mqak입니다. 다음을 포함합니다.

- 다른 모든 MQSeries Adapter Kernel 디렉토리.
- 원하면 설치 후 시스템 환경 변수를 변경하는 aqmsetenv.bat(Windows 시스템) 또는 aqmsetenv.sh(UNIX) 파일.
- readme.txt 파일.
- aqmuninstall.bat(Windows 시스템) 또는 aqmuninstall.sh(UNIX) 파일.

**bin** 다음을 포함합니다.

- 클래스 라이브러리 및 공유 라이브러리.
- 확인 전용으로 커널의 일부로서 제공되는 어댑터.
- aqmversion.bat(Windows 시스템) 또는 aqmversion.sh(UNIX 및 OS/400) 파일, 커널의 버전 번호를 표시하기 위해 실행하는 스크립트.
- aqmcrtmsg.bat(Windows 시스템) 또는 aqmcrtmsg.sh(UNIX 및 OS/400) 파일, 프로덕션 환경에 두기 전에 구성 파일을 검증하는 데 사용하는 XML 파일을 작성하기 위해 실행하는 스크립트.
- aqmsndmsg.bat(Windows 시스템) 또는 aqmsndmsg.sh(UNIX 및 OS/400) 파일, 프로덕션 환경에 두기 전에 구성 파일을 검증하기 위해 실행하는 스크립트.
- aqmstrad.bat(Windows 시스템) 또는 aqmstrad.sh(UNIX 및 OS/400) 파일, 어댑터 디면을 시작하기 위해 실행하는 스크립트.

- aqmstrtd.bat(Windows 시스템) 또는 aqmstrtd.sh(UNIX 및 OS/400) 파일, 추적 서버를 시작하기 위해 실행하는 스크립트.

### **documentation**

정보 센터를 비롯한 제품 문서를 포함합니다.

### **runtimefiles**

커널 런타임 파일을 포함합니다.

### **samples**

어댑터 및 관련 구성과 유틸리티 파일의 샘플을 포함합니다. 샘플로 실습하고 배울 수 있습니다.

주: 커널은 MQSeries Adapter Builder를 사용함으로써 빌드된 어댑터와 함께 사용하도록 되어 있습니다. 커널은 사용자 정의 코드만의 커널 API 호출에 의해 사용되도록 설계되지는 않습니다. 어댑터 샘플은 커널의 기능 방법을 이해하고 진단하는데 도움을 주기 위해서만 제공됩니다.

- 어댑터 샘플.
- 어댑터 샘플을 지원하는 값을 갖는 커널의 설정 파일 aqmsetup. 이 파일에 대해서는 73 페이지의 『설정 파일』을 참조하십시오.
- 샘플 추적 값을 포함하여 어댑터 샘플을 지원하는 값을 갖는 커널 구성 파일 aqmconfig.xml. 이 파일에 대해서는 74 페이지의 『구성 파일』을 참조하십시오.

### **toolkit**

다음으로 구성된 SDK(software development toolkit)를 포함합니다.

- 헤더 파일.
- Windows 시스템하에서 컴파일 중에 사용되는 라이브러리 파일.

### **uninstall**

커널을 설치 제거하는 데 사용되는 파일을 포함합니다.

## verification

커널의 설치 확인을 지원하는 다음 파일을 포함합니다.

- `aqmverifyinstall.bat`(Windows 시스템) 또는 `aqmverifyinstall.sh`(UNIX 및 OS/400) 파일, 컴퓨터에 커널 설치를 확인하기 위해 실행하는 스크립트.
- `aqmcreateq.bat`(Windows 시스템) 또는 `aqmcreateq.sh`(UNIX 및 OS/400) 파일, 확인용 MQSeries 큐를 작성하는 스크립트. 108 페이지의 『MQSeries 큐 작성』을 참조하십시오.
- `aqmconfig.xml` 파일. 이 파일에 대해서는 74 페이지의 『구성 파일』을 참조하십시오.
- `aqmsetup` 파일. 이 파일에 대해서는 73 페이지의 『설정 파일』을 참조하십시오.
- `aqminstalltest.xml` 파일.



---

## 제3장 커널 설치

이 장에서는 MQSeries Adapter Kernel을 설치 및 확인하는 데 필요한 단계에 대하여 설명합니다. 설치는 다음과 같은 단계로 구성됩니다.

- 단계 1. 설치를 준비하십시오. 세부사항은 46 페이지의 『설치 준비』를 참조하십시오.
- 단계 2. 커널을 설치하십시오. 세부사항은 47 페이지의 『커널 설치』를 참조하십시오.
- 단계 3. 몇 가지 설치 후 단계를 완료하십시오. 세부사항은 50 페이지의 『설치 후 작업 완료』를 참조하십시오.
- 단계 4. 설치를 검증하십시오. 세부사항은 53 페이지의 『설치 검증』을 참조하십시오.

이 장에서는 다음 주제에 대해서도 설명합니다.

- 자동 설치를 사용하여 MQSeries Adapter Kernel 설치. 세부사항은 59 페이지의 『자동 설치 사용』을 참조하십시오.
- 이전 버전에서 MQSeries Adapter Kernel 업그레이드. 세부사항은 61 페이지의 『커널 업그레이드』를 참조하십시오.
- MQSeries Adapter Kernel 설치 제거. 세부사항은 62 페이지의 『커널 제거』를 참조하십시오.

커널 설치 후 다음과 같은 추가 작업을 수행하여 커널을 사용할 수 있도록 준비하십시오.

1. 커널을 구성하십시오. 세부사항은 67 페이지의 『커널 구성』을 참조하십시오.
2. 메시징 소프트웨어와 선택적 소프트웨어를 구성하십시오. 세부사항은 101 페이지의 『MQSeries 및 MQSeries Integrator 구성』을 참조하십시오.
3. MQSeries Adapter Builder를 사용하여 어댑터를 빌드하고 테스트한 다음 전개하십시오.
4. 커널을 시작하십시오. 세부사항은 102 페이지의 『커널 시작』을 참조하십시오.

---

## 설치 준비

MQSeries Adapter Kernel을 설치할 수 있는 관리자 또는 루트 권한이 반드시 있어야 합니다. MQSeries Adapter Kernel을 설치한 위치와 두 커널 구성 파일을 놓은 위치에 파일을 작성하고 액세스하기 위한 사용권한이 필요합니다. 실행 가능 경로에 현재 디렉토리가 있어야 합니다. 커널을 실행하는 모든 사용자 ID가 읽기, 쓰기 및 실행 권한을 가지고 있는지 확인하십시오.

큐 관리자 작성 및 큐 액세스와 같은 MQSeries 조작을 수행할 수 있는 권한이 필요합니다. 이러한 조작은 여러 다른 플랫폼에서 다른 방법으로 수행됩니다. 자세한 정보는 플랫폼에 대한 *MQSeries* 관리 안내서를 참조하십시오.

커널 프로세스를 시작하는 사용자 ID는 mqm 그룹에 있어야 합니다. 두 가지 종류의 커널 프로세스가 있습니다.

- 컴퓨터가 작동하는 각 대상 응용프로그램에 대한 한 개의 어댑터 디먼
- 추적 서버(선택적)

소스 응용프로그램 프로세스 중 소스 어댑터가 실행됨에 유의하십시오. 소스 어댑터가 실행되려면 소스 어댑터를 포함하는 디먼이나 서버가 시작되어야 합니다.

빌드한 어댑터를 실행하려면 반드시 커널을 설치하고 구성해야 합니다. 그러나 MQSeries Adapter Builder를 설치하기 위해 또는 어댑터를 빌드하기 위해 설치할 필요는 없습니다.

설치를 시작하기 전에 다음 단계를 수행하십시오.

- CD-ROM이나 근거리 통신망(LAN)의 readme.txt 파일을 읽어 보십시오. 이 파일에는 이 책이 완성된 후에 사용할 수 있게 된 중요한 정보가 들어 있습니다. 이 파일은 루트 설치 디렉토리에 있습니다.
- [www.ibm.com/software/ts/mqseries/](http://www.ibm.com/software/ts/mqseries/)의 MQSeries 웹 사이트를 방문하십시오. 여기에는 이 책이 출판된 후에 사용할 수 있게 된 중요한 정보나 이 책에 대한 새로운 개정판이 들어 있습니다.
- MQSeries Adapter Kernel 이전 버전에서 업그레이드하는 중이라면 61 페이지의 『커널 업그레이드』를 참조하십시오.

- 하드웨어 및 소프트웨어 필수조건을 충족하는지 확인하십시오. 세부사항은 35 페이지의 『하드웨어』 및 36 페이지의 『소프트웨어』를 참조하십시오. MQSeries Adapter Kernel의 설치를 확인하려면 먼저 MQSeries를 설치하여 실행 중이어야 합니다. MQSeries Java 지원이 설치되고 구성되었는지 확인하십시오.

---

## 커널 설치

Windows 시스템(Windows NT 또는 Windows 2000), UNIX 플랫폼(AIX, HP-UX 또는 Solaris) 또는 OS/400에서 MQSeries Adapter Kernel을 설치하려면 운영 체제별로 다음 단계를 수행하십시오.

### Windows 시스템에서

단계 1. 다음과 같이 설치 프로그램을 시작하십시오.

- 근거리 통신망(LAN)에서 설치하는 중이라면, MQSeries Adapter Kernel 설치 파일을 포함하는 디렉토리로 변경하고 install.bat 파일을 실행하십시오.
- CD-ROM에서 설치 중이라면, MQSeries Adapter Kernel CD-ROM을 CD-ROM 드라이브에 넣으십시오. 자동실행이 사용 가능하게 되면 설치 프로그램이 자동으로 시작됩니다. 자동실행이 사용 가능하지 않으면 CD-ROM의 루트 디렉토리에 있는 install.bat 파일을 실행하여 설치 프로그램을 시작하십시오.

주: Windows 시스템에서 install.bat 파일을 실행 전에 다른 위치로 복사할 필요는 없습니다. 설치 프로세스 중에 MQSeries Adapter Kernel을 설치할 위치를 선택하도록 요청합니다.

단계 2. 설치 프로그램이 제공하는 프롬프트에 따르십시오. MQSeries Adapter Kernel을 디폴트(Windows 시스템에서는 C:\Program Files\MQAK)가 아닌 위치에 설치하도록 선택하는 경우, 상대 경로 이름이 아닌 완전한 경로 이름으로 설치 디렉토리를 지정해야 합니다.

### UNIX에서

단계 1. 다음과 같이 설치 프로그램을 시작하십시오.

- 근거리 통신망(LAN)에서 설치하는 중이라면, MQSeries Adapter Kernel 설치 파일을 포함하는 디렉토리로 변경하고 install.sh 스크립트를 실행하십시오.
- CD-ROM에서 설치 중이라면, MQSeries Adapter Kernel CD-ROM을 CD-ROM 드라이브에 넣고 필요하면 운영 체제 문서에 따라 CD-ROM 드라이브를 마운트하십시오. CD-ROM의 루트 디렉토리에 있는 install.sh 스크립트를 실행하십시오.

단계 2. 설치 프로그램이 제공하는 프롬프트에 따르십시오. MQSeries Adapter Kernel을 디폴트가 아닌 다른 위치에 설치하려는 경우 상대 경로 이름이 아닌 완전한 경로 이름으로 설치 디렉토리를 지정해야 합니다. UNIX에서의 디폴트 설치 디렉토리는 다음과 같습니다.

- AIX: /usr/lpp/mqak
- HP-UX: /MQAK
- Solaris: /opt/MQAK

## OS/400에서

단계 1. 35 페이지의 『하드웨어』, 37 페이지의 OS/400 소프트웨어 필수조건 및 38 페이지의 『OS/400 설치 필수조건』에 나열된 모든 필수조건을 충족하는지 확인하십시오. MQSeries Adapter Kernel을 OS/400에 설치하면 OS/400 시스템과 인터페이스하는 워크스테이션을 사용해야 하는 InstallShield 프로그램이 사용됩니다. 세부사항은 38 페이지의 『OS/400 설치 필수조건』을 참조하십시오.

단계 2. 제어 언어(CL) 프롬프트에서 **CRTUSRPRF** 명령을 사용하여 AS/400 시스템에서 MQAKSRV라는 이름의 사용자 프로파일을 작성하십시오.

단계 3. 리모트 AWT 또는 접속된 클라이언트 워크스테이션을 사용하여 설치를 수행하는지 여부에 따라 다음 단계를 수행하십시오.

- 리모트 AWT를 사용하여 설치를 수행하고 있다면, 다음 단계를 수행하십시오.
  - a. 리모트 AWT가 설정되어 실행 중인지 확인하십시오. 세부사항은 39 페이지의 『리모트 AWT 사용』을 참조하십시오.



- b. installAS400.jar 파일이 AS/400 시스템에 액세스할 수 있어야 합니다. 파일은 통합 파일 시스템(IFS) 또는 AS/400 시스템에 접속된 디바이스에 있어야 합니다. 파일이 접속된 디바이스인 경우 링크 작성(CRTLINK) 명령을 사용하여 파일로의 기호 링크를 작성하십시오.
- c. 설치 프로세스의 성능을 향상시키려면, Java 프로그램 작성(CRTJVAPGM) 명령을 installAS400.jar 파일에 대해 실행하십시오.
- d. Java 실행(RUNJVA) 명령을 다음과 같이 실행하십시오. 여기서 *n.n.n.n*은 리모트 AWT를 실행 중인 워크스테이션의 TCP/IP 주소를 나타냅니다.

```
RUNJVA CLASS(run)
CLASSPATH('/installAS400.jar')
PROP((os400.class.path.rawt 1) (RmtAwtServer 'n.n.n.n')
(java.version 1.2))
```

- 접속된 클라이언트 워크스테이션을 사용하여 설치를 수행하고 있다면, 다음 단계를 수행하십시오.

단계 a. 40 페이지의 『접속된 클라이언트 사용』에 지정된 요구사항을 충족하는지 확인하십시오.

단계 b. 호스트 서버 옵션이 AS/400 시스템에 설치되어 실행 중인지 확인하십시오. CL 프롬프트에서 STRHOSTSVR(호스트 서버 시작) 명령을 사용하여 호스트 서버를 시작할 수 있습니다.

단계 c. TCP/IP가 AS/400 시스템에 설치되어 실행 중인지 확인하십시오. CL 프롬프트에서 TCP/IP 시작(STRTCP) 명령을 사용하여 TCP/IP를 시작할 수 있습니다.

단계 d. 워크스테이션에서 명령 프롬프트를 열고 MQSeries Adapter Kernel 설치 미디어(근거리 통신망(LAN) 또는 CD-ROM)의 AS400 디렉토리로 변경하십시오.

단계 e. 다음 명령을 입력하십시오.

```
java -classpath installAS400.jar; run -os400
```

단계 4. 설치 프로그램이 시작하고 **AS/400**에 사인 온 패널을 표시합니다. AS/400 시스템의 TCP/IP 주소를 시스템: 필드에 입력하고 사용자 ID와 암호를 해당 필드에 입력하십시오. 디폴트 사용자 선택란은 체크표시하지 마십시오. 다음을 누르십시오

단계 5. 설치 프로그램이 제공하는 프롬프트에 따르십시오. 네트워크 및 시스템의 속도에 따라 설치 프로세스는 완료하는 데 최대 1시간까지 걸릴 수 있습니다. 워크스테이션에 표시되는 진행 표시줄은 설치 상태를 나타냅니다. OS/400에서 MQSeries Adapter Kernel은 항상 통합 파일 시스템(IFS)의 루트에 있는 /QIBM/ProdData/mqak 디렉토리에 설치됩니다.

단계 6. CLASSPATH, PATH, 및 QIBM\_MULTI\_THREADED 환경 변수를 다음과 같이 설정하십시오.

- /QIBM/ProdData/mqak/bin 디렉토리를 CLASSPATH 환경 변수에 추가하십시오.
- /QIBM/ProdData/mqak/bin 디렉토리를 PATH 환경 변수에 추가하십시오.
- QIBM\_MULTI\_THREADED 환경 변수를 Y로 설정하십시오.

단계 7. 라이브러리 MQAK를 QSYS.LIB 라이브러리 목록에 추가하십시오.

커널 설치가 완료되었습니다. 설치되면 커널이 특정 사이트에서 확인은 지원하고 프로덕션은 지원하지 않도록 구성됩니다. 53 페이지의 『설치 검증』에 나열된 단계를 수행하여 설치를 확인하십시오. 설치를 확인한 후 『설치 후 작업 완료』 단계를 수행하여 사용자 사이트에서 프로덕션을 지원하도록 환경 변수를 설정하고 몇 가지 구성 파일을 이동하십시오.

필요에 따라 다른 컴퓨터에 커널을 설치하십시오.

---

## 설치 후 작업 완료

커널 설치 후 다음 단계를 수행하십시오.

단계 1. 커널 구성에 사용되는 aqmsetup 및 aqmconfig.xml 파일을 어디에 둘 것인지 결정하십시오. 이 파일들에 대한 자세한 정보는 67 페이지의 『커널 구성』을 참조하십시오.

**주의:**

사용자의 구성 파일은 작성하지 않았지만 대신 프로덕션 **samples** 디렉토리에 제공된 구성 파일을 사용하는 경우, 새로운 커널 버전을 설치하면 사용자의 구성 파일을 겹쳐쓰고 사용자 프로덕션 구성을 파괴합니다.

단계 2. 두 개의 구성 파일용 디렉토리를 작성하십시오. 둘 다 같은 디렉토리에 있을 필요는 없지만 간단하게 하기 위해 같은 디렉토리에 두는 것이 좋습니다. MQSeries Adapter Kernel을 설치한 디렉토리 외부에 두는 경우 나중에 커널이 설치 제거될 때 몇 개의 디렉토리가 남습니다. 설치 제거 프로세스는 원래 MQSeries Adapter Kernel 파일이 아닌 다른 파일을 포함하는 디렉토리는 그대로 둡니다.

단계 3. aqmsetup 및 aqmconfig.xml 파일을 samples 디렉토리에서 사용자가 원하는 위치로 복사하십시오. 많은 컴퓨터에서 액세스하여 파일을 갱신하고 더 쉽게 백업할 수 있도록 다른 중앙 위치나 네트워크 드라이브에 넣을 수 있습니다.

aqmconfig.xml 파일의 이름을 바꾸는 경우 커널이 올바르게 작동하지 않습니다. 5 단계에서 올바르게 가리키도록 환경 변수를 설정하는 경우 aqmsetup 파일의 이름을 바꿀 수 있습니다.

단계 4. 텍스트 편집기를 사용하여 aqmsetup 파일을 aqmconfig.xml 파일의 원하는 디렉토리를 가리키도록 편집하십시오. 디렉토리 위치로 완전한 경로 이름(상대 경로 이름이 아님)을 사용하십시오. 파일 이름 자체를 경로에 포함시키지 마십시오. 예를 들면 다음과 같습니다.

```
# Location of configuration file aqmconfig.xml.  
AQMCONFIG=C:\Program Files\MQAK\Data\
```

원하는 aqmconfig.xml 위치가 같은 디렉토리(aqmsetup 파일이 있는)인 경우에도 여기에 완전한 경로 이름을 입력해야 합니다. aqmsetup 파일을 저장하고 닫으십시오.

단계 5. AQMSETUPFILE 환경 변수를 aqmsetup 파일 위치를 가리키도록 설정하십시오(예를 들어, Windows 시스템에서는 C:\Program Files\MQAK\Data\aqmsetup, UNIX에서는 /MQAK/data/aqmsetup 또

는 OS/400에서는 /home/user\_name/aqmsetup). OS/400에서 aqmsetup 파일은 항상 현재 사용자의 홈 IFS 디렉토리 즉, /home/user\_name에 있어야 합니다.

커널이 네트워크 드라이브에 설치되어 있다면, 여기에 액세스하는 각 컴퓨터에 대해 이 단계를 수행하십시오.

- 단계 6. AIX를 사용 중이고 C 프로그램에서 호출된 원시 C 언어 소스 어댑터를 사용할 계획이라면, AIXTHREAD\_SCOPE 환경 변수를 S 값으로 설정하십시오. Bourne 셸이나 Korn 셸에서 이 환경 변수를 설정하려면 다음 명령을 입력하십시오.

```
export AIXTHREAD_SCOPE=S
```

C 셸에서 이 환경 변수를 설정하려면 다음 명령을 입력하십시오.

```
setenv AIXTHREAD_SCOPE S
```

AIX에 로그인할 때 AIXTHREAD\_SCOPE 변수를 자동으로 설정하려면, 이 명령을 .profile 파일(Bourne 셸이나 Korn 셸) 또는 .cshrc 파일(C 셸)에 추가하십시오.

스케줄 정책에 대해서는 25 페이지의 7 단계를 참조하십시오.

- 단계 7. 필요하다면 THREADS\_FLAG 환경 변수를 설정하십시오. 다음 조건이 모두 참인 경우에만 이 변수를 설정해야 합니다.

- 사용 중인 운영 체제는 Solaris입니다.
- 사용 중인 JDK(Java Development Kit) 버전은 1.2.2입니다.
- 메시지 전송에 MQSeries를 사용 중입니다.
- 소스 및 대상 어댑터가 C로 작성되었습니다.

위 조건 모두가 참인 경우 THREADS\_FLAG 환경 변수를 native로 설정하십시오. Bourne 셸이나 Korn 셸에서 이 환경 변수를 설정하려면 다음 명령을 입력하십시오.

```
export THREADS_FLAG=native
```

C 셸에서 이 환경 변수를 설정하려면 다음 명령을 입력하십시오.

```
setenv THREADS_FLAG native
```

Solaris에 로그인할 때 THREADS\_FLAG 변수를 자동으로 설정하려면 이 명령을 .profile 파일(Bourne 셸 또는 Korn 셸) 또는 .cshrc 파일(C 셸)에 추가하십시오.

설치 후 단계를 완료한 후 다음 단계를 수행하여 커널 사용을 준비하십시오.

1. 프로덕션을 준비하십시오. 65 페이지의 『프로덕션 준비』를 참조하십시오.
2. 구성 파일을 편집하십시오. 세부사항은 67 페이지의 『커널 구성』을 참조하십시오.
3. MQSeries와 선택적 소프트웨어를 구성하십시오. 101 페이지의 『MQSeries 및 MQSeries Integrator 구성』을 참조하십시오.
4. 프로덕션 시스템의 경우 102 페이지의 『성능 권장사항』을 고려하십시오.
5. 커널을 시작하십시오. 102 페이지의 『커널 시작』을 참조하십시오.
6. 커널 유지보수 계획을 설정하십시오. 105 페이지의 『커널 유지보수』를 참조하십시오.

---

## 설치 검증

커널을 설치한 후, 확인 스크립트를 실행하여 올바르게 설치되었는지 확인하십시오. 스크립트는 소스 어댑터를 사용하여 소스 응용프로그램으로부터 테스트 메시지를 송신한 후 커널을 사용하여 MQSeries로 송신합니다. 그런 다음 커널을 사용하여 MQSeries에서 메시지를 수신한 다음 대상 어댑터를 호출합니다. 이러한 프로세스는 모두 단일 컴퓨터에서 실행됩니다.

이러한 확인에서 소스 응용프로그램은 MQSeries 큐 TEST1입니다. 대상 응용프로그램은 다른 MQSeries 큐 TEST2입니다.

확인에는 다음 작업을 수행합니다.

- 제공된 소스 어댑터와 대상 어댑터를 사용하는 커널이 컴퓨터 내에서 엔드-투-엔드 메시징 소프트웨어로 MQSeries를 사용하여 테스트 메시지를 올바르게 전달하고 라우트했는지 확인합니다.
- 설치시 제공된 aqmconfig.xml 및 aqmsetup 파일을 확인합니다. 커널 구성을 판별합니다. 이러한 파일에 대해서는 67 페이지의 『커널 구성』을 참조하십시오.

구성 파일을 프로덕션 환경에 두기 전에 검증할 수 있습니다. 97 페이지의 『구성 파일 검증』을 참조하십시오.

MQSeries Adapter Kernel과 함께 제공되는 설치 확인 스크립트는 스크립트가 실행될 시스템에 MQSeries가 설치 및 구성되었다고 가정합니다. MQSeries가 아닌 다른 메시징 소프트웨어를 사용 중이라면, 다음과 같이 메시징 소프트웨어를 지원 하도록 설치 확인 스크립트를 편집할 수 있습니다.

1. 커널 설치의 확인 디렉토리로 변경하십시오.
2. 텍스트 편집기에서 aqmconfig.xml 파일을 열고  
`<epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>` 행을  
`<epicmqppqueuemgr>queue_manager_name</epicmqppqueuemgr>`로 변경하십시오. 여기서 `queue_manager_name`은 큐 관리자의 이름입니다.
3. 다음과 같이 aqmverifyinstall 파일을 편집하십시오.
  - Windows 시스템에서 설치 확인을 수행 중이라면, 텍스트 편집기에서 aqmverifyinstall.bat 파일을 편집하고 aqmcreateq TEST2 행을 aqmcreateq TEST2 `queue_manager_name`으로 변경하십시오. 여기서 `queue_manager_name`은 큐 관리자의 이름입니다.
  - UNIX 또는 OS/400에서 설치 확인을 수행 중이라면, 텍스트 편집기에서 aqmverifyinstall.sh 파일을 열고 aqmcreateq.sh TEST2 행을 aqmcreateq.sh TEST2 `queue_manager_name`으로 변경하십시오. 여기서 `queue_manager_name`은 큐 관리자의 이름입니다.

이 확인은 커널의 일부가 아닌 대상 어댑터 이름 com.ibm.epic.adapters.eak.test.InstallVerificationTest와 같은 일부 구성요소를 사용합니다. 설치 확인을 위해서만 커널과 함께 제공됩니다.

확인이 완료되면 확인 어댑터 디먼이 정지합니다.

확인 중 추적은 사용할 수 없습니다.

## 확인 절차

- 단계 1. 확인은 세 개의 MQSeries 큐를 작성하고 사용합니다. 확인을 수행하기 전에 이러한 큐가 그 안에 메시지를 갖는 경우 확인은 실패합니다. 다음 큐에 있는 메시지를 지우십시오.
  - TEST2AIQ
  - TEST2AEQ
  - TEST2RPL
- 단계 2. 사용자에게 설치 권한이 있는지 확인하고 커널을 확인하십시오. 46 페이지의 『설치 준비』를 참조하십시오.
- 단계 3. 다음과 같이 확인을 시작하십시오.
  - Windows 시스템에서 verification 디렉토리에 있는 aqmverifyinstall.bat 파일을 두 번 누르십시오. 또는 명령 프롬프트 창을 열고 verification 디렉토리로 변경한 후 aqmverifyinstall.bat를 실행하십시오.
  - UNIX에서 터미널을 열고 verification 디렉토리로 변경한 다음 aqmverifyinstall.sh 파일을 실행하십시오.
  - OS/400에서 다음 단계를 수행하십시오.
    - a. **STRQSH** 명령을 입력하여 **qsh** 세션을 시작하십시오.
    - b. /QIBM/ProdData/mqak/verification/aqmsetup 파일을 홈 디렉토리(/home/user\_name)로 복사하십시오.
    - c. /QIBM/ProdData/mqak/verification 디렉토리로 변경하십시오.
    - d. aqmverifyinstall.sh 파일을 실행하십시오.

aqmverifyinstall 파일에는 기능 방법에 대한 주석이 들어 있습니다.
- 단계 4. 설치 확인 테스트가 정상적으로 완료되었습니다라는 메시지는 성공을 나타냅니다. 필요하다면 확인 창을 닫으십시오.
- 단계 5. 실패할 경우 확인 창과 로그 파일 EpicSystemExceptionFilennnnnnn.log를 조사하여 오류를 판별하십시오.
- 단계 6. 확인 중 발견할 수 있는 일반적인 문제점과 가능한 응답에 대해서는 56 페이지의 『일반적인 확인 문제점』을 참조하십시오.

- 단계 7. 원하면 선택적 확인을 수행하십시오. 세부사항은 58 페이지의 『선택적 확인』을 참조하십시오.
- 단계 8. 설치 프로시저로 되돌아가서 특정 사이트에서 조작성을 지원하도록 커널을 구성하십시오. 50 페이지의 1 단계로 가십시오.

## 일반적인 확인 문제점

이 절에서는 가능한 해결방안과 함께 확인 중에 발견할 수 있는 일반적인 문제점을 나열합니다. 예외 메시지의 중요한 정보는 굵게 강조표시됩니다.

**문제점:** aqmsetup 파일이 없습니다.

**조치:** AQMSETUPFILE 환경 변수가 verification 디렉토리의 aqmsetup 디렉토리 위치로 설정되었는지 확인하십시오.

**예외 메시지:**

```
com.ibm.epic.adapters.eak.nativeadapter.EpicNativeAdapter::main: caught
throwable with message <AQM0002: com.ibm.epic.adapters.eak.common.
AdapterDirectory::getProperties():
Received exception <com.ibm.epic.adapters.eak.common.AdapterException>
Message information: <AQM0002: com.ibm.epic.adapters.eak.common.
AdapterCfg::readConfig(String):
Received exception <java.io.FileNotFoundException> Message information:
<C:\aqmsetup> Additional program information <>.>
Additional program information <Error Reading Configuration File
[File or Keys in file may not exist]>.>
```

**문제점:** aqmconfig.xml 파일이 없습니다.

**조치:** verification 디렉토리의 aqmsetup 파일을 편집하고 AQMCONFIG= 항목이 verification 디렉토리를 가리키는지 확인하십시오. 완전한 경로 이름을 사용하십시오. 또한 aqmconfig.xml 파일이 verification 디렉토리에 있는지도 확인하십시오.

**예외 메시지:**

```
com.ibm.epic.adapters.eak.common.AdapterException: MessageID <AQM0002>
<AQM0002: com.ibm.epic.adapters.eak.common.AdapterDirectory::
getProperties(): Received exception
<java.io.FileNotFoundException> Message information:
<AQMCONFIG.xml> Additional program information <>.>
```

**문제점:** 메시지를 넣을 큐가 없습니다.



조치: MQSeries를 사용하여 예외 메시지에 이름 지정된 큐(설치가 확인되고 있는 경우 TEST2AIQ)가 있는지와 메시지를 수용할 수 있는지를 확인하십시오. 108 페이지의 『MQSeries 큐 작성』을 참조하십시오.

예외 메시지:

```
com.ibm.epic.adapters.eak.nativeadapter.EpicNativeAdapter::main: caught
throwable with message
<AQM0107: com.ibm.epic.adapters.eak.nativeadapter.LMSMQbase::
createMQOutputQueue(String):
Received MQException creating queue, QManager name <DEFAULT>
Queue name <TEST2AIQ>:
completion code <2> reason code <2085>.>
```

문제점: 대상 어댑터가 없습니다.

조치: 메시지에 지정된 대상 어댑터가 있는지 확인하십시오.

```
com.ibm.epic.adapters.eak.test.InstallVerificationTest.
CLASSPATH 환경 변수가 커널의 bin 디렉토리를 포함해야 합니다.
```

예외 메시지:

```
com.ibm.epic.adapters.eak.adapterdaemon.EpicAdapterWorker::sendException
(Throwable, String):Thread-2:
Message <<TEST2> <2000.05.18.09.41.43.781> <<Processing Messages.>
<com.ibm.epic.adapters.eak.common.AdapterException: MessageID <AQM0002>
<AQM0002: com.ibm.epic.adapters.eak.adapterdaemon.EpicAdapterWorker:
instantiateClass(String, Class[], Object[]): Received exception
<java.lang.ClassNotFoundException> Message information:
<com.ibm.epic.adapters.eak.test.InstallVerificationTest>
Additional program information <[Cannot obtain Class for class name
<com.ibm.epic.adapters.eak.test.InstallVerificationTest>]>.>>>>
```

문제점: 메시지 전달을 위해 로드할 어댑터가 없습니다. 목적지 논리 ID는 큐의 메시지에 지정된 본문 유형과 본문 범주에 대한 항목이 aqmconfig.xml 파일에 없습니다.

조치: 확인 중 이 예외 메시지의 가장 큰 원인은 확인 이전에 TEST2AIQ 큐에 메시지가 있다는 것입니다. 모든 메시지를 TEST2AIQ 큐에서 지우고 확인을 다시 시도하십시오. verification 디렉토리의 aqmconfig.xml 파일에 있는 응용프로그램 TEST2의 명령 클래스 이름에 대한 유일한 항목은 본문 유형 TESTBOD와 본문 범주 OAG용입니다.

### 예외 메시지:

```
com.ibm.epic.adapters.eak.adapterdaemon.EpicAdapterWorker::sendException
(Throwable, String):Thread-2: Message <<TEST2> <2000.05.18.10.28.43.105>
<<Processing Messages.> <com.ibm.epic.adapters.eak.common.
AdapterException:
MessageID <AQM0401> <AQM0401: com.ibm.epic.adapters.eak.
adapterdaemon.EpicAdapterWorker::processMessage(EpicMessage):
Cannot obtain Command class name to load for a received message.>>>>
```

문제점: 확인 큐 관리자가 시작되지 않았습니다.

조치: 디폴트 MQSeries 큐 관리자가 정상적으로 시작되었는지 확인하십시오.

### 예외 메시지:

```
com.ibm.epic.adapters.eak.common.AdapterException: Message ID <AQM0104>
<AQM0104: com.ibm.epic.adapters.eak.nativeAdapter.queueCollection::
constructor(String,String,boolean,String,String,int):
Received MQException creating QManager connection for
QManager name <QMGRNAME>
MQ Message information: completion code <2> reason code <2059.>.
```

문제점: 일반 MQSeries 오류가 발생했습니다.

조치: MQSeries가 올바르게 설치되어 구성되었는지와 시스템에서 실행 중 인지를 확인하십시오. MQException 이유 코드를 조사하고 MQSeries Messages 문서를 사용하여 이유 코드 원인을 판별하십시오.

### 예외 메시지:

```
Received MQException "ACTION ATTEMPTED." Message information:
completion code <completion_code> reason code <reason_code>
```

## 선택적 확인

커널이 첫번째 컴퓨터에 올바르게 설치되었음을 확인한 후 선택적으로 다음 단계를 수행할 수 있습니다.

1. 커널이 같은 확인을 사용하여 두번째 컴퓨터에 올바르게 설치되었는지 확인하십시오.
2. 한 컴퓨터에 있는 소스 어댑터에서 다른 컴퓨터에 있는 대상 어댑터로 테스트 메시지를 송신할 수 있는지 확인하십시오. 수동으로 이 확인을 구성하고 수행

하십시오. 커널과 함께 제공된 원래 확인 파일을 수정함으로써 이 확인을 개발 하도록 선택한 경우 백업을 위해 원래 확인 파일의 사본을 보존하십시오.

---

## 자동 설치 사용

MQSeries Adapter Kernel은 자동 설치를 사용하여 모든 플랫폼에서 설치할 수 있습니다. 자동 설치를 사용하면, 원하는 설치 옵션을 수동으로 선택해야 하는 MQSeries Adapter Kernel 설치 프로그램을 생략할 수 있습니다. 자동 설치 는 여러 시스템에서 디폴트 구성을 설치하고자 할 때 유용합니다.

커널을 자동으로 설치하려면 운영 체제별로 다음 단계를 수행하십시오.

### Windows 시스템에서

단계 1. 명령 프롬프트를 열고 MQSeries Adapter Kernel 설치 파일이 들어 있는 디렉토리로 이동하십시오.

단계 2. 다음 명령을 입력하십시오.

```
java -cp install.jar run -P product.installLocation="install_location" -silent
```

여기에서 *install\_location*은 원하는 설치 위치입니다(예를 들면, D:\mqak).

### UNIX에서

단계 1. 터미널에서 MQSeries Adapter Kernel 설치 파일이 들어 있는 디렉토리로 이동하십시오. CD-ROM에서 설치 중이라면, MQSeries Adapter Kernel CD-ROM을 CD-ROM 드라이브에 넣고 필요하면 운영 체제 문서에 따라 CD-ROM 드라이브를 마운트하십시오.

단계 2. 다음 명령을 입력하십시오.

```
java -cp install.jar run -P product.installLocation="install_location" -silent
```

여기에서 *install\_location*은 원하는 설치 위치입니다(예를 들면, /opt/mqak).

### OS/400에서

접속된 클라이언트를 사용하여 AS/400 시스템에 액세스하지 않는 경우에는 다음 단계를 수행하십시오.

단계 1. `installAS400.jar` 파일이 AS/400 시스템에 액세스할 수 있어야 합니다. 파일은 통합 파일 시스템(IFS) 또는 AS/400 시스템에 접속된 디바이스에 있어야 합니다. 파일이 접속된 디바이스인 경우 링크 작성(**CRTLINK**) 명령을 사용하여 파일로의 기호 링크를 작성하십시오.

단계 2. 설치 프로세스의 성능을 향상시키려면, Java 프로그램 작성(**CRTJVAPGM**) 명령을 `installAS400.jar` 파일에 대해 실행하십시오.

단계 3. CL 프롬프트나 **qsh** 세션 사용 여부에 따라 다음 명령 중 하나를 입력하십시오.

- CL 프롬프트를 사용하는 경우에는 다음 명령을 입력하십시오.

```
RUNJAVA CLASS(run)
CLASSPATH('/installAS400.jar')
PROP((java.version 1.2)) PARM('-silent')
```

- **qsh** 세션을 사용하는 경우에는 다음 명령을 입력하십시오.

```
java -Djava.version=1.2 -classpath installAS400.jar run -silent
```

접속된 클라이언트를 사용하여 AS/400 시스템과 인터페이스하는 경우 다음 단계를 수행하십시오.

단계 1. 40 페이지의 『접속된 클라이언트 사용』에 지정된 요구사항을 충족하는지 확인하십시오.

단계 2. 호스트 서버 옵션이 AS/400 시스템에 설치되어 실행 중인지 확인하십시오. 제어 언어(CL) 프롬프트에서 호스트 시작(**STRHOSTSVR**) 명령을 사용하여 호스트 서버를 시작할 수 있습니다.

단계 3. TCP/IP가 AS/400 시스템에 설치되어 실행 중인지 확인하십시오. CL 프롬프트에서 TCP/IP 시작(**STRTCP**) 명령을 사용하여 TCP/IP를 시작할 수 있습니다.

단계 4. 워크스테이션에서 명령 프롬프트를 열고 MQSeries Adapter Kernel 설치 미디어(근거리 통신망(LAN) 또는 CD-ROM)의 AS400 디렉토리로 변경하십시오.

단계 5. 다음 명령을 입력하십시오.

```
java -cp installAS400.jar run -silent -os400 machine_name user_ID password
```

여기에서 *machine\_name*은 AS/400 시스템의 TCP/IP 주소이고 *user\_ID*는 사용자의 사용자 ID이며 *password*는 사용자 암호입니다.

---

## 커널 업그레이드

CSD(Corrective Service Diskette)를 사용하거나 사용하지 않고 MQSeries Adapter Kernel 버전 1.0을 설치했거나 이전 수정 레벨이 적용된 MQSeries Adapter Kernel 버전 1.1을 설치한 경우, 현재의 수정 레벨이 적용된 MQSeries Adapter Kernel 버전 1.1을 설치하기 전에 다음 단계를 수행하십시오.

단계 1. aqmsetup 및 aqmconfig(aqmconfig.properties 또는 aqmconfig.xml) 파일을 MQSeries Adapter Kernel 설치 디렉토리 외부 위치에 백업하십시오.

단계 2. MQSeries Adapter Kernel CSD가 설치된 경우 다음과 같이 설치 제거하십시오.

- Windows NT에서 다음 메소드 중 하나를 사용하십시오.
  - Windows NT 시작 메뉴에서 프로그램 > **MQSeries Adapter Kernel** > **CSD** 제거를 누르십시오.
  - 제어판에서 프로그램 추가/제거 유틸리티를 사용하십시오.
  - 커널 루트 디렉토리에 있는 aqmuninstallCSD.bat 파일을 실행하십시오.
  - 명령 프롬프트를 열고 커널의 루트 디렉토리로 변경한 후 다음 명령을 입력하십시오.

```
java uninstallCSD
```

- AIX에서 커널의 루트 디렉토리로 변경하고 다음 명령 중 하나를 입력하십시오.

```
aqmuninstallCSD.sh
```

```
java uninstallCSD
```

단계 3. 다음과 같이 MQSeries Adapter Kernel을 설치 제거하십시오.

- Windows NT에서 다음 메소드 중 하나를 사용하십시오.

- Windows NT 시작 메뉴에서 **프로그램 > MQSeries Adapter Kernel > MQSeries Adapter Kernel** 설치 제거를 누르십시오.
- 제어판에서 **프로그램 추가/제거** 유틸리티를 사용하십시오.
- 커널 루트 디렉토리에 있는 `aqmuninstall.bat` 파일을 실행하십시오.
- 명령 프롬프트를 열고 커널의 루트 디렉토리로 변경한 후 다음 명령을 입력하십시오.

```
java uninstall
```

- AIX에서 커널의 루트 디렉토리로 변경하고 다음 명령 중 하나를 입력하십시오.

```
aqmuninstall.sh
```

```
java uninstall
```

단계 4. MQSeries Adapter Kernel 버전 1.1을 설치하십시오. 세부사항은 47 페이지의 『커널 설치』를 참조하십시오.

단계 5. `aqmsetup` 및 `aqmconfig` 파일을 MQSeries Adapter Kernel 설치 디렉토리의 이전 위치로 복원하십시오. 필요에 따라 `aqmconfig.properties` 파일을 `aqmconfig.xml` 파일로 변환하십시오. `aqmconfig.xml` 파일에 대해서는 74 페이지의 『구성 파일』을 참조하십시오.

## 커널 제거

커널을 제거하는 방법에는 여러 가지가 있습니다. 설치 제거 프로세스는 커널 설치 후 작성된 파일이나 디렉토리를 제거하지 않습니다. 사용자가 복사한 모든 로그 파일과 데이터 파일도 제거하지 않습니다.

- Windows 시스템에서 다음 메소드 중 하나를 사용하십시오.
  - 시작 메뉴에서 **프로그램 > IBM MQSeries Adapter Kernel > MQSeries Adapter Kernel** 설치 제거를 누르십시오.
  - 제어판에서 **프로그램 추가/제거** 유틸리티를 사용하십시오.
  - 커널 루트 디렉토리에 있는 `aqmuninstall.bat` 파일을 실행하십시오.

- 커널을 자동으로 설치 제거하려면(즉, 설치 제거 프로그램이 세부사항이나 확인을 위한 프롬프트를 표시하지 않도록 하고 설치 제거하려면), 명령 프롬프트를 열고 커널의 설치 디렉토리로 이동한 후 다음 명령을 입력하십시오.

```
java -cp uninstall.jar run -silent
```

- UNIX에서 커널의 루트 디렉토리로 변경하고 다음 명령을 입력하십시오.

```
aqmuninstall.sh
```

커널을 자동으로 설치 제거하려면(즉, 설치 제거 프로그램이 세부사항이나 확인을 위한 프롬프트를 표시하지 않도록 하고 설치 제거하려면), 커널의 루트 디렉토리로 이동한 후 다음 명령을 입력하십시오.

```
java -cp uninstall.jar run -silent
```

- OS/400에서는 다음 방법 중 하나를 사용하여 커널을 설치 제거하십시오.

- 리모트 AWT를 사용하여 커널을 설치 제거하려는 경우 다음 단계를 수행하십시오.

단계 1. 리모트 AWT가 설정되어 실행 중인지 확인하십시오. 세부사항은 39 페이지의 『리모트 AWT 사용』을 참조하십시오.

단계 2. 설치 제거 프로세스의 성능을 향상시키려면, Java 프로그램 작성(CRTJVAPGM) 명령을 /QIBM/ProdData/mqak/uninstall/uninstall.jar 파일에 대해 실행하십시오.

단계 3. Java 실행(RUNJVA) 명령을 다음과 같이 실행하십시오. 여기서 *n.n.n.n*은 리모트 AWT를 실행 중인 워크스테이션의 TCP/IP 주소를 나타냅니다.

```
RUNJVA CLASS(run)
CLASSPATH('/QIBM/ProdData/mqak/uninstall/uninstall.jar')
PROP((os400.class.path.rawt 1) (RmtAwtServer 'n.n.n.n')
(java.version 1.2))
```

- 접속된 클라이언트 워크스테이션을 사용하여 커널을 설치 제거하려면 다음 단계를 수행하십시오.

단계 1. 40 페이지의 『접속된 클라이언트 사용』에 지정된 요구사항을 충족하는지 확인하십시오.

단계 2. 호스트 서버 옵션이 AS/400 시스템에 설치되어 실행 중인지 확인하십시오. 제어 언어(CL) 프롬프트에서 호스트 시작 (**STRHOSTSVR**) 명령을 사용하여 호스트 서버를 시작할 수 있습니다.

단계 3. TCP/IP가 AS/400 시스템에 설치되어 실행 중인지 확인하십시오. CL 프롬프트에서 TCP/IP 시작(**STRTCP**) 명령을 사용하여 TCP/IP를 시작할 수 있습니다.

단계 4. `uninstall.jar` 및 `uninstall.dat` 파일을 AS/400 시스템의 `/QIBM/ProdData/mqak/uninstall` 디렉토리에서 클라이언트 워크스테이션의 디렉토리로 복사하십시오.

단계 5. 다음 명령을 입력하십시오.

```
java -classpath uninstall.jar; run -os400
```

커널을 자동으로 설치 제거하려면(즉, 설치 제거 프로그램이 세부사항이나 확인을 위한 프롬프트를 표시하지 않도록 하고 설치 제거하려면), 다음 명령을 입력하십시오.

```
java -cp uninstall.jar run -silent -os400 machine_name user_ID password
```

여기에서 `machine_name`은 AS/400 시스템의 TCP/IP 주소이고 `user_ID`는 사용자의 사용자 ID이며 `password`는 사용자 암호입니다.

- AS/400 시스템의 CL 프롬프트나 **qsh** 세션에서 직접 작업하고 있는 경우 커널을 자동으로 설치 제거하려면(즉, 설치 제거 프로그램이 세부사항이나 확인을 위한 프롬프트를 표시하지 않도록 하고 설치 제거하려면), 다음 명령 중 하나를 입력하십시오.

- CL 프롬프트에서:

```
RUNJAVA CLASS(run)
CLASSPATH('/uninstall.jar')
PROP((java.version 1.2)) PARM('-silent')
```

- **qsh** 세션에서:

```
java -Djava.version=1.2 -classpath uninstall.jar run -silent
```



---

## 제4장 커널 사용

이 장에는 커널 사용에 대한 다음 정보가 들어 있습니다.

- 『프로덕션 준비』
- 67 페이지의 『커널 구성』
- 101 페이지의 『MQSeries 및 MQSeries Integrator 구성』
- 102 페이지의 『커널 시작』
- 104 페이지의 『커널 정지』
- 105 페이지의 『커널 유지보수』
- 105 페이지의 『문제점 진단』

---

### 프로덕션 준비

커널을 프로덕션 환경에 두기 전, 다음 작업을 수행하십시오.

1. MQSeries Adapter Offering, MQSeries 또는 기타 메시징 소프트웨어 및 선택적으로 MQSeries Integrator를 포함한 전체 시스템 구조를 사용자 사이트의 요구사항과 상태에 기초하여 설계하십시오. 일반적으로 구조는 각 사이트에 고유합니다.
2. MQSeries Adapter Builder를 사용하여 필요한 소스 어댑터 및 대상 어댑터를 빌드한 후 테스트하고 전개하십시오.
3. MQSeries Adapter Offering 외부에서 다음 목적으로 사용할 응용프로그램 특정 인터페이스를 개발하십시오.
  - 소스 어댑터가 소스 응용프로그램으로부터 응용프로그램 데이터를 확보할 수 있게 하기 위해
  - 대상 응용프로그램이 대상 어댑터로부터 메시지를 확보할 수 있게 하기 위해

응용프로그램 특정 인터페이스의 정확한 특성은 소스 응용프로그램과 대상 응용프로그램의 특성에 따라 달라집니다. 다음은 응용프로그램 특정 인터페이스의 예입니다.

- API 호출 및 사용자 엑시트
  - 파일 읽기 및 쓰기
  - 데이터베이스 트리거
  - 메시지 큐
4. 런타임 플로우(메시지 송신, 라우팅, 추적 및 전달)를 지원하기 위한 커널을 구성하십시오. 커널 구성에 대해서는 67 페이지의 『커널 구성』을 참조하십시오.
  5. 전체 시스템 구조를 지원하도록 MQSeries 또는 메시징 소프트웨어 및 선택적으로 MQSeries Integrator를 구성하십시오. 101 페이지의 『MQSeries 및 MQSeries Integrator 구성』을 참조하십시오.
  6. 필요하다면 메시지 전달을 지원하도록 Java 로그온 클래스를 개발하십시오. 이러한 클래스는 각 대상 응용프로그램에 고유합니다. 이는 로그온하여 응용프로그램에 연결하기 위한 정보를 대상 어댑터가 필요로 하는 경우에만 필요합니다.
  7. 시스템을 프로덕션 환경에 두기 전에 시스템 전체, 즉 소스 어댑터와 대상 어댑터, 응용프로그램 특정 인터페이스 및 사용자 정의 코드를 갖춘 MQSeries Adapter Kernel을 테스트하십시오.
  8. 시스템을 프로덕션 환경에서 전개하십시오.
  9. 하나 이상의 어댑터 디먼을 시작하여 커널을 켜고 선택적으로 서버를 추적하십시오. 소스 응용프로그램이 시작되었는지 확인하십시오. 소스 응용프로그램 프로세스에서 소스 어댑터를 실행 중인 경우 소스 어댑터는 소스 응용프로그램으로 자동 시작됩니다. 소스 어댑터를 시작하는 데 추가 단계는 필요 없습니다. 소스 어댑터를 포함하는 디먼 또는 서버를 시작해야 합니다. 102 페이지의 『커널 시작』을 참조하십시오.

---

## 커널 구성

이 절에서는 사용자 환경에서 사용하기 위한 커널 구성에 대하여 논의합니다. 『구성 개요』에서는 커널 구성에 대한 개념적 개요를 설명합니다. 73 페이지의 『시작 및 구성시 사용되는 파일』에서는 MQSeries Adapter Kernel 구성을 함께 정의하는 다양한 파일에 대하여 설명합니다. 73 페이지의 『설정 파일』에서는 커널의 여러 초기 설정값을 정의하는 `aqmsetup` 파일에 대하여 설명합니다. 74 페이지의 『구성 파일』에서는 커널에 소스 및 대상 응용프로그램의 이름, 소스 및 대상 어댑터, 큐 및 큐 관리자, 통신 모드, 로그 기록 및 추적 스펙 등 특정 구성 정보를 제공하는 `aqmconfig.xml` 파일에 대하여 설명합니다.

### 구성 개요

이 절에서는 커널 구성에 대한 개념적 개요에 대하여 설명합니다. 커널을 구성하기 전에 커널의 런타임 플로우를 이해하는 것이 중요합니다. 이 절에서는 간단한 수준에서 런타임 플로우를 설명합니다. 런타임 플로우에 대한 자세한 정보는 15 페이지의 『런타임 플로우』를 참조하십시오.

가장 기본적인 레벨에서 MQSeries Adapter Kernel의 구성은 응용프로그램간에 이동하는 데이터에 의해 이루어집니다. 구성시 다음과 같은 요인을 고려해야 합니다.

- 데이터를 수신하는 응용프로그램
- 소스 측에서 필요한 어댑터와 대상 측에서 필요한 대상 어댑터, 어댑터 디먼 및 작업 프로그램
- 통신 모드, 정렬 형식 및 사용된 전송 메커니즘

구성에서 데이터 구조와 데이터 형식은 각 응용프로그램에 따라 서로 다릅니다. 예를 들어, 구성에 응용프로그램 C로 구매 주문 데이터를 각각 보내는 응용프로그램 A와 B가 있고 응용프로그램 A의 데이터 형식이 응용프로그램 B의 데이터 형식과 다르고 서로 다른 태그 의미를 갖는 경우를 생각해 봅시다. 응용프로그램 C가 두 가지 다른 응용프로그램으로부터 서로 다른 데이터 플로우를 인식하고 구문 분석할 필요가 없도록 각 응용프로그램 데이터는 통합 공통 형식(*integration-neutral format*)의 통합 메시지로 변환됩니다. 일반적으로 통합 공통 형식(*integration-neutral*

format)은 XML에 기초한 업계 표준입니다. 응용프로그램 C가 인식하고 구문 분석할 수 있는 유일한 데이터 형식은 통합 공통 형식(integration-neutral format)입니다.

그림3에서는 응용프로그램 A, B에서 응용프로그램 C, D로의 데이터 플로우를 보여줍니다. 그림 다음은 이 그림을 설명하는 여러 가지 데이터 플로우에 대한 설명입니다.

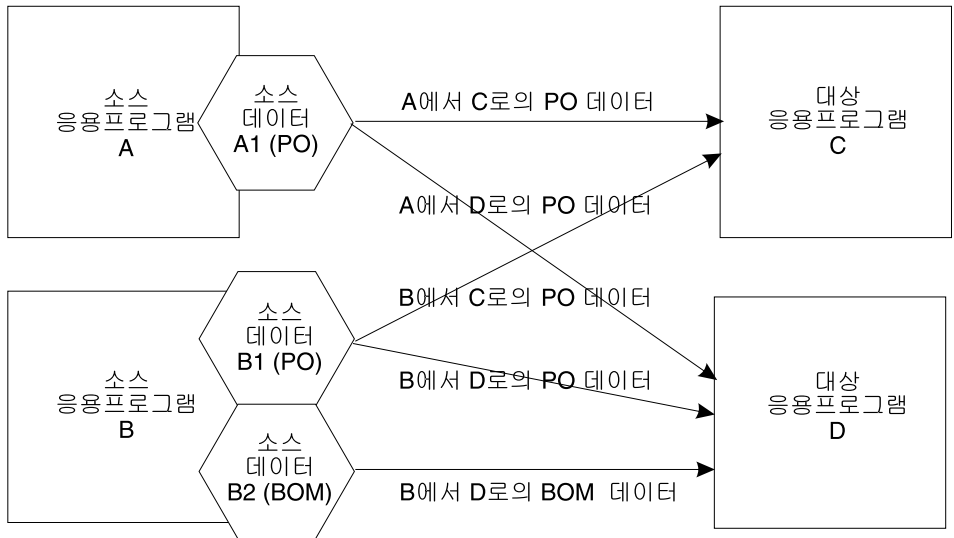


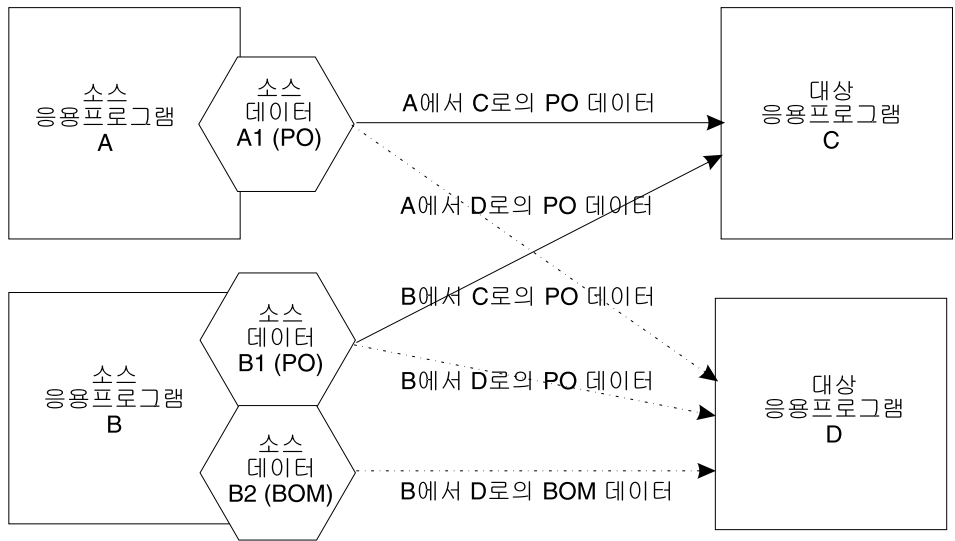
그림3. 단순한 구성에서 데이터 플로우에 의해 연결된 응용프로그램

그림3에서 응용프로그램 A의 구매 주문 데이터가 응용프로그램 C와 D로 이동하고 응용프로그램 B의 구매 주문 데이터가 역시 응용프로그램 C와 D로 이동하며, 응용프로그램 B의 BOM(Bill of Material) 데이터는 응용프로그램 D로만 이동합니다. 각각의 경우 데이터는 대상 응용프로그램으로 보내지기 전에 업계 표준 형식으로 변환됩니다. 응용프로그램 A와 B로부터 수신되는 구매 주문 데이터는 구매 주문 데이터를 표시하는 표준 XML 형식으로 변환됩니다. 응용프로그램 B로부터 수신되는 BOM 데이터는 BOM 데이터를 표시하는 표준 XML 형식으로 변환됩니다.

MQSeries나 JMS(Java Message Service) 구현과 같은 통신 전송은 데이터를 하나 이상의 대상 응용프로그램으로 송신하는 데 사용됩니다. 통합 메시지는 특정 통

신 전송에서 요구되는 정렬 형식으로 변환된 후 MQSeries 큐와 같은 통신 전송으로 전달됩니다. 각 대상 응용프로그램에서는 다른 통신 전송 및 정렬 형식을 사용하여 메시지를 수신할 수 있습니다. 예를 들면 70 페이지의 그림4에서와 같이 응용프로그램 C는 MQSeries를 사용하여 메시지를 수신하고 응용프로그램 D는 JMS를 사용할 수도 있습니다. 이 경우 응용프로그램 C로 들어가는 모든 통합 메시지(즉, 응용프로그램 A와 B의 구매 주문 데이터)는 MQSeries 정렬 형식으로 변환되고, 응용프로그램 D로 들어가는 모든 통합 메시지(즉, 응용프로그램 A와 B의 구매 주문 데이터와 응용프로그램 B의 BOM 데이터)는 JMS 정렬 형식으로 변환됩니다. MQSeries Adapter Kernel에서는 다음과 같은 메커니즘을 사용하여 이러한 변환을 수행합니다.

- 응용프로그램 데이터를 통합 메시지로 변환하는데 소스 어댑터가 사용됩니다. 소스 어댑터는 MQSeries Adapter Builder에서 작성됩니다.
- 통합 메시지를 통신 메시지로 변환하는데 원시 어댑터가 사용됩니다. 원시 어댑터는 LMS(논리 메시지 전달 서비스)를 사용하여 통신 전송에 의해 전송될 메시지를 변환합니다. LMS는 사용 중인 통신 전송에 고유합니다. 그리고나서 LMS는 형식기를 사용하여 전송시 메시지를 정렬합니다.



키:

————> = MQSeries를 통한 데이터 이동

.....> = JMS를 통한 데이터 이동

그림 4. 단순한 구성에서 다른 통신 전송에 의해 연결된 응용프로그램

응용프로그램 - 통합 메시지 - 통신 메시지로의 데이터 플로우 는 71 페이지의 그림5 및 71 페이지의 그림6에 나와 있습니다. 통신 메시지가 대상에서 수신될 때 변환은 반대가 됩니다. 원시 어댑터가 통신 메시지를 다시 통합 메시지로 변환합니다. 그리고나서 필요에 따라 대상 어댑터가 통합 메시지를 대상 응용프로그램에서 요구되는 데이터 형식으로 변환합니다.

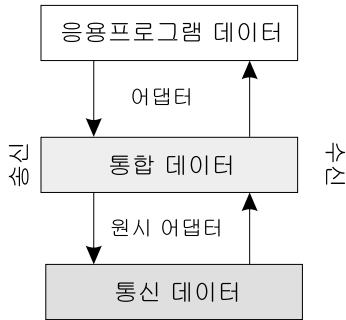


그림 5. 데이터 변환

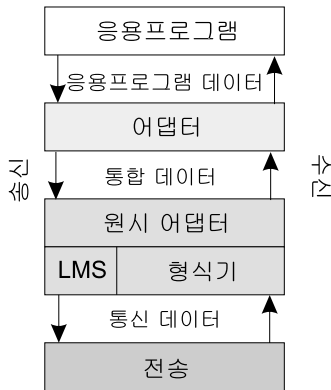


그림 6. 데이터 플로우

데이터가 전달되는 각 지점은 커널의 구성 파일에 지정해야 합니다. 응용프로그램 ID(소스 또는 대상)에 의해 나뉘는 논리 구성 요구사항이 세 가지가 있습니다. 메시지가 응용프로그램(대상)으로 들어가느냐 또는 응용프로그램(소스)으로부터 수신 되느냐 여부에 따라 소스 응용프로그램 또는 대상 응용프로그램에 대한 것일 수 있습니다. 구성 파일에는 다음과 유형의 정보가 포함되어야 합니다.

- 통신
  - 데이터가 이동할 위치
  - 사용할 통신 전송
  - 사용할 통신 정렬 메소드(형식기)

- MQSeries 큐 관리자 및 MQSeries 큐 이름 또는 JMS 큐 연결 팩토리와 JMS 큐 이름과 같은 기본 통신 요구사항
- 어댑터(대상측에만 해당)
  - 데이터 처리에 필요한 어댑터
  - 사용된 어댑터 유형(EAB 또는 EJB)
  - 사용된 특정 어댑터 유형에 대한 추가 정보
  - 독립형 MQSeries Adapter Kernel의 경우 어댑터 디먼과 작업 프로그램 정보
- 기타
  - 추적 스펙
  - 로그 기록 스펙

그림7에서는 구성의 다른 부분에 관련된 데이터 플로우를 보여줍니다.

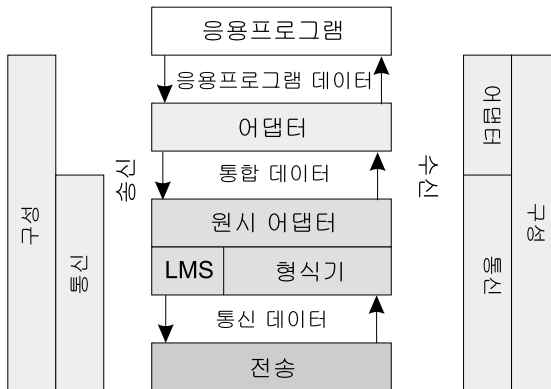


그림7. 구성과 관련된 데이터 플로우

커널 구성에서 이러한 부분을 제어하는 aqmconfig.xml 파일의 XML 요소에 이 구성 요구사항을 매핑하는 것에 대한 자세한 내용은 75 페이지의 『구성 파일의 구분 및 조직』을 참조하십시오. 88 페이지의 『공통 구성』에서는 몇 가지 공통 구성을 나열합니다.



## 시작 및 구성시 사용되는 파일

커널 구성은 여러 개의 사용자 정의 가능 파일로 판별됩니다. 표준 텍스트 편집기를 사용하여 파일을 편집함으로써 사이트용 커널을 구성하십시오. 다음 파일은 커널 구성과 관련이 있습니다.

- `aqmsetenv.bat`(Windows 시스템) 또는 `aqmsetenv.sh`(UNIX) 파일은 환경 변수를 설정합니다. 설치 후 원한다면 이 파일을 편집하여 시스템 환경 변수를 변경하십시오. 이 파일로 설정되는 환경 변수에는 `PATH`, `CLASSPATH` 및 `LIBPATH`가 포함됩니다. 이러한 변수는 Windows 시스템에서 설치 프로그램에 의해 자동으로 설정됩니다. UNIX에 로그인할 때 자동으로 이 변수들을 설정하려면 `aqmsetenv.sh` 파일에 지정된 값을 `.profile` 파일(Bourne 셸이나 Korn 셸을 사용하는 경우) 또는 `.cshrc` 파일(C 셸을 사용하는 경우)에 추가하십시오.

OS/400에서의 적절한 환경 변수 설정에 대한 정보는 50 페이지의 6 단계를 참조하십시오.

- `aqmsetup` 파일은 커널에 대한 여러 초기 설정 값을 제공합니다. 자세한 정보는 『설정 파일』을 참조하십시오.
- `aqmconfig.xml` 파일은 커널을 구성합니다. 자세한 정보는 74 페이지의 『구성 파일』을 참조하십시오. 이 파일에는 커널을 구성하는 대부분의 값이 포함됩니다.
- `aqmcreateq.bat`(Windows 시스템) 또는 `aqmcreateq.sh`(UNIX 및 OS/400) 파일은 MQSeries 큐를 작성하는 스크립트입니다. 108 페이지의 『MQSeries 큐 작성』을 참조하십시오.

이러한 모든 파일은 편집에 도움이 되는 주석을 포함합니다.

이러한 파일은 백업하는 것이 좋습니다. 자세한 정보는 105 페이지의 『커널 유지 보수』를 참조하십시오.

## 설정 파일

설정 파일 `aqmsetup`은 다음을 포함하여 커널의 여러 초기 설정을 제어합니다.

- 구성 파일의 위치. 74 페이지의 『구성 파일』을 참조하십시오.
- 현재 디렉토리에 없는 경우, XML DTD의 위치.

- 사용된 메모리의 양을 변경하기 위한 C 인터페이스의 JNI(Java Native Interface) 환경 변수. 이는 C 실행 가능 모듈이 프로세스를 시작하고 해당 프로세스가 Java 가상 시스템을 인스턴스화하는 경우에 적용됩니다. 이 경우 메모리 사용은 aqmsetup 파일에서 다음 행들의 주석 표시를 없애거나 수정하여 제어할 수 있습니다.

```
#AQM_JNI_NATIVESTACKSIZE=1048576
#AQM_JNI_JAVASTACKSIZE=4194304
#AQM_JNI_MINHEAPSIZE=16777216
#AQM_JNI_MAXHEAPSIZE=268435426
```

모든 크기는 바이트 단위입니다.

샘플 aqmsetup 파일은 137 페이지의 『부록E. 설정 파일 샘플』에 제공되고 MQSeries Adapter Kernel 설치의 samples 디렉토리에도 제공됩니다.

필요하면 MQSeries Adapter Kernel을 처음 설치하고 설정 파일을 편집하십시오. 설치 후에는, 커널이 이전 목록에 설명된 대로 Java 메모리 부족 문제점을 발견하게 되는 경우에만 파일을 편집하십시오.

## 구성 파일

이 절에서는 커널 구성을 판별하는 aqmconfig.xml 파일을 설명합니다. 75 페이지의 『구성 파일의 구문 및 조직』에서는 구성 파일 구조에 대한 정보를 제공합니다. 96 페이지의 『구성 파일 편집』에서는 구성 파일 편집을 위한 최상의 실습 제안사항을 제공합니다.

MQSeries Adapter Kernel 구성은 XML 파일 aqmconfig.xml로 판별됩니다. 샘플 구성 파일은 129 페이지의 『부록D. 구성 파일 샘플』에 포함되고 MQSeries Adapter Kernel 설치의 samples 디렉토리에도 포함됩니다.

구성 파일에 지정된 값은 다음과 같은 커널 요소를 제어합니다.

- 소스 논리 ID
- 목적지 논리 ID
- 대상측에서의 어댑터 디먼 및 작업 프로그램 정보
- 추적 클라이언트
- 추적 서버

- 다음 스펙으로 판별되는 메시지 정렬 및 라우팅
  - 리시브 큐, 오류 큐 및 리플라이 큐의 이름
  - 메시지가 송신되어야 할 하나 이상의 디폴트 목적지
  - 메시지를 수신 또는 송신하는 MQSeries 큐 관리자 또는 JMS 큐 연결 팩토리의 이름
  - 메시지 또는 응답 수신에 적용되는 시간 종료
  - 각 메시지를 처리하는 커널의 대상측에 있는 대상 어댑터 클래스
  - 대상 어댑터에 고유한 추가 정보
  - 독립형 MQSeries Adapter Kernel을 실행하는 경우 대상측에서의 작업 프로그램 최소 수
  - 추적 사용 가능 및 사용 불가능과 추적 레벨 제어
  - 감사 로그 기록 사용 가능 및 사용 불가능
- 통신 모드

#### 구성 파일의 구문 및 조직

MQSeries Adapter Kernel의 구성은 LDAP(Lightweight Directory Access Protocol)를 기초로 하므로 구성 파일의 구조는 LDAP를 미러링합니다. 최상위 레벨 XML 요소 Epic는 LDAP 디렉토리의 최상위 레벨을 나타내고 하위 LDAP 오브젝트는 최상위 레벨 요소 내에 중첩된 XML 요소로 나타납니다. 일부 XML 요소는 LDAP 정보를 나타내는 필수 속성을 가집니다. 값은 요소 내용 또는 요소 속성으로 구성에 추가됩니다. 요소의 내용으로 지정된 구성 값의 예는 `<epictracellevel>-1</epictracellevel>`입니다. 이는 값 -1(가능한 모든 메시지)을 epictracellevel 요소에 지정합니다. 요소의 속성으로 지정된 구성 값의 예는 `<ePICTraceHandler epictracelhandler="com.ibm.logging.ConsoleHandler">`이고, 이는 추적 핸들러로 사용할 com.ibm.logging.ConsoleHandler 클래스를 지정합니다.

다음은 구성 파일에 사용된 상위 레벨 요소의 목록 및 설명입니다. 77 페이지의 『구성 파일에 사용된 XML 요소』에서는 구성 파일에 사용되는 완전한 요소 세트를 나열하고 설명합니다. 다른 여러 요소가 컨텍스트에서 사용되는 방법에 대한 예를 보려면 샘플 구성 파일을 참조하십시오.

- Epic -- aqmconfig.xml 파일의 필수 최상위 레벨 요소.
- ePICApplications -- Epic 요소의 필수 하위 요소.
- ePICApplication -- ePICApplications 요소의 필수 하위 요소. 커널이 서비스할 응용프로그램을 나열하고 정의합니다. 각 응용프로그램에 대해 완전히 정의된 ePICApplication 요소(하위 요소 포함)가 필요합니다.
- AdapterRouting -- ePICApplication 요소의 선택적 하위 요소. 큐 관리자와 관련 정보를 정의합니다.
- ePICBodyCategory -- AdapterRouting 요소의 필수 하위 요소. 커널이 라우트할 메시지의 본문 범주를 설정합니다.
- ePICBodyType -- ePICBodyCategory 요소의 필수 하위 요소. 커널이 라우트할 메시지의 본문 유형을 설정합니다. 메시지 목적지, 메시지 수신을 위한 통신 모드 및 메시지 형식기와 같은 항목의 정의가 포함됩니다.
- ePICAdapterDaemonExtensions -- 어댑터 디몬을 나타내는 ePICApplication 요소의 선택적 하위 요소. 응용프로그램 ID 및 어댑터 작업 프로그램 수를 포함하여 어댑터 디몬 관련 정보를 포함합니다.
- ePICTraceExtensions -- 추적 클라이언트 응용프로그램 또는 추적 서버 요소를 나타내는 ePICApplication 요소의 선택적 하위 요소. 추적 관련 정보를 정의합니다.

77 페이지의 그림8에는 구성 파일의 상위 레벨 구조가 나와 있습니다. 이것은 구성 파일의 작업 예로서 상위 레벨 요소 사이의 관계 및 의존 상태를 보여줍니다. 완전한 예제에 대해서는 129 페이지의 『부록D. 구성 파일 샘플』을 참조하십시오.

```

<?xml version="1.0" encoding="UTF-8"?>
<Epic o="ePIC">
  <ePICApplications o="ePICApplications">
    <!-- The following <ePICApplication> tag configures the kernel to work with
    an application named APP1. -->
    <ePICApplication epicappid="APP1">
      <!-- Tags here specify logging and trace information for the APP1
      application. -->
      <AdapterRouting cn="epicadapterrouting">
        <!-- Tags here specify the queue manager and its attributes. -->
        <ePICBodyCategory epicbodycategory="DEFAULT">
          <ePICBodyType epicbodytype="DEFAULT">
            <!-- Tags here specify the details of transporting and processing messages
            from APP1. -->
            </epicBodyType>
          </ePICBodyCategory>
        </AdapterRouting>
      </ePICApplication>
      <!-- The following <ePICApplication> tag starts an adapter daemon for the
      APP1 application. -->
      <ePICApplication epicappid="APP1Daemon">
        <!-- Specifications for the APP1Daemon adapter daemon, which works with
        the APP1 application. -->
        <ePICAdapterDaemonExtensions cn="epicappextensions">
          <epicdepappid>APP1</epicdepappid>
          <epicminworkers>1</epicminworkers>
        </ePICAdapterDaemonExtensions>
      </ePICApplication>
      <!-- The following <ePICApplication> tag configures the kernel to work with
      an application named APP2. -->
      <ePICApplication epicappid="APP2">
        <!-- Tags here specify logging and trace information for the APP2
        application. -->
        <AdapterRouting cn="epicadapterrouting">
          <!-- Tags here specify the queue manager and its attributes. -->
          <ePICBodyCategory epicbodycategory="DEFAULT">
            <ePICBodyType epicbodytype="DEFAULT">
              <!-- Tags here specify the details of transporting and processing messages
              from APP2. -->
              </epicBodyType>
            </ePICBodyCategory>
          </AdapterRouting>
        </ePICApplication>
        <!-- The following <ePICApplication> tag configures a trace client named
        TraceClient. -->
        <ePICApplication epicappid="TraceClient">
          <ePICTraceExtensions cn="epicappextensions">
            <!-- Tags here specify attributes of the trace client. -->
            </ePICTraceExtensions>
          </ePICApplication>
        </ePICApplications>
      </Epic>

```

## 그림 8. 구성 파일의 상위 레벨 구조

다음은 구성 파일에 사용된 전체 요소 세트의 목록과 설명입니다. 요소가 디폴트 값을 갖는 경우, 구성 요소가 명시적으로 지정되지 않은 값을 필요로 하는 경우에 커널이 해당 값을 사용합니다.

### 구성 파일에 사용된 XML 요소

**Epic** 구성 파일의 최상위 레벨 요소.

하위 요소:

- context
- ePICApplications(필수)

속성: o="ePIC"(필수)

### **context**

JMS(Java Message Service) 오브젝트가 사용될 때 JNDI(Java Naming and Directory Interface) 파일 시스템 컨텍스트(FSContext)의 루트를 지정합니다. 디폴트는 현재 디렉토리입니다. JMS가 사용되는 경우에 필요합니다. MQSeries Adapter Kernel과 함께 오브젝트를 사용하는 경우에 대해서는 116 페이지의 『JMS 오브젝트 저장영역 사용』을 참조하십시오.

하위 요소: 없음

속성: 없음

### **ePICApplications**

커널이 서비스하는 응용프로그램에 대한 정보를 포함합니다.

하위 요소: ePICApplication(필수)

속성: o="ePICApplications"(필수)

### **ePICApplication**

커널이 서비스하는 응용프로그램에 대한 정보를 지정합니다.

하위 요소:

- epiclogging
- epictrace
- epictracelevel
- epictraceclientid
- epiclogoninfoclassname
- AdapterRouting
- ePICTraceExtensions
- ePICAdapterDaemonExtensions

속성: `epicappid="application_ID"`, 여기서 `application_ID`는 올바른 응용프로그램 ID입니다(필수).

### **epiclogging**

감사 로그 기록 수행 여부를 판별합니다. 감사 로그 기록에는 WebSphere Business Integrator 제품이 필요합니다. 디폴트는 `false`입니다.

하위 요소: 없음

속성: 없음

### **epictrace**

추적 사용 여부를 판별합니다. 디폴트는 `false`입니다.

하위 요소: 없음

속성: 없음

### **epictracelevel**

`com.ibm.logging.IRecordType` 클래스로 지정된 상수를 사용하여 추적 레벨을 설정합니다. 디폴트는 0(메시지 없음)입니다. 추적에 대한 자세한 내용 및 유효한 추적 레벨 전체 목록에 대해서는 *Problem Determination Guide*를 참조하십시오.

하위 요소: 없음

속성: 없음

### **epictraceclientid**

추적 클라이언트 응용프로그램의 이름을 지정합니다. 디폴트는 `TraceClient`입니다.

하위 요소: 없음

속성: 없음

### **epiclogoninfoclassname**

EAB 어댑터 사용시 응용프로그램에 연결하기 위해 사용되는 로그온 클래스의 이름을 지정합니다. 디폴트는 `com.ibm.epic.adapters.eak.adapterdaemon.EpicLogonDefault`입니다.

하위 요소: 없음

속성: 없음

### **AdapterRouting**

메시지 유형과 메시지의 라우팅에 대한 정보를 포함합니다.

하위 요소:

- epicmqppqueuemgr
- epicuseremotequeueanagerToSend
- epicmqppqueuemgrhostname
- epicmqppqueuemgrportnumber
- epicmqppqueuemgrchannelname
- epicjmsconnectionfactoryname
- ePICBodyCategory(필수)

속성: cn="epicadapterrouting"(필수)

### **epicmqppqueuemgr**

MQSeries가 전송 메커니즘으로 사용되고 있는 경우, 사용될 큐 관리자의 이름을 지정합니다. 지정되지 않았거나 DEFAULT로 지정된 경우 디폴트 큐 관리자가 사용됩니다.

하위 요소: 없음

속성: 없음

### **epicuseremotequeueanagerToSend**

MQSeries가 전송 메커니즘으로 사용되고 있는 경우, 리모트 큐 관리자를 사용하여 메시지를 송신할 것인지를 지정합니다. 디폴트는 false입니다.

하위 요소: 없음

속성: 없음

### **epicmqppqueuemgrhostname**

MQSeries가 전송 메커니즘으로 사용되고 있는 경우, 큐 관리자가 있는 시스템의 TCP/IP 호스트 이름을 지정합니다. MQSeries 클라이언트가 사용되는 경우에만 필요합니다.

하위 요소: 없음



속성: 없음

#### **epicmqppqueuemgrportnumber**

MQSeries가 전송 메커니즘으로 사용되고 있는 경우, 큐 관리자 서버 프로세스의 포트 번호를 지정합니다. 디폴트는 1414(MQSeries 디폴트)입니다. MQSeries 클라이언트가 사용되는 경우에만 필요합니다.

하위 요소: 없음

속성: 없음

#### **epicmqppqueuemgrchannelname**

MQSeries가 전송 메커니즘으로 사용되고 있는 경우, 큐 관리자 서버의 채널 이름을 지정합니다. MQSeries 클라이언트가 사용되는 경우에만 필요합니다.

하위 요소: 없음

속성: 없음

#### **epicjmsconnectionfactoryname**

JMS가 전송 메커니즘으로 사용되고 있는 경우, JMS 연결 팩토리 이름을 지정합니다. 값은 *attribute=object*로 지정해야 합니다. 여기서 *attribute*는 LDAP 속성이고 *object*는 JMS 연결 오브젝트입니다. 오브젝트는 AdapterRouting 요소 아래 저장될 것입니다. 예를 들어 LDAP 속성 cn을 갖는 JMS 연결 오브젝트 QCFTEST1의 경우, 이 요소로 지정된 값은 cn=QCFTEST1입니다.

하위 요소: 없음

속성: 없음

#### **ePICBodyCategory**

송신되고 있는 메시지의 본문 범주를 지정합니다.

하위 요소: ePICBodyType(필수)

속성: *epicbodycategory=body\_category*. 여기서 *body\_category*는 송신 중인 메시지의 본문 범주를 지정합니다(필수).

#### **ePICBodyType**

송신되고 있는 메시지의 본문 유형을 지정합니다.

하위 요소:

- epiccommandclassname
- epiccommandtype
- epiccommandejbmapper
- epiccommandejbmethod
- epiccommandejbmethodparmtype
- epiccommandejburl
- epiccommandejbinitialcontext
- epicdestids
- epicreceivemode
- epicmessageformatter
- epicreceivetimeout
- epicreceivemppqueue
- epicerrormppqueue
- epicreplymppqueue
- epicjmsreceivequeueenamel
- epicjmserrorqueueenamel
- epicjmsreplyqueueenamel
- epicreceivefiledir
- epiccommitfiledir
- epicerrorfiledir

속성: epicbodytype=*body\_type*. 여기서 *body\_type*은 송신되고 있는 메시지의 본문 유형을 지정합니다(필수).

### **epiccommandclassname**

메시지를 처리하기 위해 호출되는 EJB 명령이나 EAB 대상 어댑터의 이름을 지정합니다. 어댑터 디먼이나 WebSphere Application Server를 사용하여 메시지를 수신하는 경우에 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommandtype**

대상 어댑터 유형을 지정합니다. MQAKEAB 및 MQAKEJB가 가능한 값입니다. MQAKEAB는 표준 MQSeries Adapter Kernel EAB 대상 어댑터를 지정합니다. MQAKEJB는 WebSphere Application Server의 커널의 대상층에서 엔터프라이즈 bean이 사용됨을 지정합니다. 디폴트는 MQAKEAB입니다. 대상 어댑터가 엔터프라이즈 bean일 때 MQAKEJB 값이 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommandejbmapper**

입력 데이터를 맵핑하는 데 사용하는 TDCMapper 클래스 이름을 지정합니다. 디폴트는 TDCGenericMapper입니다. 대상 어댑터가 엔터프라이즈 bean일 때 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommandejbmethod**

엔터프라이즈 bean에서 호출하기 위한 메소드 이름을 지정합니다. 이 메소드는 입력으로 TerminalDataContainer 오브젝트를 받아들이고 TerminalDataContainer 오브젝트를 리턴해야 합니다. 디폴트는 execute입니다. 대상 어댑터가 엔터프라이즈 bean일 때 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommandejbmethodparmtype**

엔터프라이즈 bean에서 호출되는 메소드의 매개변수로 사용되고 있는 오브젝트의 클래스 이름을 지정합니다. 디폴트는 TDCMapper에 의해 리턴되는 오브젝트의 클래스 이름입니다. 대상 어댑터가 엔터프라이즈 bean일 때 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommandejbur1**

전개된 엔터프라이즈 bean의 URL을 `IIOP://host_name:port` 형식으로 지정하는데, 여기서 `host_name`은 EJB 서버의 호스트 이름이고 `port`는 이름 서버가 대기하는 포트(디폴트 값 900)입니다. 디폴트는 `IIOP:///`입니다. 대상 어댑터가 엔터프라이즈 bean일 때 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommandejbinitialcontext**

엔터프라이즈 bean의 호스트 이름을 검색하는 데 사용되는 초기 컨텍스트 팩토리 이름을 지정합니다. 디폴트는 `com.ibm.ejs.ns.jndi.CNInitialContextFactory`입니다. 대상 어댑터가 엔터프라이즈 bean일 때 필요합니다.

하위 요소: 없음

속성: 없음

### **epicdestids**

메시지 목적지로 사용될 하나 이상의 응용프로그램 ID를 지정합니다. 응용프로그램이 메시지를 송신하고 있고 목적지 논리 ID가 `NONE`으로 설정된 경우에 필요합니다.

하위 요소: 없음

속성: 없음

### **epicreceivemode**

사용될 통신 모드를 지정합니다. 올바른 통신 모드 목록 및 설명은 113 페이지의 『부록A. 통신 모드』를 참조하십시오. 응용프로그램이 메시지를 수신하는 경우에 필요합니다.

하위 요소: 없음

속성: 없음

### **epicmessageformatter**

`epicreceivemode` 값과 사용되는 전송 메소드에 따라 사용할 메시지 형

식기를 지정합니다. 메시지 형식기와 전송 메소드에 대해서는 115 페이지의 표10 및 115 페이지의 표11을 참조하십시오.

하위 요소: 없음

속성: 없음

#### **epicreceptivetimeout**

리시버가 시간 종료 이전에 메시지를 기다리는 기간을 밀리초 단위로 지정합니다. 디폴트는 0입니다. -1 값은 시간 종료 없음(무한 대기)을 지정합니다.

하위 요소: 없음

속성: 없음

#### **epicreceptivemqppqueue**

메시지를 수신하는 큐의 이름을 지정합니다. epicreceptivemode 요소가 MQSeries 통신 모드를 지정하는 경우 필요합니다. MQSeries 통신 모드 목록은 113 페이지의 『부록A. 통신 모드』를 참조하십시오.

하위 요소: 없음

속성: 없음

#### **epicerrormqppqueue**

오류 메시지를 넣을 큐의 이름을 지정합니다. 오류 메시지 큐잉이 사용되고 있고 epicreceptivemode 요소가 MQSeries 통신 모드를 지정하는 경우에 필요합니다. MQSeries 통신 모드 목록은 113 페이지의 『부록A. 통신 모드』를 참조하십시오.

하위 요소: 없음

속성: 없음

#### **epicreplymqppqueue**

응답 메시지를 수신하는 큐의 이름을 지정합니다. 응답 요청이 사용되고 있고 epicreceptivemode 요소가 MQSeries 통신 모드를 지정하는 경우에 필요합니다. MQSeries 통신 모드 목록은 113 페이지의 『부록A. 통신 모드』를 참조하십시오.

하위 요소: 없음

속성: 없음

#### **epicjmsreceivequeue**

메시지를 수신하는 큐의 이름을 지정합니다. JMS 통신 모드에 필요합니다. 오브젝트는 ePICBodyType 요소 아래 저장될 것입니다. 값은 *attribute=object*로 지정해야 합니다. 여기서 *attribute*는 LDAP 속성이고 *object*는 JMS 큐 오브젝트의 이름입니다. 예를 들어 LDAP 속성 *cn*을 갖는 JMS 오브젝트 TEST1AIQ의 경우, 이 요소로 지정된 값은 *cn=TEST1AIQ*입니다.

하위 요소: 없음

속성: 없음

#### **epicjmserrorqueue**

오류 메시지를 넣을 큐의 이름을 지정합니다. 오류 메시지 큐잉이 JMS 통신 모드와 함께 사용되고 있는 경우 필요합니다. 오브젝트는 ePICBodyType 요소 아래 저장될 것입니다. 값은 *attribute=object*로 지정해야 합니다. 여기서 *attribute*는 LDAP 속성이고 *object*는 JMS 큐 오브젝트의 이름입니다. 예를 들어 LDAP 속성 *cn*을 갖는 JMS 오브젝트 TEST1AEQ의 경우, 이 요소로 지정된 값은 *cn=TEST1AEQ*입니다.

하위 요소: 없음

속성: 없음

#### **epicjmsreplyqueue**

응답 메시지를 수신하는 큐의 이름을 지정합니다. 응답 요청이 JMS 통신 모드와 함께 사용되고 있는 경우 필요합니다. 오브젝트는 ePICBodyType 요소 아래 저장될 것입니다. 값은 *attribute=object*로 지정해야 합니다. 여기서 *attribute*는 LDAP 속성이고 *object*는 JMS 큐 오브젝트의 이름입니다. 예를 들어 LDAP 속성 *cn*을 갖는 JMS 오브젝트 TEST1RPL의 경우, 이 요소로 지정된 값은 *cn=TEST1RPL*입니다.

하위 요소: 없음

속성: 없음

### **epicreceivefiledir**

메시지를 수신하는 디렉토리의 이름을 지정합니다. FILE 통신 모드에 필요합니다.

하위 요소: 없음

속성: 없음

### **epiccommitfiledir**

확약될 때까지 수신된 메시지를 보유하는 디렉토리의 이름을 지정합니다. 메시지가 수신될 때 FILE 통신 모드에 필요합니다.

하위 요소: 없음

속성: 없음

### **epicerrorfiledir**

오류 메시지를 넣을 디렉토리의 이름을 지정합니다. 오류 메시지 큐잉이 FILE 통신 모드와 함께 사용되고 있는 경우 필요합니다.

하위 요소: 없음

속성: 없음

### **ePICAdapterDaemonExtensions**

어댑터 디먼 확장자에 대한 정보를 포함합니다.

하위 요소:

- epicdepappid
- epicminworkers

속성: cn="epicappextensions"(필수)

### **ePICTraceExtensions**

추적 확장자에 대한 정보를 포함합니다. 이 요소와 그 하위에 대해서는 *Problem Determination Guide*를 참조하십시오.

하위 요소:

- epicdepappid
- epictracesyncoperation
- epictracemessagefile

- epictracehandler
- ePICTraceHandler

속성: cn="epicappextensions"(필수)

### **epicdepappid**

어댑터 디먼이 서비스 중인 응용프로그램의 ID를 지정합니다. 디폴트는 어댑터 디먼이 시작된 응용프로그램 ID입니다.

하위 요소: 없음

속성: 없음

### **epicminworkers**

어댑터 디먼이 시작한 어댑터 작업 프로그램의 수를 지정합니다. 디폴트는 1입니다.

하위 요소: 없음

속성: 없음

### **공통 구성**

이 절에서는 다양한 통신 전송을 사용하여 메시지를 송수신하기 위한 값을 포함한 여러 공통 구성 시나리오의 구성 값을 나열합니다. 메시지가 송신될 때 구성 값은 소스측과 대상측 모두에서 확보합니다. 메시지가 수신될 때 구성 값은 대상측에서만 확보됩니다. 소스 및 대상은 각각의 논리 ID에 의해 표시됩니다. 이 예에서는 소스와 대상이 두 개의 별개 시스템에 있는 것으로 간주합니다. 대상 응용프로그램 ID가 아직 설정되지 않은 경우 소스 구성의 epicdestids 요소 값에서 결정됩니다.

주: 구성 시나리오에서는 적용 가능하고 설정 가능한 요소 구성 값을 나열합니다.

해당 요소에 적용되는 디폴트 값에 대해서는 77 페이지의 『구성 파일에 사용된 XML 요소』의 요소 목록을 참조하십시오.

**MQSeries 공통 구성:** 이 절에서는 MQSeries가 통신 전송으로 사용될 때의 공통 구성을 설명합니다. epicreceivemode 요소는 MQPP나 MQRFH2와 같은 MQSeries 통신 모드를 지정합니다. 다음과 같은 시나리오가 나열됩니다.



- 표2에서는 MQSeries 서버에서 MQSeries 서버로 메시지를 송신할 때 설정해야 하는 구성 요소를 보여줍니다.
- 90 페이지의 표3에서는 리모트 큐 관리자를 사용하고 있는 MQSeries 서버에서 MQSeries 서버로 메시지를 송신할 때 설정해야 하는 구성 요소를 보여줍니다.
- 90 페이지의 표4에서는 호스트 서버를 사용하고 있는 MQSeries 클라이언트로부터 MQSeries 서버로 메시지를 송신할 때 설정해야 하는 구성 요소를 보여줍니다.
- 91 페이지의 표5에서는 MQSeries 서버에서 메시지를 수신할 때 설정해야 하는 구성 요소를 보여줍니다.
- 92 페이지의 표6에서는 호스트 서버를 사용하고 있는 MQSeries 클라이언트에서 메시지를 수신할 때 설정해야 하는 구성 요소를 보여줍니다.

표2. 공통 구성: MQSeries 서버에서 다른 MQSeries 서버로 메시지 송신

소스 구성	대상 구성
	epicreivemode 요소는 MQSeries 통신 모드를 지정합니다.
epicmqppqueuemgr 요소는 큐 관리자 이름을 지정합니다. 이 큐 관리자는 소스 응용프로그램 시스템에 있어야 합니다.	
	epicreivemqppqueue 요소는 리시브 큐 이름을 지정합니다. 이 큐는 대상 응용프로그램 시스템의 MQSeries 리모트 큐이거나 MQSeries 클러스터의 일부여야 합니다.
epicreplymqppqueue 요소는 리플라이 큐 이름을 지정합니다. 이 큐는 송신자 시스템에 있는 MQSeries 로컬 큐이거나 MQSeries 클러스터의 일부여야 합니다. 동기 요청 및 응답에만 사용됩니다.	
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.
epicreivetimeout 요소는 시간 종료가 되기 전에 수신자가 응답을 대기하는 시간을 지정합니다.	

표 3. 공통 구성: 리모트 큐 관리자를 통해 MQSeries 서버에서 MQSeries 서버로 메시지 송신

소스 구성	대상 구성
	epicreceivemode 요소는 MQSeries 통신 모드를 지정합니다.
epicmqppqueuemgr 요소는 큐 관리자 이름을 지정합니다. 이 큐 관리자는 소스 응용프로그램 시스템에 있어야 합니다.	epicmqppqueuemgr 요소는 큐 관리자 이름을 지정합니다. 이 큐 관리자는 대상 응용프로그램 시스템에 있어야 합니다. 이름을 지정해야 합니다. 디폴트 값을 사용할 수 없습니다.
epicremotequeueanagertosend 요소는 메시지 송신에 리모트 큐 관리자가 사용되고 있음을 지정합니다.	
	epicreceivemqppqueue 요소는 리시브 큐 이름을 지정합니다. 이 큐는 대상 응용프로그램 시스템에 있는 MQSeries 로컬 큐이거나 MQSeries 클러스터의 일부여야 합니다.
epicreplymqppqueue 요소는 리플라이 큐 이름을 지정합니다. 이 큐는 송신자 시스템에 있는 MQSeries 로컬 큐이거나 MQSeries 클러스터의 일부여야 합니다. 동기 요청 및 응답에서만 사용 됩니다.	
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.
epicreceive timeout 요소는 시간 종료가 되기 전에 수신자가 응답을 대기하는 시간을 지정합니다.	

표 4. 공통 구성: 호스트 서버를 사용하고 있는 MQSeries 클라이언트에서 MQSeries 서버로 메시지 송신

소스 구성	대상 구성
	epicreceivemode 요소는 MQSeries 통신 모드를 지정합니다.
epicmqppqueuemgr 요소는 큐 관리자 이름을 지정합니다. 이 큐 관리자는 송신자 클라이언트의 호스트 시스템에 있어야 합니다.	
epicmqppqueuemgrhostname 요소는 MQSeries 서버 시스템의 호스트 이름을 지정합니다.	
epicmqppqueuemgrportnumber 요소는 서버 시스템에서 큐 관리자 서버 프로세스의 포트 번호를 지정합니다.	

표 4. 공통 구성: 호스트 서버를 사용하고 있는 MQSeries 클라이언트에서 MQSeries 서버로 메시지 송신 (계속)

소스 구성	대상 구성
epicmqppqueuemgrchannelnumber 요소는 큐 관리자 서버의 채널 번호를 지정합니다.	
	epicreceivemqppqueue 요소는 리시브 큐 이름을 지정합니다. 이 큐는 대상 응용프로그램 시스템의 MQSeries 리모트 큐이거나 MQSeries 클러스터의 일부여야 합니다.
epicreplymqppqueue 요소는 리플라이 큐 이름을 지정합니다. 이 큐는 송신자 클라이언트의 호스트 시스템에 있는 MQSeries 로컬 큐이거나 MQSeries 클러스터의 일부여야 합니다. 동기 요청 및 응답에만 사용됩니다.	
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.
epicreceivevtimeout 요소는 시간 종료가 되기 전에 수신자가 응답을 대기하는 시간을 지정합니다.	

표 5. 공통 구성: 메시지를 수신하는 MQSeries 서버

소스 구성	대상 구성
적용할 수 없음.	epicreceivevmode 요소는 MQSeries 통신 모드를 지정합니다.
	epicmqppqueuemgr 요소는 큐 관리자 이름을 지정합니다. 이 큐 관리자는 대상 응용프로그램 시스템에 있어야 합니다.
	epicreceivemqppqueue 요소는 리시브 큐 이름을 지정합니다. 이 큐는 대상 시스템에 있는 MQSeries 로컬 큐여야 합니다.
	epicerrormqppqueue 요소는 오류 큐 이름을 지정합니다. 이 큐는 대상 시스템에 있는 MQSeries 로컬 큐이거나 클러스터의 일부여야 합니다. 어댑터 작업 프로그램을 사용하는 경우에만 필요합니다.
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.

표 5. 공통 구성: 메시지를 수신하는 MQSeries 서버 (계속)

소스 구성	대상 구성
	epicreivetimeout 요소는 시간 종료가 되기 전에 수신자가 메시지 수신을 대기하는 시간을 지정합니다.

표 6. 공통 구성: 메시지를 수신하는 호스트 서버를 사용하고 있는 MQSeries 클라이언트

소스 구성	대상 구성
적용할 수 없음.	epicreivemode 요소는 MQSeries 통신 모드를 지정합니다.
	epicmqqpqueuemgr 요소는 큐 관리자 이름을 지정합니다. 이 큐 관리자는 수신자의 클라이언트 호스트 시스템에 있어야 합니다.
	epicmqqpqueuemgrhostname 요소는 MQSeries 서버 시스템의 호스트 이름을 지정합니다.
	epicmqqpqueuemgrportnumber 요소는 서버 시스템에서 큐 프로세스의 서버 프로세스 포트 번호를 지정합니다.
	epicmqqpqueuemgrchannelnumber 요소는 큐 관리자 서버의 채널 번호를 지정합니다.
	epicreivemqqpqueue 요소는 리시브 큐 이름을 지정합니다. 이 큐는 수신자 클라이언트 호스트 시스템에 있는 MQSeries 로컬 큐여야 합니다.
	epicerrormqqpqueue 요소는 오류 큐 이름을 지정합니다. 이 큐는 수신자 클라이언트 호스트 시스템에 있는 MQSeries 로컬 큐이거나 클러스터의 일부여야 합니다. 어댑터 작업 프로그램을 사용하는 경우에만 필요합니다.
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.
	epicreivetimeout 요소는 시간 종료가 되기 전에 수신자가 메시지 수신을 대기하는 시간을 지정합니다.

**JMS 공통 구성:** 이 절에서는 통신 전송으로 JMS를 사용하는 경우의 공통 구성을 설명합니다. epicreivemode 요소는 JMS를 지정합니다.

MQSeries JMS 구현이 사용되고 있는 경우, 적절한 MQSeries 오브젝트가 있어야 합니다. 예를 들어, JMS 큐 연결 팩토리는 MQSeries 서버의 큐 관리자와 관련되어야 하고, JMS 큐는 MQSeries 큐와 관련되어야 합니다. MQSeries 오브젝트는 구성 목록에 나열될 필요는 없으나 지원하는 MQSeries 오브젝트는 존재해야 합니다.

다음과 같은 시나리오가 나열됩니다.

- 표7에서는 JMS를 통해 메시지를 송신할 때 설정되어야 하는 구성 요소를 보여줍니다.
- 표8에서는 JMS를 통해 메시지를 수신할 때 설정되어야 하는 구성 요소를 보여줍니다.

표7. 공통 구성: JMS를 통해 메시지 송신

소스 구성	대상 구성
	epicreceivemode 요소는 JMS 통신 모드를 지정합니다.
epicjmsconnectionfactoryname 요소는 JMS 큐 연결 팩토리 이름을 지정합니다. 참조된 오브젝트는 구성에 존재해야 합니다.	
	epicjmsreceivequeueenamel 요소는 JMS 리시브 큐 이름을 지정합니다. 참조된 오브젝트는 구성에 존재해야 합니다.
epicjmsreplyqueueenamel 요소는 JMS 리플라이 큐 이름을 지정합니다. 참조된 오브젝트는 구성에 존재해야 합니다. 동기 요청 및 응답에만 사용됩니다.	
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.
epicreceivetimeout 요소는 시간 종료가 되기 전에 수신자가 응답을 대기하는 시간을 지정합니다.	

표8. 공통 구성: JMS를 통한 메시지 수신

소스 구성	대상 구성
적용할 수 없음.	epicreceivemode 요소는 JMS 통신 모드를 지정합니다.

표 8. 공통 구성: JMS를 통한 메시지 수신 (계속)

소스 구성	대상 구성
	epicjmsconnectionfactoryname 요소는 JMS 큐 연결 팩토리 이름을 지정합니다. 참조된 오브젝트는 구성에 존재해야 합니다.
	epicjmsreceivequeueenamenam은 JMS 리시브 큐 이름을 지정합니다. 참조된 오브젝트는 구성에 존재해야 합니다.
	epicjmserrorqueueenamenam 요소는 JMS 오류 큐 이름을 지정합니다. 참조된 오브젝트는 구성에 존재해야 합니다.
	epicmessageformatter 요소는 사용할 형식기 이름을 지정합니다.
	epicreceivevertimeout 요소는 시간 종료가 되기 전에 수신자가 메시지 수신을 대기하는 시간을 지정합니다.

**어댑터 공통 구성:** 이 절에서는 대상측에서 어댑터가 호출될 때의 공통 구성을 설명합니다. 대상이 EAB(Enterprise Access Builder) 대상 어댑터(MQAKEAB의 epiccommandtype 값에 의해 지정됨)인지 또는 EJB 서비스 세션 bean 대상 어댑터(MQAKEJB의 epiccommandtype 값에 의해 지정됨)인지에 따라 다른 구성 값이 사용됩니다.

**주:** EJB 서비스 세션 bean 대상 어댑터는 WebSphere Application Server에서 만 지원됩니다.

MQAKEAB의 epiccommandtype 값의 경우 다음과 같은 추가 요소 값을 지정합니다.

- epiclogoninfoclassname
- epiccommandclassname

MQAKEJB의 epiccommandtype 값의 경우 다음과 같은 추가 요소 값을 지정합니다.

- epiccommandclassname
- epiccommandejbmethod

- epiccommandejbmethodparatype
- epiccommandejburl
- epiccommandejbinitialcontext
- epiccommandejbmapper

### 구성에 어댑터 정보 추가

새 어댑터가 커널 구성에 추가되면 최소한 여러 스펙을 구성 파일에 추가해야 합니다. 최소 구성 파일의 예에 대해서는 aqmconfig.minimum.xml 파일을 참조하십시오. 이 파일은 129 페이지의 『부록D. 구성 파일 샘플』에 나와 있으며 MQSeries Adapter Kernel 설치의 samples 디렉토리에도 포함되어 있습니다.

다음 스펙은 새 어댑터가 추가될 때 구성에 추가해야 하는 최소 정보량을 나타냅니다.

- 소스 어댑터 (메시지 송신)
  - 소스 어댑터가 실행 중인 응용프로그램의 ID.
  - 디폴트 큐 관리자. MQSeries가 전송 메커니즘으로 사용되고 소스 어댑터와 같은 시스템에 설치되어 실행되고 있는 경우, 큐 관리자를 구성할 필요가 없습니다.
  - 메시지의 목적지 논리 ID. 모든 메시지가 같은 목적지로 이동되면 본문 범주 DEFAULT와 본문 유형 DEFAULT를 사용하십시오.
  - 소스 어댑터가 메시지를 송신하고 있는 각 목적지 논리 ID의 리시브 큐.
- 대상 어댑터 (메시지 수신)
  - 대상 어댑터가 실행 중인 응용프로그램의 ID.
  - 디폴트 큐 관리자. MQSeries가 전송 메커니즘으로 사용되고 소스 어댑터와 같은 시스템에 설치되어 실행되고 있는 경우, 큐 관리자를 구성할 필요가 없습니다.
  - MQSeries의 수신 모드. 일반적으로 모든 메시지에 대해 동일합니다. 이 경우 본문 범주 DEFAULT와 본문 유형 DEFAULT를 사용하십시오.
  - 리시브 큐. 모든 메시지에 대해 같을 경우, 본문 범주 DEFAULT와 본문 유형 DEFAULT를 사용하십시오.

- 대상 어댑터가 메시지를 처리할 때 오류가 발생하는 경우 오류 큐. 일반적으로 모든 메시지에 대해 동일합니다. 이 경우 본문 범주 DEFAULT와 본문 유형 DEFAULT를 사용하십시오.
- 메시지가 수신될 때 호출될 대상 어댑터 클래스 이름. 이것은 본문 범주와 본문 유형에 고유합니다.
- 수신 시간 종료 값. 높은 CPU 사용을 방지하기 위해 적절한 값을 설정하는 것이 권장됩니다. 일반적으로 모든 메시지에 대해 동일합니다. 이 경우 본문 범주 DEFAULT와 본문 유형 DEFAULT를 사용하십시오.

추가 대상 어댑터의 경우, 같은 리시브 큐가 사용되고 있다면 같은 정보로 충분할 수 있습니다. 이 경우 달리 지정해야 하는 정보는 특정 본문 범주와 본문 유형에 대해 호출할 대상 어댑터 클래스 이름뿐입니다.

- 스펙 추적

- 추적이 켜져 있는지 또는 꺼져 있는지 여부.
- 추적 레벨.
- 소스 어댑터 및 대상 어댑터에 대한 추적 목적지를 포함한 추가 추적 스펙. 디폴트로 추적은 커널이 시작된 명령 프롬프트 창이나 터미널에 표시됩니다.

### 구성 파일 편집

텍스트 편집기나 전용 XML 편집기를 사용하여 구성 파일을 편집하십시오. DTD 파일 aqmconfig.dtd는 XML 편집기 사용자를 위한 커널 설치 samples 디렉토리에 제공됩니다. XML 편집기 Xeena는 IBM alphaWorks 웹 사이트 [www.alphaworks.ibm.com](http://www.alphaworks.ibm.com)에서 다운로드할 수 있습니다. 다음 권장사항이 구성 파일 편집에 적용됩니다.

- 구성 파일을 편집하기 전에 원하는 구성에 대한 모든 관련 정보를 모으십시오. 구성에 속하는 응용프로그램 및 큐의 이름, 교환 중인 메시지 유형, 사용 중인 통신 모드 및 추적 프로그램과 기타 확장자에 대한 정보가 포함됩니다.
- 샘플 aqmconfig.xml 파일을 samples 디렉토리에서 원하는 위치로 복사하십시오. 구성 파일의 사본 이름을 바꾸십시오. 사본을 편집하십시오.
- 주석을 사용하여 구성 파일의 다른 섹션을 식별하고 구성에 사용되는 특정 값을 문서화하십시오(예를 들어, 응용프로그램 ID, 메시지 큐 이름 및 시간 종료).



값). XML에서 주석은 문자 `<!--`로 시작하고 문자 `-->`로 끝납니다. 주석은 다음 예에서와 같이 여러 줄에 걸쳐 있을 수 있습니다.

```
<!--  
    주석 텍스트  
-->
```

XML은 다른 주석 내의 주석을 허용하지 않습니다.

- 응용프로그램 ID에 따라 구성 파일을 조직하십시오. 각 응용프로그램 ID의 항목을 함께 보존하십시오.
- 전용 XML 편집기를 사용 중이 아니라면 파일이 저장될 때 행 끝을 유지하고 행을 분할하지 않는 텍스트 편집기를 사용하십시오. 이러한 종류의 텍스트 편집기 예제는 Windows 시스템의 Notepad와 UNIX의 vi 또는 Emacs입니다.
- XML은 대소문자를 구분함을 기억하십시오. 모든 태그(요소) 이름 및 속성에 올바른 대소문자를 사용하도록 주의를 기울여야 합니다. 태그에 틀린 대소문자를 사용하면 구성 파일이 무효화될 수 있습니다. 전용 XML 편집기를 사용하면 대소문자 오류를 막는 데 도움이 됩니다.
- 본문 범주와 본문 유형의 디폴트 값을 사용하려고 하고 값이 아직 디폴트 값으로 설정되지 않은 경우, 구성 파일의 각각에 대해 DEFAULT 값을 구성해야 합니다. 그렇지 않으면 커널이 디폴트 값을 사용하지 않습니다.
- 구성 파일을 프로덕션 환경에 두기 전에 검증하십시오. 『구성 파일 검증』을 참조하십시오.
- 구성 파일 변경사항은 다음 번에 커널 프로세스가 시작될 때 적용됩니다. 구성 파일이 변경될 때 프로세스가 실행 중이라면 변경사항이 적용되도록 프로세스를 정지한 다음 재시작해야 합니다. 현재 프로덕션 환경에 있는 구성 파일을 편집하는 경우 매우 주의해야 합니다.
- 구성 파일은 편집할 때마다 백업하십시오.

### 구성 파일 검증

구성 파일을 편집하고 프로덕션 환경에 배치하기 전에 검증할 것을 권장합니다. 구성 파일을 검증하려면 다음과 같은 일반 단계를 수행하십시오.

1. 테스트를 검증하고 설정하는 구성 파일 검증 디렉토리를 작성하십시오.
2. 검증 XML 메시지를 작성하십시오.

3. 검증 테스트를 지원하기 위해 메시지 큐를 설정하십시오.
4. 메시지를 송신하고 메시지를 수신하는 구성 파일 검증을 설정한 다음 실행하십시오.
5. 구성 파일이 올바른지 판별하기 위해 테스트 결과를 조사하십시오.

검증 XML 메시지를 작성하는 데 도움이 되는 유틸리티와 구성 파일 검증 테스트는 둘 다 커널의 일부로 제공됩니다.

구성 파일 검증 테스트는 `sendMsg` 메소드를 호출하고 커널의 소스측에 있는 원시 어댑터에서 커널의 대상측에 있는 어댑터 디먼으로 검증 XML 메시지를 송신합니다. 소스 어댑터와 대상 어댑터는 필요 없습니다. 그러나 대상 어댑터가 있다면 대상 응용프로그램으로의 메시지 송신을 테스트할 수도 있습니다.

프로시저는 다음과 같습니다.

**주:** 해당 프로시저에서 사용하기 편리하도록 여러 스크립트가 제공됩니다. 원하면 스크립트를 복사한 다음 사본을 편집하여 사용자 소유 버전을 작성하십시오. OS/400을 사용 중이라면 스크립트의 UNIX 버전이 **qsh** 세션에서 실행될 수 있습니다. 제어 언어(CL) 프롬프트에서 **QSH** 시작(**STRQSH**) 명령을 입력하여 **qsh** 세션을 시작할 수 있습니다.

- 단계 1. 명령 프롬프트 창을 여십시오.
- 단계 2. 구성 파일 검증 디렉토리를 작성하십시오. 구성 파일과 설정 파일을 이 디렉토리에 복사하십시오.
- 단계 3. 검증 디렉토리로 변경하십시오.
- 단계 4. 다음 명령을 입력하여 검증 XML 메시지를 작성하십시오.
  - `aqmcrmsg.bat` (Windows 시스템)
  - `aqmcrmsg.sh` (UNIX 및 OS/400)
- 단계 5. 옵션 목록이 표시됩니다. 옵션을 선택하고 Enter를 누르십시오. 각각의 값을 입력하십시오. 값이 입력되는 순서는 중요하지 않습니다. 옵션 예는 `set sourcelogicalid`, `set msgtype` 및 `set bodycategory`입니다. 옵션 20, 21, 22 및 23에 대한 값을 입력해야 합니다. 옵션 24 또는 241을 사용하여 메시지 본문 데이터를 제공할 수 있습니다. 다른 값은 필요없습니다.

단계 6. 옵션 1을 입력하여 검증 XML 파일을 작성하십시오. 검증 XML 파일이 현재 디렉토리에 작성되어 EpicMessagenn.xml로 이름 지정됩니다. 여기서 *nn*은 XML 파일의 개수입니다.

단계 7. 검증 유틸리티를 종료하려면 옵션 0을 입력하십시오.

단계 8. 검증을 지원하기 위해 적절한 메시지 큐를 설정하십시오.

단계 9. 검증 디렉토리에서 일시적으로 설정 파일을 가리키도록 AQMSETUPFILE 환경 변수를 설정하십시오.

- Windows 시스템의 명령 프롬프트에 다음을 입력하십시오:

```
set AQMSETUPFILE=E:\run_time_files\aqmsetup
```

여기서 E:\는 올바른 드라이브이고 *run\_time\_files*는 검증 디렉토리입니다.

- UNIX 및 OS/400에서 다음 명령을 입력하십시오. 명령 예에서는 Korn 셸을 사용하고 있다고 가정합니다. 다른 셸을 사용하고 있다면 명령을 변경하십시오.

```
export AQMSETUPFILE=root_directory/run_time_files/aqmsetup
```

여기서 *root\_directory*는 커널의 설치 디렉토리이고 *run\_time\_files*는 검증 디렉토리입니다. OS/400에서 *aqmsetup* 파일은 항상 IFS 홈 디렉토리(/home/user\_name)에 있어야 합니다.

필요하면 검증되고 있는 구성 파일을 가리키도록 검증 디렉토리의 설정 파일을 편집하십시오.

단계 10. 테스트하기 위해 다음 중 하나를 선택하십시오.

- 커널의 소스측만.
- 대상 응용프로그램으로의 모든 경로에서 메시지를 라우트할 수 있는지 여부. 이 테스트의 경우 대상 어댑터가 이미 있어야 합니다.
- 추적.

먼저 소스측을 테스트한 다음 대상측을 테스트하십시오. 소스측만 테스트하려면 어댑터 디면을 끄십시오. 대상측도 테스트하려면 어댑터 디면을 켜십시오. 대상 어댑터가 아직 없다면 적절한 대상 어댑터에 대한 명

령을 호출하려는 시점까지 어댑터 디먼이 메시지를 처리하는지 여부를 테스트할 수 있습니다. 대상 어댑터가 아직 사용 중이 아니라면 특히 추적 을 사용 가능화하십시오.

단계 11. 검증 테스트를 실행하십시오. 임의의 디렉토리에서 다음 명령을 입력하십시오.

- Windows 시스템에서:

```
aqmsndmsg.bat -a source_logical_identifier -f XML_message_file
```

- UNIX 및 OS/400에서:

```
aqmsndmsg.sh -a source_logical_identifier -f XML_message_file
```

설명:

*source\_logical\_identifier*

소스 논리 ID를 나타냅니다. 이 값은 98 페이지의 5 단계에서 옵션 20에 대해 입력한 소스 논리 ID 값과 일치해야 합니다.

*XML\_message\_file*

XML 메시지 파일을 나타냅니다.

주: 이 테스트에 대한 모든 옵션 목록은 다음 명령을 입력하여 표시할 수 있습니다.

Windows 시스템에서:

```
aqmsndmsg.bat -?
```

UNIX 및 OS/400에서:

```
aqmsndmsg.sh -?
```

-?는 Korn 셸에서만 작동합니다. Bourne 셸 또는 C 셸과 같은 다른 UNIX 셸을 사용하는 경우, 물음표 앞에 백슬래시를 사용하십시오(즉, -\?).

단계 12. 결과를 조사하십시오. 검증 메시지는 올바른 본문 범주, 본문 유형 및 데이터를 포함합니다.

- 커널의 소스측만 테스트하는 중이라면(즉, 어댑터 디먼이 시작되지 않은 경우) 메시지가 라우트된 큐를 조사하십시오.

- 해당 큐의 검증 메시지를 보는 경우, 구성 파일의 항목이 검증됩니다.
- 해당 큐의 검증 메시지를 보지 않을 경우, 실행 파일을 점검하십시오. 추적이 사용 가능하게 되면 추적 메시지를 점검하십시오.
- 커널의 대상측을 테스트하고 있고 대상 어댑터가 있는 경우, 대상 응용프로그램을 점검하십시오.
  - 대상 응용프로그램이 검증 메시지를 수신하는 경우, 구성 파일의 해당 항목이 검증됩니다.
  - 대상 응용프로그램이 검증 메시지를 수신하지 않을 경우, 예외 파일을 점검하십시오. 추적이 사용 가능하게 되면 추적 메시지를 점검하십시오.
- 커널의 대상측을 테스트하고 있고 대상 어댑터가 없는 경우, 검증 메시지에 대해 오류 큐를 점검하고 예외 메시지에 대해 예외 파일을 점검하십시오. 추적이 사용 가능하게 되면 추적 메시지를 점검하십시오.
  - 오류 큐의 검증 메시지와 예외 메시지를 보는 경우, 구성 파일의 항목이 검증됩니다.
  - 오류 큐의 검증 메시지를 보지 않을 경우, 예외 파일을 점검하십시오. 추적이 사용 가능하게 되면 추적 메시지를 점검하십시오.

단계 13. 필요하면 구성 파일을 수정하고 다시 검증하십시오.

---

## MQSeries 및 MQSeries Integrator 구성

다음과 같이 커널을 지원하도록 MQSeries 및 MQSeries Integrator와 같은 선택적 소프트웨어를 구성하십시오.

MQSeries에서

- 설치를 확인하는 데 여러 큐가 사용됩니다. 이러한 큐를 테스트 또는 프로덕션 환경에서 사용하는 경우, 설치를 확인하기 위해 큐를 지워야 합니다. 설치 확인에 사용되는 큐에 대해서는 55 페이지의 『확인 절차』를 참조하십시오.
- 설계한 라우팅 설계에 따라 메시지 전송을 지원하도록 큐를 설정하십시오.

- 큐 작성시 MAX\_QUEUE\_DEPTH 환경 변수를 허용되는 최대 큐 용량으로 설정하십시오.

MQSeries Integrator에서 구성 파일에 구성된 큐에 해당하는 규칙(버전 1.1)이나 메시지 플로우(버전 2)에 입력 및 출력 큐를 설정하십시오.

---

## 성능 권장사항

다음과 같은 성능 권장사항이 MQSeries Adapter Kernel에 적용됩니다.

- XML DTD가 분석되는 경우 DTD 파일이 이를 분석하는 프로세스와 같은 디렉토리에 있는지 확인하십시오. 같은 디렉토리에 있으면 프로세스가 DTD를 찾을 필요가 없습니다.
- 큰 메시지가 전송되고 수신되는 경우 메시지 유형 RFH2를 사용하면 메시지 유형 XML을 사용할 때보다 성능이 향상됩니다.

성능 향상을 위한 일반적인 권장사항은 MQSeries 문서를 참조하십시오.

---

## 커널 시작

커널을 시작하려면 다음 항목을 시작하십시오.

- 각 대상 응용프로그램의 어댑터 디먼
- 추적 서버(선택적)

소스 어댑터가 소스 응용프로그램 프로세스시 실행 중인 경우, 소스 어댑터는 소스 응용프로그램으로 자동 시작됩니다. 소스 어댑터를 시작하는 데 추가 단계는 필요가 없습니다. 소스 어댑터를 포함하는 디먼 또는 서버를 시작해야 합니다. 사용자가 직접 소스 어댑터를 시작하지 않습니다.

다음 단계를 수행하여 각 어댑터 디먼과 추적 서버를 시작하십시오.

주: 해당 프로시저에서 사용하기 편리하도록 여러 스크립트가 제공됩니다. 원하면 스크립트를 복사한 다음 사본을 편집하여 사용자 소유 버전을 작성하십시오. OS/400을 사용 중이라면 스크립트의 UNIX 버전이 **qsh** 세션에서 실행될 수 있습니다. 제어 언어(CL) 프롬프트에서 **QSH** 시작(**STRQSH**) 명령을 입력하여 **qsh** 세션을 시작할 수 있습니다.

- 단계 1. MQSeries 및 MQSeries Integrator와 같은 다른 메시징 소프트웨어 및 선택적 소프트웨어를 시작하십시오.
- 단계 2. 사이트에 필요한 기타 관련 소프트웨어를 시작하십시오. 예를 들어 큐에서 추적 메시지를 읽는 커널 외부의 응용프로그램 등이 있습니다.
- 단계 3. 명령 프롬프트를 여십시오. 각 어댑터 디먼에 대해 다음 명령을 입력하십시오.

- Windows 시스템에서:

```
aqmstrad.bat -a application_identifier [-bc body_category
-bt body_type] [-noretry]
```

- UNIX 및 OS/400에서:

```
aqmstrad.sh -a application_identifier [-bc body_category
-bt body_type] [-noretry]
```

설명:

**-a *application\_identifier***

어댑터 디먼이 작동하는 목적지 논리 ID를 식별합니다.

**-bc *body\_category***

어댑터 디먼 작업 프로그램이 통신 모드 및 메시지 수신에 대한 관련 정보를 판별하는 데 사용하는 본문 범주를 지정합니다. 값을 제공하지 않으면 어댑터 디먼이 DEFAULT 값을 사용합니다.

**-bt *body\_type***

어댑터 디먼 작업 프로그램이 통신 모드 및 메시지 수신에 대한 관련 정보를 판별하는 데 사용하는 본문 유형을 지정합니다. 값을 제공하지 않으면 어댑터 디먼이 DEFAULT 값을 사용합니다.

**-noretry**

추가 메시지가 없는 경우 작업 프로그램이 자동으로 정지하도록 지정합니다. **-noretry**를 지정하지 않으면, 작업 프로그램이 계속해서 메시지 큐를 폴링하고 어댑터 디먼은 수동으로 정지해야 합니다.

주: Java 시동 매개변수를 수정해야 하는 경우, aqmstrad.bat (Windows systems) 또는 aqmstrad.sh(UNIX 및 OS/400) 파일을 편집하십시오. 세부사항은 파일 내의 주석을 참조하십시오.

단계 4. 각 추적 서버의 경우 다음 명령을 입력하십시오.

- Windows 시스템에서:

```
aqmstrtd.bat -how -a source_application_identifier
```

- UNIX 및 OS/400에서:

```
aqmstrtd.sh -how -a source_application_identifier
```

설명:

**-how**

추적 메시지가 수신되는 방법을 나타냅니다. 다음은 가능한 값입니다.

- socket
- ena, 즉 원시 어댑터

**-a source\_application\_identifier**

소스 응용프로그램 ID. 값을 제공하지 않으면 구성 파일의 디폴트 TraceServer가 사용됩니다.

추적 서버에 대해서는 *Problem Determination Guide*를 참조하십시오.

단계 5. 어댑터 디먼이나 추적 서버가 시작된 후 프로세스 창은 사용자가 어댑터 디먼을 정지할 때까지 열려 있습니다. 프로세스 창은 예외를 표시할 수 있습니다. 106 페이지의 『예외 메시지』를 참조하십시오.

## 커널 정지

커널을 정지하려면 각각의 어댑터 디먼과 추적 서버를 정지하십시오. 정지하는 방법에는 여러가지가 있습니다.

- 어댑터 디먼을 시작할 때 매개변수 **-noretry**를 설정하십시오. 102 페이지의 『커널 시작』을 참조하십시오.
- 어댑터 디먼 또는 추적 서버가 시작된 명령 프롬프트(Windows 시스템) 또는 터미널(UNIX)로 가서 **Ctrl-C**를 누르십시오. 각 어댑터 디먼이나 추적 서버에 대해 이 단계를 수행하십시오.



- Windows 시스템에서 작업 관리자를 사용하여 프로세스를 종료할 수 있습니다.
- UNIX에서 **ps** 명령을 사용하여 프로세스의 수를 알아낸 다음 **kill** 명령을 사용하여 프로세스를 종료하십시오.

---

## 커널 유지보수

커널 유지보수 계획을 설정하십시오. 다음 항목은 정기적으로 백업하는 것이 좋습니다.

- 다음 파일에 지정된 구성
  - aqmconfig.xml
  - aqmsetup
- 빌드한 어댑터와 관련 파일

커널의 처리를 지원하기 위해 커널이 사용하는 추적 및 기타 파일의 내용을 백업하거나 정기적으로 삭제할 필요는 없습니다. 원하면 이러한 파일을 백업하십시오. 추적 메시지가 여러 개의 파일 대신 한 개의 파일로 라우트되는 경우 하나의 추적 파일이 매우 커질 수 있습니다. 추적 레벨이 세부사항 상위 레벨(예를 들어, 모든 추적 메시지 또는 정보 메시지)을 캡처하도록 설정된 경우, 추적 파일의 정기적인 삭제를 고려해 보십시오.

---

## 문제점 진단

문제점 진단에 예외 메시지, 추적 메시지 및 MQSeries 오류 큐를 사용할 수 있습니다. MQSeries Adapter Kernel은 예외 메시지를 생성하고, 추적이 사용 가능하게 되면 추적 메시지도 생성합니다. MQSeries Adapter Kernel 환경에서의 문제점을 진단하는 방법에 대해서는 *Problem Determination Guide*를 참조하십시오.

예외 메시지와 추적 메시지를 이해하려면 커널 작동 방법을 이해해야 합니다. 커널은 오류 큐를 사용하여 일부 오류를 처리합니다. 10 페이지의 『커널 작업 방법』을 참조하십시오.

고유한 메시지 ID와 고유한 트랜잭션 ID를 결합하여 예외 메시지 및 추적 메시지의 원인이 되는 메시지를 식별할 수 있습니다.

오류 큐와 커널 둘 다에서 같은 메시지를 식별할 수 있게 해주는 ID는 없습니다. 그러나 수동으로 오류 큐의 메시지와 해당 예외 메시지, 추적 메시지 또는 둘 다를 관련시킬 수 있습니다. 다음 중 하나 이상을 비교할 수 있습니다.

- 시간 소인
- 소스 논리 ID 큐
- 목적지 논리 ID 큐
- 본문 범주
- 본문 유형
- 고유 메시지 ID
- 고유 트랜잭션 ID

일치할 경우 오류 큐의 메시지를 해당 예외 메시지 또는 추적 메시지와 관련시켰을 수 있습니다.

## 버전 번호

bin 디렉토리의 aqmversion.bat(Windows 시스템) 또는 aqmversion.sh(UNIX 및 OS/400)를 실행하십시오.

---

## 예외 메시지

커널이 다음 유형의 예외 메시지를 생성합니다.

- 커널의 소스측에 있는 원시 어댑터가 예외를 소스 어댑터로 전달합니다. 소스 어댑터가 이러한 예외를 처리하는 방법에 대해서는 MQSeries Adapter Builder 문서를 참조하십시오.
- 커널의 대상측에 있는 원시 어댑터가 예외를 원시 어댑터를 관리하는 작업 프로그램으로 전달합니다.
- 작업 프로그램이 예외를 작업 프로그램과 같은 디렉토리에 있는 EpicSystemExceptionFile nnnnnnnn.log 파일에 기록합니다.
- 어댑터 디먼이 예외 메시지를 어댑터 디먼과 같은 디렉토리에 있는 예외 파일 EpicSystemExceptionFilennnnnnn.log에 기록합니다. 어댑터 디먼과 작업 프로그램이 같은 디렉토리에 있으므로 둘 다 같은 예외 파일에 기록합니다. 어

댁터 디먼은 예외 메시지를 콘솔에도 기록합니다(즉, 창에서 시작된 경우 시작하는 데 사용된 터미널이나 명령 프롬프트 창).

커널의 추적 예외 메시지는 MQSeries 예외 메시지와 다릅니다. 다음은 커널로부터의 예외 메시지의 예입니다.

```
2000.10.26 19:38:20.929 com.ibm.epic.adapters.eak.nativeadapter.LMSMQ
Thread Name=main receiveRequest(ENAService) ePIC TEST2
TYPE_ERROR_EXC AQM5004: Received exception <com.ibm.epic.adapters.eak.common.
AdapterException> Message information: <AQM0114: com.ibm.epic.adapters.eak.
nativeadapter.MQNMRFH2Formatter::convertMessage(MQMessage): Expecting a message
with an MQHRF2 format and received a message with format <MQSTR >.>
for <unmarshall Message(>> having invalid data <(null)>
```

예외 메시지의 값은 잠재적으로 다음 항목을 포함한 메시지의 특성에 따라 달라집니다.

- 시간 소인
- 소스 논리 ID
- 목적지 논리 ID
- 본문 범주
- 본문 유형
- 고유 메시지 ID
- 고유 트랜잭션 ID
- 예외 정보

설치 확인 중 발견할 수 있는 일반적인 문제점과 가능한 응답에 대해서는 56 페이지의 『일반적인 확인 문제점』을 참조하십시오.

---

## 추적 메시지

추적 메시지를 생성하도록 커널을 구성할 수 있습니다. 추적에 대해서는 *Problem Determination Guide*를 참조하십시오.

### MQSeries 큐 작성

배치 파일이나 셸 스크립트를 사용하여 MQSeries 큐의 작성을 자동화할 수 있습니다. 응용프로그램 이름을 매개변수로 사용하여 `aqmcreateq.bat`(Windows 시스템) 또는 `aqmcreateq.sh`(UNIX 및 OS/400)를 실행하십시오. 이러한 파일은 각 응용프로그램에 대해 다음과 같은 큐를 작성합니다.

- 리시브 큐 `application_nameAIQ`.
- 오류 큐 `application_nameAEQ`.
- 리플라이 큐 `application_nameRPL`.

---

## 제5장 MQSeries Adapter Kernel API 사용

커널은 메시지 송신 및 수신, XML 작성 및 분석, 커널 구성 관리와 같은 기능에 사용되는 API를 포함합니다. 이러한 API는 MQSeries Adapter Builder를 사용하여 작성된 어댑터에 의해 사용됩니다. MQSeries Adapter Kernel 정보 센터는 Javadoc HTML 형식의 관련 온라인 API 문서를 포함합니다.

커널은 MQSeries Adapter Builder를 사용하여 빌드된 어댑터와 함께 사용하도록 되어 있습니다. 커널은 사용자 정의 코드만의 커널 API 호출에 의해 사용되도록 설계되지는 않습니다. 온라인 API 문서는 커널의 기능 방법을 이해하고 진단 하는데 도움을 주기 위해서만 제공됩니다.

커널 온라인 API 문서는 documentation 디렉토리에 있습니다.



---

## 제6장 추가 정보 확보

MQSeries Adapter Offering을 사용할 때 유용한 여러 정보 소스가 있습니다. MQSeries Adapter Kernel에 대한 추가 정보는 제품과 함께 제공된 MQSeries Adapter Kernel 정보 센터에서 사용 가능한 *Problem Determination Guide* 문서를 참조하십시오. *Problem Determination Guide*에서는 커널 사용시 발생할 수 있는 특정 문제점을 해결하는 정보를 제공합니다. MQSeries Adapter Builder에 대해서는 해당 제품의 정보 센터와 온라인 도움말 시스템을 참조하십시오.

---

### 인터넷에서 사용 가능

MQSeries 제품군 웹 사이트는 [www.ibm.com/software/ts/mqseries/](http://www.ibm.com/software/ts/mqseries/)에 있습니다. 이 웹 사이트에서 링크를 따라 다음을 수행할 수 있습니다.

- MQSeries Adapter Offering을 비롯한 MQSeries 제품군에 대한 최신 정보를 얻을 수 있습니다.
- 이 책의 최신 개정판을 포함하여 HTML 및 PDF 형식의 MQSeries 서적에 액세스할 수 있습니다. MQSeries 라이브러리 페이지로의 직접 링크는 [www.ibm.com/software/ts/mqseries/library/manualsa/](http://www.ibm.com/software/ts/mqseries/library/manualsa/)입니다.
- MQSeries 지원팩을 다운로드할 수 있습니다.

OS/400에서의 MQSeries 사용에 대해서는 [www.ibm.com/servers/eserver/series/library/](http://www.ibm.com/servers/eserver/series/library/)의 OS/400 라이브러리를 참조하십시오. 또한 MQSeries 라이브러리 웹 사이트([www.ibm.com/software/ts/mqseries/library/manualsa/](http://www.ibm.com/software/ts/mqseries/library/manualsa/))에서 사용 가능한 OS/400 관련 서적도 참조하십시오.

---

### 참조서

다음 참조 자료는 이 책에서 다룬 주제를 설명합니다.

- Open Applications Group 웹 사이트([www.openapplications.org/](http://www.openapplications.org/))

- XML(Extensible Markup Language) 1.0 W3C Recommendation([www.w3.org/TR/1998/Rec-xml-19980210](http://www.w3.org/TR/1998/Rec-xml-19980210))

이들은 IBM 웹 사이트가 아닙니다.



---

## 부록A. 통신 모드

이 부록에서는 MQSeries Adapter Kernel이 지원하는 통신 모드에 대한 정보 및 통신 모드 지원에 사용되는 Java 클래스에 대한 정보를 제공합니다. 일부 통신 모드는 디폴트 형식기에서 편의 모드로 제공됩니다. 편의 모드로 사용되는 디폴트 형식기에 대해서는 115 페이지의 표10을 참조하십시오.

다음과 같은 통신 모드가 지원됩니다.

- MQPP**            커널이 MQSeries 기본 서비스를 사용하여 메시지를 전송합니다. 이것은 편의 모드입니다.
- MQRFH1**        커널이 MQSeries를 사용하여 메시지를 전송하고 MQSeries Integrator 버전 1.1을 사용하여 메시지를 중개합니다. 이것은 편의 모드입니다.
- MQRFH2**        커널이 MQSeries를 사용하여 메시지를 전송하고 MQSeries Integrator 버전 2를 사용하여 메시지를 중개합니다. 이것은 편의 모드입니다.
- MQBD**            커널이 MQSeries 기본 서비스를 사용하여 메시지를 전송하지만 본문 데이터만을 송신하고 수신합니다. 이것은 편의 모드입니다. 다음과 같은 특성은 이 모드에 고유합니다.
- 메시지 헤더 값이 아닌 본문 데이터만을 송신할 수 있습니다.
  - 본문 데이터만을 포함하는 메시지를 수신할 수 있습니다. 수신된 메시지에 대해 다음과 같은 디폴트 메시지 헤더 값을 사용합니다.
    - SourceLogicalApplicationID -- 수신 메소드 호출에 사용되는 ENAService 오브젝트의 값.
    - BodyCategory -- 수신 메소드 호출에 사용되는 ENAService 오브젝트의 값.
    - BodyType -- 수신 메소드 호출에 사용되는 ENAService 오브젝트의 값.

- Acknowledgment -- 수신된 MQMessage가 MQSeries REQUEST인 경우 Acknowledgment는 1로 설정됩니다.
  - BodyData -- MQseries에서 수신된 메시지 데이터.
- 다른 모든 헤더 값은 정상 디폴트를 사용합니다.

- MQ**            커널이 MQSeries 기본 서비스를 사용하여 메시지를 전송합니다.
- JMS**            커널은 JMS(Java Message Service)를 사용하여 메시지를 전송합니다. MQSeries Adapter Kernel과 함께 JMS 오브젝트를 사용하는 경우에 대해서는 116 페이지의 『JMS 오브젝트 저장영역 사용』을 참조하십시오.
- FILE**            커널이 메시지를 파일에 놓고 파일에서 가져옵니다. 이 모드는 진단 목적으로만 제공됩니다.

표9에서는 통신 모드 및 통신 모드를 지원하는 Java 클래스를 나열합니다. 모든 Java 클래스는 Java 클래스 com.ibm.epic.adapters.eak.nativeadapter에서 옵니다. LMS(논리 메시지 전달 서비스)를 지원하는 Java 클래스는 통신 모드로 지정할 수 있습니다. 이 경우 클래스 자체는 통신을 지원하는 데 사용됩니다.

표9. 통신 모드 및 지원 Java 클래스

통신 모드	Java 클래스	참고
MQPP	LMSMQBindingMQPP	MQSeries 설치 필수
MQRFH1	LMSMQBindingMQRFH1	MQSeries 설치 필수
MQRFH2	LMSMQBindingMQRFH2	MQSeries 설치 필수
MQBD	LMSMQMQBD	MQSeries 설치 필수
MQ	LMSMQBinding	MQSeries 설치 필수
JMS	LMSJMS	JMS 설치 필수
FILE	LMSFile	없음.

115 페이지의 표10에서는 통신 모드 및 관련 형식기 인터페이스를 나열합니다. 115 페이지의 표11은 형식기 인터페이스, 형식기 클래스 이름 및 그 사용을 상호 참조합니다. 모든 형식기는 Java 패키지 com.ibm.epic.adapters.eak.nativeadapter에서 옵니다. 통신 모드에서는 아무 형식기 클래스나 지정할 수 있음에 유의하십시오. 이 경우 지정된 형식기 클래스를 형식기로 사용합니다.

표 10. 통신 모드 및 형식기 인터페이스

통신 모드	형식기 인터페이스	디폴트 형식기
MQPP	MQFormatterInterface	MQNMXLFormatter
MQRFH1	MQFormatterInterface	MQNMRFH1Formatter
MQRFH2	MQFormatterInterface	MQNMRFH2Formatter
MQBD	MQFormatterInterface	MQNMBDFormatter
MQ	MQFormatterInterface	MQNMXLFormatter
JMS	JMSFormatterInterface	JMSNMRFH2Formatter
FILE	StringFormatterInterface	NMXLFormatter

표 11. 형식기 인터페이스, 형식기 클래스 이름 및 목적

형식기 인터페이스	형식기 클래스 이름	목적
MQFormatterInterface	MQNMXLFormatter	XML로 EpicMessage
	MQNMRFH1Formatter	RFH1으로 EpicMessage
	MQNMRFH2Formatter	RFH2로 EpicMessage
	MQNMBDFormatter	본문 데이터만
JMSFormatterInterface	JMSNMXLFormatter	XML로 EpicMessage
	JMSNMRFH2Formatter	RFH2로 EpicMessage
	JMSNMBDFormatter	본문 데이터만
StringFormatterInterface	NMXLFormatter	XML로 EpicMessage

표12는 지원되는 LMS 클래스와 그 트랜잭션 지원 정도를 나열합니다. MQSeries Adapter Kernel과 함께 트랜잭션 사용에 대해서는 30 페이지의 『트랜잭션 기능』을 참조하십시오.

표 12. LMS 클래스 및 트랜잭션 지원

LMS 클래스	트랜잭션 지원
LMSMQBindingMQPP	단일 단계
LMSMQBindingMQRFH1	단일 단계
LMSMQBindingMQRFH2	단일 단계
LMSMQMQBD	단일 단계
LMSMQBinding	단일 단계
LMSJMS	단일 단계
LMSFILE	지원되지 않음

---

## JMS 오브젝트 저장영역 사용

JMS 오브젝트 이름은 MQSeries JMS SupportPac의 일부로 제공되는 JNDI의 FSContext 파일 구현을 사용하여 저장됩니다. 커널이 FSContext에 사용하는 컨텍스트(디렉토리 구조)는 디렉토리 이름에 대한 관련 값을 갖는 구별되는 속성을 사용하여 LDAP 계층 구조를 따릅니다. 예를 들어 LDAP 디렉토리 o=ePIC, o=ePICApplications, epicappid=TEST1의 경우, 디렉토리 구조는 o-ePIC/o-ePICApplications/epicappid-TEST1입니다.

컨텍스트와 오브젝트를 작성하려면 JMS 설치시 제공되는 JMS 관리 도구를 사용하십시오. 기본 단계는 컨텍스트를 정의한 다음 컨텍스트를 변경하는 것입니다. 컨텍스트 변경은 사용자를 컨텍스트로 이동시킵니다. JMS 오브젝트를 적절한 위치에 작성하십시오. 다음은 컨텍스트 구조 및 JMS 오브젝트를 작성하는 예제 명령입니다. 이 예에서 응용프로그램 ID는 TEST1입니다.

```
#
# aqmjmscreatesample.scp 1.00 09Mar01
# Used for MQSeries Adapter Kernel
# Sample AQM JMS Configuration.
#
# Copyright (c) 2001 International Business Machines. All Rights Reserved.
#
# This configuration file is as an example only.
#
# IBM MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THIS
# SAMPLE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
# IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
# PURPOSE, OR NON-INFRINGEMENT.
#
# CopyrightVersion 1.0
#
#
# This is a script to use with the JMS administration (JMSAdmin) tool
# which comes with MQSeries Support pac MA88.
# This tool requires the JMSAdmin.config to be set to use either
# FSCONTEXT (file) or LDAP. This script is setup to work with
# FSCONTEXT, but will work with LDAP with the following changes:
# - Change the "-" signs to "=" . Example: define ctx(o=ePIC)
# becomes define ctx(o=ePIC)
# - In LDAP the contexts have to already be defined using the
# LDAP administration tool. For example you do not need
# to "define ctx(o=ePIC) but only change into it with the
# "change ctx(o=ePIC)" command.
# - There are some notes in the following script which highlight
# differences when using LDAP.
#
#
# Example usage: MQSeries root\java\bin\jmsadmin.bat < aqmjmscreatesample.scp
#
# Some helpful commands:
# "display ctx" will display context you are
# currently in.
```

```

# "=UP" means return to the parent context. Example: change ctx(=UP)
# "=INIT" means return to root context. In this example one directory level
# above o-ePIC. Example: change ctx(=INIT)
# "define xxx" is for creating either a context or object.
# "change xxx" is for changing/moving into the context.
#
# Always required.
define ctx(o-ePIC)
change ctx(o-ePIC)
# Always required.
define ctx(o-ePICApplications)
change ctx(o-ePICApplications)
# Application id is TEST1, requires a context.
define ctx(epicappid-TEST1)
change ctx(epicappid-TEST1)
# Always required.
define ctx(cn-epicadapterrouting)
change ctx(cn-epicadapterrouting)
# This will hold the JMS QueueConnectionFactory object.
# Note: These two steps are not required for LDAP.
define ctx(cn-QCFTEST1)
change ctx(cn-QCFTEST1)
# Create the JMS QueueConnectionFactory object whose name is QCFTEST1
# Using MQSeries in server (bindings) mode.
define qcf(QCFTEST1) qmgr(yourQManagerName) tran(BIND)
change ctx(=UP)
# BodyCategory is DEFAULT
define ctx(epicbodycategory-DEFAULT)
change ctx(epicbodycategory-DEFAULT)
# BodyType is DEFAULT
define ctx(epicbodytype-DEFAULT)
change ctx(epicbodytype-DEFAULT)
# This will hold the JMS Queue object whose name is TEST1AIQ.
# Note: These two steps are not required for LDAP.
define ctx(cn-TEST1AIQ)
change ctx(cn-TEST1AIQ)
# Create the JMS Queue object whose name is TEST1AIQ
# q(JMS Q Object Name) queue(MQSeries Queue name)
define q(TEST1AIQ) queue(TEST1AIQ)
# Can move up and define other contexts and JMS objects.
# Quit the administration tool.
end

```



---

## 부록B. 검증 구성

MQSeries, MQSeries Adapter Offering 및 MQSeries Integrator의 많은 가능한 구성과 조합이 있습니다. 이러한 각각의 MQSeries 제품 계열 구성원은 기능과 구성면에서 풍부합니다. MQSeries, MQSeries Adapter Offering 및 MQSeries Integrator에서 기능을 조합할 수도 있습니다. MQSeries 제품군 중 한 구성원의 일부 기능은 부분적으로 다른 계열 구성원이 제공하는 기능과 겹칠 수 있습니다. MQSeries, MQSeries Adapter Offering 및 MQSeries Integrator의 여러 다른 메시지 라우팅과 전달 기능을 사용하고 조합하는 방법을 알아내야 합니다.

다음과 같은 MQSeries, MQSeries Adapter Offering 및 MQSeries Integrator의 구성은 프로덕션시 검증되었습니다. 검증된 최신 구성에 대해서는 MQSeries 웹 사이트를 참조하십시오.

### **MQSeries Adapter Kernel:**

- 요청된 수신확인 메시지 송신 및 요청된 수신확인 없이 메시지 송신.
- MQSeries 또는 JMS 통신 모드 사용. 유효한 통신 모드에 대한 자세한 정보는 113 페이지의 『부록A. 통신 모드』를 참조하십시오.
- 메시지 라우팅 및 전달:
  - 한 소스 어댑터에서 한 대상 어댑터로 메시지 송신
  - 한 소스 어댑터에서 여러 대상 어댑터로 메시지 송신
  - 멀티스레드 메시지 전달, 즉 여러 작업 프로그램
  - 메시지에서 NONE으로 설정된 목적지 논리 ID. 커널 구성 파일을 사용하여 본문 범주, 본문 유형 및 소스 논리 ID를 기초로 목적지 논리 ID를 판별합니다.
  - 전달 PUSH 모델
  - 추적을 사용 가능으로 설정

주: 121 페이지의 『부록C. 메시지 헤더』를 참조하십시오. 커널이 데이터를 입력하고 처리하는 MQSeries Adapter Kernel 메시지 헤더를 포함합니다.

- 35 페이지의 『하드웨어』 및 36 페이지의 『소프트웨어』에 표시된 필수 조건.
- 구성 정의에 LDAP가 아닌 구성 파일 사용.

### **MQSeries:**

- MQSeries 클러스터 사용 안함.

주: 121 페이지의 『부록C. 메시지 헤더』를 참조하십시오. 커널이 데이터를 입력하고 처리하는 MQSeries Adapter Kernel 메시지 헤더 필드를 포함합니다.

### **MQSeries Integrator:**

- MQSeries Adapter Kernel 및 MQSeries는 메시지를 MQSeries Integrator로 라우트하고 전달할 수 있습니다. 이러한 메시지를 중개하는 기능을 알아내려면 MQSeries Integrator 정보를 참조하십시오.
- MQSeries 및 MQSeries Integrator 버전 2를 통해 커널의 소스측에서 메시지를 송신하여 직접 커널의 대상측으로 라우팅. MQSeries Integrator를 통해 정적으로 라우트하도록 메시지 플로우가 구성됩니다. 플로우의 MQInput 노드에 도달하는 모든 메시지는 직접 특정 MQOutput 큐로 라우트됩니다.

주: 121 페이지의 『부록C. 메시지 헤더』를 참조하십시오. 커널이 데이터를 입력하고 처리하는 MQSeries Adapter Kernel 메시지 헤더 필드를 포함합니다.



## 부록C. 메시지 헤더

MQSeries Adapter Offering은 여러 메시지 헤더를 사용합니다. 어떤 헤더가 어떤 상황에서 사용되는지에 대해서는 13 페이지의 『메시지 및 메시지 형식』을 참조하십시오.

이 부록에서는 메시지 헤더 필드를 나열하고 설명합니다.

### MQSeries Adapter Kernel 메시지 설명자 헤더

MQSeries Adapter Kernel이 사용하는 헤더 값. 이 값은 메시지 홀더 오브젝트에 위치합니다. **응답에 전달?** 열에서는 소스 응용프로그램이 응답을 요청할 때 해당 값이 응답 메시지의 소스 응용프로그램으로 다시 전달되는지 여부를 나열합니다. 일부 값은 WebSphere Business Integrator와 함께만 사용됩니다.

표 13. MQSeries Adapter Kernel 헤더

헤더 이름	응답에 전달?	의미 또는 사용법
UniqueID	아니오	각 메시지에 대한 고유 ID.
TransactionID	예	각 메시지와 해당 응답에 의해 공유되는 트랜잭션 ID. PublicProcessID 또는 DI(Data Interchange) ApplicationID에 해당됩니다.
MessageType	아니오	게이트웨이 및 log/trace/exception 메시지에 사용됩니다.
SourceLogicalID	아니오	소스 응용프로그램의 논리 ID. DI, PAM(Partner Agreement Manager) 및 BFM(Business Flow Manager)의 예약 이름에 해당됩니다.
DestinationLogicalID	아니오	대상 응용프로그램의 논리 ID. DI와 PAM의 경우 디폴트 값은 none이지만 다른 값으로 대체할 수 있습니다.

표 13. MQSeries Adapter Kernel 헤더 (계속)

헤더 이름	응답에 전달?	의미 또는 사용법
RespondToLogicalID	예	응답 메시지가 송신될 논리 ID. DI의 경우 DestinationLogicalID로 복사되고 PAM의 경우 SourceLogicalID로 복사됩니다.
CorrelationID	아니오	사용 예약.
GroupStatus	아니오	사용 예약.
ProcessingCategory	아니오	PAM 퍼블릭 프로세스 ID 또는 DI 명령 프로세스 ID에 해당됩니다.
QosPolicy	아니오	사용 예약.
DeliveryCategory	아니오	DI RequestorProfileID에 해당됩니다.
AckRequested	아니오	소스 응용프로그램의 응답 메시지 요청 여부를 판별합니다.
PublicationTopic	아니오	사용 예약.
SessionID	아니오	DI BatchID에 해당됩니다.
EncryptionStatus	아니오	본문 암호화 유형 및 서명을 판별합니다.
TimeStampCreated	아니오	메시지가 작성된 시간 및 날짜.
TimeStampExpired	아니오	메시지가 더 이상 의미없게 된 시간 및 날짜. -1 값은 만기가 설정되지 않았음을 나타냅니다.
Size	아니오	사용 예약.
BodyType	아니오	메시지의 특정 목적을 나타냅니다.
BodyCategory	아니오	메시지의 응용프로그램 유형을 나타냅니다.
BodySecondaryType	아니오	사용 예약.
UserArea	아니오	사용자 데이터용 일반 영역.
RelatedSubjectID	아니오	프로세스간 상호관계에 사용됩니다.
ExternalID	아니오	응용프로그램 환경 외부의 현재 소유자(예를 들면, 사용자나 거래 파트너)의 ID.

표 13. MQSeries Adapter Kernel 헤더 (계속)

헤더 이름	응답에 전달?	의미 또는 사용법
InternalID	아니오	응용프로그램 환경 내부의 현재 소유자(예를 들면, 사용자나 거래 파트너)의 ID.
BodySignature	아니오	사용 예약.
TransportCorrelationID	예	사용 예약.

## MQSeries 메시지 설명자 헤더

필드 내용은 MQSeries로 판별합니다. MQSeries Adapter Offering은 메시지를 메시지 제어 값에 의해 판별된 큐에 넣습니다. 세부사항은 17 페이지의 『메시지 제어 값』을 참조하십시오.

표 14. MQSeries 헤더

섹션 또는 필드	의미 또는 사용
Revision	고정
UniqueID	각 메시지가 고유한 ID를 가집니다.
TransactionID	메시지와 응답이 같은 트랜잭션 ID를 가집니다.
MessageType	사용 예약.
SourceLogicalID	소스 응용프로그램의 논리 ID.
DestinationLogicalID	대상 응용프로그램의 논리 ID.
RespondToLogicalID	응답 메시지를 송신할 논리 ID.
CorrelationID	사용 예약.
GroupStatus	사용 예약.
ProcessingCategory	사용 예약.
QosPolicy	사용 예약.
DeliveryCategory	사용 예약.
AckRequested	소스 응용프로그램이 응답을 요청하는지 여부를 판별합니다.
PublicationTopic	사용 예약.
SessionID	사용 예약.
EncryptionStatus	사용 예약.
TimeStampCreated	메시지가 작성된 시간 및 날짜.
TimeStampExpired	메시지가 더 이상 의미없게 된 시간 및 날짜.
Size	사용 예약.

표 14. MQSeries 헤더 (계속)

BodyCategory	메시지 응용프로그램 유형, 예를 들어 OAG 또는 RosettaNet를 나타냅니다.
BodyType	메시지의 특정 목적, 예를 들어 판매 주문 추가 또는 재고 동기화를 나타냅니다.
BodySecondaryType	예약되었습니다.
UserArea	사용자 데이터용 일반 영역.
BodyData	메시지 본문 데이터.

## MQSeries Integrator가 없는 MQSeries

커널 헤더 값과 본문 데이터는 XML 문서에 놓입니다. 다음은 XML 문서를 설명하는 DTD의 예입니다.

```
<!ELEMENT EPICHEADER (HEADER, EPICBODY,USERAREA*)>
<!ELEMENT HEADER (#PCDATA)>
<!ATTLIST HEADER Revision CDATA #FIXED "001">
<!ATTLIST HEADER UniqueID CDATA #REQUIRED>
<!ATTLIST HEADER TransactionID CDATA #REQUIRED>
<!ATTLIST HEADER MessageType CDATA #REQUIRED>
<!ATTLIST HEADER SourceLogicalID CDATA #REQUIRED>
<!ATTLIST HEADER DestinationLogicalID CDATA #REQUIRED>
<!ATTLIST HEADER RespondToLogicalID CDATA #IMPLIED>
<!ATTLIST HEADER CorrelationID CDATA #IMPLIED>
<!ATTLIST HEADER GroupStatus CDATA #IMPLIED>
<!ATTLIST HEADER ProcessingCategory CDATA #IMPLIED>
<!ATTLIST HEADER QosPolicy CDATA #IMPLIED>
<!ATTLIST HEADER DeliveryCategory CDATA #IMPLIED>
<!ATTLIST HEADER AckRequested CDATA #IMPLIED>
<!ATTLIST HEADER PublicationTopic CDATA #IMPLIED>
<!ATTLIST HEADER SessionID CDATA #IMPLIED>
<!ATTLIST HEADER EncryptionStatus CDATA #IMPLIED>
<!ATTLIST HEADER TimeStampCreated CDATA #REQUIRED>
<!ATTLIST HEADER TimeStampExpired CDATA #REQUIRED>
<!ATTLIST HEADER Size CDATA #IMPLIED>
```

```

<!ELEMENT EPICBODY (#PCDATA)> <!-- The data will be escaped -->
<!ATTLIST EPICBODY Size CDATA #IMPLIED>
<!ATTLIST EPICBODY BodyType CDATA #REQUIRED>
<!ATTLIST EPICBODY BodyCategory CDATA #REQUIRED>
<!ATTLIST EPICBODY BodySecondaryType CDATA #IMPLIED>
<!ELEMENT USERAREA (#PCDATA) >

```

## MQSeries Integrator 버전 1 헤더

MQSeries Integrator 버전 1 헤더 RFH1은 다음 항목으로 구성됩니다.

1. 고정 부분
2. Neon 헤더
3. 커널 헤더와 메시지 본문 데이터를 포함하는 데이터 섹션

표 15. MQSeries Integrator 버전 1 헤더 -- RFH1

섹션 또는 필드	의미 또는 사용
고정 부분	MQSeries Integrator 버전 1.1에 지정된 대로 사용됩니다.
Neon 헤더	Neon 헤더 형식을 따릅니다.
OPT_APP_GRP	SourceLogicalId 값. 커널 헤더에서 가져옵니다.
OPT_MSG_TYPE	BodyCategory+BodyType. 커널 헤더에서 가져옵니다.  예: BodyCategory가 OAG이고 BodyType이 SyncItem 인 경우 이 값은 OAG+SyncItem입니다.
데이터 섹션	커널 헤더 값과 메시지 본문 데이터로 구성됩니다.
커널 헤더	커널 헤더는 태그 <EPICHEADER>header</EPICHEADER> 안에 묶입니다.  커널 헤더 값은 XML 구문 형식입니다. 값을 갖는 속성만 표시됩니다. 실제 데이터는 별개의 행에 없습니다. 값 형식 예: <MessageType>value</MessageType>.
MessageType	사용 예약.
SourceLogicalID	소스 응용프로그램의 논리 ID.
DestinationLogicalID	대상 응용프로그램의 논리 ID.
RespondToLogicalID	응답 메시지를 수신할 논리 ID.
TimeStampCreated	메시지가 작성된 시간 및 날짜.

표 15. MQSeries Integrator 버전 1 헤더 -- RFH1 (계속)

TimeStampExpired	메시지가 더 이상 의미없게 된 시간 및 날짜.
TransactionID	메시지와 응답이 같은 트랜잭션 ID를 가집니다.
UniqueID	각 메시지가 고유한 ID를 가집니다.
AckRequested	소스 응용프로그램이 응답을 요청하는지 여부를 판별합니다.
ProcessingCategory	예약되었습니다.
BodyCategory	메시지 응용프로그램 유형, 예를 들어 OAG 또는 RosettaNet를 나타냅니다.
BodyType	메시지의 특정 목적, 예를 들어 판매 주문 추가 또는 재고 동기화를 나타냅니다.
BodySecondaryType	예약되었습니다.
UserArea	사용자 통합 특정 응용프로그램 데이터.
MsgHeaderVersion	커널 헤더 버전(예약).
CorrelationID	특정 사용자 통합.
GroupStatus	특정 사용자 통합.
QosPolicy	예약되었습니다.
DeliveryCategory	예약되었습니다.
PublicationTopic	예약되었습니다.
SessionID	예약되었습니다.
EncryptionStatus	예약되었습니다.
메시지 본문 데이터	메시지 본문 데이터.

## MQSeries Integrator 버전 2 헤더

MQSeries Integrator 버전 2 헤더 RFH2는 다음 항목으로 구성됩니다.

1. 고정 부분
2. <mcd> 폴더 -- 메시지 내용 설명자
3. <usr> 폴더 -- 응용프로그램(사용자) 정의 등록 정보
4. 커널 헤더와 메시지 본문 데이터를 포함하는 데이터 섹션

표 16. MQSeries Integrator 버전 2 헤더 -- RFH2

섹션 또는 필드	의미 또는 사용
고정 부분	MQSeries Integrator 버전 2에 지정된 대로 사용됩니다.
<mcld>	메시지가 XML인 경우 XML. MQSeries Integrator 버전 2 규칙을 따릅니다.
set	커널에서 사용하지 않습니다.
type	커널에서 사용하지 않습니다.
format	메시지가 XML인 경우 XML. MQSeries Integrator 버전 2 규칙을 따릅니다.
<usr> 폴더 -- 응용프로그램(사용자) 정의 등록 정보	커널 헤더 값들로 구성됩니다.
커널 헤더	값을 갖는 속성만 표시됩니다. 실제 데이터는 별개의 행에 없습니다.
SourceLogicalID	소스 응용프로그램의 논리 ID.
DestinationLogicalID	대상 응용프로그램의 논리 ID.
MessageType	사용 예약.
RespondToLogicalID	응답 메시지를 수신할 논리 ID.
TimeStampCreated	메시지가 작성된 시간 및 날짜.
TimeStampExpired	메시지가 더 이상 의미없게 된 시간 및 날짜.
TransactionID	메시지와 응답이 같은 트랜잭션 ID를 가집니다.
UniqueID	각 메시지가 고유한 ID를 가집니다.
ProcessingCategory	예약되었습니다.
BodyCategory	메시지 응용프로그램 유형, 예를 들어 OAG 또는 RosettaNet를 나타냅니다.
BodyType	메시지의 특정 목적, 예를 들어 판매 주문 추가 또는 재고 동기화를 나타냅니다.
BodySecondaryType	예약되었습니다.
AckRequested	소스 응용프로그램이 응답을 요청하는지 여부를 판별합니다.
UserArea	사용자 통합 특정 응용프로그램 데이터.
MsgHeaderVersion	커널 헤더 버전(예약).
CorrelationID	특정 사용자 통합.
GroupStatus	특정 사용자 통합.
QosPolicy	예약되었습니다.
DeliveryCategory	예약되었습니다.
PublicationTopic	예약되었습니다.

표 16. MQSeries Integrator 버전 2 헤더 -- RFH2 (계속)

SessionID	예약되었습니다.
EncryptionStatus	예약되었습니다.
데이터 섹션	메시지 본문 데이터.



---

## 부록D. 구성 파일 샘플

이 절에서는 이 서적의 작성 시점에서의 현재 aqmconfig.xml 파일 버전을 나열합니다. 134 페이지의 『최소 구성 파일 샘플』에는 이 서적의 작성 시점에서의 현재 aqmconfig.minimum.xml 파일 버전이 나열됩니다. 가장 최신 버전은 커널 설치 samples 디렉토리에 있는 aqmconfig.xml 및 aqmconfig.minimum.xml 파일을 참조하십시오. 여기에 나열된 예는 최신이 아닐 수도 있습니다.

구성 파일 해석 및 편집에 대해서는 74 페이지의 『구성 파일』을 참조하십시오.

여러 응용프로그램 ID가 이 예제 구성 파일에 포함됩니다. 항목 세트가 각 응용프로그램 ID 아래 나열됩니다. 샘플 구성 파일은 다음과 같은 응용프로그램 ID를 포함합니다.

- TEST1
- TEST1Daemon
- TEST2
- TEST3
- TraceClient
- TraceServer

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- aqmconfig.xml 1.01 09Mar01 -->
<!-- Used for MQSeries Adapter Kernel -->
<!-- Sample AQM Configuration. -->
<!-- -->
<!-- Copyright (c) 2001 International Business Machines. All Rights Reserved. -->
<!-- -->
<!-- This configuration file is as an example only. -->
<!-- -->
<!-- IBM MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THIS -->
<!-- SAMPLE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE -->
<!-- IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR -->
<!-- PURPOSE, OR NON-INFRINGEMENT. -->
<!-- -->
<!-- CopyrightVersion 1.0 -->
<!-- -->
<Epic o="ePIC">
  <!-- If getObject is called this indicates the top level directory -->
  <!-- where the JNDI file system context will retrieve objects from. -->
  <!-- This defaults to the current directory if this key is not present. -->
  <!-- All applications share this context root. -->
  <context>file:///epic/configContext</context>
  <!-- Example using a drive letter 'c' -->
  <!-- -->
  <context>file://c:/E/runtimefiles</context>
  <!-- -->
</EpicApplications o="ePICApplications">
```

```

<!-- The following is for sample Test Application ID: TEST1 with a -->
<!-- sample AdapterDaemon named TEST1Daemon -->
<ePICApplication epicappid="TEST1">
<!-- Audit Logging on/off. Requires WebSphere Business Integrator product. -->
<!-- If no entry defaults to false. -->
<epiclogging>>false</epiclogging>
<!-- Tracing on/off. If no entry defaults to false. -->
<epictrace>>false</epictrace>
<!-- Trace levels - Uses the jlog com.ibm.logging.IRecordType constants, -->
<!-- common constants: -->
<!-- 0=TYPE_NONE (No messages), 1=TYPE_INFO, 512=TYPE_ERROR_EXC (Exceptions), -->
<!-- 513=TYPE_INFO | TYPE_ERROR_EXC, -1=TYPE_ALL (All possible messages). -->
<!-- No entry defaults to TYPE_NONE -->
<epictracelevel>-1</epictracelevel>
<!-- Name of the Trace application id. Will be used for -->
<!-- trace configuration information. Defaults to TraceClient -->
<epictraceclientid>TraceClient</epictraceclientid>
<!-- When processing messages into the application. -->
<!-- LogonInfo class name used for connecting to an application. -->
<!-- Will be used by the AdapterDaemon. If no entry will default -->
<!-- to com.ibm.epic.adapters.eak.adapterdaemon.EpicLogonDefault. -->
<epiclogoninfoclassname>com.ibm.epic.adapters.eak.adapterdaemon.EpicLogonDefault
</epiclogoninfoclassname>
<AdapterRouting cn="epicadapterrouting">
<!-- MQSeries Q Manager for this application use, no entry -->
<!-- uses the default Q Manager. A value of DEFAULT means -->
<!-- use the default Q Manager. -->
<epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>
<!-- Use the remote Q Manager for sending messages. Remote queue -->
<!-- definitions are not required. true - use remote Q Manager, -->
<!-- false - do not use remote Q Manager. No entry defaults to false -->
<epicuseremotequeue MANAGERTOSEND>>false</epicuseremotequeue MANAGERTOSEND>
<!-- MQSeries Client hostname for where the MQSeries server -->
<!-- resides for TEST1. Required if using MQSeries Client -->
<!--
<!--
<epicmqppqueuemgrhostname>localhost</epicmqppqueuemgrhostname>
-->
<!-- MQSeries Client port to use for where the MQSeries server -->
<!-- resides for TEST1. No entry defaults to MQSeries default -->
<!--
<epicmqppqueuemgrportnumber>1414</epicmqppqueuemgrportnumber>
-->
<!-- MQSeries Client channel name to use for the MQSeries server, required -->
<!--
<epicmqppqueuemgrchannelname>xyz</epicmqppqueuemgrchannelname>
-->
<!-- JMS example for TEST1. Refers to the JMS Connection factory name. -->
<!-- Requires the attribute describing the object plus the attributes value. -->
<!-- For JMS the attribute is 'cn'. -->
<!--
<epicjmsconnectionfactoryname>cn=QCFTST1</epicjmsconnectionfactoryname>
-->
<ePICBodyCategory epicbodycategory="DEFAULT">
<ePICBodyType epicbodytype="DEFAULT">
<!-- Contains the Command selection criteria when processing -->
<!-- a message into an application. Will be used by the -->
<!-- AdapterDaemon - Command to invoke. -->
<epiccommandclassname>com.ibm.epic.adapters.eak.samples.SampleCAdapterWrapper
</epiccommandclassname>
<!-- Represents the type of command the "epiccommandclassname" -->
<!-- represents. MQAKEAB is the EAB style interface. MQAKEJB -->
<!-- is an EJB Service Session Bean. No entry defaults to MQAKEAB -->
<epiccommandtype>MQAKEAB</epiccommandtype>
<!-- If the "epiccommandtype" is "MQAKEJB" this entry is the -->
<!-- method name to invoke. No entry defaults to "execute". -->
<epiccommandejbmethod>execute</epiccommandejbmethod>
<!-- If the "epiccommandtype" is "MQAKEJB" this entry is the -->
<!-- parameter type for the method specified by "epiccommandejbmethod". -->
<!-- This will be the same datatype returned by the -->
<!-- TerminalDataContainerMapper. No entry defaults -->
<!-- to "com.ibm.mqao.mqak.ejbclient.TDCGenericMapper". -->
<epiccommandejbmethodparmttype>com.ibm.mqao.mqak.ejbclient.TDCGenericMapper
</epiccommandejbmethodparmttype>
<!-- If the "epiccommandtype" is "MQAKEJB" this entry is the -->
<!-- URL where the EJB specified in "epiccommandclassname" -->
<!-- has been deployed in the form "IIOP://hostname:900/". -->
<!-- No entry defaults to "IIOP://". -->
<epiccommandejburl>IIOP://</epiccommandejburl>
<!-- If the "epiccommandtype" is "MQAKEJB" this entry is the -->
<!-- name of the Initial Context Factory used to to lookup the -->
<!-- home name for the EJB specified in "epiccommandclassname". -->

```

```

<!-- No entry defaults to "com.ibm.ejs.ns.jndi.CNInitialContextFactory". -->
<epiccommandejbinitialcontext>com.ibm.ejs.ns.jndi.CNInitialContextFactory
</epiccommandejbinitialcontext>
<!-- If the "epiccommandtype" is "MQAKEJB" this entry is the -->
<!-- name of the Mapper the Worker uses for creating the -->
<!-- "epiccommandejbmethodparmtype" object passed in to the -->
<!-- "epiccommandejbmethod" in to the "epiccommandclassname". -->
<!-- No entry defaults to "com.ibm.mqao.mqak.ejbcclient.TDCGenericMapper". -->
<epiccommandejbmapper>com.ibm.mqao.mqak.ejbcclient.TDCGenericMapper
</epiccommandejbmapper>
<!-- Default destinations to send messages to. -->
<!-- Single destination. -->
<epicdestids>TEST2</epicdestids>
<!-- Multiple destinations. -->
<!--
<epicdestids>
  <Value>TEST2</Value>
  <Value>TEST3</Value>
</epicdestids>
-->
<!-- Receive transport communications mode this application -->
<!-- wants for receiving messages. -->
<!-- For MQSeries normal mode use MQPP. -->
<!-- For MQSeries using an RFH1 header format use MQRFH1, -->
<!-- when using MQSeries Integrator V1 -->
<!-- For MQSeries using an RFH2 header format use MQRFH2, -->
<!-- when using MQSeries Integrator V2 -->
<!-- For file normal mode use FILE. -->
<epicreceivemode>MQPP</epicreceivemode>
<!-- How to format the message for the receive mode. -->
<!-- Entry is the class name of the formatter which -->
<!-- must be for the receive mode -->
<!-- Receive modes MQPP, MQRFH1, MQRFH2, FILE have -->
<!-- default receive modes -->
<epicmessageformatter>com.ibm.epic.adapters.eak.nativeadapter.MQNMDBFormatter
</epicmessageformatter>
<!-- JMS formatter for mode for MQSeries provider implementation -->
<!--
<epicmessageformatter>com.ibm.epic.adapters.eak.nativeadapter.JMSNMRFH2Formatter
</epicmessageformatter>
-->
<!-- Receive Time out in milliseconds ie. 1000 = 1 second, -->
<!-- -1 means never ending. No entry defaults to 0 -->
<!-- milliseconds. Used when receiving messages. -->
<epicreceivetimeout>30000</epicreceivetimeout>
<!-- MQSeries queue for this application to receive messages -->
<!-- from for receive modes MQPP, MQRFH1, MQRFH2 -->
<epicreceivemqppqueue>TEST1AIQ</epicreceivemqppqueue>
<!-- MQSeries queue required by the AdapterWorker when -->
<!-- errors encountered processing a message -->
<!-- for receive modes MQPP, MQRFH1, MQRFH2 -->
<epicerrormqppqueue>TEST1AEQ</epicerrormqppqueue>
<!-- MQSeries reply queue required for synchronous request/replies -->
<!-- for receive modes MQPP, MQRFH1, MQRFH2 -->
<epicreplymqppqueue>TEST1RPL</epicreplymqppqueue>
<!-- JMS receive mode, refers to the JMS queue object name for -->
<!-- this application to receive messages from. -->
<!-- Requires the attribute describing the object plus the attribute's value. -->
<!-- For JMS the attribute is 'cn'. -->
<epicjmsreceivequeueenamel>cn=TEST1AIQ</epicjmsreceivequeueenamel>
<!-- JMS receive mode, refers to the JMS queue object name for -->
<!-- errors required by the AdapterWorker when errors -->
<!-- encountered processing a message. -->
<!-- Requires the attribute describing the object plus the attribute's value. -->
<!-- For JMS the attribute is 'cn'. -->
<epicjmserrorqueueenamel>cn=TEST1AEQ</epicjmserrorqueueenamel>
<!-- JMS receive mode, refers to the JMS queue object name for -->
<!-- the reply queue, required for synchronous request/replies -->
<!-- Requires the attribute describing the object plus the attribute's value. -->
<!-- For JMS the attribute is 'cn'. -->
<epicjmsreplyqueueenamel>cn=TEST1RPL</epicjmsreplyqueueenamel>
<!-- In FILE receive mode, directory for this application to receive messages from -->
<epicreceivefiledir>./TEST1AID</epicreceivefiledir>
<!-- In FILE receive mode, interim directory for this application to -->
<!-- hold received messages until committed. -->
<epiccommitfiledir>./TEST1ACD</epiccommitfiledir>
<!-- In FILE receive mode, directory for this application to put error messages -->
<!-- File receive mode, directory required by the AdapterWorker when -->
<!-- errors encountered processing a message -->
<epicerrorfiledir>./TEST1AED</epicerrorfiledir>
</ePICBodyType>

```

```

    </ePICBodyCategory>
  </AdapterRouting>
</ePICApplication>
<!-- The following is for sample AdapterDaemon 'TEST1Daemon' -->
<!-- for the 'TEST1' application -->
<ePICApplication epicappid="TEST1Daemon">
  <epictrace>>false</epictrace>
  <epictracelevel>-1</epictracelevel>
  <ePICAdapterDaemonExtensions cn="epicappextensions">
    <!-- Dependency appid, if no entry then will default -->
    <!-- to the application id of the daemon. -->
    <epicdepappid>TEST1</epicdepappid>
    <!-- Minimum number of workers the AdapterDaemon will start. -->
    <!-- No entry defaults to 1. -->
    <epicminworkers>1</epicminworkers>
  </ePICAdapterDaemonExtensions>
</ePICApplication>
<!-- The following is for Test Application ID: TEST2 -->
<!-- Refer to TEST1 for explanations and possible additional entries. -->
<ePICApplication epicappid="TEST2">
  <epictrace>>true</epictrace>
  <epictracelevel>512</epictracelevel>
  <AdapterRouting cn="epicadapterrouting">
    <epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>
    <ePICBodyCategory epicbodycategory="DEFAULT">
      <ePICBodyType epicbodytype="DEFAULT">
        <epiccommandclassname>com.ibm.epic.adapters.eak.test.InstallVerificationTest
        </epiccommandclassname>
        <epicreceivemode>MQPP</epicreceivemode>
        <epicreceivemppqueue>TEST2AIQ</epicreceivemppqueue>
        <epicerrormppqueue>TEST2AEQ</epicerrormppqueue>
        <epicreplymppqueue>TEST2RPL</epicreplymppqueue>
      </ePICBodyType>
    </ePICBodyCategory>
  </AdapterRouting>
</ePICApplication>
<!-- The following is for Test Application ID: TEST3 -->
<!-- Refer to TEST1 for explanations and possible additional entries. -->
<ePICApplication epicappid="TEST3">
  <AdapterRouting cn="epicadapterrouting">
    <epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>
    <ePICBodyCategory epicbodycategory="DEFAULT">
      <ePICBodyType epicbodytype="DEFAULT">
        <epicdestids>TEST1</epicdestids>
        <epicreceivemode>MQPP</epicreceivemode>
        <epicreceivemppqueue>TEST3AIQ</epicreceivemppqueue>
      </ePICBodyType>
    </ePICBodyCategory>
  </AdapterRouting>
</ePICApplication>
<!-- The following is for sample Trace Client Application ID: TraceClient -->
<!-- Contains the TraceClient configuration information for doing tracing. -->
<!-- This is the application id value in the 'epictraceclientid' element -->
<!-- configured for the application wanting to do tracing -->
<ePICApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    <!-- Dependency Trace Server application id used for SocketHandler -->
    <!-- and ENAHandler (uses MQSeries), defaults to TraceServer -->
    <epicdepappid>TraceServer</epicdepappid>
    <!-- Write messages synchronously (true) or asynchronously (false), -->
    <!-- defaults to false (write messages asynchronously). This is -->
    <!-- used when giving the messages to the handlers. -->
    <epictracesyncoperation>>false</epictracesyncoperation>
    <!-- Default Trace message file to use if none passed in to the -->
    <!-- writeTrace method call. Defaults to this file if not indicated -->
    <epictracemessagefile>com.ibm.epic.trace.client.TraceMessage</epictracemessagefile>
    <!-- Handlers to load. Handlers do the actual processing of the -->
    <!-- Trace message. If the default trace client id 'TraceClient' -->
    <!-- is used then the handler defaults to the -->
    <!-- com.ibm.logging.ConsoleHandler. If the default trace client -->
    <!-- id 'TraceClient' is not used, the handler has to be specified. -->
    <!-- A Single Trace Handler -->
    <epictracehandler>com.ibm.logging.ConsoleHandler</epictracehandler>
    <!-- Multiple Trace Handlers -->
    <!--
    <epictracehandler>
      <Value>com.ibm.logging.ConsoleHandler</Value>
      <Value>com.ibm.logging.SocketHandler</Value>
    </epictracehandler>
    -->
    <!-- Handler definitions. Available definitions depend on the -->
  </ePICTraceExtensions>
</ePICApplication>

```

```

<!-- handler. Formatters are used for formatting the trace message.-->
<ePICTraceHandler epictracehandler="com.ibm.logging.ConsoleHandler">
  <!-- ConsoleHandler formatter to use, defaults to this formatter if none provided. -->
  <epictraceformatter>com.ibm.epic.trace.client.EpicTraceFormatter</epictraceformatter>
</ePICTraceHandler>
<ePICTraceHandler epictracehandler="com.ibm.logging.FileHandler">
  <!-- FileHandler formatter to use, defaults to this formatter if none provided. -->
  <epictraceformatter>com.ibm.epic.trace.client.EpicTraceFormatter</epictraceformatter>
  <!-- Trace filename to use, defaults to trc.log in the current directory. -->
  <epictracefilename>trc.log</epictracefilename>
</ePICTraceHandler>
<ePICTraceHandler epictracehandler="com.ibm.epic.trace.client.ENAHandler">
  <!-- ENAHandler formatter to use, defaults to this formatter if none provided. -->
  <epictraceformatter>com.ibm.epic.trace.client.EpicXMLFormatter</epictraceformatter>
</ePICTraceHandler>
<ePICTraceHandler epictracehandler="com.ibm.logging.SocketHandler">
  <!-- SocketHandler formatter to use, defaults to this formatter if none provided. -->
  <epictraceformatter>com.ibm.epic.trace.client.EpicXMLFormatter</epictraceformatter>
</ePICTraceHandler>
</ePICTraceExtensions>
</ePICAApplication>
<!-- The following is for sample Trace Server Application ID: TraceServer -->
<!-- Contains the TraceServer configuration information. -->
<!-- This is the application id pointed to by the trace client -->
<!-- epicdeppappid value. Definitions are similar to TraceClient example. -->
<ePICAApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    <epicmqppqueueemgr>DEFAULT</epicmqppqueueemgr>
    <ePICBodyCategory epicbodycategory="DEFAULT">
      <ePICBodyType epicbodytype="DEFAULT">
        <epicreceiveivemode>MQPP</epicreceiveivemode>
        <epicreceiveivemqppqueue>TraceServerAIQ</epicreceiveivemqppqueue>
      </ePICBodyType>
    </ePICBodyCategory>
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    <!-- Write messages synchronously/asynchronously (true/false (default)). -->
    <epictracesyncoperation>>false</epictracesyncoperation>
    <!-- Trace message file. Defaults to this file if not indicated -->
    <epictracemessagefile>com.ibm.epic.trace.server.TraceServerMessage</epictracemessagefile>
    <!-- Handlers to load, for multiple handlers see TraceClient example. -->
    <!-- If the default trace server id 'TraceServer' is used then the handler -->
    <!-- defaults to the com.ibm.logging.MultiFileHandler. -->
    <!-- Note: Do not use SocketHandler or ENAHandler for the trace server. -->
    <epictracehandler>com.ibm.logging.MultiFileHandler</epictracehandler>
    <!-- Handler definitions for com.ibm.logging.SocketHandler -->
    <!-- Formatter to use, defaults to this formatter if none provided.-->
    <ePICTraceHandler epictracehandler="com.ibm.logging.SocketHandler">
<!-- Entries when using socket handler from the TraceClient and -->
<!-- starting the Trace Server in socket receive mode. -->
<!-- SocketHandler host machine, defaults to localhost -->
    <epictracesocketsserverhost>localhost</epictracesocketsserverhost>
    <!-- SocketHandler port number, defaults to 8181 -->
    <epictraceportnumber>8181</epictraceportnumber>
  </ePICTraceHandler>
  <!-- Formatter to use, defaults to this formatter if none provided.-->
  <ePICTraceHandler epictracehandler="com.ibm.logging.ConsoleHandler">
    <!-- ConsoleHandler formatter to use, defaults to this formatter if none provided.-->
    <epictraceformatter>com.ibm.epic.trace.client.ReFormatter</epictraceformatter>
  </ePICTraceHandler>
  <ePICTraceHandler epictracehandler="com.ibm.logging.MultiFileHandler">
<!-- MultiFileHandler formatter to use, defaults to this formatter if none provided. -->
    <epictraceformatter>com.ibm.epic.trace.client.ReFormatter</epictraceformatter>
  <!-- MultiFileHandler trace base filename to use, defaults to trc.log in the -->
  <!-- current directory. The actual filename will be for this -->
  <!-- example trcx.log, where x is a numeric number starting at -->
  <!-- 0 and going up to the number of trace files specified. -->
    <epictracefilename>trc.log</epictracefilename>
  <!-- MultiFileHandler number of trace files, defaults to 3 -->
    <epictracefilenumber>3</epictracefilenumber>
  <!-- MultiFileHandler file size in number of bytes, defaults to -->
    <epictracefilesize>1000000</epictracefilesize>
  </ePICTraceHandler>
</ePICTraceExtensions>
</ePICAApplication>
</ePICAApplications>

```

## 최소 구성 파일 샘플

이 절에서는 MQSeries Adapter Kernel과 함께 사용할 최소 구성 파일 예제를 제공합니다. 최소 구성 파일에 대해서는 95 페이지의 『구성에 어댑터 정보 추가』를 참조하십시오.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- aqmconfig.minimum.xml 1.00 00/11/07 -->
<!-- Used for MQSeries Adapter Kernel -->
<!-- Sample AQM Configuration showing a minimum configuration for the -->
<!-- following conditions: -->
<!-- 1) Going from applicationid TEST1 to TEST2. TEST1 is not receiving -->
<!-- messages. -->
<!-- 2) TEST2 has no special application requirements. -->
<!-- 3) TEST2 is using 1 worker. -->
<!-- 4) Using MQSeries with the default QManager installed on each machine. -->
<!-- and using default format. -->
<!-- 5) No specific body category and body type being used. -->
<!-- 6) Using default tracing to the console. -->
<!-- -->
<!-- -->
<!-- -->
<!-- Copyright (c) 2000 International Business Machines. All Rights Reserved. -->
<!-- -->
<!-- This configuration file is as an example only. -->
<!-- -->
<!-- IBM MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THIS -->
<!-- SAMPLE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE -->
<!-- IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR -->
<!-- PURPOSE, OR NON-INFRINGEMENT. -->
<!-- -->
<!-- CopyrightVersion 1.0 -->
<!-- -->
<Epic o="ePIC">
  <ePICApplications o="ePICApplications">
    <!-- The following is for sample Test Application ID: TEST1 -->
    <ePICApplication epicappid="TEST1">
      <!-- Tracing on/off. If no entry defaults to false. -->
      <epictrace>>false</epictrace>
      <!-- Trace levels - 512=TYPE_ERROR_EXC (Exceptions),-1=TYPE_ALL (All possible messages). -->
      <epictracelevel>0</epictracelevel>
      <AdapterRouting cn="epicadaptrerouting">
        <epicmqqpqueuemgr>DEFAULT</epicmqqpqueuemgr>
        <ePICBodyCategory epicbodycategory="DEFAULT">
          <ePICBodyType epicbodytype="DEFAULT">
            <!-- Default destinations to send messages to. -->
            <epicdestids>TEST2</epicdestids>
          </ePICBodyType>
        </ePICBodyCategory>
      </AdapterRouting>
    </ePICApplication>
    <!-- The following is for Test Application ID: TEST2 -->
    <ePICApplication epicappid="TEST2">
      <epictrace>>false</epictrace>
      <epictracelevel>512</epictracelevel>
      <AdapterRouting cn="epicadaptrerouting">
        <epicmqqpqueuemgr>DEFAULT</epicmqqpqueuemgr>
        <ePICBodyCategory epicbodycategory="DEFAULT">
          <ePICBodyType epicbodytype="DEFAULT">
            <!-- AdapterDaemon - Command to invoke. -->
            <epiccommandclassname>com.ibm.epic.adapters.eak.samples.SampleCAdapterWrapper
            </epiccommandclassname>
            <epicreceivemode>MQ</epicreceivemode>
            <!-- Receive Time out in milliseconds ie. 1000 = 1 second, -->
            <!-- -1 means never ending. No entry defaults to 0. -->
            <!-- milliseconds. Used when receiving messages. -->
            <epicreceivetimeout>30000</epicreceivetimeout>
            <epicreceivemppqueue>TEST2AIQ</epicreceivemppqueue>
            <epicerrormppqueue>TEST2AEQ</epicerrormppqueue>
            <epicreplymppqueue>TEST2RPL</epicreplymppqueue>
          </ePICBodyType>
        </ePICBodyCategory>
      </AdapterRouting>
    </ePICApplication>
  </ePICApplications>
</Epic>
```

```
    </AdapterRouting>  
  </ePICApplication>  
</ePICApplications>  
</Epic>
```





---

## 부록E. 설정 파일 샘플

다음은 여러 환경 변수를 포함하여 커널의 여러 초기 구성 값을 정의하는 aqmsetup 파일의 예입니다. 이 파일에 대해서는 73 페이지의 『설정 파일』을 참조하십시오. aqmsetup 파일은 커널 루트 설치 디렉토리의 samples 디렉토리에 있습니다.

```
#
# aqmsetup 1.01 01/03/27
# Sample AQM Adapter runtime parameter configuration file entries.
#
# Copyright (c) 2001 International Business Machines. All Rights Reserved.
#
# This configuration file is as an example only.
#
# IBM MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THIS
# SAMPLE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
# IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
# PURPOSE, OR NON-INFRINGEMENT.
#
# CopyrightVersion 1.0
#
#
# Pound (#) signs are comments.
#
#####
#
# Use Websphere Business Integrator(WSI) product 5724-A78 LDAP
# Directory Services or configuration file. No entry defaults to
# true (use configuration file). To use the WSI directory service
# set the value to false. Refer to the WSI documentation for
# specifics on using the directory service.
#AdapterDirectoryUseFileFlag=true
# When using Websphere Business Integrator(WSI) product 5724-A78 LDAP
# Directory Services this additional entry is required. Refer to the
# WSI documentation for specifics on using the directory service.
#DirectoryServices=ChangeToDestDir/samples/DirectoryServices.properties
# Location of configuration file aqmconfig.xml when not using
# the Websphere Business Integrator(WSI) product 5724-A78 LDAP
# Directory Services.
# No entry defaults to current directory.
#AQMConfig=ChangeToDestDir/samples
#
#####
#####
# XML DTD Catalogs and Directories - where to locate DTD's if not
# in the current directory.
# Format: XML_DTD_DIRECTORY_x=ddd where x is a numeric suffix to
# be incremented for each key and ddd is the directory.
# The numeric suffix's must start with 1 and be contiguous.
#####
XML_DTD_DIRECTORY_1=ChangeToDestDir/runtimefiles/oag
```

```

#XML_DTD_DIRECTORY_2=ChangeToDestDir/runtimefiles
#
#####
# Java JNI Environment Variables for C Interface for increasing
# the amount of memory used. This applies to when a C module
# is instantiating a JVM. When a C Interface is being called
# from within JAVA the JVM is already established.
#####
# The stack memory is used for holding local function, function
# parameters, local variable references.
# Native stack is used for non-Java calls from within Java such
# as to C code. Stack size in bytes to use.
# Default is 128 kilobytes on NT.
#AQM_JNI_NATIVESTACKSIZE=1048576
# Java stack is for Java method calls and local variables.
# Stack size in bytes to use.
# Default is 400 kilobytes on NT.
#AQM_JNI_JAVASTACKSIZE=4194304
# The heap memory is used for storing instantiated Java objects
# Minimum heap size in bytes to start with.
# Default is 1 megabyte on NT.
#AQM_JNI_MINHEAPSIZE=16777216
# Maximum heap size in bytes which can be used.
# Default is 16 megabytes on NT.
#AQM_JNI_MAXHEAPSIZE=268435426#
#####
# Designate end of configuration file
#####
*ENDCFG

```

---

## 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 개발되었습니다. IBM은 다른 나라에서는 이 문서에서 언급한 제품, 서비스 또는 부속 기능을 제공하지 않을 수도 있습니다. 현재 사용 가능한 제품 및 서비스에 대한 정보는 한국 IBM 영업대표에 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 반드시 IBM 제품, 프로그램 또는 서비스 만을 사용하라는 의미는 아닙니다. IBM의 지적 재산권을 침해하지 않는 한, 타사의 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나, 비IBM 제품, 프로그램 또는 서비스 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 문서에서 다루어지는 내용에 대해 특허를 보유하고 있거나 특허 출원 중일 수 있습니다. 이 문서를 제공한다고 해서 특허에 대한 사용권까지 제공하는 것은 아닙니다. 사용권에 대한 의문사항은 다음으로 문의하시기 바랍니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2 바이트(DBCS) 정보에 관한 사용권 문의는 한국 IBM 고객만족센터 또는 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 단락은 해당 규정이 현지법과 상충되는 영국이나 기타 나라에서는 적용되지 않습니다. IBM은 이 책을 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 어떠한 종류의 묵시적 또는 명시적 보증없이 “현

상태대로” 제공됩니다. 일부 국가에서는 특정 거래에 있어서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 정기적으로 변경되며 변경사항은 신판에 통합됩니다. IBM은 이 책에 기술된 제품 및/또는 프로그램을 사전 통고 없이 언제든지 변경하거나 개선 할 수 있습니다.

이 정보에서 언급되는 비 IBM 웹 사이트는 편의상 제공된 것으로 이들 웹 사이트를 옹호하기 위한 것은 절대 아닙니다. 이들 웹 사이트에 나오는 자료들은 이 IBM 제품에 대한 자료의 일부가 아니며, 이들 웹 사이트의 사용은 사용자 본인의 책임입니다.

IBM은 적당하다고 생각하는 방식으로 사용자가 제공하는 정보를 사용하거나 배포할 수 있으며, 사용자는 이에 대해 책임을 지지 않습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함) 사이의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 정보를 필요로 하는 이 프로그램의 사용권자는 다음으로 문의해야 합니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들면, 사용료 지불 등)에 따라 사용할 수 있습니다.

이 정보에 기술된 사용권 프로그램과 이에 사용할 수 있는 모든 사용권 자료는 IBM 기본 계약, IBM 프로그램 사용권 계약 또는 이에 준하는 당사와의 계약을 통해 제공됩니다.

여기에 포함된 모든 성능 데이터는 제어된 환경에서 판별되었습니다. 따라서 다른 운영 환경에서 얻은 결과는 상당히 다를 수 있습니다. 개발 레벨 시스템에서 일부

측정이 이루어졌을 수 있으며 이러한 측정이 일반적으로 사용할 수 있는 시스템에서도 동일하다고 보증할 수 없습니다. 더우기 일부 측정치는 보외법을 통해 이루어졌습니다. 실제 결과는 다를 수 있습니다. 이 문서의 사용자는 특정 환경에 적용 가능한 데이터를 확인해야 합니다.

비IBM 제품에 대한 정보는 이들 제품의 공급업체, 이들 업체의 공개 자료 또는 범용 자료로부터 입수한 것입니다. IBM은 이들 제품을 테스트하지 않았으며 비IBM 제품과 관련한 배상 청구, 호환성 또는 성능의 정확성 등에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대해서는 이들 제품의 공급업체에 문의하십시오.

---

## 상표

다음 용어는 미국 또는 기타 국가에서 사용되는 International Business Machines Corporation의 상표입니다.

AIX	OS/400
AS/400	RISC System/6000
IBM	RS/6000
MQSeries	WebSphere

Lotus 및 LotusScript는 미국 또는 기타 국가에서 사용되는 Lotus Development Corporation의 상표입니다.

Java 및 모든 Java 관련 상표와 로고는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표 또는 등록상표입니다.

Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.



## 용어집

이 용어집에는 MQSeries Adapter Kernel 문서에서 사용된 핵심 용어와 그 의미가 포함되어 있습니다.

특정 개념이나 용어가 한 섹션에만 나타나는 경우에는 용어집에 포함시키지 않았습니다. 그러나 149 페이지의 『색인』에는 있을 수 있습니다.

용어집에 MQSeries와 같은 기타 IBM 제품에 대한 용어는 없습니다.

**구성 파일(configuration file).** 커널의 구성 값 대부분을 포함하는 aqmconfig.xml 파일. 세부사항은 74 페이지의 『구성 파일』을 참조하십시오.

**논리 메시지 전달 서비스.** 원시 어댑터가 통신 전송에 의한 전송을 위해 메시지를 변환하기 위해 사용되는 구성요소.

**대상 논리 ID.** 대상 어댑터와 연관된 대상 응용프로그램을 나타내는 값. 『대상 논리 ID』 및 145 페이지의 『응용프로그램 논리 ID』를 참조하십시오.

**대상 어댑터(target adapter).** 다음 작업을 수행하는 어댑터.

- 소스 어댑터가 송신한 메시지(커널 및 MQSeries 또는 다른 메시징 소프트웨어로부터)를 수신합니다.
- 어댑터가 모델화된 방법에 따라 통합 메시지를 처리합니다.
- 통합 메시지를 대상 응용프로그램이 수신할 수 있는 응용프로그램 특정 형식 메시지로 변환합니다.
- 응용프로그램 특정 인터페이스를 사용하여 메시지를 대상 응용프로그램으로 송신합니다.
- 작업 프로그램이 수신확인을 송신할 수 있도록 작업 프로그램으로 하여금 대상 응용프로그램으로의 메시지 송신을 완료한 시기를 알게 합니다.

대상 응용프로그램이 통합 메시지를 수신할 수 있다면 대상 어댑터가 필요없을 수 있습니다.

각 메시지 유형에 대해 대상 어댑터는 한 개뿐입니다. 일반적으로 대상 응용프로그램은 여러 메시지 유형을 송신할 수 있으므로, 대부분의 경우 여러 대상 어댑터가 대상 응용프로그램을 지원합니다. 145 페이지의 『어댑터』를 참조하십시오.

**대상 응용프로그램.** 컴퓨터 네트워크를 통해 데이터를 일반적으로 다른 컴퓨터에 상주하는 프로그램(소스 응용프로그램)으로부터 수신하는 데 필요한 프로그램.

**로그온 클래스(logon class).** 각 대상 응용프로그램 고유의 Java 클래스로서 대상 응용프로그램으로의 메시지 전달을 돕는 데 사용할 수 있습니다. 로그온 클래스는 대상 어댑터가 메시지를 전달하기 전에 대상 응용프로그램에 로그온해야 하는 경우에만 필요합니다. 각 로그온 클래스는 사용자가 작성합니다. 작업 프로그램은 로그온 클래스를 인스턴스화합니다. 로그온 클래스는 대상 어댑터가 대상 응용프로그램에 대한 응용프로그램 특정 인터페이스를 지원하는 데 필요로 하는 값을 찾기 위해 구성 파일을 참조합니다. 일반적으로 이러한 값은 로그온 매개변수입니다. 따라서 값은 대상 어댑터에서 사용 가능하게 됩니다.

아무 것도 하지 않는 데미 로그온이 커널과 함께 제공됩니다.

**리시브 큐(receive queue).** MQSeries Adapter Offering 용어로서 메시지를 수신하는 데 기본 입력 큐로서 사용되는 메시지 큐. 대상 응용프로그램마다 여러 개의 리시브 큐가 있을 수 있지만, 응용프로그램 ID, 본문 범주 및 본문 유형의 각 조합에 대해서는 리시브 큐가 한 개뿐입니다.

**리플라이 큐(reply queue).** 응답을 수신하는 데 사용되는 메시지 큐. 커널의 sendRequestResponse 메소드와 함께 사용됩니다.

**메시지 유형(message type).** 본문 범주와 본문 유형의 고유 조합으로 지정되는 메시지. 『본문 범주』 및 『본문 유형』을 참조하십시오.

**메시지 제어 값.** 커널이 메시지의 정렬 및 라우팅을 제어하는데 사용하고 각 어댑터가 부분적으로 기능 수행 방법을 제어하는 데 사용하는 구성 파일과 메시지(본문 및 헤더)의 값 세트 집합 용어.

**메시지 홀더 오브젝트.** 커널이 통합 메시지와 기타 제어 데이터를 캡슐화하는데 커널이 사용하는 메타 데이터용 컨테이너.

**메시지(message).** MQSeries에서 MQSeries Adapter Offering을 포함하여 한 프로그램이 송신하고 다른 프로그램용으로 설계된 데이터 콜렉션.

**목적지 논리 ID(destination logical identifier).** 대상 응용프로그램을 나타내는 값. 커널이 메시지를 라우트하고 메시지를 정렬할 때 다른 메시지 제어 값과 함께 사용됩니다. 『메시지 제어 값』을 참조하십시오.

**본문 범주(body category).** 메시지 응용프로그램 유형. 예를 들어 OAG 또는 RosettaNet를 나타내는 메시지에 포함된 데이터. 메시지 제어 값 세트에 속합니다. 『메시지 제어 값』을 참조하십시오.

본문 범주는 메시지 유형을 지정하는 데도 도움이 됩니다. 『메시지 유형』을 참조하십시오.

**본문 유형(bodytype).** 메시지의 특정 목적, 예를 들어 판매 주문 추가 또는 재고 동기화를 나타내는 메시지에 포함된 데이터. 메시지 제어 값 세트에 속합니다. 『메시지 제어 값』을 참조하십시오.

본문 유형은 메시지 유형을 지정하는 데도 도움이 됩니다. 『메시지 유형』을 참조하십시오.

**설정 파일(setup file).** 커널의 여러 초기 설정을 포함하는 파일. 디폴트 파일 이름은 aqmsetup입니다.

**소스 논리 ID(source logical identifier).** 소스 응용프로그램을 나타내는 값. 커널이 메시지를 라우트하고 메시지를 정렬할 때 다른 메시지 제어 값과 함께 사용됩니다. 『메시지 제어 값』, 145 페이지의 『응용프로그램 논리 ID』 및 143 페이지의 『대상 논리 ID』를 참조하십시오.

**소스 어댑터(source adapter).** 다음 작업을 수행하는 어댑터.

- 소스 응용프로그램에서 구조 데이터를 승인하거나 그렇지 않으면 확보합니다(일반적으로 어댑터 외부에서 개발된 응용프로그램 특정 인터페이스를 사용하여).
- 어댑터가 모델화된 방법에 따라 구조 데이터를 처리합니다.
- 구조 데이터를 통합 메시지 형식으로 변환합니다.
- 커널을 사용하여 하나 이상의 대상 어댑터로 전달하기 위해 메시지를 메시지 큐에 놓은 다음 대상 응용프로그램으로 전달합니다.

각 메시지 유형에 대해 소스 어댑터는 한 개뿐입니다. 일반적으로 소스 응용프로그램은 여러 메시지 유형을 송신할 수 있으므로, 대부분의 경우 여러 소스 어댑터가 소스 응용프로그램을 지원합니다.

145 페이지의 『어댑터』를 참조하십시오.

**소스 응용프로그램.** 컴퓨터 네트워크를 통해 데이터를 일반적으로 다른 컴퓨터에 상주하는 프로그램(대상 응용프로그램)으로 송신하는 데 필요한 프로그램.



**어댑터 디먼(adapter daemon).** 커널의 일부인 실행 가능 소프트웨어. 어댑터 디먼은 PUSH 전달 모델에서만 사용됩니다. 그 목적은 작업 프로그램을 인스턴스화하는 것입니다. 시작후 어댑터 디먼은 활동 상태로 남습니다. 각 대상 응용프로그램의 경우, 하나 이상의 어댑터 디먼이 있을 수 있습니다.

대부분의 경우 어댑터 디먼은 대상 응용프로그램의 역할을 수행합니다. 필요한 기능, 예를 들어 대상 어댑터를 사용하여 전자 우편 송신 또는 파일에 레코드 쓰기를 수행합니다.

**어댑터.** MQSeries Adapter Builder의 출력. 일반적으로 사용자는 각 어댑터를 응용프로그램으로(부터) 송신된 한 가지 메시지 유형에 고유하게 빌드합니다. 따라서 어댑터는 MQSeries Adapter Offering의 일부가 아닙니다. 어댑터는 공유 라이브러리로 컴파일되는 C 또는 Java 소스 코드로 구성됩니다. 어댑터 및 MQSeries Adapter Kernel 이 함께 실행되는 경우 MQSeries Adapter Offering의 런타임 기능을 수행합니다. MQSeries Adapter Builder에서 사용자에 의해 모델화되는 방법에 따라, 어댑터는 제어 플로우, 데이터 플로우, 순차 탐색, 결정 및 반복을 포함한 조건부 분기, 데이터 입력, 데이터 컨텍스트 저장, 데이터 요소 변환, 트랜잭션 제어, 논리 조작 및 사용자 정의 코드와 같은 다양한 기능을 포함할 수 있습니다. 사용자가 작성한 어댑터는 재사용할 수 있습니다.

144 페이지의 『메시지 유형』, 144 페이지의 『소스 응용 프로그램』 및 143 페이지의 『대상 응용프로그램』을 참조하십시오.

**오류 큐(error queue).** MQSeries Adapter Offering 용어로서 리시브 큐에서 가져온 메시지를 처리할 수 없을 때 사용되는 메시지 큐.

**원시 어댑터(native adapter).** 메시지 홀더 오브젝트를 송신하고 수신하는 데 사용되는 소프트웨어.

**응답 대상 논리 ID.** 응답이 요청될 때 응답이 송신될 응용프로그램의 논리 ID. 디폴트는 메시지의 소스 논리 ID입니다.

**응용프로그램 공통 형식(application-neutral format).** 146 페이지의 『통합 메시지』를 참조하십시오.

**응용프로그램 논리 ID(application logical identifier).** 어댑터(소스 어댑터 또는 대상 어댑터)가 연관된 응용프로그램을 나타내는 ID. 144 페이지의 『소스 논리 ID』 및 143 페이지의 『대상 논리 ID』를 참조하십시오.

**응용프로그램 특정 인터페이스(application-specific interface).** 다음 중 하나의 목적으로 MQSeries Adapter Offering 외부에서 개발된 인터페이스.

- 소스 어댑터가 소스 응용프로그램으로부터 메시지를 확보할 수 있게 하기 위해.
- 대상 응용프로그램이 대상 어댑터로부터 메시지를 확보할 수 있게 하기 위해.

**작업 프로그램 메시지 bean.** WebSphere Application Server가 커널의 대상측에서 사용될 때 작업 프로그램의 기능을 수행하는 엔터프라이즈 bean.

**작업 프로그램(worker).** 커널의 일부인 소프트웨어. 작업 프로그램은 PUSH 전달 모델에서만 사용됩니다. 어댑터 디먼은 작업 프로그램을 시작하고 작성합니다. 각 작업 프로그램은 하나의 원시 어댑터를 관리합니다. 작업 프로그램은 각 메시지를 적절한 대상 어댑터로 전달합니다.

**전달 모델(delivery models).** 커널이 대상 응용프로그램으로 인터페이스하는 두 가지 모델이 있습니다. 다음은 이러한 모델입니다.

**PUSH** 커널이 대상 응용프로그램으로의 메시지 전달을 시작하고 관리해야 합니다. 이 모델은 일반적으로 MQSeries Adapter Offering을 지원할 때 대상 응용프로그램의 변경을 요구하지 않습니다.

**PULL** 대상 응용프로그램은 메시지 전달 관리를 담당

합니다. 이 모델의 경우 MQSeries Adapter Offering을 지원할 때 대상 응용프로그램을 변경해야 합니다. 대상 응용프로그램은 대상 응용프로그램에 대한 커널 인터페이스를 관리해야 합니다.

**전달의 PULL 모델(pull model of delivery).** 145 페이지의 『전달 모델』을 참조하십시오.

**전달의 PUSH 모델(push model of delivery).** 145 페이지의 『전달 모델』을 참조하십시오.

**종속 응용프로그램 ID(dependency application identifier).** 작업 프로그램이 서비스하는 응용프로그램 이름. 작업 프로그램이 어댑터 디먼 이름을 기초로 구성 과일에서 종속 응용프로그램 ID를 가져옵니다.

**추적 메시지(trace messages).** 커널의 특정 지점에서의 메시지 처리 상태를 포함하는 메시지. 추적 메시지를 사용하여 커널 또는 어댑터에 대한 문제점을 진단하는 데 도움을 줄 수 있습니다.

『추적』을 참조하십시오.

**추적(tracing).** 커널이 추적 메시지를 쓰는 데 사용하는 프로세스의 콜렉션. 『추적 메시지』를 참조하십시오.

**커널의 대상측(target side of the kernel).** 메시지를 메시지 큐에서 가져올 때 시작되고 대상 어댑터로 메시지를 송신할 때 끝나는 커널 기능의 일부.

**커널의 소스측(source side of the kernel).** 소스 어댑터로부터 메시지가 수신될 때 시작되고 메시지가 메시지 큐에 놓일 때 끝나는 커널 기능의 일부.

**커널(kernel).** MQSeries Adapter Kernel의 동의어.

**통신 메시지(communications message).** 사용 중인 통신 전송 고유의 메시징 형식으로 변환된 메시지 홀더 오브젝트와 통신 전송 고유 정보.

**통신 모드.** 커널이 메시지를 전송하고 중개자 서비스를 수행하는 데 사용하는 모드.

**통합 메시지.** 통합을 위해 응용프로그램 공통 형식의 응용프로그램 데이터로 구성된 메시지. 예는 소스 어댑터가 소스 응용프로그램의 형식에서 XML로 변환하는 XML 문서입니다.

**트랜잭션(transaction).** 분리할 수 없는 작업 단위로서 실행해야 하는 조각 세트입니다. 트랜잭션을 구성하는 모든 조각이 성공적이라면 트랜잭션은 확약됩니다. 즉 모든 조각이 수행됩니다. 트랜잭션을 구성하는 하나 이상의 조각이 실패할 경우 트랜잭션은 롤백됩니다. 즉 어떤 조각도 수행되지 않습니다.

**추적 클라이언트(trace client).** 추적 메시지를 쓰는 커널의 구성요소.

**aqmconfig.xml 파일.** 143 페이지의 『구성 파일』을 참조하십시오.

**aqmsetup 파일.** 144 페이지의 『설정 파일』을 참조하십시오.

**BOD.** 비즈니스 오브젝트 문서. 한 조직 내에서 또는 조직들간에 이동하는 표준 비즈니스 프로세스의 표현. 구매 주문 추가, 제품 사용 가능성 표시 및 판매 주문 추가 등이 예입니다. BOD는 OAG가 XML을 사용하여 정의합니다. 147 페이지의 『OAG』 및 147 페이지의 『XML』을 참조하십시오.

BOD는 MQSeries Adapter Offering이 메시지 본문을 통합 메시지에서 정의하는 데 사용할 수 있습니다.

**DTD.** 문서 유형 정의. XML에서는 보통 특정 문서 유형의 정식 정의를 포함하는 파일(또는 함께 사용되는 여러 파일). DTD 내에서 요소에 사용할 수 있는 이름, 요소가 DTD 내에서 발생할 수 있는 위치 및 요소가 함께 적용되는 방법을 지정합니다. MQSeries Adapter Offering

에서 DTD를 사용하여 메시지 본문을 정의할 수 있습니다. 『XML』 및 146 페이지의 『통합 메시지』를 참조하십시오.

**Java 서비스 어댑터.** JMS 리스너 환경에서 어댑터 더먼, 작업 프로그램 및 대상 어댑터 기능을 제공하는 Java 언어 어댑터 유형.

**JMS 리스너.** MQSeries Adapter Kernel과 WebSphere Application Server 확장판간의 확고한 통합을 가능케 하는 WebSphere Business Integrator 제품에서 제공하는 구성요소.

**MQSeries Adapter Builder.** 그래픽 사용자 인터페이스(GUI)를 사용하여 가상적으로 응용프로그램에 대한 어댑터를 빌드할 수 있게 해주는 소프트웨어.

**MQSeries Adapter Kernel 원시 어댑터.** 원시 어댑터의 동의어.

**MQSeries Adapter Kernel.** API, C 및 Java로 구성된 여러 실행 가능 프로그램, 여러 구성 파일의 세트. 커널은 어댑터로 작업하고 지원합니다. 145 페이지의 『어댑터』를 참조하십시오. 커널은 직접 어댑터를 지원하는 것 외에도 MQSeries 또는 다른 메시징 소프트웨어와 인터페이스, 메시지 작성 및 추적과 같은 하부구조 서비스와 메시지 라우팅을 포함한 관련 기능을 수행합니다.

**MQSeries Adapter Offering.** MQSeries Adapter Builder와 MQSeries Adapter Kernel로 구성된 응용프로그램 통합 제품 세트.

**OAG.** 개방형 응용프로그램 그룹. 비즈니스 소프트웨어 구성요소 상호 운영성 계통의 많은 탁월한 웨이크홀더로 구성된 비영리 산업 컨소시엄. OAG는 비즈니스 오브젝트 문서(BOD)를 정의합니다.

**WebSphere Application Server 확장판.** Sun Microsystems EJB(Enterprise JavaBeans) 스펙을 사용할 수 있도록 하는 IBM의 소프트웨어 제품. WebSphere

Application Server 확장판에는 엔터프라이즈 bean이 실행될 수 있는 EJB 서버가 포함됩니다. 엔터프라이즈 bean은 EJB 클라이언트에 의해 사용되고 공유되는 비즈니스 논리 및 데이터를 캡슐화합니다. 엔터프라이즈 bean에는 수명이 짧고 클라이언트 고유의 작업과 오브젝트를 캡슐화하는 세션 bean과 지속적인 데이터를 캡슐화하는 개체 bean의 두 가지 유형의 bean이 있습니다. 작업 프로그램 메시지 bean이라고도 하는 세션 bean 유형은 MQSeries Adapter Kernel의 대상측에서 사용될 수 있습니다.

**XML.** Extensible Markup Language. 데이터 표현을 위한 W3C 표준.



# 색인

## [ 가 ]

개방형 응용프로그램 그룹  
정보 13

구성  
개요 67  
수신 시간 종료 기간 20  
추적 레벨 18

구성 구성요소  
정보 12

구성 파일  
검증 97  
구문 75  
상위 레벨 요소 75  
샘플 129  
정보 74  
정보 추가 95  
조직 75  
편집 96  
XML 요소 77

구성 파일 검증  
XML 메시지 98

권한  
필수조건 46

## [ 나 ]

논리 메시지 전달 서비스  
런타임 플로우 19

## [ 다 ]

대상 어댑터  
기능 8  
명령 26, 27  
정보 12

대상 어댑터 (계속)  
Epic.Message.createReplyMsg 28

데이터 변환  
상위 레벨 9

데이터 조정  
상위 레벨 9

디스크 공간 요구사항 35

디폴트  
본문 범주 18  
본문 유형 18

## [ 라 ]

라우팅  
단계 14  
단순 15  
메시지 제어 값 15  
복합 9  
판별 15

런타임 플로우  
개요 4  
자세히 15

리시브 큐  
커널의 대상측 25

## [ 마 ]

메모리 활용  
C 언어 73  
Java 73

메소드  
대상 어댑터 27  
큐 관계 9  
sendMessage 7, 18, 20, 28  
sendRequestResponse 7, 18, 20  
sendResponse 7

메시지  
메시지 제어 값 6, 15  
본문 13  
수신확인 8, 18  
오브젝트 18  
응용프로그램 공통 13  
정보 13  
확인 BOD 메시지 18

메시지 유형  
데이터그램 9  
어댑터 3  
요청 9  
응답 9

메시지 전달  
단일 스레드 11  
멀티스레드 11

메시지 제어 값  
세부사항 17

메시지 헤더 121

메시지 홀더  
정보 11

메시지 홀더 오브젝트  
정의 14

## [ 바 ]

비즈니스 오브젝트 문서 13

## [ 사 ]

설정 파일  
편집 73  
설치 47  
프로시저 45  
소스 어댑터  
기능 6  
정보 10

소스 응용프로그램  
 형식 5  
 소프트웨어 전제조건 36  
 AIX 37  
 HP-UX 37  
 OS/400 37  
 Solaris 37  
 Windows 36  
 스레드  
 정책 스케줄 25  
 스케줄 정책 52

## [ 아 ]

어댑터  
 기능 2  
 예 2  
 유형 3  
 어댑터 다면  
 시작 24  
 이름 24  
 정보 11  
 어댑터 작업 프로그램  
 정보 11  
 예외 파일  
 EpicSystemExceptionFile.log 28  
 원시 어댑터  
 정보 11  
 웹 사이트  
 개방형 응용프로그램 그룹 111  
 관련 정보 ix  
 서적 ix  
 MQSeries 35  
 MQSeries SupportPacs ix  
 MQSeries 제품군 111  
 XML 111  
 유지보수 계획 105  
 응용프로그램 특정 인터페이스  
 예 5  
 정보 5

## [ 자 ]

작업 프로그램  
 인스턴스화 24  
 최소 수 25  
 플래그 29  
 전제조건  
 소프트웨어 36  
 하드웨어 35  
 정보 센터  
 MQSeries Adapter Kernel 111  
 정책 스케줄  
 스레드 25  
 종속 응용프로그램 ID  
 정보 25

## [ 차 ]

추적  
 런타임 플로우 18  
 시작 104  
 정보 31  
 추적 사용 가능 18  
 추적 구성요소  
 정보 13

## [ 카 ]

캐널  
 라우팅 6  
 사용 42  
 전달 모델 7  
 정렬 6  
 측면 5  
 큐  
 수신 9  
 오류 9  
 응답 9  
 응답 메시지 확보 28

큐 처리  
 요약 8

## [ 타 ]

통신 메시지  
 정의 14  
 통신 모드  
 런타임 플로우 19  
 목록 19  
 통합 메시지  
 정의 13  
 트랜잭션 가능 30

## [ 파 ]

파일  
 목록 41  
 위치 41  
 프로시저  
 상위 레벨 ix

## [ 하 ]

하드웨어 전제조건 35  
 확인 문제점  
 대상 어댑터 57  
 큐 56  
 큐 관리자 58  
 환경 변수 56  
 aqmconfig.xml 파일 56  
 aqmsetup 파일 56  
 MQSeries 오류 58  
 환경 변수  
 설치시 51  
 임시로 검증 설정 99  
 AIXTHREAD\_SCOPE 52  
 OS/400 설정 50  
 THREADS\_FLAG 52  
 환경 변수 파일 73

# [ 숫자 ]

1 단계 요약 30

## A

AIX

소프트웨어 전제조건 37

aqmconfig.xml 파일

샘플 129

위치 50

이름 51

정보 74

편집 96

aqmcreateq 파일 73

사용 108

aqmcrormsg 파일

사용 98

aqmsetenv 파일 73

aqmsetup 파일

위치 50

이름 51

편집 73

환경 변수 51

aqmsndmsg 파일

사용 100

aqmstrad 파일

사용 103

aqmstrtd 파일

사용 104

aqmverifyinstall 파일

사용 55

aqmversion 파일

사용 106

## B

BOD

예제 13

정보 13

## D

DTD

정보 13

## E

Epic

의미 xi

Epic.Message.createReplyMsg 28

## H

HP-UX

소프트웨어 전제조건 37

## J

Java

메모리 부족 조건 29

시동 매개변수 104

Java 로그인 클래스 66

Java 서비스 어댑터

정보 12

## M

MAX\_QUEUE\_DEPTH

설정 102

MQSeries

검증된 구성 119

역할 9

큐 9

확약 제어 25

MQSeries Adapter Builder

정보 17

MQSeries Adapter Kernel

정보 센터 111

MQSeries Adapter Offering

구성요소 2

서비스 제공 2

이점 1

MQSeries Adapter Offering (계속)

정보 소스 111

타어 4

MQSeries Integrator

검증된 구성 119

역할 9

통신 모드와의 관계 19

## O

OS/400

설치 필수조건 38

소프트웨어 전제조건 37

환경 변수 설정 50

## S

SDK

정의 42

Solaris

소프트웨어 전제조건 37

## W

Windows

소프트웨어 전제조건 36

## X

XML

정보 13

XML 요소

구성 파일 77









GA30-1268-05

