Dell™ PowerVault™ Encryption Key Manager

사용자 안내서

Dell™ PowerVault™ Encryption Key Manager

사용자 안내서

목차

그림

표

서문

이 책에 대한 정보

이 안내서에는 Dell™ Encryption Key Manager의 설치 및 작동에 필요한 정보와 지시사항이 포함되어 있습니다. 또한 다음에 관한 개념 및 관련 프로시저가 있습니다.

- 암호화 가능 LTO 4 및 LTO 5 테이프 드라이브
- 암호화 키
- 디지털 인증서

이 책의 독자

이 책은 중요한 데이터의 보안 및 백업을 담당하는 스토리지 및 보안 관리자와 운영 환경에서 Encryption Key Manager 서버의 설치 및 유지보수를 지원하는 사용자를 대상으로 합니다. 이 책에서는 독자가 스토리지 장치 및 네트워크에 대한 실제적인 지식을 갖추었다고 가정합니다.

이 책에 사용된 규칙 및 용어

이 책에서는 다음 인쇄 규칙을 사용합니다.

표 1. 이 책에 사용된 인쇄 규칙

규칙	사용법
굵은체	**굵은체**로 표시된 단어나 문자는 명령어, 파일 이름, 플래그 이름, 경로 이름 및 선택된 메뉴 옵션과 같이 문자 그대로 사용해야 하는 시스템 요소를 나타냅니다.
고정 글꼴	예제, 사용자가 지정한 텍스트 및 시스템에서 표시하는 정보는 고정 글꼴로 표시합니다.
기울임체	*기울임체*로 표시된 단어나 문자는 사용자가 제공해야 하는 변수값을 나타냅니다.
[항목]	선택적 항목을 나타냅니다.
{항목}	형식 및 구문 설명에서 항목을 선택해야 하는 목록을 묶습니다.
|	세로 막대는 선택사항 목록에서 항목을 구분합니다.
<키>	누르는 키를 나타냅니다.

주의사항

주의사항은 프로그램, 장치, 시스템 또는 데이터가 손상될 가능성이 있음을 나타냅니다. 주의사항에는 느낌표 기호가 함께 표시될 수 있지만 항상 표시되지는 않습니다. 다음은 주의사항에 대한 샘플입니다.

경고: 드라이버를 사용하여 이 절차를 수행하는 경우 테이프가 손상될 수 있습니다.

관련 서적

자세한 정보는 다음 서적을 참조하십시오.

- *Dell™ PowerVault™ TL2000 및 TL4000 테이프 라이브러리 시작하기*에서는 설치 정보를 제공합니다.
- *Dell™ PowerVault™ TL2000 Tape Library and TL4000 Tape Library SCSI Reference*에서는 SCSI 인터페이스 동작을 제어하는 지원되는 SCSI 명령 및 프로토콜을 제공합니다.

Linux 정보

Red Hat 정보

다음은 Red Hat Linux® 시스템과 관련된 URL입니다.

- **http://www.redhat.com**

SuSE 정보

다음은 SuSE Linux 시스템과 관련된 URL입니다.

- **http://www.suse.com**

Microsoft Windows 정보

다음은 Microsoft® Windows® 시스템에 대한 정보에 액세스할 수 있는 URL입니다.

- **http://www.microsoft.com**

온라인 지원

다음 관련 서적을 보려면 **http://support.dell.com**을 방문하십시오.

*Dell Encryption Key Manager 빠른 시작 안내서*에서는 기본 구성 설정에 대한 정보를 제공합니다.

다음 관련 서적을 보려면 **http://www.dell.com**을 방문하십시오.

Library Managed Encryption for Tape 백서에서는 LTO 테이프 암호화에 대한 모범 사례를 제공합니다.

먼저 읽어야 할 사항

Dell 연락처

미국에 거주하는 고객은 800-WWW-DELL(800-999-3355)로 전화하십시오.

주: 인터넷 연결이 활성화 되어 있지 않은 경우에는 구매 송장, 포장 명세서, 청구서 또는 Dell 제품 카탈로그에서 담당자 정보를 찾을 수 있습니다.

Dell은 여러 가지 온라인, 전화 기반 지원 및 서비스 옵션을 제공합니다. 이러한 옵션은 국가 및 제품에 따라 다르며, 한국에서는 일부 서비스를 사용하지 못할 수도 있습니다. 영업, 기술 지원 또는 고객 서비스 문제로 Dell에 연락하려면 다음을 수행하십시오.

1. **http://supportapj.dell.com/support/index.aspx**을 방문하십시오.
2. 페이지 맨 아래에 있는 **Choose A Country/Region** 드롭 다운 메뉴에서 해당 국가 또는 지역을 확인하십시오.
3. 페이지의 왼쪽에서 **Contact Us**를 클릭하십시오.
4. 필요에 따라 적절한 서비스 또는 지원 링크를 선택하십시오.
5. 사용자에게 편리한 Dell 연락 방법을 선택하십시오.

제 1 장 테이프 암호화 개요

데이터는 경쟁력 있는 비즈니스 환경에서 매우 중요한 가치를 지니는 자원 중 하나입니다. 보안의 중요성이 부각되는 현실에서 이러한 데이터를 보호하고 이에 대한 액세스를 제어하며 데이터 출처를 확인하는 동시에 가용성을 유지보수하는 기능이 가장 우선시되고 있습니다. 데이터 암호화는 이러한 많은 요구에 대해 답을 줄 수 있는 도구입니다. Dell Encryption Key Manager(이후로는 Encryption Key Manager로 지칭함))는 암호화 작업을 단순화합니다.

LTO 4 및 LTO 5 드라이브는 LTO 4 및 LTO 5 데이터 카트리지에 데이터를 쓰는 경우 해당 데이터를 암호화할 수 있습니다. 이 새로운 기능은 서버에서 수행되는 암호화로 인한 성능 저하 및 처리 오버헤드의 발생 및 전용 기구에 대한 비용을 지출하지 않고도 저장된 데이터에 대한 강력한 보안 수단을 추가합니다.

테이프 드라이브 암호화 솔루션은 다음과 같은 세 가지 주요 요소로 구성됩니다.

암호화 가능한 테이프 드라이브

모든 LTO 4 및 LTO 5 테이프 드라이브는 라이브러리 인터페이스를 통해 사용될 수 있어야 합니다.

테이프 드라이브에 대한 자세한 정보는 2-2 페이지의 『하드웨어 및 소프트웨어 요구사항』을 참조하십시오.

암호화 키 관리

암호화는 일련의 연속된 계층에서 여러 종류의 키 사용과 관련이 있습니다. 이러한 키의 생성, 유지보수, 제어 및 전송은 암호화하는 테이프 드라이브가 설치된 운영 환경에 따라 다릅니다. 일부 응용프로그램은 키 관리를 수행할 수 있습니다. 이러한 응용프로그램이 없는 환경 또는 응용프로그램에 종속되지 않는 암호화가 필요한 환경의 경우, Dell Encryption Key Manager에서 필요한 모든 키 관리 작업을 수행할 수 있습니다. 1-3 페이지의 『암호화 관리』에서 이러한 작업을 자세히 설명합니다.

암호화 정책

암호화를 구현하는 데 사용되는 방법입니다. 여기에는 키 선택 메커니즘 및 암호화하는 볼륨을 제어하는 규칙이 포함됩니다. 이러한 규칙을 설정하는 방법 및 위치는 운영 환경에 따라 다릅니다. 자세한 정보는 1-3 페이지의 『암호화 관리』를 참조하십시오.

컴포넌트

Encryption Key Manager는 Java 환경에 포함되어 있으며 암호 기능을 위해 Java Security 컴포넌트를 사용합니다. Java Security 컴포넌트에 대한 자세한 정보는 관련 서적을 참조하십시오. Encryption Key Manager에는 해당 동작을 제어하는 데 사용되는 세 가지 기본 컴포넌트가 있습니다. 이 컴포넌트는 다음과 같습니다.

Java Security 키 저장소

키 저장소는 Java Security의 요소 및 JCE(Java Cryptography Extension)의 일부로 정의됩니다. 또한 Java Security는 Java Runtime Environment의 일부로 포함됩니다. 키 저장소는 Encryption Key Manager에서 암호화 조작을 수행하기 위해 사용하는 인증서 및 키를 보유하고 있거나 해당 항목으로 포인터하는 것을 말합니다. 사용자의 요구를 충족시키기 위해 다양한 조작 특성을 제공하는 여러 가지 종류의 Java 키 저장소가 지원됩니다. 이러한 특성은 2-4 페이지의 『키 저장소 고려사항』에서 자세히 설명합니다.

키 저장소 데이터를 보존하는 작업은 매우 중요합니다. 키 저장소에 액세스하지 않으면 암호화된 테이프의 암호를 해독할 수 없습니다. 아래 주제를 자세히 읽고 키 저장소 데이터를 보호하는 데 사용할 수 있는 방법을 검토하십시오.

구성 파일

구성 파일을 사용하면 조직 요구를 만족하도록 Encryption Key Manager의 동작을 사용자 정의할 수 있습니다. 이러한 동작에 대한 옵션은 이 문서에서 자세히 설명합니다. 2-1 페이지의 제 2 장 『Encryption Key Manager 환경 계획』, 4-1 페이지의 제 4 장 『Encryption Key Manager 구성』에서 차례로 설명하고 마지막에는 부록 B에서 구성 옵션의 전체 세트를 설명합니다.

테이프 드라이브 테이블

테이프 드라이브 테이블은 Encryption Key Manager에서 지원하는 테이프 장치를 추적할 때 사용됩니다. 테이프 드라이브 테이블은 편집할 수 없는 2진 파일이며 해당 위치는 구성 파일에 지정되어 있습니다. 이 위치는 사용자 요구에 맞게 변경 가능합니다.

KeyGroups.xml 파일

이 암호로 보호되는 파일에는 모든 암호화 키 그룹의 이름 및 각 키 그룹과 연관된 암호화 키 별명이 포함됩니다.

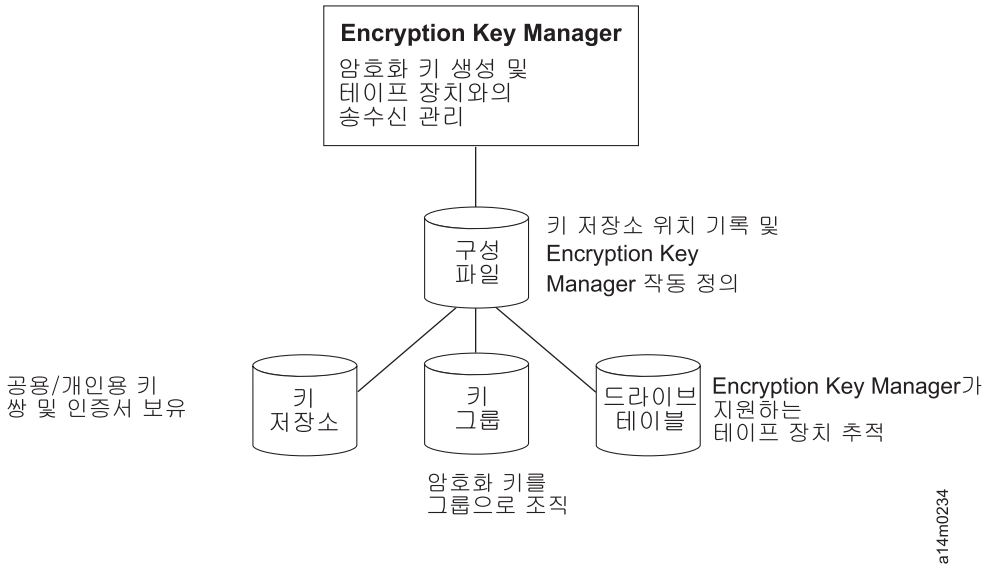

그림 1-1. Encryption Key Manager의 네 가지 기본 컴포넌트

암호화 관리

Dell Encryption Key Manager는 테이프 매체(테이프 및 카트리지 형식)에 쓸 정보를 암호화하거나 여기에서 읽으려는 정보를 암호 해독하는 데 사용되는 암호화 키를 생성, 보호, 저장 및 유지보수하는 경우 암호화 가능 테이프 드라이브를 지원하는 Java™ 소프트웨어 프로그램입니다. Encryption Key Manager는 Linux(SLES 및 RHEL) 및 Windows에서 작동하며, 엔터프라이즈 내 여러 위치에 배포된 공유 자원으로 백그라운드에서 실행하도록 설계되어 있습니다. 명령행 인터페이스 클라이언트는 사용자 환경에 맞게 Encryption Key Manager를 사용자 정의하고 이의 작동을 모니터할 수 있는 명령 세트를 제공합니다. Dell Encryption Key Manager GUI(Graphical User Interface)에서도 많은 사용자 정의 및 모니터링 기능을 사용할 수 있습니다. Encryption Key Manager는 모든 암호화 작업에 필요한 인증서 및 키를 보유하거나 해당 항목으로 포인터하기 위해 하나 이상의 키 저장소를 사용합니다. 자세한 내용은 2-4 페이지의 『키 저장소 고려사항』을 참조하십시오.

중요 Encryption Key Manager 호스트 서버 구성 정보: Dell Encryption Key Manager 프로그램을 호스트하는 시스템은 데이터 유실의 위험을 최소화하기 위해 ECC 메모리를 사용할 것을 권장합니다. Encryption Key Manager는 암호화 키의 생성을 요청하고 이들 키를 LTO 4 및 LTO 5 테이프 드라이브에 전달하는 기능을 수행합니다. 랩된(암호화된) 키는 처리 중 Encryption Key Manager가 시스템 메모리에 배치합니다. 카트리지에 기록된 데이터를 복구(암호 해독)할 수 있으려면 오류 없이 키를 적절한 테이프 드라이브에 전송해야 합니다. 시스템 메모리의 비트 오류로 키가 손상되었고 카트리지에 데이터를 쓸 때 이 키를 사용하는 경우, 해당 카트리지에 쓰여진 데이터는 복구할 수 없습니다. (즉, 차후에 암호를 해독할 수 없습니다.) 이러한 데이터 오류가 발생하지 않도록 하는 보호 장치가 마련되어 있습니다. 그러나 Encryption Key Manager를 호스트하는 시스템이 ECC(Error Correction Code) 메모리를 사용하지 않는 경우, 시스템 메모리에 있는 동안 키가 손상되고 이러한 손상으로 데이터 유실이 발생할 수 있는 가능성은 남아 있습니다. 이와 같은 데이터 유실 가능성은 적지만 중요 응용프로그램(예: Encryption Key Manager)을 호스트하는 시스템은 ECC 메모리를 사용할 것을 권장합니다.

Encryption Key Manager는 테이프 라이브러리와의 TCP/IP 통신 경로를 통해 전송된 키 도착 요청 또는 키 생성을 기다리는 백그라운드 프로세스 역할을 합니다. 테이프 드라이브에서 암호화된 데이터를 쓰는 경우 먼저 Encryption Key Manager에 암호화 키를 요청합니다. 요청이 도착하면 Encryption Key Manager는 다음 작업을 수행합니다.

Encryption Key Manager는 키 저장소에서 기존의 AES 키를 가져와 보안 전송을 위해 도착 시 이 키를 랩핑 해제하고 이 키를 사용하여 테이프에 쓸 데이터를 암호화하는 테이프 드라이브에 별도로 랩핑합니다.

LTO 4 또는 LTO 5 드라이브에서 암호화된 테이프를 읽는 경우 Encryption Key Manager는 테이프의 키 ID에 포함된 정보에 따라 키 저장소에서 필요한 키를 가져온 다음 보안 전송을 위해 랩핑된 테이프 드라이브에 제공합니다.

선택할 수 있는 암호화 관리 방법은 두 가지가 있습니다. 이 방법은 암호화 정책 엔진이 있는 위치와 암호화 솔루션에서 키 관리를 수행하는 위치, 드라이브와 Encryption Key Manager의 연결 방법에 따라 다릅니다. 최상의 방법은 운영 환경에서 결정합니다. 키 관리 및 암호화 정책 엔진은 다음 두 환경 계층 중 하나에 있습니다.

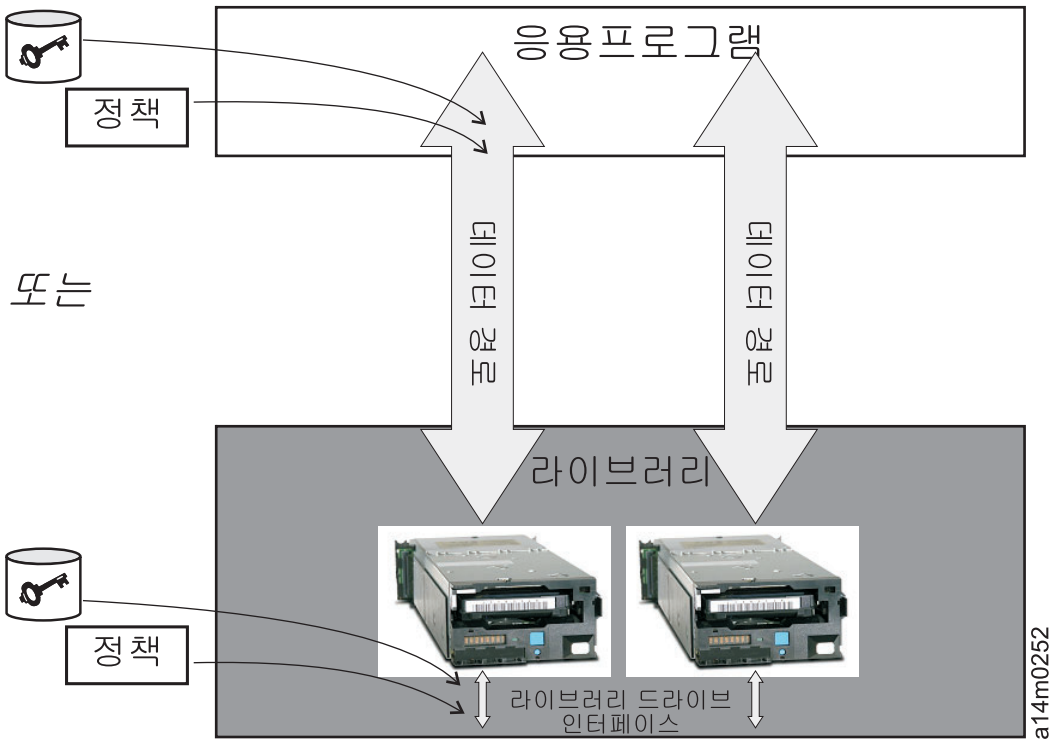

그림 1-2. 암호화 정책 엔진 및 키 관리가 있을 수 있는 두 위치

응용프로그램 계층

Key Manager와는 별도로 응용프로그램은 테이프 스토리지에서 데이터 전송을 시작합니다. 지원되는 응용프로그램은 『응용프로그램에서 관리하는 테이프 암호화』를 참조하십시오.

라이브러리 계층

테이프 스토리지의 격납장치(예: Dell PowerVault TL2000/TL4000 및 ML6000 제품군이 이에 해당합니다. 요즘 제공되는 테이프 라이브러리에는 내부에 각 테이프 드라이브에 대한 내부 인터페이스가 있습니다.

응용프로그램에서 관리하는 테이프 암호화

이 방법은 운영 환경에서 이미 암호화 정책 및 키를 생성하고 관리할 수 있는 응용프로그램을 실행하는 경우에 가장 적합합니다. 암호화를 사용할 시기를 지정하는 정책은 응용프로그램 인터페이스를 통해 정의됩니다. 정책 및 키는 응용프로그램 계층과 암호화한 테이프 드라이브 간 데이터 경로를 통해 전달됩니다. 암호화는 응용프로그램 및 암호화 가능한 테이프 드라이브 사이에서 상호 작용한 결과이며 암호화 때문에 시스템 및 라이브러리 계층을 변경할 필요는 없습니다. 응용프로그램에서 암호화 키를 관리하므로 응용프로그램을 사용하여 쓰고 암호화된 볼륨은 응용프로그램에서 볼륨을 쓴 방식대로 응용프로그램에서 관리하는 암호화 방법을 사용해야만 읽을 수 있습니다.

Encryption Key Manager는 응용프로그램에서 관리하는 테이프 암호화에서 사용되거나 요청되는 항목은 아닙니다.

암호화를 관리하는 경우 다음과 같은 최소 버전의 응용프로그램을 사용할 수 있습니다.

- CommVault Galaxy 7.0 SP1
- Symantec Backup Exec 12

응용프로그램에서 관리하는 테이프 암호화는 다음과 같은 LTO 4 및 LTO 5 테이프 드라이브에서 지원됩니다.

- Dell™ PowerVault™ TL2000 테이프 라이브러리
- Dell™ PowerVault™ TL4000 테이프 라이브러리
- Dell™ PowerVault™ ML6000 테이프 라이브러리

암호화 정책 및 키 관리 방법을 알려면 테이프 백업 소프트웨어 응용프로그램 문서를 참조하십시오.

라이브러리에서 관리하는 암호화

Dell™ PowerVault™ TL2000 테이프 라이브러리, Dell™ PowerVault™ TL4000 테이프 라이브러리 또는 Dell™ PowerVault™ ML6000 테이프 라이브러리의 LTO 4 및 LTO 5 테이프 드라이브에서 이 방법을 사용합니다. 키 생성 및 관리는 라이브러리에 연결된 호스트에서 실행되는 Java 응용프로그램인 Encryption Key Manager에서 수행합니다. 정책 제어 및 키는 라이브러리에서 드라이브로 연결된 인터페이스를 통해 전달되므로 암호화는 응용프로그램에 대해 투명합니다.

암호화 키 정보

암호화 키는 데이터를 스크램블(일종의 암호화)하고 이를 해제하기 위해 특별히 생성된 무작위 비트 문자열입니다. 암호화 키는 각 키가 고유하고 키를 예상할 수 없도록 설계된 알고리즘을 사용하여 작성됩니다. 키에서 이러한 방식으로 구성된 길이가 길수록 암호화 코드를 깨기 더 어렵습니다. IBM 및 T10의 암호화 방법 모두 데이터를 암호화하는 데 256비트 AES 알고리즘을 사용합니다. 256비트 AES는 현재 미국 정부에서 인정 및 권장하는 암호화 표준으로 세 가지 서로 다른 키 길이를 허용합니다. 256비트 키는 AES에서 허용한 키 중 가장 길이가 깁니다.

Encryption Key Manager에서는 대칭 알고리즘 및 비대칭 알고리즘과 같은 두 가지 유형의 암호화 알고리즘을 사용합니다. 대칭 또는 비밀 키 암호화에서는 암호화 및 암호 해독에 단일 키를 사용합니다. 보통 대칭 키 암호화는 많은 양의 데이터를 효율적으로 암호화하는 경우에 사용됩니다. 256비트 AES 키는 대칭 키입니다. 비대칭 또는 공용/개인용 키 암호화는 한 쌍의 키를 사용합니다. 키 하나를 사용하여 암호화된 데이터는 공용/개인용 키 쌍의 다른 키를 사용해야만 암호를 해독할 수 있습니다. 비대칭 키 쌍이 생성되면 공용 키는 암호화에, 개인용 키는 암호 해독에 사용됩니다.

Encryption Key Manager에서는 대칭 및 비대칭 키를 모두 사용합니다. 대칭 암호화는 사용자 또는 호스트 데이터를 빨리 암호화하는 경우에, 비대칭 암호화는 대칭 키를 보호하는 경우(속도가 더 느림)에 사용됩니다.

keytool과 같은 유틸리티를 사용하여 Encryption Key Manager에 대한 암호화 키를 생성할 수 있습니다. AES 키를 생성하는 책임과 이를 테이프 드라이브에 전송하는 방식은 암호화 관리 방법에 따라 다릅니다. 그러나 Encryption Key Manager에서 암호화 키를 사용하는 방법과 다른 응용프로그램에서 사용하는 방법 사이의 차이를 이해하면 도움이 될 수 있습니다.

Dell Encryption Key Manager에서의 암호화 키 처리

라이브러리에서 관리하는 테이프 암호화의 경우 암호화되지 않은 데이터는 LTO 4 또는 LTO 5 테이프 드라이브로 전송되어 Encryption Key Manager에서 사용 가능한 키 저장소의 사전에 생성된 대칭 데이터 키(DK)를 사용하여 암호 텍스트로 변환된 후 다시 테이프에 기록됩니다. Encryption Key Manager는 라운드 로빈 방식으로 사전에 생성된 DK를 선택합니다. 여러 개의 테이프 카트리지에서 사전 생성된 DK 수가 부족한 경우 DK를 다시 사용합니다. DK는 Encryption Key Manager에서 암호화 또는 랩핑된 양식으로 LTO 4 또는 LTO 5 테이프 드라이브로 전송됩니다. LTO 4 및 LTO 5 테이프 드라이브는 이러한 DK를 랩핑 해제하여 암호화 또는 암호 해독에 사용합니다. 그러나 랩핑 해제되지 않은 키는 LTO 4 또는 LTO 5 테이프 카트리지의 임의의 위치에 저장됩니다. 암호화된 볼륨이 기록되면 DK는 볼륨에서 읽은 순서대로 별명 또는 키 레이블에 따라 액세스 가능하고 Encryption Key Manager에서 사용할 수 있어야 합니다. 1-8 페이지의 그림 1-3에서 이 프로세스를 보여줍니다.

Dell Encryption Key Manager에서는 LTO 암호화를 위한 대칭 키를 키 그룹으로 조직하는 기능도 제공합니다. 이때 암호화하는 데이터 유형, 이 데이터에 액세스하는 사용자 또는 다른 중요한 특징에 따라 키를 그룹화할 수 있습니다. 자세한 정보는 3-16 페이지의 『키 그룹 작성 및 관리』를 참조하십시오.

다른 응용프로그램에서 암호화 키 처리

응용프로그램에서 관리하는 테이프 암호화의 경우 암호화되지 않은 데이터는 LTO 4 및 LTO 5 테이프 드라이브로 전송되어 해당 응용프로그램에서 제공하는 대칭 DK를 사용하여 암호 텍스트로 변환된 후 다시 테이프에 기록됩니다. DK는 테이프 카트리지의 임의의 위치에 저장되지 않습니다. 암호화된 볼륨이 기록되면 DK는 예를 들어 볼륨에서 읽은 순서대로 응용프로그램에서 사용 가능한 위치, 서버 데이터베이스에 있어야 합니다.

응용프로그램에서 관리하는 암호화의 경우 LTO 4 및 LTO 5 테이프 드라이브는 Yosemite(Dell PowerVault TL2000 및 TL4000 테이프 라이브러리용), CommVault 및 Symantec Backup Exec와 같은 응용프로그램을 사용할 수 있습니다.

또는 암호화를 수행하는 T10 명령 세트를 사용하는 응용프로그램의 경우 LTO 4 및 LTO 5 테이프 드라이브를 사용할 수 있습니다. T10 명령 세트는 응용프로그램에서 제공하는 대칭 256비트 AES 키를 사용합니다. T10은 테이프 카트리지당 고유한 여러 DK를 사용할 수 있으며 암호화된 데이터 및 일반 데이터를 같은 테이프 카트리지에 쓸 수 있습니다. 응용프로그램에서 테이프 카트리지를 암호화하는 경우 응용프로그램에서 판별한 방법을 사용하여 DK를 선택하거나 생성한 후 테이프 드라이브로 전송합니다. 키는 비대칭 공용 키를 사용하여 랩핑되지 **않으며** 테이프 카트리지에 저장되지 **않습니다**. 암호화된 데이터가 테이프에 기록되면 DK는 데이터를 읽은 순서대로 응용프로그램에서 사용 가능한 위치에 있어야 합니다.

응용프로그램에서 관리하는 암호화 및 라이브러리에서 관리하는 암호화에 관한 테이프 암호화에 대한 프로세스는 그림 1-3에 나와 있습니다.

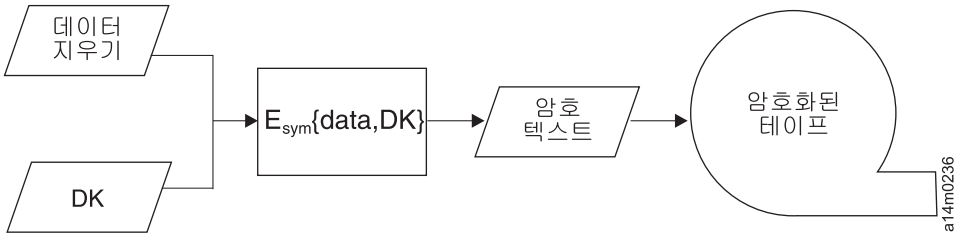

그림 1-3. 대칭 암호화 키를 사용하여 암호화. LTO 4 및 LTO 5 테이프 드라이브에서의 라이브러리에서 관리하는 암호화 및 응용프로그램에서 관리하는 암호화

요약

각 볼륨에서 사용할 수 있는 암호화 키의 수는 암호화를 관리하는 데 사용하는 방법, 암호화 표준 및 테이프 드라이브에 따라 다릅니다. LTO 4 및 LTO 5의 투명 암호화(즉, Encryption Key Manager에서 라이브러리 관리 암호화 사용)의 경우 DK의 고유함은 Encryption Key Manager에서 충분한 수의 사전 생성된 키를 사용할 수 있는지에 종속되어 있습니다.

표 1-1. 암호화 키 요약

암호화 관리 방법	사용되는 키	
	IBM 암호화	**T10 암호화**
라이브러리에서 관리하는 암호화	단일 DK/카트리지	N/A
응용프로그램에서 관리하는 암호화	복수 DK/카트리지	복수 DK/카트리지
DK = 대칭 AES 256비트 DK		

제 2 장 Encryption Key Manager 환경 계획

이 섹션에서는 사용자 요구에 맞도록 Encryption Key Manager를 구성하는 데 도움이 되는 정보를 제공합니다. 암호화 전략을 설정하는 방법을 계획하려면 많은 요소를 고려해야 합니다.

암호화 설정 작업 개요

테이프 드라이브의 암호화 기능을 사용하려면 먼저 특정 소프트웨어 및 하드웨어 요구사항을 만족해야 합니다. 이러한 요구사항을 만족하는 데 도움이 되도록 다음 점검 목록이 제공됩니다.

Encryption Key Manager 설정 작업

테이프를 암호화하려면 먼저 암호화한 테이프 드라이브와 통신할 수 있도록 Encryption Key Manager를 구성 및 실행해야 합니다. 테이프 드라이브를 설치하는 동안에는 Encryption Key Manager를 실행할 필요가 없지만 암호화를 수행하려면 이를 실행해야 합니다.

- Encryption Key Manager 서버로 사용할 시스템 플랫폼을 결정하십시오.
- 필요한 경우 서버 운영 체제를 업그레이드하십시오. 2-2 페이지의 『하드웨어 및 소프트웨어 요구사항』을 참조하십시오.
- Java UR(Unrestricted) 정책 파일을 설치하십시오. 2-2 페이지의 『하드웨어 및 소프트웨어 요구사항』을 참조하십시오.
- Encryption Key Manager JAR을 업그레이드하십시오. 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.
- 키, 인증서 및 키 그룹을 작성하십시오.

  3-6 페이지의 『GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성』

  3-16 페이지의 『키 그룹 작성 및 관리』
- 추가 구성 옵션을 활용하지 않는 경우 3-6 페이지의 『GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성』의 절차를 따랐다면 이 단계는 필요하지 않습니다.
  - 필요한 경우 키 및 인증서를 가져오십시오. 3-14 페이지의 『Keytool -importseckey를 사용하여 데이터 키 가져오기』를 참조하십시오.
  - 구성 등록 정보 파일을 정의하십시오. 4-1 페이지의 제 4 장 『Encryption Key Manager 구성』을 참조하십시오.
  - Encryption Key Manager에 테이프 드라이브를 정의하거나 **drive.acceptUnknownDrives** 구성 등록 정보 값을 설정하십시오. 드라이브를 명

시적으로 정의하려면 5-10 페이지의 『adddrive』를 참조하거나 4-1 페이지의 『테이프 드라이브 테이블 자동 업데이트』를 참조하십시오.

– Encryption Key Manager 서버를 시작하십시오. 5-1 페이지의 『Key Manager 서버 시작, 새로 고침 및 중지』를 참조하십시오.

– 명령행 인터페이스 클라이언트를 시작하십시오. 5-6 페이지의 『명령행 인터페이스 클라이언트』를 참조하십시오.

라이브러리에서 관리하는 테이프 암호화 계획

암호화를 수행하려면 다음이 필요합니다.

- 암호화 가능 LTO 4 및 LTO 5 테이프 드라이브
- 키 저장소
- Dell Encryption Key Manager

라이브러리에서 관리하는 테이프 암호화 작업

1. LTO 4 및 LTO 5 테이프 드라이브를 설치하고 연결하십시오.
   - 라이브러리 펌웨어(필요한 경우 TL2000, TL4000, ML6000)를 업데이트하십시오. **http://supportapj.dell.com/support/index.aspx**을 방문하십시오.
     – Dell™ PowerVault™ TL2000 테이프 라이브러리 최소 필수 펌웨어 버전은 5.xx입니다.
     – Dell™ PowerVault™ TL4000 테이프 라이브러리 최소 필수 펌웨어 버전은 5.xx입니다.
     – Dell™ PowerVault™ ML6000 테이프 라이브러리 최소 필수 펌웨어 버전은 415G.xxx입니다.
   - 필요한 경우 테이프 드라이브 펌웨어를 업데이트하십시오. 최소 필수 펌웨어 버전은 77B5입니다.
2. 라이브러리에서 관리하는 테이프 암호화를 위해 LTO 4 및 LTO 5 테이프 드라이브 및 테이프 라이브러리를 설정하십시오(자세한 내용은 Dell 테이프 라이브러리 정보 참조).
   - Encryption Key Manager 서버 IP 주소를 추가합니다.
3. 라이브러리 진단 프로그램 기능을 사용하여 Encryption Key Manager 경로 및 암호화 구성을 확인하십시오(자세한 내용은 Dell 테이프 라이브러리 정보 참조).

하드웨어 및 소프트웨어 요구사항

주: 다음의 각 플랫폼에서는 JRE(Java Runtime Environment)의 IBM 버전만 Encryption Key Manager를 지원합니다.

Linux 솔루션 컴포넌트

운영 체제

- RHEL 4
- RHEL 5
- SLES 9
- SLES 10
- SLES 11

Encryption Key Manager(Linux에서 실행 중)

표 2-1. Linux에서의 최소 소프트웨어 요구사항

플랫폼	IBM Software Developer Kit	자원 웹 사이트
64비트 AMD/Opteron/EM64T	Java 6.0 SR5	**http://support.dell.com**
32비트 Intel® 호환		

테이프 라이브러리

Dell PowerVault TL2000 테이프 라이브러리, TL4000 테이프 라이브러리 및 ML6000 테이프 라이브러리의 펌웨어 레벨이 최신인지 확인하십시오. 펌웨어를 업데이트하려면 **http://support.dell.com**을 방문하십시오.

테이프 드라이브

LTO 4 및 LTO 5 테이프 드라이브의 펌웨어 레벨이 최신인지 확인하십시오. 펌웨어를 업데이트하려면 **http://support.dell.com**을 방문하십시오.

Windows 솔루션 컴포넌트

운영 체제

Windows Server 2003, 2008 및 2008 R2

Dell Encryption Key Manager

Encryption Key Manager의 최소 필수 버전은 빌드 날짜가 2007년 9월 14일(20070914) 이후인 2.1 버전이며, 다음 IBM Runtime Environment 중 하나입니다.

표 2-2. Windows에서의 최소 소프트웨어 요구사항

운영 체제	IBM Runtime Environment
Windows 2003	• AMD64/EM64T 구조의 Windows에 대한 IBM® 64비트 Runtime Environment, Java 2 Technology Edition, 버전 5.0 SR5 • Windows에 대한 IBM 32비트 Runtime Environment, Java 2 Technology Edition, 버전 5.0 SR5

표 2-2. Windows에서의 최소 소프트웨어 요구사항 (계속)

운영 체제	IBM Runtime Environment
Windows 2008 및 2008 R2	AMD64/EM64T 구조의 Windows에 대한 IBM 64비트 Runtime Environment, Java 2 Technology Edition, 버전 6.0 SR5

테이프 라이브러리

Dell™ PowerVault™ TL2000 테이프 라이브러리, Dell™ PowerVault™ TL4000 테이프 라이브러리 및 Dell™ PowerVault™ ML6000 테이프 라이브러리의 펌웨어 레벨이 최신인지 확인하십시오. 펌웨어를 업데이트하려면 **http://support.dell.com**을 방문하십시오.

테이프 드라이브

LTO 4 및 LTO 5 테이프 드라이브의 펌웨어 레벨이 최신인지 확인하십시오. 펌웨어를 업데이트하려면 **http://support.dell.com**을 방문하십시오.

키 저장소 고려사항

키 저장소 데이터를 보존하는 작업은 매우 중요합니다. 키 저장소에 액세스하지 않으면 암호화된 테이프의 암호를 해독할 수 없습니다. 아래 주제를 자세히 읽고 키 저장소 데이터를 보호하는 데 사용할 수 있는 방법을 검토하십시오.

JCEKS 키 저장소

EKM에서는 JCEKS 키 저장소 유형을 지원합니다.

JCEKS(Unix System Services 파일 기반) 는 EKM을 실행하는 모든 플랫폼에서 지원되는 파일 기반 키 저장소입니다. 따라서 상대적으로 백업 및 복구 시 이 키 저장소 내용을 복사하고 장애 복구 시 두 EKM 인스턴스를 동기화하는 작업이 용이합니다. JCEKS는 보안을 위해 키 저장소 내용에 대한 암호 기반 보호를 제공하며 상대적으로 양호한 성능을 제공합니다. FTP와 같은 파일 복사 방법을 사용할 수도 있습니다.

암호화 키 및 LTO 4 및 LTO 5 테이프 드라이브

Dell Encryption Key Manager 및 모든 지원되는 테이프 드라이브는 데이터를 암호화할 때 대칭 256비트 AES 키를 사용합니다. 이 주제에서는 이러한 키 및 인증서와 관련하여 알아야 하는 정보를 설명합니다.

LTO 테이프 카트리지의 LTO 4 또는 LTO 5 테이프 드라이브에서 암호화 작업을 수행하는 경우 Encryption Key Manager에서는 256비트 AES 대칭 데이터 키만 사용합니다.

LTO 4 또는 LTO 5가 키를 요청하면 Encryption Key Manager에서는 테이프 드라이브에 지정된 별명을 사용합니다. 테이프 드라이브에 지정된 별명이 없으면 symmetricKeySet 구성 등록 정보에 지정된 키 별명 범위, 키 별명 목록 또는 키 그룹의 별명이 사용됩니다. 테이프 드라이브에 대한 특정 별명이 부족하면 키를 균등하게 사용하기 위해 차례로 다른 엔티티에서 별명을 선택합니다.

선택된 별명은 미리 키 저장소에 로드된 대칭 데이터 키(DK)와 연관됩니다. Encryption Key Manager에서는 테이프 드라이브가 암호 해독할 수 있는 다른 키를 사용하여 랩핑된 이 DK를 LTO 4 또는 LTO 5 테이프 드라이브에 전송하여 데이터를 암호화합니다. DK는 일반 텍스트 형식으로 TCP/IP를 통해 전송되지 않습니다. 선택된 별명은 데이터 키 ID(DKi)라고 하는 엔티티로 변환되고 이는 암호화된 데이터를 사용하여 테이프에 기록됩니다. 이 방식을 사용하면 Encryption Key Manager에서는 DKi를 사용하여 LTO 4 또는 LTO 5 테이프를 읽을 때 데이터 암호를 해독하는 데 필요한 올바른 DK를 식별할 수 있습니다.

5-9 페이지의 『CLI 명령』의 **adddrive** 및 **moddrive** 주제에서는 테이프 드라이브에 대한 별명을 지정하는 방법을 보여줍니다. symmetricKeySet 구성 등록 정보에서 키 가져오기, 키 내보내기 및 기본 별명 지정에 관한 정보는 3-11 페이지의 『LTO 4 및 LTO 5에서 암호화에 대한 키 및 별명 생성』을 참조하십시오. 3-16 페이지의 『키 그룹 작성 및 관리』에서는 키 그룹을 정의하고 이를 키 저장소의 별명으로 채우는 방법을 보여줍니다.

그림 2-1에서는 암호화된 쓰기 조작에서 키를 처리하는 방법을 보여줍니다.

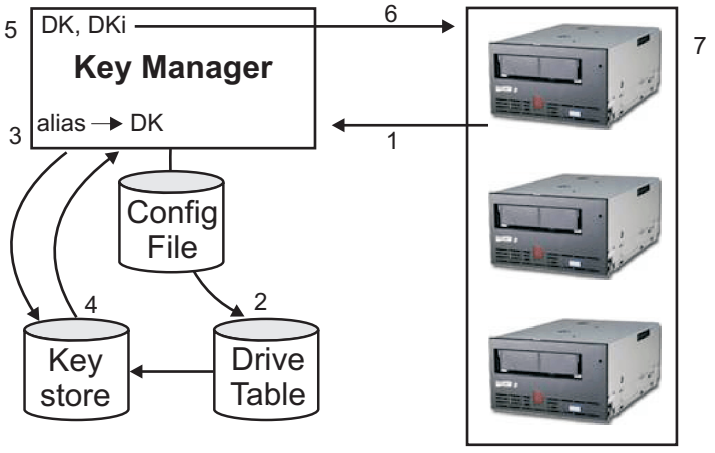

그림 2-1. LTO 4 또는 LTO 5 테이프 드라이브의 암호화 쓰기 조작 요청

1. 테이프 드라이브에서 테이프를 암호화하는 키를 요청합니다.
2. Encryption Key Manager는 드라이브 테이블의 테이프 장치를 확인합니다.

3. 요청에서 별명을 지정하지 않고 드라이브 테이블에 지정된 별명이 없으면 Encryption Key Manager는 keyAliasList에 있는 키 그룹 또는 별명 세트에서 별명을 선택합니다.
4. Encryption Key Manager는 이에 대응하는 DK를 키 저장소에서 가져옵니다.
5. Encryption Key Manager는 별명을 DKi로 변환하고 드라이브가 암호 해독할 수 있는 키를 사용하여 DK를 랩핑합니다.
6. Encryption Key Manager는 DK 및 DKi를 테이프 드라이브에 전송합니다.
7. 테이프 드라이브가 DK를 랩핑 해제하고 암호화된 데이터 및 DKi를 테이프에 씁니다.

그림 2-2에서는 암호화된 읽기 조작에서 키를 처리하는 방법을 보여줍니다.

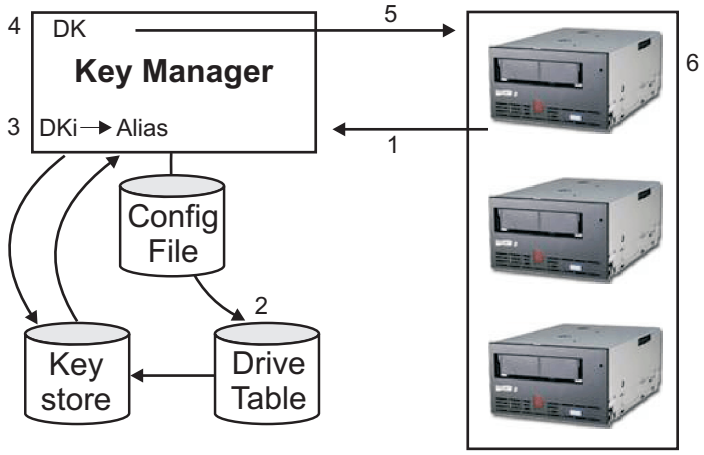

그림 2-2. LTO 4 또는 LTO 5 테이프 드라이브의 암호화 읽기 조작 요청

1. 테이프 드라이브는 읽기 요청을 수신하며 Encryption Key Manager에 DKi를 전송합니다.
2. Encryption Key Manager는 드라이브 테이블의 테이프 장치를 확인합니다.
3. Encryption Key Manager는 DKi를 별명으로 변환하고 이에 대응하는 DK를 키 저장소에서 가져옵니다.
4. Encryption Key Manager는 드라이브가 암호 해독할 수 있는 키를 사용하여 DK를 랩핑합니다.
5. Encryption Key Manager는 랩핑된 DK를 테이프 드라이브에 전송합니다.
6. 테이프 드라이브가 DK를 랩핑 해제하고 이를 사용하여 데이터 암호를 해독합니다.

키 저장소 데이터 백업

주: 키 저장소에 있는 키의 중요한 특징 때문에 암호화되지 않은 장치에서 이 데이터를 백업하는 것이 중요합니다. 그래야만 필요한 경우 이를 복구하여 해당 테이프 드라

이브 또는 라이브러리와 연관된 해당 인증서를 사용하여 암호화된 테이프를 읽을 수 있습니다. 키 저장소 백업에 실패하면 암호화된 데이터에 대한 모든 액세스를 유실할 수 있습니다.

이 키 저장소 정보를 백업하는 데에는 여러가지 방법이 있습니다. 각 키 저장소 유형마다 고유한 특징이 있습니다. 다음의 일반 지침은 모든 경우에 적용됩니다.

- 키 저장소로 로드한 인증서(보통 PKCS12 형식 파일) 모두의 사본을 보관합니다.
- RACF와 같은 시스템 백업 기능을 사용하여 키 저장소 정보의 백업 사본을 작성합니다. 암호화한 테이프 드라이브를 사용하여 이 사본을 암호화하지 마십시오. 그러면 복구 시 암호를 해독할 수 없습니다.
- 백업 및 장애 복구 시 여유 자원으로 기본 및 보조 Encryption Key Manager 및 키 저장소 사본을 유지보수합니다. 추가된 여유 자원으로 기본 및 보조 모두에서 키 저장소를 백업합니다.
- JCEKS 키 저장소의 경우 키 저장소 파일을 복사하고 볼트와 같은 안전한 위치에 암호화하지 않은 일반 사본을 저장합니다. 암호화한 테이프 드라이브를 사용하여 이 사본을 암호화하지 마십시오. 그러면 복구 시 암호를 해독할 수 없습니다.

최소한 변경할 때마다 키 저장소를 백업해야 합니다. Encryption Key Manager에서는 키 저장소 데이터를 수정하지 않습니다. 키 저장소가 변경되는 경우는 사용자가 변경할 때뿐이므로 사용자가 키 저장소를 변경하면 바로 키 저장소를 복사해야 합니다.

GUI를 사용하여 파일 백업

1. GUI를 아직 시작하지 않은 경우 GUI를 여십시오.

   Windows의 경우

   c:₩ekm₩gui를 탐색하여 **LaunchEKMGui.bat**를 클릭하십시오.

   Linux 플랫폼의 경우

   /var/ekm/gui를 탐색하여 `. ./LaunchEKMGui.sh`를 입력하십시오.

2. Encryption Key Manager GUI 왼쪽에 있는 탐색기에서 **중요 파일 백업(Backup Critical Files)**을 선택하십시오.
3. 표시된 대화 상자에 백업 데이터에 대한 경로를 입력하십시오(2-8 페이지의 그림 2-3).

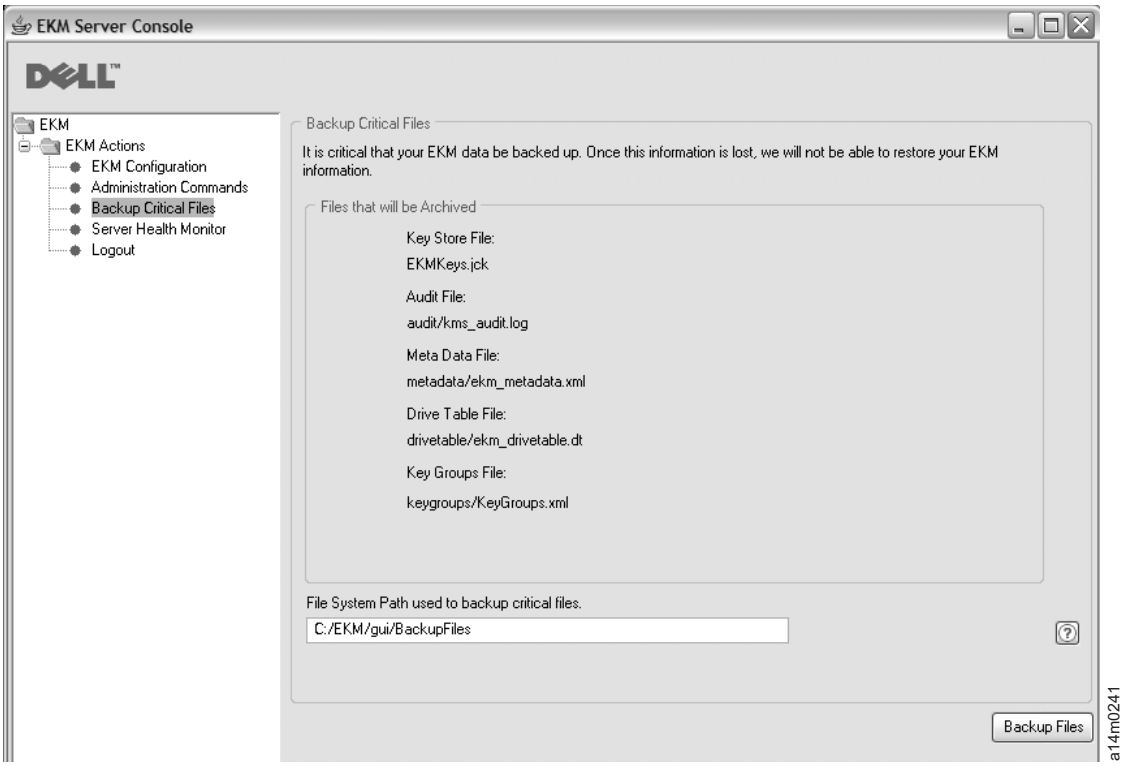

그림 2-3. 중요 파일 백업 창

4. **파일 백업(Backup Files)**을 클릭하십시오.
5. 정보 메시지에 결과가 표시됩니다.

백업용으로 여러 개의 Key Manager 지원

Encryption Key Manager는 테이프 드라이브 및 라이브러리에서 여유 자원을 사용하고 고가용성을 보장하도록 설계되었습니다. 따라서 같은 테이프 드라이브 및 라이브러리를 지원하는 여러 개의 Key Manager를 사용할 수 있습니다. 이러한 Key Manager가 테이프 드라이브 및 라이브러리와 같은 시스템에 있을 필요는 없습니다. 최대 Key Manager 수는 라이브러리 또는 프록시에 따라 다릅니다. TCP/IP 연결을 통해 테이프 드라이브에서 사용할 수 있기만 하면 됩니다.

이를 통해 한 개의 Key Manager가 사용 불가능한 경우, 장애 복구 조치는 물론 키 저장소의 중요한 정보의 기본 백업을 제공하는 서로에 대한 미러 이미지로 두 개의 Encryption Key Manager 사용이 가능합니다. 장치(또는 프록시)를 구성하는 경우 두 개의 Key Manager를 가리킬 수 있습니다. 한 Key Manager가 사용 불가능한 경우에 장치(또는 라이브러리)는 대체 Key Manager를 사용합니다.

또한 두 개의 Encryption Key Manager를 동기화하는 기능도 제공됩니다. 필요한 경우 테이프 조작의 중단을 방지하기 위해 장애 복구 기능 및 중요한 데이터의 기본 백업 수단으로 이 기능을 활용하는 것이 중요합니다. 4-2 페이지의 『두 개의 Key Manager 서버 사이에서 데이터 동기화』를 참조하십시오.

주: 동기화에서는 키 저장소를 포함하지 않습니다. 수동으로 복사해야 합니다.

Encryption Key Manager 서버 구성

Encryption Key Manager는 단일 서버 또는 여러 대의 서버에 설치할 수 있습니다. 다음 예제에서는 하나의 Key Manager 및 두 개의 Key Manager 구성을 보여주지만 라이브러리에서는 그 이상을 허용할 수도 있습니다.

단일 서버 구성

단일 서버 구성(그림 2-4 참조)은 가장 단순한 형태의 Encryption Key Manager 구성입니다. 그러나 여유 자원이 부족하기 때문에 권장하지는 않습니다. 이 구성에서는 모든 테이프 드라이브가 백업 없이 하나의 Key Manager 서버에 의존합니다. 서버가 중단되면 키 저장소, 구성 파일, KeyGroups.xml 파일 및 드라이브 테이블을 사용할 수 없으므로 암호화된 테이프를 읽을 수 없습니다. 단일 서버 구성에서는 서버 사본이 유실된 경우에도 키 저장소, 구성 파일, KeyGroups.xml 파일 및 드라이브 테이블이 대체 서버에서 해당 기능을 다시 빌드할 수 있도록 Encryption Key Manager 이외의 안전한 위치에서 유지보수되어야 합니다.

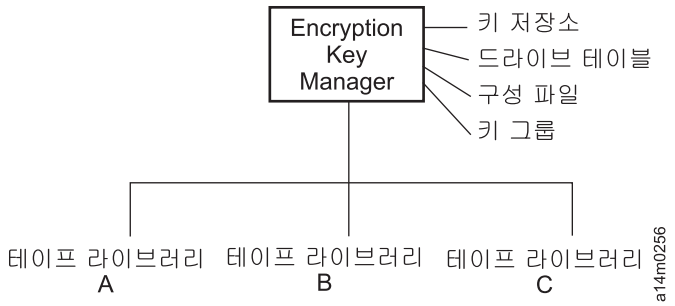

그림 2-4. 단일 서버 구성

두 개의 서버 구성

두 개의 서버 구성은 권장 사항입니다. Encryption Key Manager 구성은 어떤 이유에서든 기본 Encryption Key Manager에 액세스할 수 없는 경우에 장애 복구 조치로 보조 Encryption Key Manager를 사용합니다.

주: 다른 Encryption Key Manager 서버를 사용하여 같은 테이프 드라이브 세트의 요청을 처리하는 경우 연관된 키 저장소의 정보는 서로 같아야 합니다. 그래야만 연결된 Key Manager 서버와 상관없이 테이프 드라이브의 요청을 지원하는 데 필요한 정보를 사용할 수 있습니다.

동일 구성: 두 Encryption Key Manager 서버의 구성이 서로 같은 환경(예: 2-10 페이지의 그림 2-5 참조)에서 기본 서버가 중단되는 경우, 자동으로 장애 복구가 보조 Key Manager에서 처리됩니다. 이러한 구성에서는 두 Key Manager 서버를 동기화하는 것

이 매우 중요합니다. 한 Key Manager 서버의 구성 파일 및 드라이브 테이블에서 업데이트된 사항은 **sync** 명령을 사용하여 자동적으로 다른 서버에 복제될 수 있습니다. 그러나 한 키 저장소에서 업데이트된 사항은 사용 중인 키 저장소에 특정한 방법을 사용하여 다른 키 저장소로 복사해야 합니다. 키 저장소 및 키 그룹 XML 파일을 수동으로 복사해야 합니다. 자세한 정보는 4-2 페이지의 『두 개의 Key Manager 서버 사이에서 데이터 동기화』를 참조하십시오.

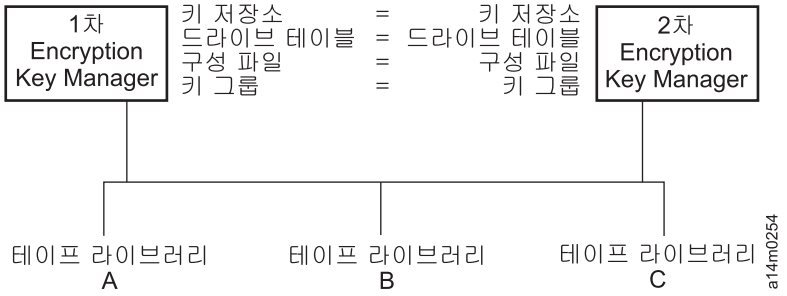

그림 2-5. 구성을 공유하는 두 개의 서버

별도 구성: 두 개의 Encryption Key Manager 서버가 하나의 공통 키 저장소를 공유할 수 있습니다. 그러나 드라이브 테이블에는 두 개의 서로 다른 구성 파일과 해당 XML 파일에 정의된 두 개의 서로 다른 키 그룹 세트가 포함되어 있습니다. 공통 테이프 드라이브를 지원하는 데 사용되는 키가 각 서버에서 동일하기만 하면 가능합니다. 또한 각 Key Manager 서버에서 고유한 등록 정보 세트를 보유할 수도 있습니다. 이러한 유형의 구성(그림 2-6 참조)에서는 Key Manager 서버 사이에서 드라이브 테이블만 동기화해야 합니다. 자세한 정보는 4-2 페이지의 『두 개의 Key Manager 서버 사이에서 데이터 동기화』를 참조하십시오. 구성 파일을 겹쳐쓰지 않도록 하려면 `sync.type = drivetab`을 지정해야 합니다. 구성 파일 또는 모두를 지정하지 않도록 하십시오.

주: 서버 사이에서 구성을 부분적으로 공유하는 방법은 없습니다.

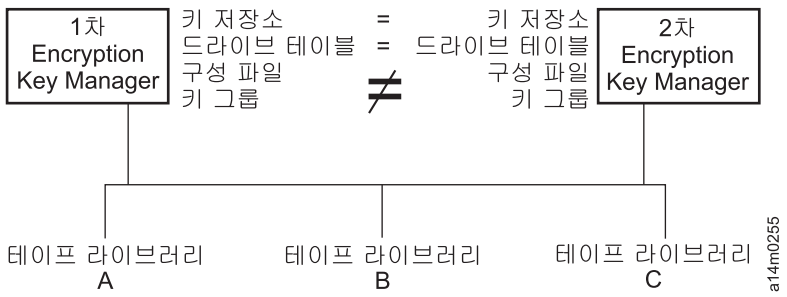

그림 2-6. 같은 장치에 액세스하는 서로 다른 구성의 두 개의 서버

장애 복구 사이트 고려사항

장애 복구(DR) 사이트를 사용하려는 경우 Encryption Key Manager에서는 해당 사이트에서 암호화된 테이프를 읽고 쓸 수 있도록 지원하는 많은 옵션을 제공합니다. 옵션은 다음과 같습니다.

- DR 사이트에서 중복 Encryption Key Manager 항목을 작성합니다.

  DR 사이트에서 로컬 Encryption Key Manager 항목과 같은 정보(구성 파일, 테이프 드라이브 테이블, 키 그룹 XML 파일 및 키 저장소)를 사용하는 중복 Encryption Key Manager 항목을 설정합니다. 그러면 이 Key Manager는 해당 위치에서 암호화된 테이프를 읽고 쓰던 기존의 제품 Key Manager 중 하나를 대신할 수 있습니다.

- 필요한 경우 복구할 수 있도록 세 개의 Encryption Key Manager 데이터 파일 백업 사본을 작성합니다.

  Encryption Key Manager에 필요한 네 개의 데이터 요소(구성 파일, 테이프 드라이브 테이블, 키 그룹 XML 파일 및 키 저장소)의 현재 사본을 작성하면 언제라도 Key Manager를 시작하여 DR 사이트에서 중복 항목의 역할을 수행할 수 있습니다. 기능 Key Manager 없이 암호를 해독할 수 없는 경우에는 이러한 파일 사본을 암호화하는 데 Encryption Key Manager 사용을 권장하지 않습니다. DR 사이트에서 기본 사이트와 다른 테이프 드라이브를 사용하는 경우 구성 파일 및 테이프 드라이브 테이블에는 DR 사이트에 대한 올바른 정보가 포함되어야 합니다.

오프사이트에서 암호화된 테이프를 공유하는 경우 고려사항

주: 인증서의 신뢰 관계를 처음으로 추적하여 궁극적으로 인증서에 서명한 인증 권한(CA)부터 확인하는 작업을 통해 비즈니스 파트너로부터 받은 인증서의 유효성을 확인하는 것이 중요합니다. CA를 신뢰하는 경우 해당 인증서도 신뢰할 수 있습니다. 또는 인증서의 이동 경로가 안전하게 보안된 경우 인증서의 유효성을 확인할 수 있습니다. 이러한 방법을 사용하여 인증서의 유효성을 확인하지 못하면 『MITM(Man-in-the-Middle)』 공격에 노출되었을 수 있습니다.

LTO 4 및 LTO 5 테이프 공유

LTO 4 또는 LTO 5 테이프에서 암호화된 데이터를 공유하려면 테이프에서 데이터를 암호화하는 데 사용되는 대칭 키 사본을 다른 조직에서 사용할 수 있어야만 테이프를 읽을 수 있습니다. 대칭 키를 공유하려면 다른 조직에서 자신의 공용 키를 사용자 조직과 공유해야 합니다. keytool을 사용하여 Encryption Key Manager 키 저장소에서 대칭 키를 내보내는 경우 대칭 키를 랩핑하는 데 이 공용 키를 사용합니다(3-14 페이지의 『Keytool -exportseckey를 사용하여 데이터 키 내보내기』 참조). 다른 조직에서 대칭 키를 자신의 Encryption Key Manager 키 저장소로 가져오는 경우 이에 대응하는

개인용 키를 사용하여 랩핑 해제됩니다(3-14 페이지의 『Keytool -importseckey를 사용하여 데이터 키 가져오기』 참조). 그러면 개인용 키의 보유자만 대칭 키를 랩핑 해제할 수 있으므로 전송 시 대칭 키를 안전하게 보안할 수 있습니다. 자신의 Encryption Key Manager 키 저장소에서 데이터를 암호화하는 데 사용되는 대칭 키를 사용하면 다른 조직도 테이프의 데이터를 읽을 수 있습니다.

FIPS(Federal Information Processing Standard) 140-2 고려사항

현재 FIPS(Federal Information Processing Standard) 140-2는 연방 정부가 모든 암호 제공자에게 FIPS 140을 인증하도록 규정하면서 매우 중요해졌습니다. 점차 늘어나는 민간 부문의 커뮤니티에서도 이 기준을 채택하고 있습니다. 정부 기준에 따라 타사에 의해 암호화 기능을 인증하는 이 제도는 오늘날 보안이 중요하게 부각된 현실에서 그 가치를 더하고 있습니다.

Encryption Key Manager에서는 자체적으로 암호화 기능을 제공하지 않습니다. 따라서 FIPS 140-2 인증이 필요하거나 이를 확보하지 않아도 됩니다. 그러나 Encryption Key Manager에서는 IBM Java Cryptographic Extension 컴포넌트의 IBM JVM에서 제공하는 암호화 기능의 장점을 활용하고 FIPS 140-2 레벨 1 인증을 보유한 IBMJCEFIPS 암호 제공자를 선택하여 사용할 수 있습니다. 구성 등록 정보 파일에서 **fips** 구성 매개변수를 **on**으로 설정하면 Encryption Key Manager에서 모든 암호 기능에 대한 IBMJCEFIPS 제공자를 사용할 수 있습니다.

제품이 FIPS 140-2 인증을 받았는지 확인하려면 특정 하드웨어 및 소프트웨어 암호 제공자 문서를 참조하십시오.

제 3 장 Encryption Key Manager 및 키 저장소 설치

Encryption Key Manager는 IBM Java Virtual Machine 설치와 함께 제공되며 Linux용 IBM Software Developer Kit 및 Windows용 IBM Runtime Environment이 필요합니다(2-2 페이지의 『하드웨어 및 소프트웨어 요구사항』 참조). 다음 중 사용자 운영 체제에 맞는 절차를 따르십시오.

- 『Linux에 Encryption Key Manager 설치』
- 3-3 페이지의 『Windows에 Encryption Key Manager 설치』

Encryption Key Manager가 최신 버전인지 불확실한 경우, 『최신 버전 Key Manager ISO 이미지 다운로드』에서 새 버전이 사용 가능한지 판별하는 방법을 설명합니다. Java 설치에 포함되지 않을 수도 있으므로 최신 Encryption Key Manager 버전을 설치하는 것이 좋습니다. 자세한 정보는 **http://support.dell.com**을 방문하십시오.

중요한 Encryption Key Manager 호스트 서버 구성 정보: Dell Encryption Key Manager 프로그램을 호스트하는 시스템은 데이터 유실의 위험을 최소화하기 위해 ECC 메모리를 사용할 것을 권장합니다. Encryption Key Manager는 암호화 키의 생성을 요청하고 이들 키를 LTO 4 및 LTO 5 테이프 드라이브에 전달하는 기능을 수행합니다. 랩된(암호화된) 키는 처리 중 Encryption Key Manager가 시스템 메모리에 배치합니다. 카트리지에 기록된 데이터를 복구(암호 해독)할 수 있으려면 오류 없이 키를 적절한 테이프 드라이브에 전송해야 합니다. 시스템 메모리의 비트 오류로 키가 손상되었고 카트리지에 데이터를 쓸 때 이 키를 사용하는 경우, 해당 카트리지에 쓰여진 데이터는 복구할 수 없습니다. (즉, 차후에 암호를 해독할 수 없습니다.) 이러한 데이터 오류가 발생하지 않도록 하는 보호 장치가 마련되어 있습니다. 그러나 Encryption Key Manager를 호스트하는 시스템이 ECC(Error Correction Code) 메모리를 사용하지 않는 경우, 시스템 메모리에 있는 동안 키가 손상되고 이러한 손상으로 데이터 유실이 발생할 수 있는 가능성은 남아 있습니다. 이와 같은 데이터 유실 가능성은 적지만 중요 응용프로그램(예: Encryption Key Manager)을 호스트하는 시스템은 ECC 메모리를 사용할 것을 권장합니다.

최신 버전 Key Manager ISO 이미지 다운로드

Dell ISO 이미지 최신 버전을 다운로드하려면 http://support.dell.com으로 이동하십시오.

Linux에 Encryption Key Manager 설치

CD를 사용하여 Linux에 Encryption Key Manager 설치

1. Dell Encryption Key Manager CD를 삽입하거나 CD 루트 디렉토리에서 `Install_Linux`를 입력하십시오.

설치 프로그램이 CD에서 하드 드라이브로 운영 체제에 적합한 모든 내용(문서, GUI 파일 및 구성 등록 정보 파일)을 복사합니다. 설치 중 시스템에서는 올바른 IBM JRE(Java Runtime Environment)가 있는지 확인합니다. 발견되지 않으면 자동으로 설치됩니다.

설치가 완료되면 GUI(Graphical User Interface)가 시작됩니다.

Linux에서 수동으로 Software Developer Kit 설치

CD에서 설치하지 않은 경우 다음 단계를 수행하십시오.

1. **http://support.dell.com**에서 운영 체제에 기반한 올바른 Java Runtime Environment를 다운로드하십시오.
   - Java 6 SR 5(32비트) 이상
   - Java 6 SR 5(64비트) 이상
2. 작업 디렉토리에 Java linux rpm 파일을 배치하십시오.

```
mordor:~ #/tape/Encryption/java/1.6.0# pwd
/tape/Encryption/java/1.6.0
mordor:~ #/tape/Encryption/java/1.6.0# ls
ibm-java-i386-jre-6.0-5.0.i386.rpm
```

3. rpm 패키지를 설치하십시오.

```
mordor:~ #rpm -ivh -nodeps ibm-java-i386-jre-6.0-5.0.i386.rpm
```

그러면 **/opt/ibm/java-i386-60/** 디렉토리에 파일이 배치됩니다.

```
mordor:~ #/opt/ibm/java-i386-60/jre # ls
.systemPrefs  bin  javaws  lib
```

4. 설치한 Java에 대한 JAVA_HOME, CLASSPATH 및 bin 디렉토리를 포함하는 **/etc/profile.local** 파일을 편집하십시오(필요한 경우 작성함). 다음의 세 행을 추가하십시오.

```
JAVA_HOME=/opt/ibm/java-i386-60/jre
CLASSPATH=/opt/ibm/java-i386-60/jre/lib
PATH=$JAVA_HOME:opt/ibm/java-i386-60/jre/bin/:$PATH
```

5. **/etc/profile.local** 항목을 적용하려면 로그아웃한 후 다시 로그인하거나 export 명령행 명령을 실행하십시오.

```
mordor:~ # export JAVA_HOME=/opt/ibm/java-i386-60/jre
mordor:~ # export CLASSPATH=/opt/ibm/java-i386-60/jre/lib
mordor:~ # export PATH=/opt/ibm/java-i386-60/jre/bin/:$PATH
```

6. 다시 로그인한 후 **java -version** 명령을 실행하십시오. 다음과 같은 결과가 표시됩니다.

```
mordor:~ # java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pmz60sr5-20090529(SR5))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Linux x86-32 jvmxi3260-20090519_35743 (JIT enabled)
...
mordor:~ # which java
/opt/ibm/java-i386-60/jre/bin/java
```

Windows에 Encryption Key Manager 설치

1. Dell Encryption Key Manager CD를 삽입하십시오.

   설치 프로그램이 CD에서 하드 드라이브로 운영 체제에 적합한 모든 내용(문서, GUI 파일 및 구성 등록 정보 파일)을 복사합니다. 설치 중 시스템에서는 올바른 IBM JRE(Java Runtime Environment)가 있는지 확인합니다. 발견되지 않으면 자동으로 설치됩니다.

   설치가 완료되면 GUI(Graphical User Interface)가 시작됩니다.
2. InstallShield 마법사가 열리면 **다음(Next)**을 클릭하십시오.
3. 라이센스 계약을 읽고 **예(Yes)**를 클릭하십시오.
4. 대상 위치 선택(Choose Destination Location) 창이 열리면(그림 3-1) 폴더를 선택하고 이를 기록해두십시오. Encryption Key Manager 실행 시 이 Java 경로가 필요합니다.

그림 3-1. 대상 위치 선택(Choose Destination Location) 창

다음(Next)을 클릭하십시오.

5. 이 Java Runtime Environment를 기본 시스템 JVM(그림 3-2)으로 사용할 것인지 묻는 창이 열립니다.

그림 3-2. 기본값으로 이 JVM 버전 설정

아니오(No)를 클릭하십시오.

6. 파일 복사 시작(Start Copying Files) 창이 열립니다(그림 3-3). 대상 디렉토리를 기록해두어야 합니다.

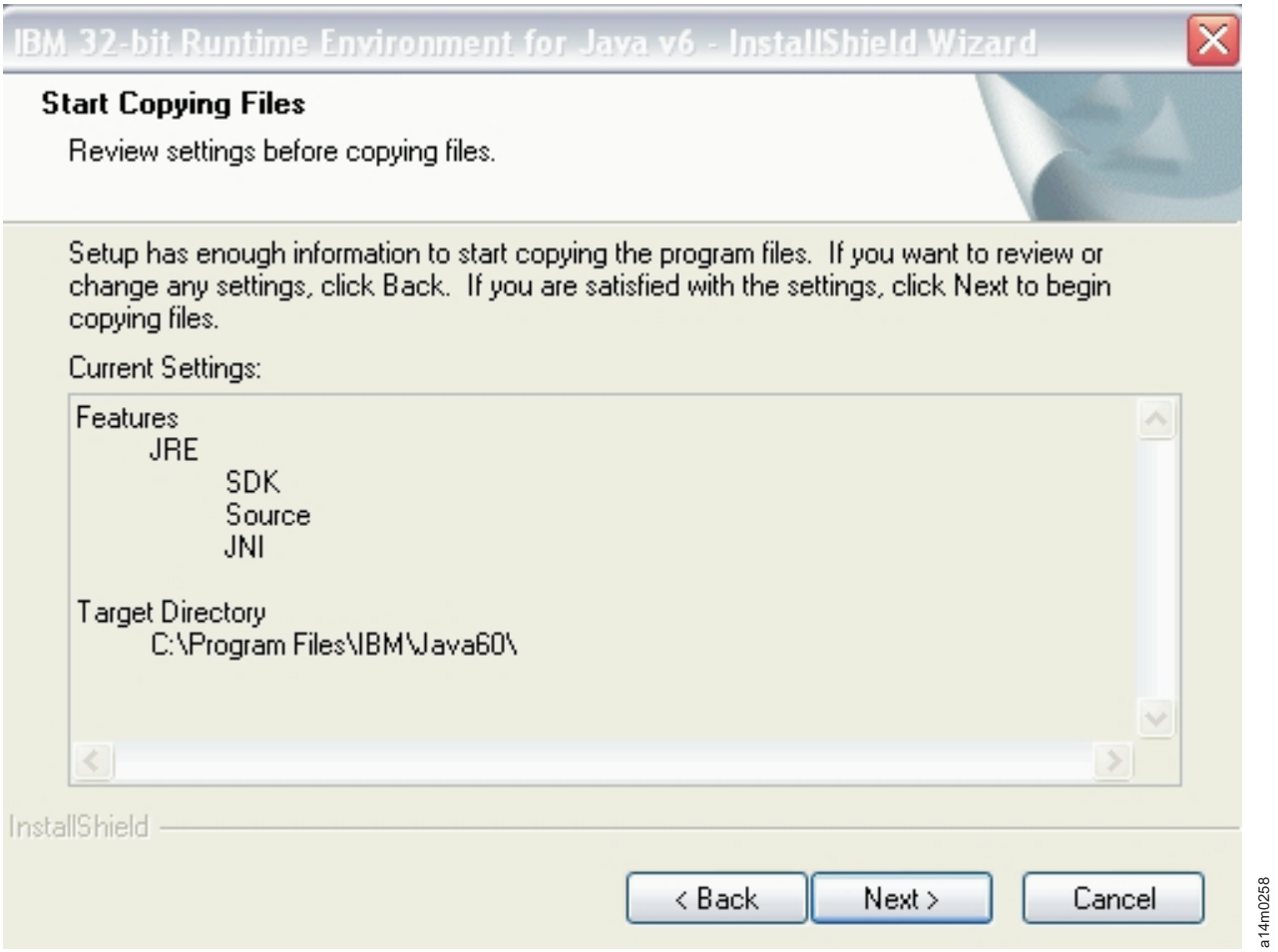

그림 3-3. 파일 복사 시작 창

다음(Next)을 클릭하십시오.

7. 상태 창에서 설치 진행을 표시합니다.
8. 브라우저 등록(Browser Registration) 창이 열립니다. Encryption Key Manager에서 사용할 브라우저를 선택하십시오. **다음(Next)**을 클릭하십시오.
9. InstallShield 마법사 완료(InstallShield Wizard Complete) 창이 열리면 **완료(Finish)**를 클릭하십시오.

   설치 후 다음과 같이 명령 프롬프트를 열어 설치된 Java 버전을 조회할 수 있습니다.

```
C:₩WinEKM>C:₩"Program Files"₩IBM₩Java60₩jre₩bin₩java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pwi3260sr5-20090529_04(SR5))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Windows Server 2003 x86-32 j9vmwi3223-20090
519_35743 (JIT enabled, AOT enabled)
...
```

10. 다음과 같이 PATH 변수를 갱신하십시오(Encryption Key Manager 2.1에는 필수이나 빌드 날짜가 05032007 이전인 경우에는 선택적임).

    명령 창에서 Java SDK를 호출하려는 경우 명령의 전체 경로를 입력하지 않고도 임의의 디렉토리에서 Java JRE 실행 파일(java.exe)을 실행하도록 PATH 변수를 설정할 수 있습니다. PATH 변수를 설정하지 않으면 실행 파일을 실행할 때마다 다음과 같이 전체 경로를 지정해야 합니다.

```
C:>₩Program Files₩IBM₩Java60₩jre₩bin₩java ...
```

    영구적으로 PATH를 설정하려면(Encryption Key Manager 2.1에는 필수) PATH 변수에 Java 바이너리 디렉토리의 전체 경로를 추가하십시오. 보통 이 전체 경로는 다음과 비슷합니다.

```
C:₩Program Files₩IBM₩Java60₩jre₩bin
```

    Microsoft Windows 2003, 2008 및 2008 R2에서 영구적으로 PATH를 설정하려면 다음을 수행하십시오.

    주: 명령행에서 PATH 변수를 설정하면 작동하지 않습니다.

    a. 시작 메뉴에서 **설정**을 선택한 후 **제어판**을 선택하십시오.
    b. **시스템**을 두 번 클릭하십시오.
    c. **고급** 탭을 클릭하십시오.
    d. **환경 변수**를 클릭하십시오.
    e. 시스템 변수 목록을 스크롤 다운하여 경로 변수로 이동한 다음 **편집**을 클릭하십시오.
    f. 경로 변수의 시작 부분에 IBM JVM 경로를 추가하십시오.

       기본 설치 디렉토리는 C:₩PROGRA~1₩IBM₩Java60₩jre₩bin입니다.

중요: 경로 목록의 다른 디렉토리와 구분할 수 있도록 경로의 끝에 세미콜론을 삽입하십시오.

g. **확인**을 클릭하십시오.

GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성

Encryption Key Manager를 시작하기 전에 먼저 각각 하나 이상의 키 저장소 및 자체 서명된 인증서를 작성해야 합니다. Dell Encryption Key Manager 서버 GUI(Graphical User Interface)를 사용하면 Encryption Key Manager 구성 등록 정보 파일, 키 저장소, 인증서 및 키를 작성할 수 있습니다. 이 프로세스를 수행하면 단순한 CLI 구성 등록 정보 파일도 작성됩니다.

1. GUI를 아직 시작하지 않은 경우 GUI를 여십시오.

   Windows의 경우

   c:₩ekm₩gui를 탐색하여 **LaunchEKMGui.bat**를 클릭하십시오.

   Linux 플랫폼의 경우

   /var/ekm/gui를 탐색하여 `. ./LaunchEKMGui.sh`를 입력하십시오.

2. GUI 왼쪽에 있는 탐색기에서 **EKM 구성(EKM Configuration)**을 선택하십시오.
3. 『EKM 서버 구성(EKM Server Configuratio)』 페이지(3-7 페이지의 그림 3-4)에서 별표(*)로 표시된 모든 필수 필드에 데이터를 입력하십시오. 편의를 위해 일부 필드는 작성되어 있습니다. 필드에 대한 설명을 보려면 데이터 필드 오른쪽에 있는 물음표를 클릭하십시오. **다음(Next)**을 클릭하십시오.

   주: 키 저장소 암호를 설정하고 나면 보안 위반이 발생하지 않는 한 **변경하지 마십시오**. 보안 노출 위험을 없애기 위해 암호는 인식하기 어렵게 만들어집니다. 키 저장소 암호를 변경하려면 **keytool** 명령을 사용하여 개별적으로 해당 키 저장소의 모든 암호를 변경해야 합니다. 3-14 페이지의 『키 저장소 암호 변경』을 참조하십시오.

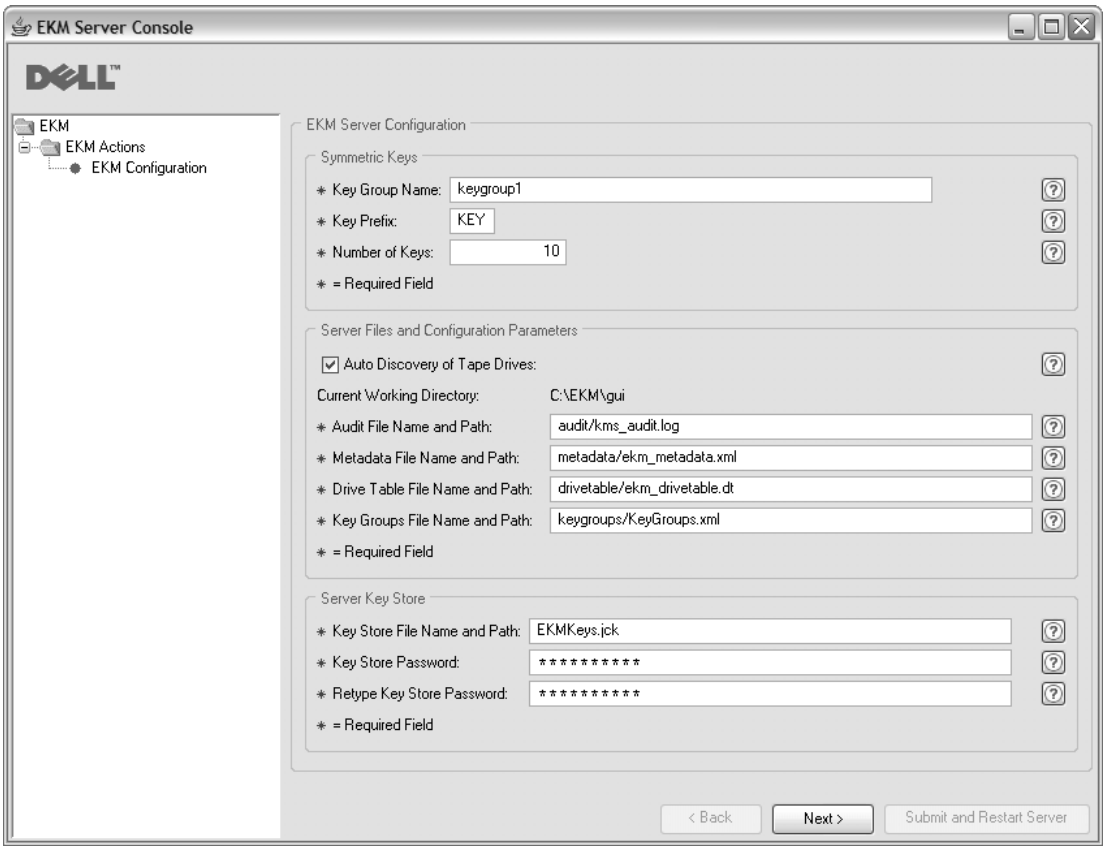

그림 3-4. EKM 서버 구성(EKM Server Configuration) 페이지

Dell Encryption Key Manager 키 저장소용으로 생성할 수 있는 키의 수에 제한이 없다고 하더라도 요청되는 키의 수에 따라 키 생성에 드는 시간이 증가합니다. Encryption Key Manager에서 10개의 키를 생성하는 데에 15초가 소요되며 10000개의 키를 생성하는 데에는 30분 이상이 소요됩니다. 키의 수는 호스트 서버 자원(서버의 메모리)에 의해 제한된다는 점에 유의하십시오. Encryption Key Manager 응용프로그램은 라이브러리가 드라이브의 키 요청을 전송할 때 키에 빠르게 액세스할 수 있도록 실행 중에 시스템 메모리의 키 저장소 목록을 유지보수합니다.

주: 키 생성 중에 Encryption Key Manager GUI를 인터럽트하면 Encryption Key Manager를 다시 설치해야 합니다.

Encryption Key Manager의 키 생성 프로세스가 완료되기 전에 이를 중지하는 경우, 키 저장소 파일이 손상될 수 있습니다. 이러한 손상을 복구하려면 다음 단계를 따르십시오.

- Encryption Key Manager의 최초 설치 중에 Encryption Key Manager가 인터럽트된 경우 Encryption Key Manager 디렉토리가 위치한 디렉토리(예: x:₩ekm)로 이동하십시오. 디렉토리를 삭제하고 설치를 다시 시작하십시오.

- 새 키 그룹을 추가하는 중에 Encryption Key Manager가 인터럽트된 경우 Encryption Key Manager 서버를 중지하고 최신 백업 키 저장소 (x:₩ekm₩gui₩backupfiles 폴더에 위치함)를 사용하여 키 저장소 파일을 복원하십시오. 백업 파일의 파일 이름에는 날짜와 시간 소인 (예: 2007_11_19_16_38_31_EKMKeys.jck)이 포함되어 있습니다. 파일이 x:₩ekm₩gui 디렉토리에 복사되면 날짜와 시간 소인을 제거해야 합니다. Encryption Key Manager 서버를 다시 시작하고 이전에 인터럽트된 키 그룹을 추가하십시오.

4. 『EKM 서버 인증서 구성(EKM Server Certificate Configuration)』 페이지(그림 3-5)에서 키 저장소 별명 및 원하는 추가 데이터를 입력하십시오. **제출 및 서버 다시 시작(Submit and Restart Server)**을 클릭하십시오.

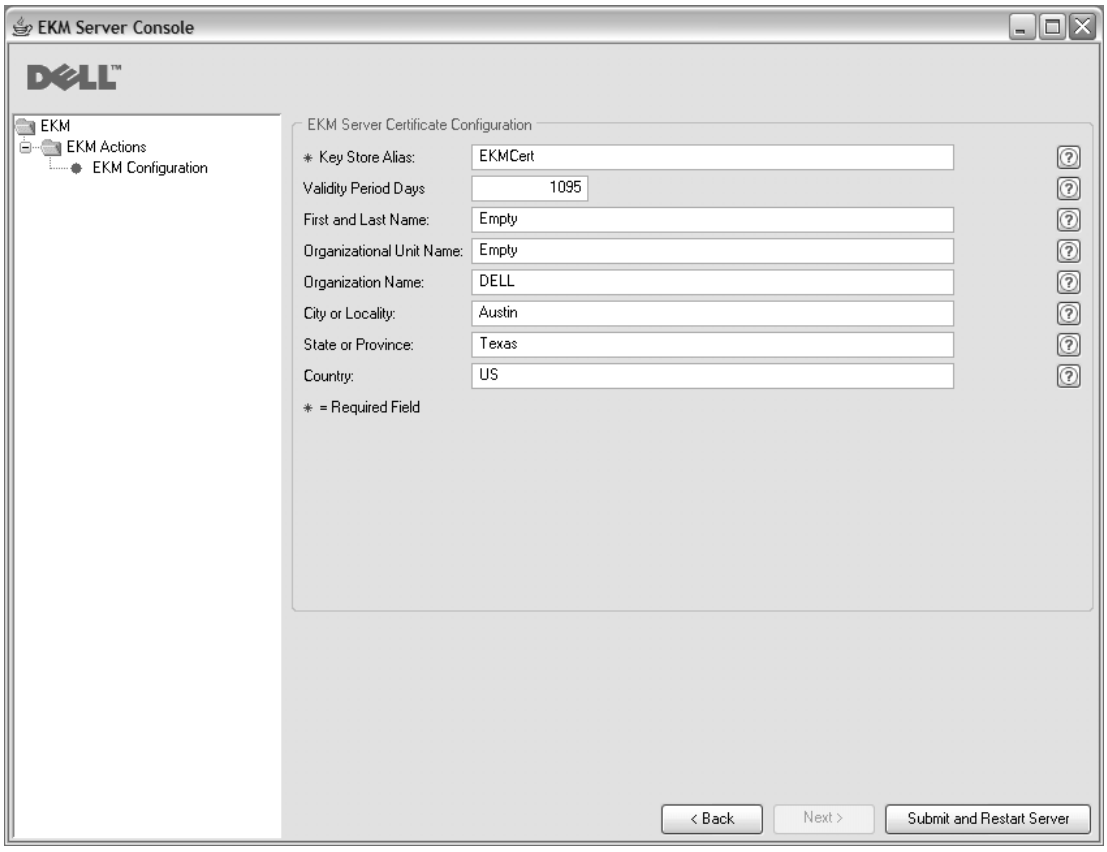

그림 3-5. EKM 서버 인증서 구성(EKM Server Certificate Configuration) 페이지

5. Encryption Key Manager 데이터 파일을 백업해야 함을 알리는 『중요 파일 백업(Backup Critical Files)』 창(3-9 페이지의 그림 3-6)이 열립니다.

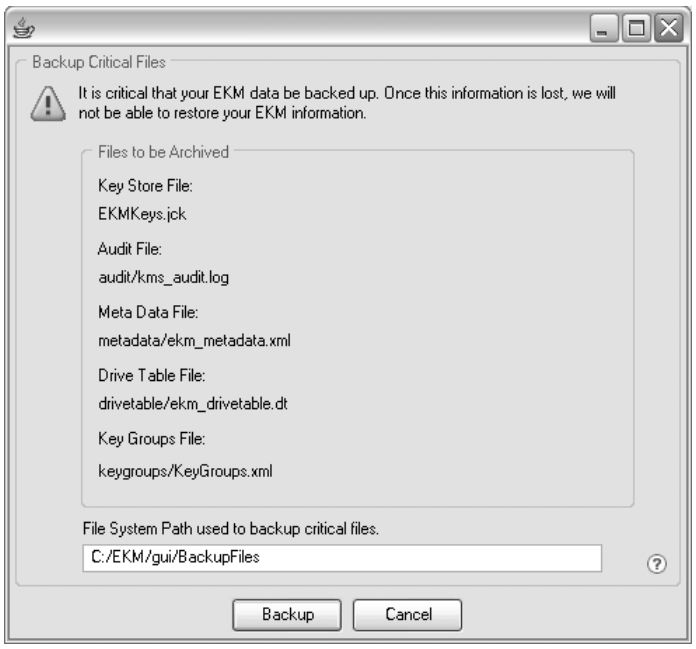

그림 3-6. 중요 파일 백업 창

경로를 확인하고 **백업(Backup)**을 클릭하십시오. Dell Encryption Key Manager 서버는 백그라운드에서 실행됩니다.

『중요 파일 백업(Backup Critical Files)』 창에서 Encryption Key Manager 서버 구성 또는 **백업(Backup)**을 변경할 때 **확인(OK)**을 누를 때마다 Encryption Key Manager가 백업 파일 세트를 생성합니다. *아카이브할 파일* 목록의 파일은 c:/ekm/gui/BackupFlies 디렉토리에 저장됩니다. 각 파일 이름에는 날짜와 시간이 첨부됩니다. 예를 들어 2007년 11월 26일 오후 2시 58분 46초에 백업된 파일 세트에는 이름 앞에 ″2007_11_26_14_58_46_FileName″와 같은 날짜와 시간 소인이 첨가됩니다. 백업 파일은 겹쳐쓰이지 않습니다.

6. GUI 탐색기에서 **서버 상태 모니터(Server Health Monitor)**를 선택하고 Encryption Key Manager 서버가 가동 중인지 확인하십시오.

기존 키 저장소에 키를 추가하려면 3-17 페이지의 『GUI를 사용하여 키 그룹 정의 및 키 작성』을 참조하십시오.

올바른 호스트 IP 주소를 찾는 방법:

현재 Encryption Key Manager GUI에 적용되는 제한사항 때문에 서버 상태 모니터(Server Health Monitor)에 Encryption Key Manager 호스트 IP 주소를 표시할 수 없습니다.

- IPv6 주소를 사용하여 호스트를 구성하는 경우 Encryption Key Manager 응용프로그램이 IP 주소를 표시할 수 없습니다.

- Encryption Key Manager 응용프로그램이 Linux 시스템에 설치된 경우 Encryption Key Manager 응용프로그램은 로컬 호스트 주소는 표시하지만 실제의 활성 IP 포트는 표시하지 않습니다.

1. 호스트 시스템의 실제 IP 주소를 검색하려면 네트워크 구성에 액세스하여 IP 포트 주소를 찾으십시오.
   - Windows 시스템의 경우 명령 창을 열고 `ipconfig`를 입력하십시오.
   - Linux의 경우 `isconfig`를 입력하십시오.

EKM SSL 포트를 식별하는 방법

1. 명령행을 사용하여 Encryption Key Manager 서버를 시작하십시오.
   - Windows의 경우 CD의 `c:₩ekm`으로 이동하여 **startServer.bat**을 클릭하십시오.
   - Linux 플랫폼의 경우 `/var/ekm`으로 이동하여 `startServer.sh`를 입력하십시오.
   - 자세한 정보는 5-1 페이지의 『Key Manager 서버 시작, 새로 고침 및 중지』를 참조하십시오.
2. 명령행을 사용하여 CLI 클라이언트를 시작하십시오.
   - Windows의 경우 CD의 `c:₩ekm`으로 이동하여 **startClient.bat**을 클릭하십시오.
   - Linux 플랫폼의 경우 `/var/ekm`으로 이동하여 `startClient.sh`를 입력하십시오.
   - 자세한 정보는 5-6 페이지의 『명령행 인터페이스 클라이언트』를 참조하십시오.
3. 다음 명령을 사용하여 Encryption Key Manager 서버의 CLI 클라이언트에 로그인하십시오.

   ```
   login –ekmuser userID –ekmpassword password
   ```

   여기서, *userID*는 EKMAdmin이고 *password*는 changeME입니다. (이것이 기본 암호입니다. 이전에 기본 암호를 변경한 경우 새 암호를 사용하십시오.)

   로그인에 성공하면 `User successfully logged in`이 표시됩니다.
4. 다음 명령을 입력하여 SSL 포트를 식별하십시오.

   ```
   status
   ```

   표시되는 응답이 다음과 유사해야 합니다. `server is running. TCP port: 3801, SSL port: 443`.

   SSL 구성 포트를 기록해두고 라이브러리 관리 암호화 설정을 구성할 때 이 포트를 사용했는지 확인하십시오.
5. 명령행에서 로그아웃하십시오. 다음 명령을 입력하십시오.

   ```
   exit
   ```

명령 창을 닫으십시오.

LTO 4 및 LTO 5에서 암호화에 대한 키 및 별명 생성

대칭 암호화 키를 가장 쉽게 생성하는 방법은 Dell Encryption Key Manager 서버 GUI를 사용하는 것입니다(3-6 페이지의 『GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성』 참조). 또한 Keytool 유틸리티를 사용해도 대칭 암호화 키를 생성할 수 있습니다. Keytool은 특히 서로 다른 키 저장소 사이에서 키를 가져오고 내보낼 때 유용합니다. 자세한 내용은 3-14 페이지의 『Keytool -importseckey를 사용하여 데이터 키 가져오기』 및 3-14 페이지의 『Keytool -exportseckey를 사용하여 데이터 키 내보내기』를 참조하십시오.

Keytool은 키, 인증서 및 별명을 관리하는 유틸리티입니다. 이 유틸리티를 사용하면 암호화 데이터 키 생성, 가져오기 및 내보내기와 키 저장소에 암호화 데이터 키 저장과 같은 작업을 수행할 수 있습니다.

키 저장소의 각 데이터 키는 고유한 별명을 통해 액세스합니다. 별명은 `123456tape`와 같은 문자열입니다. JCEKS 키 저장소에서 `123456Tape`는 `123456tape`와 같으며 키 저장소의 같은 항목에 액세스할 수 있습니다. **keytool -genseckey** 명령을 사용하여 데이터 키를 생성하는 경우 같은 명령에서 이에 대응하는 별명을 지정합니다. 별명을 사용하면 LTO 4 및 LTO 5 테이프에서 암호화된 데이터를 읽고 쓰는 경우 사용할 올바른 키 그룹 및 키 저장소의 올바른 키를 식별할 수 있습니다.

주: 개별 별명 및 별명 범위가 고유해야 합니다. 지정된 키 저장소/Encryption Key Manager 인스턴스에서 키를 생성한 경우 이를 준수해야 합니다. 그러나 여러 개의 Encryption Key Manager/키 저장소 환경에서 참조 시 고유함을 유지하고 인스턴스 사이에서 키를 전송할 때 복수 인스턴스 사이에서 고유함을 유지하려면 이름 지정 규칙을 사용해야 합니다.

키 및 별명을 생성한 후 새 별명, 별명 범위 또는 키 그룹의 그룹 ID, 대칭 키가 저장된 파일 이름 및 키 그룹이 정의된 파일 이름을 지정하도록 KeyManagerConfig.properties에서 symmetricKeySet 등록 정보를 업데이트해야 합니다. 자세한 내용은 3-16 페이지의 『키 그룹 작성 및 관리』를 참조하십시오. symmetricKeySet에 이름이 지정된 키만 유효성을 검증합니다(*크기*와 알고리즘이 올바른지에 대해 기존 별명 및 대칭 키를 검사함). 이 등록 정보에 올바르지 않은 키가 지정되면 Key Manager가 시작되지 않고 감사 레코드가 작성됩니다.

Keytool 유틸리티도 다른 키 저장소 사이에서 데이터 키를 가져오고 내보내는 방법을 제공합니다. 각 작업에 대한 개요를 제공합니다. **keytool -ekmhelp**를 실행하면 다음에 논의할 Key Manager와 관련된 모든 매개변수를 표시할 수 있습니다.

구성 등록 정보 파일 편집

KeyManagerConfig.properties 또는 ClientKeyManagerConfig.properties 파일을 변경하려면 다음을 수행하십시오.

1. Encryption Key Manager 서버를 중지하십시오.
2. 원하는 텍스트 편집기를 사용하여 KeyManagerConfig.properties 파일(서버 구성을 변경하는 경우) 또는 ClientKeyManagerConfig.properties 파일(클라이언트 구성을 변경하는 경우)을 여십시오. ^M 때문에 Linux 시스템용 파일을 편집할 때에는 Windows를 사용하지 마십시오. Windows를 사용하는 경우 gvim/vim을 사용하여 파일을 편집하십시오.
3. 이 문서의 지시대로 등록 정보 값을 변경하십시오.
4. 파일을 저장하십시오.
5. Encryption Key Manager 서버를 다시 시작하십시오.

Keytool을 사용하지 않는 경우

Keytool이나 GUI를 사용하지 않고 키와 별명을 생성할 경우에는 Encryption Key Manager와 호환 가능한 다양한 키를 생성할 수 없습니다. Encryption Key Manager와 호환 가능한 개별 키를 생성하려면 다음 형식 중 하나를 사용하여 별명을 지정해야 합니다.

- 12자 이하의 인쇄 가능한 문자(예: `abcdefghijk`)
- 인쇄 가능한 문자 3자, 0 2자, 16진수 16자가 차례로 구성된 총 21자리의 문자(예: `ABC000000000000000001`)

Keytool -genseckey를 사용하여 데이터 키 및 별명 생성

주: 세션에서 **keytool** 명령을 처음으로 사용하는 경우 먼저 `updatePath` 스크립트를 실행하여 올바른 환경을 설정합니다.

Windows의 경우

c:₩ekm을 탐색하여 **updatePath.bat**를 클릭하십시오.

Linux 플랫폼의 경우

/var/ekm을 탐색하여 `. ./updatePath.sh`를 입력하십시오.

Keytool 유틸리티는 LTO 4 및 LTO 5 테이프를 사용하여 LTO 4 및 LTO 5 테이프 드라이브에서 암호화를 수행하는 경우 별명 및 대칭 키를 생성합니다. **keytool -genseckey** 명령을 사용하여 하나 이상의 비밀 키를 생성하고 지정된 키 저장소에 저장합니다. **keytool -genseckey**에서는 다음 매개변수를 사용합니다.

```
-genseckey     [-v] [-protected]
                 [-alias <alias> | aliasrange <aliasRange>] [-keypass <keypass>]
                 [-keyalg <keyalg>] [-keysize <keysize>]
```

[-keystore <keystore>] [-storepass <storepass>]
[-storetype <storetype>] [-providerName <name>]
[-providerClass <provider_class_name> [-providerArg <arg>] ...
[-providerPath <pathlist>]

이 매개변수는 Encryption Key Manager에서 데이터 키를 생성하여 테이프 암호화를 위해 LTO 4 및 LTO 5 드라이브에 제공하는 경우 특히 중요합니다.

-alias

단일 데이터 키에 인쇄 가능한 문자를 사용하여 최대 12자로 *alias* 값을 지정합니다(예: `abcfrg` 또는 `key123tape`).

-aliasrange

여러 개의 데이터 키를 생성하는 경우 *aliasrange*는 처음 3자리 접두부는 영문자이고 다음에 상한과 하한을 나타내는 일련의 16자(16진) 문자열(자동으로 앞에 0이 채워짐)이 이어집니다. 그러면 자동으로 21자 별명이 구성됩니다. 예를 들어 `key1-a`를 지정하면 `KEY00000000000000001`부터 `KEY0000000000000000A`까지 범위의 별명이 생성됩니다. *aliasrange* 값을 `xyz01-FF`로 지정하면 `XYZ00000000000000001`부터 `XYZ000000000000000FF`까지가 생성되며 255개의 대칭 키가 생성됩니다.

-keypass

데이터 키를 보호하는 데 사용되는 암호를 지정합니다. 이 암호는 키 저장소 암호와 **같아야 합니다**. 암호를 지정하지 않으면 암호를 묻는 프롬프트가 표시됩니다. 프롬프트에서 **Enter**를 누르면 키 저장소에서 사용하는 암호가 키 암호로 설정됩니다. *keypass*는 6자 이상이어야 합니다.

주: 키 저장소 암호를 설정하고 나면 보안 위반이 발생하지 않는 한 **변경하지 마십시오**. 3-14 페이지의 『키 저장소 암호 변경』을 참조하십시오.

-keyalg

데이터 키를 생성하는 데 사용되는 알고리즘을 지정합니다. 이 값은 `AES`로 지정해야 합니다.

-keysize

생성할 데이터 키 *크기*를 지정합니다. 키 *크기*를 `256`으로 지정해야 합니다.

다음은 대칭 키와 연관될 수 있는 허용되는 별명에 대한 예제입니다.

```
abc000000000000000001
abc00a0120fa000000001
```

다음은 Key Manager에서 승인하지 않는 별명에 대한 예제입니다.

```
abcefghij1234567 ? wrong length
abcg0000000000000001 ? prefix is longer than 3 characters
```

키 저장소에 이미 별명이 있으면 Keytool에서 예외가 발생하고 Keytool이 중지됩니다.

키 저장소 암호 변경

주: 키 저장소 암호를 설정하고 나면 보안 위반이 발생하지 않는 한 **변경하지 마십시오**. 보안 노출 위험을 없애기 위해 암호는 인식하기 어렵게 만들어집니다. 키 저장소 암호를 변경하려면 다음 **keytool** 명령을 사용하여 개별적으로 해당 키 저장소의 모든 키에 대한 암호를 변경해야 합니다.

키 저장소 암호를 변경하려면 다음을 입력하십시오.

```
keytool -keypasswd  -keypass old_passwd -new new_passwd -alias alias
        -keystore keystorename -storetype keystoretype
```

또한 다음 방법 중 하나를 사용하여 암호를 지정한 경우 모든 서버 구성 파일 등록 정보에서 키 저장소 암호를 변경하도록 KeyManagerConfig.properties를 편집해야 합니다.

- 인식하기 어렵도록 만들어진 모든 암호를 삭제하고 다음 시작 시 Encryption Key Manager에서 암호를 묻는 프롬프트를 표시하게 합니다.
- 인식하기 어렵도록 만들어진 모든 암호를 삭제하고 일반 텍스트로 새 암호를 입력합니다. 다음 시작 시 인식하기 어렵게 만들어집니다.

Keytool -importseckey를 사용하여 데이터 키 가져오기

keytool -importseckey 명령을 사용하여 가져온 파일에서 비밀 키 또는 비밀 키 세트를 가져옵니다. **keytool -importseckey**에서는 다음 매개변수를 사용합니다.

-importseckey [-v]
[-keyalias <keyalias>] [-keypass <keypass>]
[-keystore <keystore>] [-storepass <storepass>]
[-storetype <storetype>] [-providerName <name>]
[-importfile <importfile>] [-providerClass <provider_class_name>]
[providerArg <arg>]

이 매개변수는 Encryption Key Manager에서 데이터 키를 가져와서 테이프 암호화를 위해 LTO 4 및 LTO 5 드라이브에 제공하는 경우 특히 중요합니다.

-keyalias
*importfile*의 모든 데이터 키를 암호 해독하는 키 저장소에 있는 개인용 키의 별명을 지정합니다.

-importfile
가져올 데이터 키를 포함하는 파일을 지정합니다.

Keytool -exportseckey를 사용하여 데이터 키 내보내기

keytool -exportseckey 명령을 사용하여 비밀 키 또는 비밀 키 세트를 내보내는 파일로 내보냅니다. **keytool -exportseckey**에서는 다음 매개변수를 사용합니다.

-exportseckey [-v]
[-alias <alias> | aliasrange <aliasRange>] [-keyalias <keyalias>]
[-keystore <keystore>] [-storepass <storepass>]
[-storetype <storetype>] [-providerName <name>]
[-exportfile <exportfile>] [-providerClass <provider_class_name>]
[providerArg <arg>]

이 매개변수는 Encryption Key Manager에서 데이터 키를 내보내기 하여 테이프 암호화를 위해 LTO 4 및 LTO 5 드라이브에 제공하는 경우 특히 중요합니다.

-alias

단일 데이터 키에 인쇄 가능한 문자를 사용하여 최대 12자로 *alias* 값을 지정합니다(예: `abcfrg` 또는 `key123tape`).

-aliasrange

여러 개의 데이터 키를 내보내는 경우 *aliasrange*는 처음 3자리 접두부는 영문자이고 다음에 상한과 하한을 나타내는 일련의 16자(16진) 문자열(자동으로 앞에 0이 채워짐)이 이어집니다. 그러면 자동으로 21자 별명이 구성됩니다. 예를 들어 `key1-a`를 지정하면 `KEY0000000000000000001`부터 `KEY000000000000000000A`까지 범위의 별명이 생성됩니다. *aliasrange* 값을 `xyz01-FF`로 지정하면 `XYZ0000000000000000001`부터 `XYZ000000000000000FF`까지가 생성됩니다.

-exportfile

데이터 키를 내보낼 때 이를 저장할 파일을 지정합니다.

-keyalias

모든 데이터 키를 암호화하는 키 저장소에 있는 공용 키의 별명을 지정합니다. 대칭(데이터) 키를 가져오는 키 저장소에 이에 해당하는 개인용 키가 반드시 있어야 합니다.

JCEKS 키 저장소를 사용하는 LTO 4 및 LTO 5 암호화에 대한 샘플 별명 및 대칭 키 설정

`-aliasrange` 옵션을 사용하여 **KeyTool**을 호출합니다.

키 알고리즘(-keyalg)은 AES로 지정하고 키 크기(–keysize)는 `256`으로 지정해야 합니다. 다음과 같습니다.

```
/bin/keytool –genseckey –v –aliasrange AES01-FF –keyalg AES –keysize 256
–keypass password -storetype jceks –keystore path/filename.jceks
```

이러한 KeyTool 호출을 수행하면 범위가 AES0000000000000000001부터 AES000000000000000FF까지인 255개의 순차적 별명과 이와 연관된 AES 256비트 대칭 키가 생성됩니다. 강력한 키 관리 조작에 적합한 전체 범위의 독립형 키 별명을 설정하도록 필요한 만큼 누적식으로 반복할 수도 있습니다. 예를 들어, LTO 4 및 LTO 5에서 추가 별명 및 대칭 키를 생성하는 경우 다음과 같습니다.

```
/bin/keytool –genseckey –v –alias abcfrg –keyalg AES –keysize 256
–keypass password -storetype jceks –keystore path/filename.jceks
```

이와 같이 호출하면 독립형 별명 abcfrg가 누적식으로 이름이 지정된 키 저장소에 추가됩니다. 이 키 저장소에는 이미 호출로 생성된 255개 별명을 포함하고 있습니다. 이 호출로 -keystore 옵션에서 이름을 지정한 jceks 파일에는 256개 대칭 키도 생성됩니다.

위에서 사용한 별명 범위 중 일부 또는 모든 항목 및 대칭 키가 저장된 파일 이름을 일치시키려면 KeyManagerConfig.properties 파일에서 다음 행을 추가하여 symmetricKeySet 등록 정보를 업데이트하십시오. 올바르지 않은 별명을 지정하면 Encryption Key Manager가 시작되지 않을 수도 있습니다. 유효성 검사에 실패하는 또 다른 이유로 비트 *크기*가 올바르지 않거나(AES의 경우 키 *크기*는 256이어야 함) 플랫폼의 알고리즘이 올바르지 않을 수 있습니다. -keyalg는 AES여야 하며 -keysize는 256이어야 합니다. **config.keystore.file**에 지정된 파일 이름은 Keytool 호출 시 –keystore <filename>에 지정된 이름과 일치해야 합니다.

symmetricKeySet = AES01-FF,abcfrg
config.keystore.file = <filename>.jceks

symmetricKeySet에 이름이 지정된 키만 유효성을 검증합니다(*크기*와 알고리즘이 올바른지에 대해 *기존* 별명 및 대칭 키를 검사함). 이 등록 정보에 올바르지 않은 키를 지정하면 Encryption Key Manager가 시작되지 않고 감사 레코드가 작성됩니다.

키 그룹 작성 및 관리

Encryption Key Manager에서는 LTO 4 및 LTO 5 암호화를 위한 대칭 키를 키 그룹으로 조직하는 *기능*을 제공합니다. 이때 암호화하는 데이터 유형, 이 데이터에 액세스하는 사용자 또는 다른 중요한 특징에 따라 키를 그룹화할 수 있습니다. 키 그룹을 작성하면 **adddrive** 명령에서 -symrec 키워드를 사용하여 특정 테이프 드라이브와 이 키 그룹을 연관시킬 수 있습니다. 자세한 구문은 5-10 페이지의 『adddrive』를 참조하십시오.

키 그룹을 빌드하려면 KeyGroups.xml 파일에서 키 그룹을 정의해야 합니다. 3-6 페이지의 『GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성』의 절차를 따랐다면 이 파일 위치는 EKM 구성 페이지에 지정되어 있습니다. 구성 파일을 수동으로 작성한 경우 KeyGroups.xml 파일 위치는 다음과 같이 구성 등록 정보 파일에 지정됩니다.

```
config.keygroup.xml.file = FILE:KeyGroups.xml
```

이 매개변수를 지정하지 않으면 기본적으로 Encryption Key Manager 실행 위치의 작업 디렉토리에 있는 KeyGroups.xml 파일을 사용합니다. 이 파일이 없는 경우 공백의 KeyGroups.xml 파일이 작성됩니다. 다음에 Encryption Key Manager 서버를 시작하면 [Fatal Error] :-1:-1: Premature end of file. 메시지가 **native_stderr.log**

에 표시됩니다. 이것은 공백 KeyGroups.xml 파일을 구문 분석하는 중 발생한 오류로, 키 그룹을 사용하도록 Encryption Key Manager 서버를 구성하지 않았다면 Encryption Key Manager 서버를 시작하는 데 방해가 되지는 않습니다.

키 그룹은 Dell Encryption Key Manager 서버 GUI 또는 다음과 같은 CLI 클라이언트 명령을 사용하여 빌드됩니다(구문은 5-9 페이지의 『CLI 명령』 참조).

GUI를 사용하여 키 그룹 정의 및 키 작성

GUI를 사용하여 키 그룹을 관리하는 데 필요한 모든 작업을 수행할 수 있습니다. 또한 추가 키를 작성할 수도 있습니다.

주: 다음 작업 중 하나를 수행하는 동안 **변경 사항 제출(Submit Changes)**을 누르면 Encryption Key Manager 데이터 파일을 백업해야 함을 알리는 백업 대화 상자 창(3-9 페이지의 그림 3-6)이 열립니다. 백업 데이터를 저장할 경로를 입력하십시오. **제출(Submit)**을 클릭하십시오. 그 다음 백업 경로를 확인하고 **확인(OK)**을 클릭하십시오.

키 그룹을 작성하고 이 키 그룹을 키로 채우거나 기존 키 그룹에 키를 추가하려면 다음을 수행하십시오.

1. GUI를 아직 시작하지 않은 경우 GUI를 여십시오.

   Windows의 경우

   c:₩ekm₩gui를 탐색하여 **LaunchEKMGui.bat**를 클릭하십시오.

   Linux 플랫폼의 경우

   /var/ekm/gui를 탐색하여 `. ./LaunchEKMGui.sh`를 입력하십시오.

2. GUI 왼쪽에 있는 탐색기에서 **관리 명령(Administration Commands)**을 클릭하십시오.
3. 관리 명령(Administration Commands) 영역에서 **키 그룹 작성(Create a Group of Keys)**을 선택하십시오(3-18 페이지의 그림 3-7).

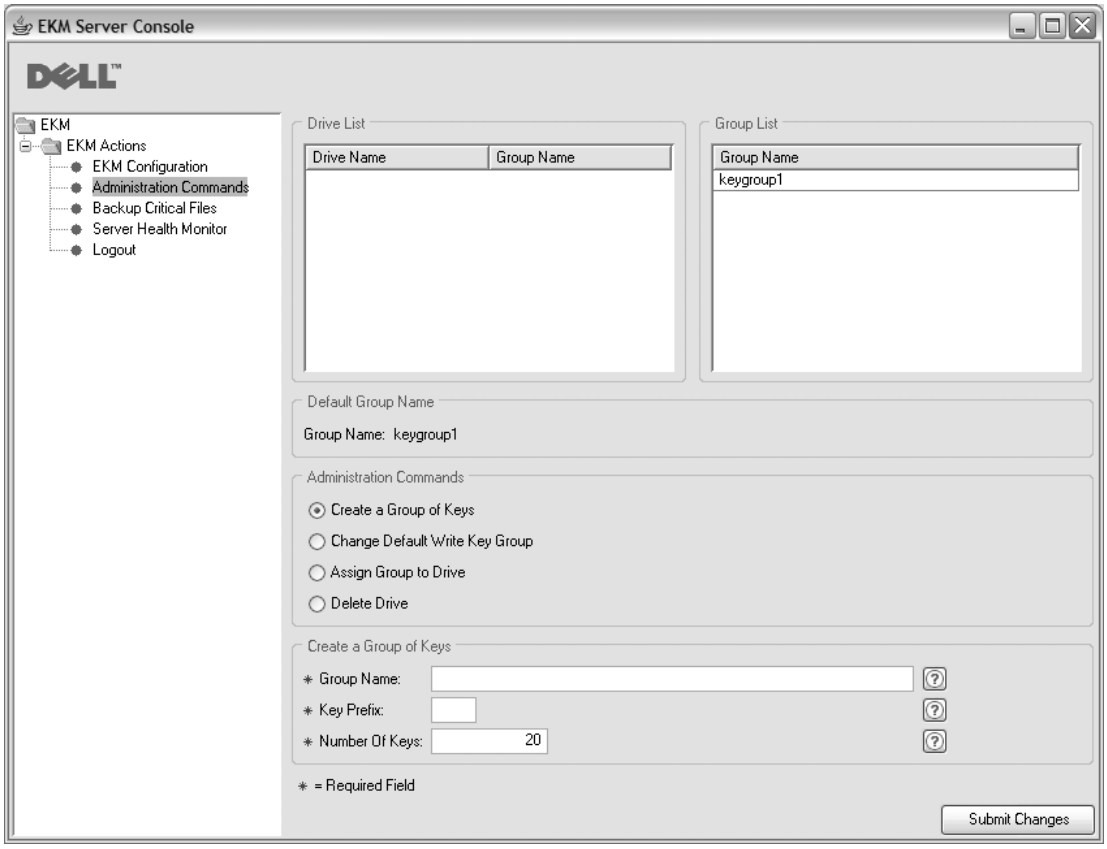

그림 3-7. 키 그룹 작성

4. 새 키 그룹 이름, 키 별명으로 사용할 접두부 및 그룹에 포함할 키 수를 입력하십시오. **변경 사항 제출(Submit Changes)**을 클릭하십시오.

기본 키 그룹을 변경하려면

1. GUI 왼쪽에 있는 탐색기에서 **관리 명령(Administration Commands)**을 클릭하십시오.
2. 관리 명령(Administration Commands) 영역에서 **기본 쓰기 키 그룹 작성(Change Default Write Key Group)**을 선택하십시오(3-19 페이지의 그림 3-8).

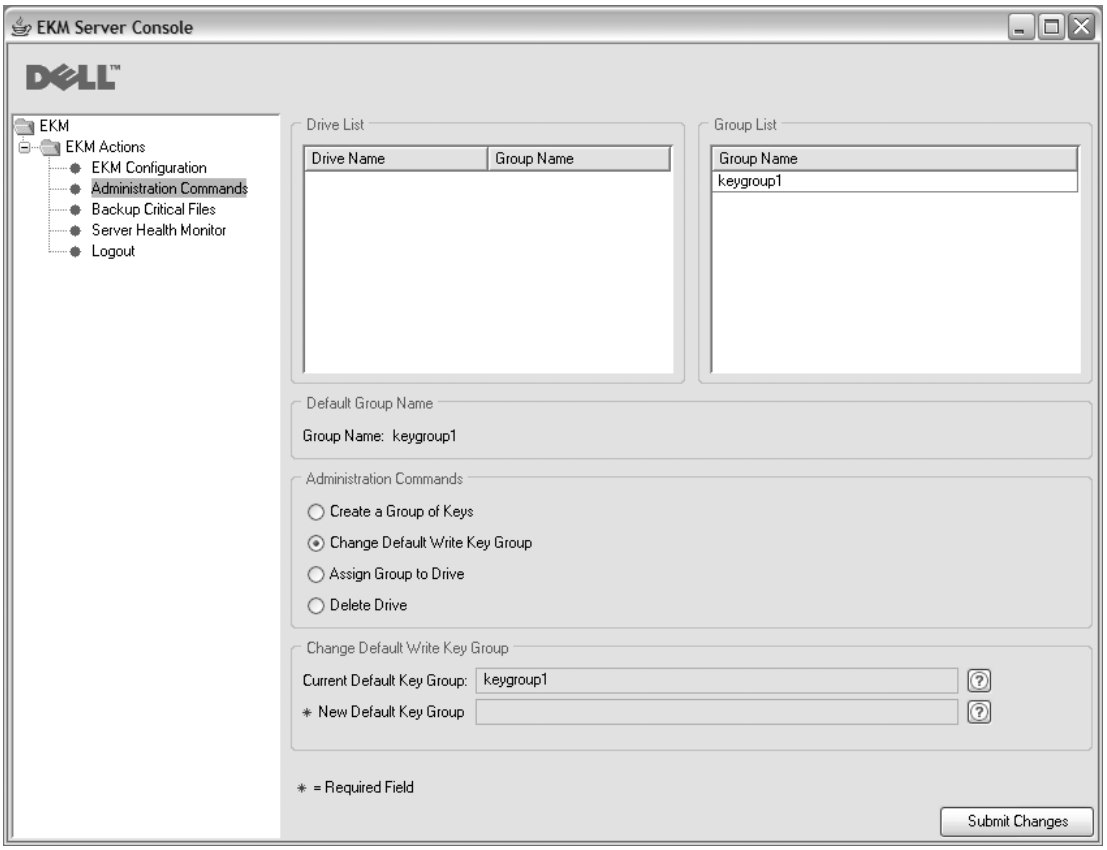

그림 3-8. 기본 쓰기 키 그룹 변경

3. 오른쪽에 있는 그룹 목록에서 새 기본 키 그룹을 선택하십시오.
4. 창의 맨 아래에서 현재 키 그룹 및 새 기본 키 그룹을 확인하고 **변경 사항 제출(Submit Changes)**을 클릭하십시오.

특정 키 그룹을 특정 테이프 드라이브에 지정하려면

1. GUI 왼쪽에 있는 탐색기에서 **관리 명령(Administration Commands)**을 클릭하십시오.
2. 관리 명령(Administration Commands) 영역에서 **드라이브에 그룹 지정(Assign Group to Drive)**을 선택하십시오(3-20 페이지의 그림 3-9).

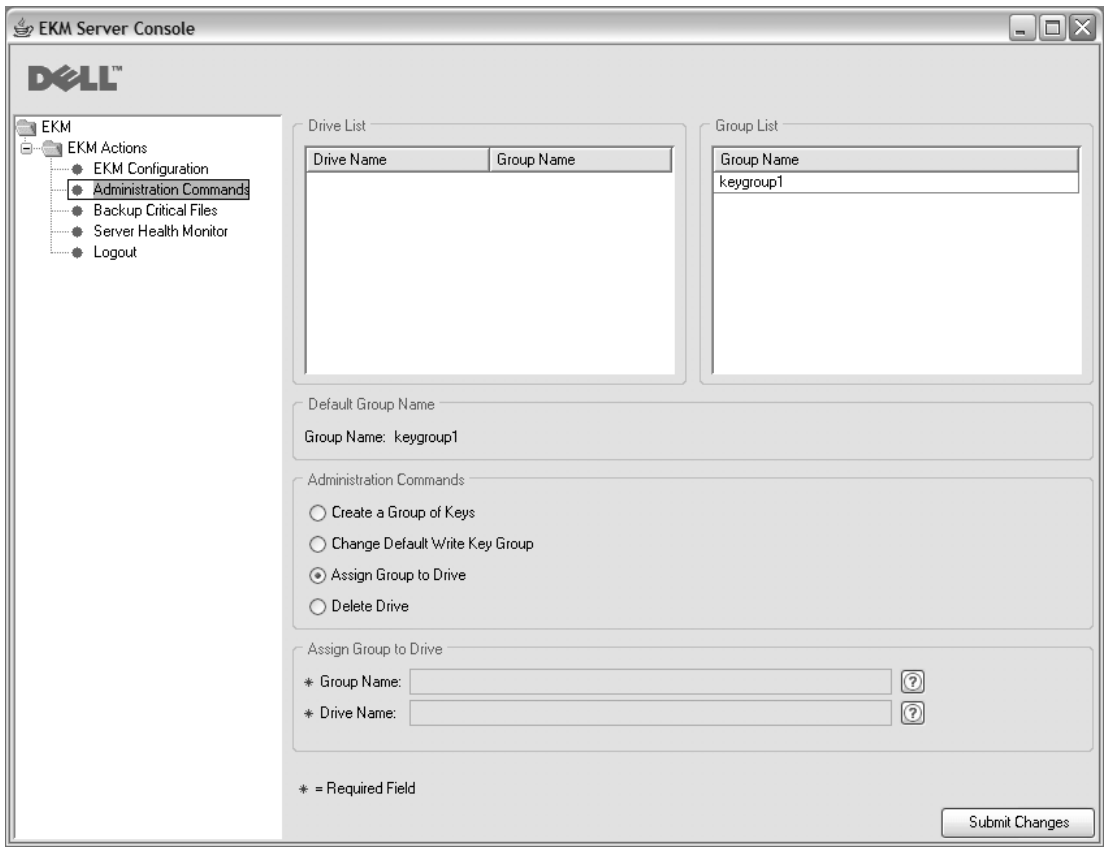

그림 3-9. 드라이브에 그룹 지정

3. 드라이브 목록에서 테이프 드라이브를 선택하십시오.
4. 그룹 목록에서 키 그룹을 선택하십시오.
5. 창의 맨 아래에서 드라이브 및 키 그룹을 확인하고 **변경 사항 제출(Submit Changes)**을 클릭하십시오.

드라이브 테이블에서 테이프 드라이브를 삭제하려면

1. GUI 왼쪽에 있는 탐색기에서 **관리 명령(Administration Commands)**을 클릭하십시오.
2. 관리 명령(Administration Commands) 영역에서 **드라이브 삭제(Delete Drive)**를 선택하십시오(3-21 페이지의 그림 3-10).

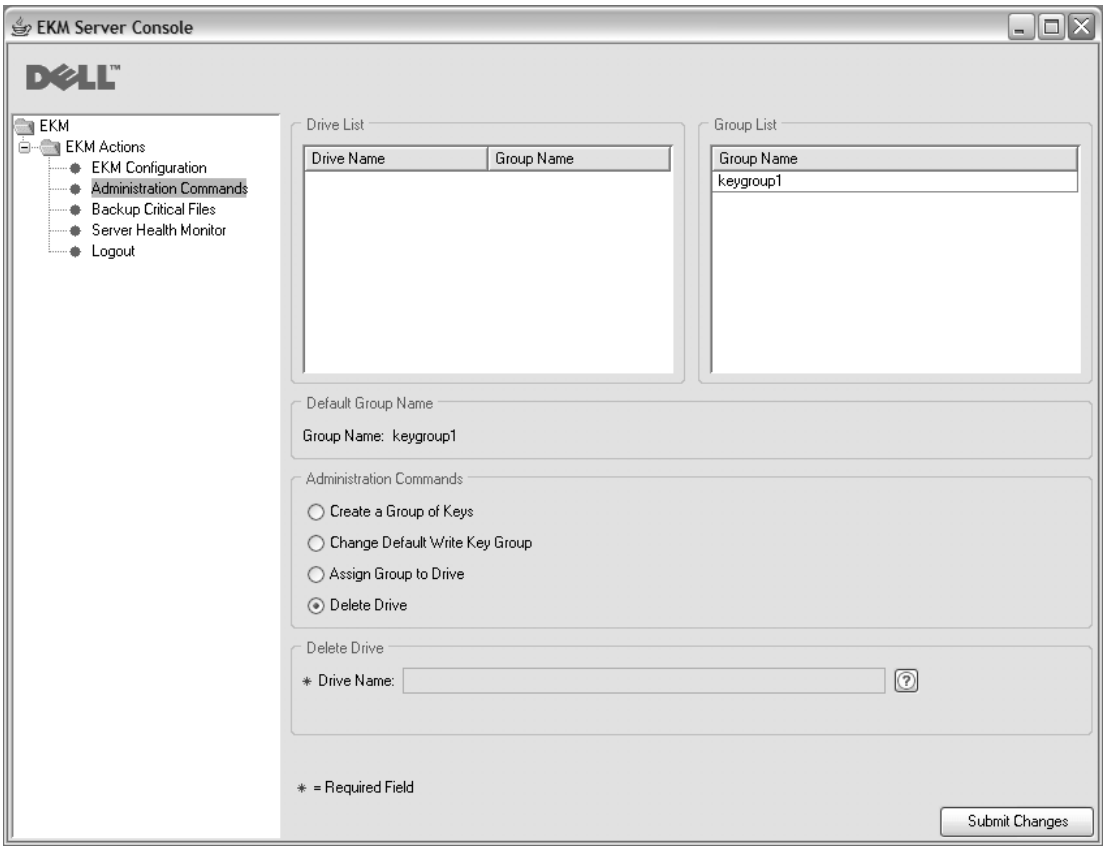

그림 3-10. 드라이브 삭제

3. 드라이브 목록에서 테이프 드라이브를 선택하십시오.
4. 창의 맨 아래에서 드라이브 이름을 확인하고 **변경 사항 제출(Submit Changes)**을 클릭하십시오.

CLI 명령을 사용하여 키 그룹 정의

Encryption Key Manager에서는 키 세트를 그룹화할 수 있는 키 그룹 기능을 제공합니다.

Encryption Key Manager 응용프로그램을 설치하여 구성하고(키 저장소 및 키가 생성됨) Encryption Key Manager 서버를 시작한 후 클라이언트를 사용하여 서버에 로그인하고 다음 단계를 수행하십시오.

1. **createkeygroup** 명령을 실행하십시오.

   이 명령은 KeyGroups.xml 파일에서 최초의 키 그룹 오브젝트를 작성합니다. 이 명령은 한 번만 실행합니다.

   구문: **createkeygroup -password** *password*

   -password
   나중에 검색하기 위해 KeyGroups.xml 파일에서 키 저장소 암호를 암호화하는

데 사용되는 *password*입니다. 키 저장소는 키 그룹의 키를 암호화한 후 차례대로 각 개별 키 그룹 별명 암호를 암호화합니다. 따라서 KeyGroups.xml 파일에는 일반 형식의 키는 존재하지 않습니다.

예제: createkeygroup -password a75xynrd

2. **addkeygroup** 명령을 실행하십시오.

   이 명령은 KeyGroups.xml에서 고유한 그룹 ID를 사용하여 키 그룹의 인스턴스를 작성합니다.

   구문: **addkeygroup -groupID** *groupname*

   -groupID
   KeyGroups.xml 파일에서 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

   예제: addkeygroup -groupID keygroup1

3. **addkeygroupalias** 명령을 실행하십시오.

   이 명령은 특정 키 그룹 ID 이외에도 키 저장소에 있는 기존 키 별명에 대한 새 별명을 작성합니다.

   구문: **addkeygroupalias -alias** *aliasname* **-groupID** *groupname*

   -alias
   키의 새 *aliasname*입니다. 전체 키 이름이어야 합니다(즉, Key00을 key00000000000000000으로 입력해야 함).

   -groupID
   KeyGroups.xml 파일에서 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

   예제: addkeygroupalias -alias key000000000000000000 -groupID keygroup1

   주: 이 CLI 명령을 사용하는 경우 한 번에 하나의 키만 추가할 수 있습니다. 이 명령은 키 그룹에 추가해야 하는 모든 개별 키에서 실행해야 합니다.

4. 키 그룹을 새 테이프 드라이브 또는 기존 테이프 드라이브에 연관시킵니다.
   a. 키 그룹을 기존 테이프 드라이브에 연관시키려면 **moddrive** 명령을 실행하십시오.

      이 명령은 드라이브 테이블에서 테이프 드라이브 정보를 수정합니다.

      구문: **moddrive -drivename** *drivename* **-symrec** *alias*

      -drivename
      *drivename*은 테이프 드라이브의 일련 번호를 지정합니다.

-symrec

테이프 드라이브에서 대칭 키의 *alias* 또는 키 그룹 이름을 지정합니다.

예제: moddrive -drivename 000123456789 -symrec keygroup1

b. 드라이브 테이블에 새 테이프 드라이브를 추가하고 이를 키 그룹과 연관시키려면 **adddrive** 명령을 실행하십시오.

이 명령을 사용하면 드라이브를 추가한 후 특정 키 그룹과 연관시킬 수 있습니다.

구문: **adddrive -drivename** *drivename* **-symrec** *alias*

-drivename

*drivename*은 추가할 드라이브의 12자리 일련 번호를 지정합니다.

주: 총 12자리가 되도록 일련 번호 10자리 앞에 0을 두 개 추가합니다.

-symrec

테이프 드라이브에서 대칭 키의 *alias* 또는 그룹 ID를 지정합니다.

예제: adddrive -drivename 000123456789 -symrec keygroup1

테이프 드라이브에 대한 별명이 정의되지 않은 경우 사용할 기본값으로 키 그룹을 지정하려면 구성 등록 정보 파일의 symmetrickeySet 등록 정보를 사용하려는 키 그룹의 그룹 ID로 설정합니다. 예를 들어 다음과 같습니다.

```
symmetricKeySet = keygroup1
```

GroupID는 KeyGroups.xml 파일의 기존 키 그룹 ID와 일치해야 합니다. 그렇지 않으면 Encryption Key Manager 서버가 시작되지 않습니다. Encryption Key Manager에서는 키 그룹 내의 키 사용을 추적합니다. 올바른 그룹 ID를 지정하면 Encryption Key Manager에서 마지막으로 사용한 키를 기록하고 지정된 키 그룹 내에서 무작위로 키를 선택합니다.

한 키 그룹에서 다른 키 그룹으로 키 복사

addaliastogroup 명령을 실행하십시오.

이 명령은 기존(소스) 키 그룹에서 새(대상) 키 그룹으로 특정 별명을 복사합니다.

구문: **addaliastogroup -aliasID** *aliasname* **-sourceGroupID** *groupname* **-targetGroupID** *groupname*

-aliasID

추가할 키의 *aliasname*입니다.

-sourceGroupID

별명을 복사할 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

-targetGroupID

별명이 추가될 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

예제: addaliastogroup -aliasID aliasname -sourceGroupID keygroup1 -targetGroupID keygroup2

주: 키는 두 키 그룹에서 모두 사용 가능합니다.

제 4 장 Encryption Key Manager 구성

GUI를 사용하여 Encryption Key Manager 구성

구성 등록 정보 파일을 작성하는 가장 쉬운 방법은 3-6 페이지의 『GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성』에서 관련 절차에 따라 Dell Encryption Key Manager GUI를 사용하는 것입니다. 이 절차를 따른 경우에는 이미 구성 파일을 작성한 상태이므로 추가 구성이 필요하지 않습니다. 다음 정보는 추가 Encryption Key Manager 구성 옵션의 장점을 활용하려는 경우 유용합니다.

구성 전략

KeyManagerConfig.properties 파일의 일부 구성 설정에서는 단축 설정을 제공합니다. 사용자는 이로 인한 영향을 알고 있어야 합니다.

테이프 드라이브 테이블 자동 업데이트

Encryption Key Manager에서는 값이 `true`로 설정되는 경우 새 테이프 드라이브가 Dell Encryption Key Manager에 연결되면 테이프 드라이브 테이블을 자동으로 채우는 구성 파일의 변수(drive.acceptUnknownDrives)를 제공합니다. 그러면 각 테이프 드라이브 또는 라이브러리에 대해 **adddrive** 명령을 사용하지 않아도 됩니다. 이 모드에서는 CLI 클라이언트 명령을 사용하여 이러한 각 장치의 10자리 일련 번호를 입력하지 않아도 됩니다. 새 드라이브는 테이프 장치의 정체를 확인하기 위해 일반 공용/개인용 키 암호 교환을 수행합니다. 이 확인이 끝나면 새 장치에서 EEDK에 저장된 키 ID에 기반하여 기존 테이프를 읽을 수 있습니다(이에 대응하는 키 정보가 구성된 키 저장소에 있다고 가정함).

주: 드라이브를 자동으로 추가한 후 드라이브 테이블에 저장하려면 GUI 또는 5-16 페이지의 『refresh』 명령을 사용하여 Encryption Key Manager 서버를 새로 고쳐야 합니다.

LTO 4 및 LTO 5 드라이브의 경우 새로 추가된 장치에서 기본 대칭 키 풀 (symmetricKeySet)을 암호화하도록 설정할 수 있습니다. 즉, 장치를 연결하면 Encryption Key Manager에서 키 자료와 연관된 장치를 전부 구성할 수 있습니다. 장치를 드라이브 테이블에 추가할 때 이를 수행하지 않으려면 **moddrive** 명령을 사용하여 테이프 드라이브를 테이프 드라이브 테이블에 추가하면 됩니다.

관리자는 Encryption Key Manager에서 제공하는 각 테이프 드라이브의 10자리 일련 번호를 입력하는 수고를 덜 뿐만 아니라 대규모 시스템 구성 시 기본 환경을 허용합니다.

단, 이러한 편의를 위해 보안 정도가 줄어든다는 점을 감안하십시오. 장치가 자동으로 추가되어 인증서 별명과 연관될 수 있기 때문에(즉, 해당 인증서 별명을 사용하여 테이프에 쓸 수 있음), 장치를 수동으로 추가하는 작업을 건너뛸 때 관리자가 수행하는 추가된 보안 검사도 건너뜁니다. 따라서 테이프 드라이브 정보를 드라이브 테이블에 자동으로 추가하고 새 장치가 인증서 정보에 액세스할 수 있는 권한을 암시적으로 부여하는 작업이 허용되는 보안 위험인지 판별하려면 이 옵션의 장단점을 신중하게 평가해야 합니다.

주: drive.acceptUnknownDrives 등록 정보는 기본적으로 `false`로 설정되어 있습니다. 따라서 Encryption Key Manager에서는 새 드라이브를 드라이브 테이블에 자동으로 추가하지 않습니다. 사용하려는 모드를 선택하고 이에 따라 구성을 변경합니다. 자세한 내용은 부록 B를 참조하십시오.

두 개의 Key Manager 서버 사이에서 데이터 동기화

드라이브 테이블 및 구성 등록 정보 파일은 두 Encryption Key Manager 서버 사이에서 동기화할 수 있습니다. CLI 클라이언트 **sync** 명령을 사용하여 수동으로 동기화하거나 KeyManagerConfig.properties 파일에 네 개의 등록 정보를 설정하여 자동으로 동기화할 수 있습니다.

참고

두 동기화 방법 모두 키 저장소 또는 키 그룹 XML 파일에서는 작동하지 않습니다. 수동으로 복사해야 합니다.

자동 동기화 기능은 KeyManagerConfig.properties 파일의 sync.ipaddress 등록 정보에 올바른 IP 주소를 지정한 경우에만 사용할 수 있습니다. 4-3 페이지의 『자동 동기화』를 참조하십시오.

수동 동기화

수동 방법에서는 CLI 클라이언트 **sync** 명령을 실행하십시오. 구문은 다음과 같습니다.

sync {**-all** | **-config** | **-drivetab**} **-ipaddr** *ip_addr* :*sslport* [**-merge** | **-rewrite**]

이 명령은 구성 파일 등록 정보 또는 드라이브 테이블 정보 또는 둘 모두를 **–ipaddr** 매개변수로 지정된 소스(발신) 서버에서 대상(수신) 서버로 전송합니다. 수신측 Encryption Key Manager 서버는 가동되어 실행 중이어야 합니다.

필수 필드

-all

구성 등록 정보 파일 및 드라이브 테이블 정보 모두를 **-ipaddr**에서 지정한 서버로 전송합니다.

-config

구성 등록 정보 파일만 **-ipaddr**에서 지정한 서버로 전송합니다.

-drivetab

드라이브 테이블 정보만 **-ipaddr**에서 지정한 서버로 전송합니다.

-ipaddr

*ip_addr:sslport*는 수신측 서버의 주소 및 SSL 포트를 지정합니다. *sslport*는 수신측 서버 KeyManagerConfig.properties 파일의 『TransportListener.ssl.port』에 지정된 값과 일치해야 합니다.

선택적 필드

-merge

수신측 서버에서 새 드라이브 테이블 데이터를 현재 데이터와 병합(추가)합니다. 구성 파일은 항상 다시 쓰기가 가능합니다. 기본값입니다.

-rewrite

수신측 서버에서 현재 데이터를 새 데이터로 바꿉니다.

자동 동기화

드라이브 테이블 및 등록 정보 파일은 기본 Key Manager 서버에서 보조 서버로 자동 전송될 수 있습니다. 보조 서버는 데이터 동기화를 수행하는 경우 실행해야 합니다. 기본 서버에서 보조 서버로 데이터를 자동으로 동기화하려면 기본 서버의 KeyManagerConfig.properties에서 다음 네 가지 등록 정보를 지정해야 합니다. 보조 서버 또는 수신측 서버 등록 정보 파일은 변경하지 않아도 됩니다.

sync.ipaddress

수신측 서버의 주소 및 SSL 포트를 지정합니다. 예를 들어 다음과 같습니다.

```
sync.ipaddress = backupekm.server.ibm.com:1443
```

이 등록 정보를 지정하지 않거나 잘못 지정하면 자동 동기화를 사용할 수 없습니다.

sync.action

수신측 서버의 기존 데이터를 병합하거나 다시 씁니다. 올바른 값은 **merge**(기본값) 및 **rewrite**입니다. 구성 등록 정보를 동기화하면 항상 다시 씁니다.

sync.timeinhours

데이터를 전송해야 하는 간격입니다. 이 값은 정수(시간)로 지정됩니다. 서버를 시작하는 시점에서 시간 간격이 시작됩니다. 즉, 지정된 시간 동안 서버를 실행한 후 동기화가 수행됩니다. 기본값은 24입니다.

sync.type

전송해야 하는 데이터입니다. 올바른 값은 **drivetab**(기본값), **config** 및 **all**입니다.

구성 기본 사항

주: 3-6 페이지의 『GUI를 사용하여 구성 파일, 키 저장소 및 인증서 작성』의 절차를 수행한 경우 기본 구성이 이미 작성되었으므로 아래 단계를 수행하지 않아도 됩니다. 이 정보에서는 GUI를 사용하지 않고 해당 작업을 수행하는 방법을 보여줍니다. 추가 구성 옵션을 활용하려는 경우 도움이 될 수 있습니다.

Windows 사용자에 대한 참고: Windows의 경우 명령에 공백을 포함한 디렉토리 경로를 사용할 수 없습니다. 따라서 명령을 입력할 때 `Program Files` 대신 `progra~1`과 같이 생성된 디렉토리의 짧은 이름을 지정해야 합니다. 디렉토리의 짧은 이름을 나열하려면 **dir /x** 명령을 실행하십시오.

이 절차에서는 Encryption Key Manager 구성에 필요한 최소 단계를 포함합니다. 부록 A에는 서버 구성 등록 정보 파일의 예가 들어 있습니다. 서버와 클라이언트 구성의 모든 등록 정보 목록을 보려면 부록 B를 참조하십시오.

1. **keytool**을 사용하여 JCEKS 키 저장소를 관리하십시오. 키 저장소 작성 시 경로 및 파일 이름과 함께 인증서 및 키에 지정된 이름을 기록해두십시오. 나중에 이 정보를 사용합니다.
2. 키 저장소가 없으면 하나를 작성하십시오. 테이프 드라이브에서 사용할 인증서 및 키를 이 새 키 저장소에 추가하거나 가져오십시오. 3-11 페이지의 『LTO 4 및 LTO 5에서 암호화에 대한 키 및 별명 생성』을 참조하십시오. 인증서 및 키에 지정된 이름을 기록해두십시오. 나중에 이 정보를 사용합니다.
3. 키 그룹을 작성하고 키 별명을 작성하십시오. 3-16 페이지의 『키 그룹 작성 및 관리』를 참조하십시오.
4. 원하는 텍스트 편집기를 사용하여 **KeyManagerConfig.properties**를 열어 다음 등록 정보를 지정하십시오. 서버의 현재 디자인이 매우 엄격하다는 점을 명심하십시오. ^M 때문에 Linux 시스템용 파일을 편집하는 데 Windows를 사용하지 마십시오. Windows를 사용하는 경우 gvim/vim을 사용하여 파일을 편집하십시오.

   Windows 사용자에 대한 참고: Java SDK에서는 Windows에서 실행하는 경우에도 슬래시를 사용합니다.
   KeyManagerConfig.properties 파일에서 경로를 지정할 때 슬래시를 사용해야 합니다. 명령 창에 완전한 경로 이름을 지정하는 경우 보통 Windows 방식대로 백슬래시를 사용합니다.

   a. **Audit.Handler.File.Directory** - 감사 로그를 저장할 위치를 지정합니다.
   b. **Audit.metadata.file.name** - 메타데이터 XML 파일의 완전한 경로 및 파일 이름을 지정합니다.

c. **Config.drivetable.file.url** – Encryption Key Manager에 알려진 드라이브에 대한 정보가 있는 위치를 지정합니다. 이 파일은 서버 또는 CLI 클라이언트를 시작하기 전에는 필요하지 않습니다. 파일이 없는 경우 Encryption Key Manager 서버의 시스템 종료 중 작성됩니다.

d. **TransportListener.ssl.keystore.name** - 1 단계에서 작성한 키 저장소의 파일 이름 및 경로를 지정합니다.

e. **TransportListener.ssl.truststore.name** - 1 단계에서 작성한 키 저장소의 파일 이름 및 경로를 지정합니다.

f. **Admin.ssl.keystore.name** - 1 단계에서 작성한 키 저장소의 파일 이름 및 경로를 지정합니다.

g. **Admin.ssl.truststore.name** - 1 단계에서 작성한 키 저장소의 파일 이름 및 경로를 지정합니다.

h. **config.keystore.file** - 1 단계에서 작성한 키 저장소의 파일 이름 및 경로를 지정합니다.

i. **drive.acceptUnknownDrives** - `true` 또는 `false`를 지정합니다. 값이 true이면 Encryption Key Manager에 연결한 새 테이프 드라이브를 드라이브 테이블에 자동으로 추가할 수 있습니다. 기본값은 false입니다.

5. 다음의 선택적 암호 항목은 추가하거나 생략할 수 있습니다. 이 항목이 **KeyManagerConfig.properties**에 지정되지 않은 경우 Encryption Key Manager에서 서버를 시작하는 중 키 저장소 암호를 묻는 프롬프트를 표시합니다.

   a. **Admin.ssl.keystore.password** - 1 단계에서 작성한 키 저장소 암호를 지정합니다.

   b. **config.keystore.password** - 1 단계에서 작성한 키 저장소 암호를 지정합니다.

   c. **TransportListener.ssl.keystore.password** - 1 단계에서 작성한 키 저장소 암호를 지정합니다.

   KeyManagerConfig.properties 파일에 추가되면 Encryption Key Manager는 추가 보안을 위해 이러한 암호를 인식할 수 없게 만듭니다.

6. 선택적으로 로컬 운영 체제 레지스트리에서 CLI 클라이언트 인증을 수행할 경우 **Server.authMechanism** 등록 정보를 `LocalOS` 값으로 설정하십시오. 이를 지정하지 않거나 `EKM`으로 설정하면 기본적으로 CLI 클라이언트 사용자가 Key Manager 서버에 사용자 이름/암호로 EKMAdmin/changeME를 사용하여 로그인합니다. 이 암호는 **chgpasswd** 명령으로 변경할 수 있습니다.

   Server.authMechanism 등록 정보를 LocalOS로 설정한 경우 Linux 플랫폼에서는 추가 설정이 필요합니다. 자세한 정보는 **http://support.dell.com** 또는 제품과

함께 제공되는 Dell Encryption Key Manager 매체에 있는 readme 파일을 참조하십시오. 5-6 페이지의 『CLI 클라이언트 사용자 인증』에서 자세한 정보를 제공합니다.

7. **KeyManagerConfig.properties**에서 변경한 사항을 저장하십시오.
8. Encryption Key Manager 서버를 시작하십시오. GUI를 사용하지 않고 서버를 시작하려면

   Windows의 경우

   cd c:₩ekm₩ekmserver를 탐색하여 **startServer.bat**를 클릭하십시오.

   Linux 플랫폼의 경우

   /var/ekm/ekmserver를 탐색하여 `. ./startServer.sh`를 입력하십시오.

   자세한 내용은 5-1 페이지의 『Key Manager 서버 시작, 새로 고침 및 중지』를 참조하십시오.
9. CLI 클라이언트를 시작하려면

   Windows의 경우

   cd c:₩ekm₩ekmclient를 탐색하여 **startClient.bat**를 클릭하십시오.

   Linux 플랫폼의 경우

   /var/ekm/ekmclient를 탐색하여 `. ./startClient.sh`를 입력하십시오.

   자세한 내용은 5-6 페이지의 『명령행 인터페이스 클라이언트』를 참조하십시오.
10. 4(i)단계에서 **drive.acceptUnknownDrives = false**를 지정한 경우 # 프롬프트에 다음을 입력하여 드라이브를 구성하십시오.

    `adddrive -drivename` *`drive_name`* `-rec1` *`cert_name`* `-rec2` *`cert_name`*

    예를 들어 다음과 같습니다.

    ```
    # adddrive -drivename 000001365054 -rec1 key1c1 -rec2 key1c2
    ```

    뒤에 다음이 나옵니다.

    ```
    # listdrives -drivename 000001365054
    ```

    다음을 리턴합니다.

    ```
    Entry Key:  SerialNumber = 000001365054

    Entry Key:  AliasTwo = key1c2

    Entry Key:  AliasOne = key1c1
     Deleted : false
     Updated : true
     TimeStamp : Sun Jul 03 17:34:44 MST 2007
    ```
11. # 프롬프트에 **listdrives** 명령을 입력하여 드라이브가 제대로 추가되었는지 확인하십시오.

제 5 장 Encryption Key Manager 관리

Key Manager 서버 시작, 새로 고침 및 중지

Encryption Key Manager 서버는 매우 쉽게 시작하고 중지할 수 있습니다.

서버를 새로 고치면 Encryption Key Manager에서 메모리에 있는 키 저장소, 드라이브 테이블 및 구성 정보의 현재 내용을 개별 파일로 덤프한 후 메모리로 다시 로드합니다. CLI 클라이언트를 사용하여 이러한 컴포넌트를 변경한 후에 새로 고치기를 실행하는 것이 좋습니다. 이러한 변경사항은 Encryption Key Manager 서버 시스템 종료 시 자동으로 저장되지만 서버를 새로 고치는 명령을 실행하면 시스템 충돌 또는 정전 시 변경 사항을 유실할 위험을 방지할 수 있습니다.

Dell Encryption Key Manager GUI에서 Encryption Key Manager 서버를 시작하십시오.

1. GUI를 아직 시작하지 않은 경우 GUI를 여십시오.

   Windows의 경우

   c:₩ekm₩gui를 탐색하여 **LaunchEKMGui.bat**를 클릭하십시오.

   Linux 플랫폼의 경우

   /var/ekm/gui를 탐색하여 `. ./LaunchEKMGui.sh`를 입력하십시오.
2. GUI 왼쪽에 있는 탐색기에서 **서버 상태 모니터(Server Health Monitor)**를 클릭하십시오.
3. 『서버 상태(Server Status)』 페이지(5-2 페이지의 그림 5-1)에서 **서버 시작(Start Server)** 또는 **서버 새로 고침(Refresh Server)**을 클릭하십시오.

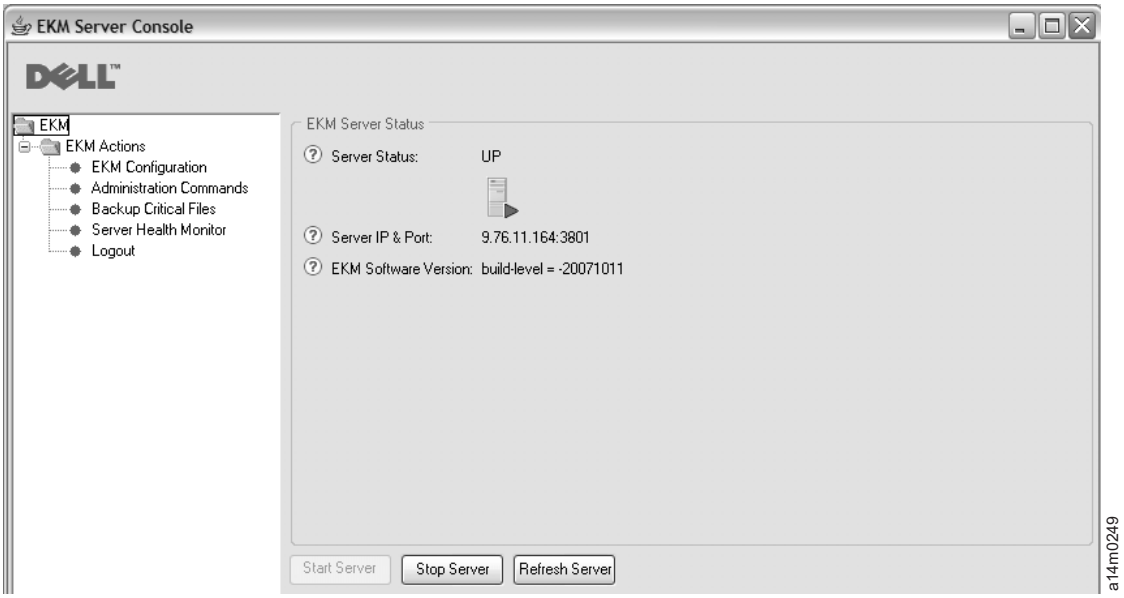

그림 5-1. 서버 상태

4. 서버 상태가 변경되면 서버 상태(Refresh Server) 창에 반영됩니다. 그림 5-1을 참조하십시오.
5. 로그인 창이 표시됩니다(그림 5-2).

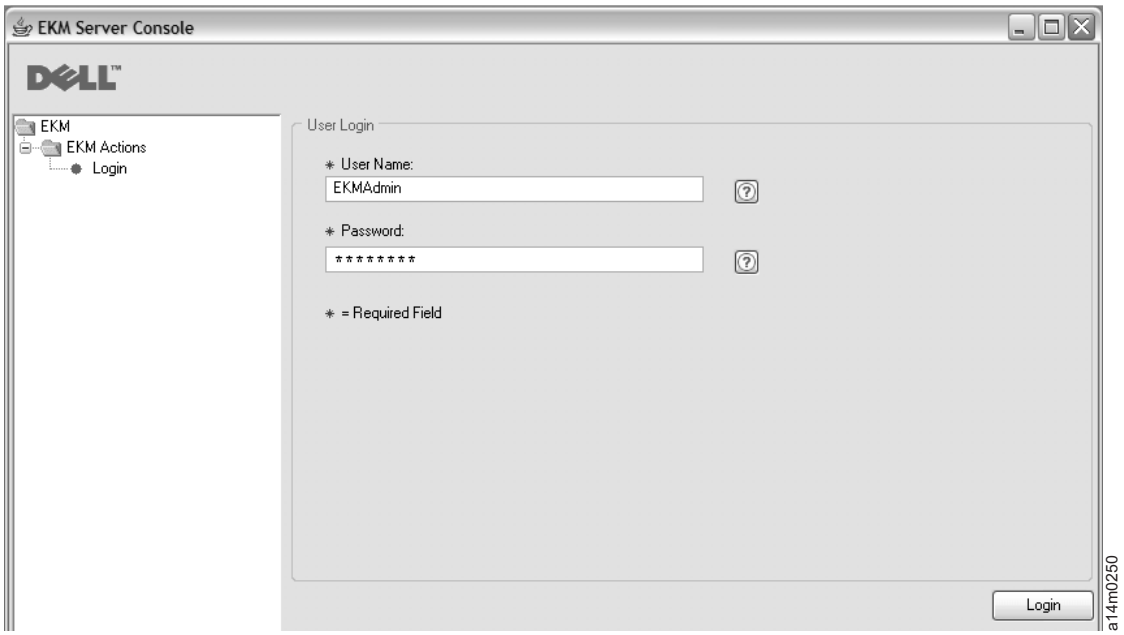

그림 5-2. 로그인 창

사용자 이름에 EKMAdmin을 입력합니다. 초기 암호는 changeME입니다. 로그인한 후 **chgpasswd** 명령을 사용하여 암호를 변경할 수 있습니다. 5-11 페이지의 『chgpasswd』를 참조하십시오.

주: • Dell Encryption Key Manager GUI가 호스트 IP 주소를 표시하지 못할 수 있습니다.

현재 GUI에 적용되는 다음과 같은 두 가지 제한사항 때문에 **서버 상태 모니터(Server Health Monitor)**에 Encryption Key Manager 호스트 IP 주소를 표시할 수 없습니다.

- 현재 응용프로그램이 IPV6를 인식하지 못하는 경우입니다. IPV6 주소를 사용하여 호스트를 구성하는 경우, Encryption Key Manager 응용프로그램이 IP 주소를 표시할 수 없습니다.
- Encryption Key Manager 응용프로그램이 Linux 시스템에 설치된 경우 응용프로그램은 로컬 호스트 주소는 표시하지만 실제의 활성 IP 포트는 표시하지 않습니다.

호스트 시스템의 실제 IP 주소를 검색하려면 네트워크 구성에 액세스하여 IP 포트 주소를 찾으십시오. Windows 시스템의 경우 명령 창을 열고 `ipconfig`를 입력하십시오. Linux의 경우 `isconfig`를 입력하십시오.

6. **로그인(Login)**을 클릭하십시오.

서버 중지 또한 서버 상태 페이지에서 할 수 있습니다.

스크립트를 사용하여 Key Manager 서버 시작

Windows의 경우

cd c:₩ekm₩ekmserver를 탐색하여 **startServer.bat**를 클릭하십시오.

Linux 플랫폼의 경우

/var/ekm/ekmserver를 탐색하여 `. ./startServer.sh`를 입력하십시오.

서버를 중지하려면 아래 5-6 페이지의 『명령행 인터페이스 클라이언트』에서 설명한 방법 중 하나를 사용하여 **stopekm** 명령을 실행하십시오. Key Manager 프로세스에 **sigterm**을 전송하는 것도 또 하나의 방법입니다. 그러면 서버가 시스템 종료되고 정상적으로 종료될 수 있습니다. Key Manager 프로세스에 **sigkill**은 전송하지 마십시오. **sigkill**은 프로세스를 정상적으로 종료시키지 못합니다. 예를 들어 Linux 시스템에서 `kill -SIGTERM` *pid* 또는 `kill -15` *pid*를 입력하십시오.

명령 프롬프트에서 Key Manager 서버 시작 및 중지

명령 창 또는 쉘을 사용하여 Encryption Key Manager 서버를 시작하려면 다음을 입력하십시오.

```
java com.ibm.keymanager.EKMLaunch KeymanagerConfig.properties
```

그러면 백그라운드에서 Encryption Key Manager 서버가 시작됩니다. 제대로 시작되면 `ps -ef | grep java` 명령(Linux 플랫폼) 또는 Windows 작업 관리자를 사용하

여 Encryption Key Manager Java 프로세스를 표시할 수 있습니다. Windows 서비스로 실행하면 LaunchEKMService로 표시됩니다.

서버를 중지하려면 아래 5-6 페이지의 『명령행 인터페이스 클라이언트』에서 설명한 방법 중 하나를 사용하여 **stopekm** 명령을 실행하십시오. Key Manager 프로세스에 **sigterm**을 전송하는 것도 또 하나의 방법입니다. 그러면 서버가 시스템 종료되고 정상적으로 종료될 수 있습니다. Key Manager 프로세스에 **sigkill**은 전송하지 마십시오. **sigkill**은 프로세스를 정상적으로 종료시키지 못합니다. 예를 들어 Linux 시스템에서 `kill -SIGTERM` *pid* 또는 `kill -15` *pid*를 입력하십시오.

Windows 플랫폼에서 Dell Encryption Key Manager가 Windows 서비스로 시작되는 경우 제어판에서 중지할 수 있습니다.

Key Manager 서버를 Windows 서비스로 설치

Encryption Key Manager 서버를 호스트 서버에서 서비스로 설치하는 경우 호스트 서버가 재부트될 때 Encryption Key Manager 서버 응용프로그램이 시작됩니다.

1. Dell 지원 웹 사이트(**http://support.dell.com**) 웹 사이트에서 다운로드한 릴리스의 LaunchEKMService.exe 실행 파일을 임시 디렉토리로 추출하십시오.
2. 서비스가 올바로 실행되게 하려면 일부 환경 변수를 설정해야 합니다.
   a. 시작 메뉴에서 **제어판**을 클릭하십시오.
   b. **시스템**을 두 번 클릭하십시오.
   c. **고급** 탭을 클릭하십시오.
   d. **환경 변수**를 클릭하십시오.
   e. 시스템 변수 목록에서 **새로 만들기**를 클릭하십시오.
   f. 변수 이름으로 `JAVA_HOME`을 지정하고 IBM JVM 디렉토리를 입력하십시오. 기본 설치 디렉토리는 `C:¥PROGRA~1¥IBM¥Java60`입니다.
   g. **확인**을 클릭하십시오.
3. 이 절차를 사용하여 시스템 PATH 변수를 편집하십시오.

   주: 명령행에서 PATH 변수를 설정하면 작동하지 않습니다.
   a. 시작 메뉴에서 **제어판**을 클릭하십시오.
   b. **시스템**을 두 번 클릭하십시오.
   c. **고급** 탭을 클릭하십시오.
   d. **환경 변수**를 클릭하십시오.
   e. 시스템 변수 목록을 스크롤하여 **Path** 변수를 찾은 다음 **편집**을 클릭하십시오.
   f. Path 변수의 시작 부분에 `IBM JVM` 경로를 추가하십시오. 기본 설치 디렉토리는 `C:¥PROGRA~1¥IBM¥Java60¥jre¥bin`입니다.

주: 경로 목록의 다른 디렉토리와 구분할 수 있도록 경로의 끝에 세미콜론을 삽입하십시오.

g. **확인**을 클릭하십시오.

4. Encryption Key Manager 서버 구성 등록 정보 파일의 경로가 완전한지 확인하십시오. 이 파일의 이름은 KeyManagerConfig.properties로 지정되어 C:₩ekm₩gui 디렉토리에 놓입니다. 다음 모든 파일의 경로를 확인하여 완전한 경로가 되도록 갱신해야 합니다. (예를 들어 gui₩EKMKeys.jck가 아니라 c:₩ekm₩gui₩EKMKeys.jck가 되어야 합니다.) 기본 설치 사용 시의 경로 변경 방법은 다음 예를 참조하십시오.

   이들은 기본 설치 및 키 저장소 이름을 사용할 때 가리켜야 하는 등록 정보 및 완전한 경로입니다. KeyManagerConfig.properties 파일에서 이들 각 항목을 찾아볼 수 있습니다.

   config.keygroup.xml.file
   올바른 경로: FILE:C:/ekm/gui/keygroups/KeyGroups.xml

   Admin.ssl.keystore.name
   올바른 경로: C:/ekm/gui/EKMKeys.jck

   TransportListener.ssl.truststore.name
   올바른 경로: C:/ekm/gui/EKMKeys.jck

   Audit.metadata.file.name
   올바른 경로: C:/ekm/gui/metadata/ekm_metadata.xml

   Audit.handler.file.directory
   올바른 경로: C:/ekm/gui/audit

   config.keystore.file
   올바른 경로: C:/ekm/gui/EKMKeys.jck

   TransportListener.ssl.keystore.name
   올바른 경로: C:/ekm/gui/EKMKeys.jck

   config.drivetable.file.url
   올바른 경로: FILE:C:/ekm/gui/drivetable/ekm_drivetable.dt

   Admin.ssl.truststore.name
   올바른 경로: C:/ekm/gui/EKMKeys.jck

5. **LaunchEKMServices.exe** 파일은 명령 프롬프트를 통해 실행해야 합니다. Windows에서 **시작 > 프로그램 > 보조 프로그램 > 명령 프롬프트** 순으로 이동하면 됩니다.

6. 명령 프롬프트에서, **LaunchEKMService.exe**를 추출한 임시 디렉토리로 이동하십시오. 다음 옵션을 참조하여 **LaunchEKMService.exe** 파일을 실행하십시오.

   LaunchEKMService {**-help** | **-i** *config_file* | **-u**}

-help

Displays사용법 정보를 표시합니다.

-i Encryption Key Manager를 Windows 서비스로 설치합니다. 이 옵션의 경우 구성 등록 정보 파일의 전체 경로 이름을 인수로 전달해야 합니다. 기본 경로 및 파일 이름은 C:₩ekm₩gui₩KeyManagerConfig.properties입니다.

-u 더 이상 Key Manager Windows 서비스를 서비스로 실행할 필요가 없는 경우 이를 설치 제거합니다. 설치 제거하기 전에 반드시 EKMServer 서비스를 중지해야 합니다. 이 명령을 실행하면 Could not remove EKMServer. Error 0 과 같은 오류 메시지가 표시될 수 있습니다. 이 메시지가 표시되도 서비스를 설치 제거할 수 있습니다.

Encryption Key Manager를 Windows 서비스로 설치하려면 다음 명령을 실행하십시오.

```
LaunchEKMService.exe -i config file
```

7. 위의 명령을 사용하여 서비스가 설치되면 서비스 제어판에 EKMServer가 표시되며 서비스 제어판을 사용하여 Encryption Key Manager를 시작 및 중지할 수 있습니다.

   주: 제어판을 사용하여 Windows 서비스를 처음 사용하는 경우 수동으로 Windows 서비스를 시작해야 합니다.

명령행 인터페이스 클라이언트

Encryption Key Manager 서버가 시작되면 로컬 또는 원격으로 클라이언트 인터페이스를 통해 CLI 명령을 실행할 수 있습니다. CLI 명령을 실행하려면 먼저 CLI 클라이언트를 시작해야 합니다.

CLI 클라이언트 사용자 인증

로컬/원격 클라이언트에서 사용할 인증 메커니즘은 구성 파일의 Server.authMechanism 등록 정보에서 지정합니다. 값이 EKM으로 설정되면 CLI 클라이언트 사용자는 사용자 이름/암호를 EKMAdmin/changeME로 사용하여 서버에 로그인해야 합니다. 이 암호는 **chgpasswd** 명령으로 변경할 수 있습니다. 5-11 페이지의 『chgpasswd』를 참조하십시오. Server.authMechanism property의 기본 설정은 EKM입니다.

KeyManagerConfig.properties 파일에서 Server.authMechanism 등록 정보 값이 LocalOS로 지정되면 클라이언트 인증은 로컬 운영 체제 레지스트리에서 수행됩니다. CLI 클라이언트 사용자는 OS 사용자 이름/암호를 사용하여 서버에 로그인해야 합니다. 서버에 로그인하여 명령을 제출할 수 있도록 허용된 사용자/암호만이 서버를 실행하고 superuser/루트 권한을 갖습니다.

중요사항: Encryption Key Manager 구성 파일을 이와 같이 변경하는 경우에는 Encryption Key Manager 서버를 중지하고 GUI를 닫아야 합니다.

Windows의 운영 체제 기반 인증의 경우 KeyManagerConfig.properties에서 Server.authMechanism=LocalOS를 다음과 같이 설정하십시오.

1. KeyManagerConfig.properties 파일(c:₩ekm₩gui 디렉토리)을 찾으십시오.
2. 원하는 텍스트 편집기(WordPad가 권장됨)를 사용하여 파일을 여십시오.
3. Server.authMechanism 문자열을 찾으십시오. 이 문자열이 없는 경우 Server.authMechanism=LocalOS 형식으로 파일에 문자열을 추가하십시오.
4. 파일을 저장하십시오.

이제 Encryption Key Manager 서버의 사용자 ID와 암호가 OS 사용자 계정과 일치합니다. 서버에 로그인하여 명령을 제출할 수 있고 관리자 특권을 갖는 사용자만이 Encryption Key Manager 서버를 관리할 수 있습니다.

Linux 플랫폼에서의 로컬 운영 체제에 기반한 인증 시 다음과 같은 추가 단계를 수행하십시오.

1. **http://support.dell.com**에서 Dell 릴리스 R175158(EKMServicesAndSamples)을 다운로드하여 파일을 원하는 디렉토리에 추출하십시오.
2. 다운로드에서 LocalOS 디렉토리를 찾으십시오.
3. 플랫폼에 적합한 JVM-JaasSetup 디렉토리에서 *java_home*/jre/bin으로 libjaasauth.so 파일을 복사하십시오.
   - 32비트 Intel Linux 환경에서 LocalOS-setup/linux_ia32/libjaasauth.so 파일을 *java_home*/jre/bin/ 디렉토리로 복사하십시오. 여기서 *java_home*은 보통 1.6 JVM을 실행하는 32비트 Intel Linux 커널의 경우 *java_install_path*/IBMJava-i386-60입니다.
   - 64비트 AMD64 Linux 환경에서 LocalOS-setup/linux-x86_64/libjaasauth.so 파일을 *java_home*/jre/bin/ 디렉토리로 복사하십시오. 여기서 *java_home*은 보통 1.6 JVM을 실행하는 64비트 Linux 커널의 경우 *java_install_path*/IBMJava-x86_64-60입니다.

Windows 플랫폼의 경우 이 파일은 필요하지 않습니다.

설치가 완료되면 Encryption Key Manager 서버를 시작할 수 있습니다. 이제 Encryption Key Manager 클라이언트는 운영 체제 기반 사용자/암호를 사용하여 로그인할 수 있습니다. 서버에 로그인하여 명령을 제출할 수 있도록 허용된 사용자 ID만이 서버를 실행하고 superuser/루트 권한을 갖습니다.

Dell 제품 매체에 포함되어 있으며 **http://support.dell.com**에도 제공되는 readme 파일에서는 추가 설치 세부사항을 제공합니다.

명령행 인터페이스 클라이언트 시작

주: Encryption Key Manager 서버와 Encryption Key Manager CLI 클라이언트 등록 정보 파일의 TransportListener.ssl.port 등록 정보가 모두 같은 값으로 설정되어 있어야 합니다. 그렇지 않으면 서로 통신할 수 없습니다. 문제점이 발생하면 6-2 페이지의 『CLI 클라이언트 및 EKM 서버 간 통신 문제점 디버깅』을 참조하십시오.

Encryption Key Manager CLI 클라이언트와 Encryption Key Manager 서버는 통신 보안에 SSL을 사용합니다. 클라이언트 인증이 없는 기본 JSSE 구성을 사용하는 경우 Encryption Key Manager 서버의 TransportListener.ssl.keystore에 있는 인증이 TransportListener.ssl.truststore에 존재해야 합니다. 이러한 방식으로 클라이언트는 서버를 신뢰할 수 있음을 알게 됩니다. Encryption Key Manager CLI 클라이언트가 Encryption Key Manager 서버와 동일한 시스템에서 실행되는 경우 동일한 구성 등록 정보 파일을 사용할 수 있습니다. 이로써 Encryption Key Manager CLI 클라이언트는 Encryption Key Manager 서버와 동일한 키 저장소/신뢰 저장소 구성을 사용할 수 있습니다. 이 클라이언트와 서버가 동일한 시스템에 없거나 클라이언트가 다른 키 저장소를 사용하게 하려는 경우 Encryption Key Manager 서버 구성 등록 정보 파일에 지정된 TransportListener.ssl.keystore에서 인증서를 내보내십시오. 이들 인증서는 Encryption Key Manager CLI 등록 정보 파일의 TransportListener.ssl.truststore에서 지정하는 신뢰 저장소로 가져와야 합니다.

네 가지 방법으로 CLI 클라이언트를 시작하고 CLI 명령을 실행할 수 있습니다. 선택한 방법에 상관없이 CLI 구성 파일의 이름은 반드시 지정해야 합니다. 자세한 내용은 부록 B를 참조하십시오.

스크립트 사용

Windows의 경우

cd c:₩ekm₩ekmclient를 탐색하여 **startClient.bat**를 클릭하십시오.

Linux 플랫폼의 경우

/var/ekm/ekmclient를 탐색하여 `. ./startClient.sh`를 입력하십시오.

Interactively

명령 창 또는 쉘에서 대화식으로 명령을 실행하려면 다음을 입력하십시오.

```
java com.ibm.keymanager.KMSAdminCmd CLIconfiglfile_name -i
```

프롬프트가 나타납니다. 명령을 제출하기 전에 먼저 다음 명령을 사용하여 CLI 클라이언트로 Key Manager 서버에 로그인해야 합니다.

```
#login –ekmuser EKMAdmin –ekmpassword changeME
```

CLI 클라이언트가 Key Manager 서버에 로그인하면 CLI 명령을 실행할 수 있습니다. 작업을 마치고 CLI 클라이언트를 종료하려면 **quit** 또는 **logout** 명령을 사용

합니다. Encryption Key Manager 서버는 기본적으로 10분이 경과한 후에 사용하지 않은 클라이언트의 통신 소켓을 닫습니다. 그 이후에 명령을 입력하려고 하면 클라이언트가 종료됩니다. Encryption Key Manager 서버-클라이언트 소켓의 시간 종료 기간을 더 길게 지정하려면 KeyManagerConfig.properties 파일의 TransportListener.ssl.timeout 등록 정보를 수정하십시오.

명령 파일 사용

파일의 일괄처리 명령을 Key Manager 서버에 제출하려면 실행할 명령을 포함하는 파일(예: *clifile*)을 작성합니다. 명령을 실행하려면 먼저 클라이언트가 로그인해야 하므로 이 파일의 첫 번째 명령은 **login** 명령이어야 합니다. 예를 들어 clifile에는 다음 내용이 포함될 수 있습니다.

```
login -ekmuser EKMAdmin -ekmpassword changeME
listdrives
```

그 다음 이 명령 파일을 실행하려면 CLI 클라이언트를 시작합니다.

```
java com.ibm.keymanager.admin.KMSAdminCmd CLIconfiglfile_name –filename clifile
```

한 번에 하나의 명령

각 명령에서 CLI userid_ID 및 암호를 지정하여 한 번에 하나의 명령을 실행할 수 있습니다. 명령 창 또는 쉘에서 다음을 입력하십시오.

```
java com.ibm.keymanager.KMSAdminCmd ClientConfig.properties_name -listdrives
    -ekmuser EKMAdmin -ekmpassword changeME
```

이 암호는 **chgpasswd** 명령으로 변경할 수 있습니다. 명령이 실행되고 클라이언트 세션이 종료됩니다.

CLI 명령

Encryption Key Manager에서는 다음 명령을 포함하여 명령행 인터페이스 클라이언트에서 Encryption Key Manager 서버와 상호 작용할 때 사용할 수 있는 명령 세트를 제공합니다.

addaliastogroup

기존(소스) 키 그룹에서 새(대상) 키 그룹으로 특정 별명을 복사하십시오. 하나의 키 그룹에 있는 별명을 다른 키 그룹에 추가하려는 경우 유용합니다.

addaliastogroup -aliasID *aliasname* **-sourceGroupID** *groupname* **-targetGroupID** *groupname*

-aliasID

추가할 키의 *aliasname*입니다.

-sourceGroupID

별명을 복사할 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

-targetGroupID
별명이 추가될 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

예제: addaliastogroup -aliasID aliasname -sourceGroupID keygroup1 -targetGroupID keygroup2

adddrive

새 드라이브를 Key Manager 드라이브 테이블에 추가합니다. 테이프 드라이브를 드라이브 테이블에 자동으로 추가하는 방법은 4-1 페이지의 『테이프 드라이브 테이블 자동 업데이트』를 참조하십시오. 별명 요구사항에 대한 정보는 2-4 페이지의 『암호화 키 및 LTO 4 및 LTO 5 테이프 드라이브』를 참조하십시오.

adddrive -drivename *drivename* [**-rec1** *alias*] [**-rec2** *alias*][**-symrec** *alias*]

-drivename
*drivename*은 추가할 드라이브의 12자리 일련 번호를 지정합니다.

주: 총 12자리가 되도록 일련 번호 10자리 앞에 0을 두 개 추가합니다.

-rec1
드라이브 인증서의 *alias*(또는 키 레이블)를 지정합니다.

-rec2
드라이브 인증서의 두 번째 *alias*(또는 키 레이블)를 지정합니다.

-symrec
테이프 드라이브에서 대칭 키의 *alias* 또는 키 그룹 이름을 지정합니다.

예제: adddrive -drivename 000123456789 -rec1 alias1 -rec2 alias2

addkeygroup

키 그룹 XML에서 고유한 그룹 ID를 사용하여 키 그룹의 인스턴스를 작성합니다.

addkeygroup -groupID *groupname*

-groupID
키 그룹 XML 파일에서 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

예제: addkeygroup -groupID keygroup1

addkeygroupalias

특정 키 그룹 ID 이외에도 키 저장소에 있는 기존 키 별명에 대한 새 별명을 작성합니다.

addkeygroupalias -alias *aliasname* **-groupID** *groupname*

-alias

키의 새 *aliasname*입니다.

-groupID

키 그룹 XML 파일에서 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

예제: addkeygroupalias -alias aliasname -groupID keygroup1

chgpasswd

CLI 클라이언트 사용자(EKMAdmin) 기본 암호를 변경합니다.

chgpasswd -new *password*

-new

이전 암호를 바꾸는 새 *password*입니다.

예제: chgpasswd -new ebw74jxr

createkeygroup

KeyGroups.xml 파일에서 최초의 키 그룹 오브젝트를 작성합니다. 한 번만 실행하십시오.

createkeygroup -password *password*

-password

나중에 검색하기 위해 KeyGroups.xml 파일에서 키 저장소 암호를 암호화하는 데 사용되는 *password*입니다. 키 저장소는 키 그룹의 키를 암호화한 후 차례대로 각 개별 키 그룹 별명 암호를 암호화합니다. 따라서 KeyGroups.xml 파일에는 일반 형식의 키는 존재하지 않습니다.

예제: createkeygroup -password password

deletedrive

Key Manager 드라이브 테이블에서 드라이브를 삭제합니다. 이와 동일한 명령으로 **deldrive** 및 **removedrive**가 있습니다.

deletedrive -drivename *drivename*

-drivename

*drivename*은 삭제할 드라이브의 일련 번호를 지정합니다.

예제: deletedrive -drivename 000123456789

delgroupalias

키 그룹에서 키 별명을 삭제합니다.

delgroupalias -groupID *groupname* **-alias** *aliasname*

-groupID
KeyGroups.xml 파일에서 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

-alias
제거할 키 별명의 *aliasname*입니다.

예제: delgroupalias -groupID keygroup1 -alias aliasname

delkeygroup

전체 키 그룹을 삭제합니다.

delkeygroup -groupID *groupname*

-groupID
KeyGroups.xml 파일에서 그룹을 식별하는 데 사용되는 고유한 *groupname*입니다.

예제: delkeygroup -groupID keygroup1

exit

CLI 클라이언트를 종료하고 Encryption Key Manager 서버를 중지합니다. 이와 같은 명령으로 **quit**가 있습니다.

예제: exit

export

드라이브 테이블 또는 Encryption Key Manager 서버 구성 파일을 지정된 URL로 내보냅니다.

export {**-drivetab**|**-config**} **-url** *urlname*

-drivetab
드라이브 테이블을 내보냅니다.

-config
Encryption Key Manager 서버 구성 파일을 내보냅니다.

-url
*urlname*은 파일을 쓸 위치를 지정합니다.

예제: export -drivetab -url FILE:///keymanager/data/export.table

help

명령행 인터페이스의 명령어 및 구문을 표시합니다. 이와 같은 명령으로 **?**가 있습니다.

help

import

지정된 URL에서 드라이브 테이블 또는 구성 파일을 가져옵니다.

import {**-merge**|**-rewrite**} {**-drivetab**|**-config**} **-url** *urlname*

-merge
새 데이터를 현재 데이터와 병합합니다.

-rewrite
현재 데이터를 새 데이터로 바꿉니다.

-drivetab
드라이브 테이블을 가져옵니다.

-config
구성 파일을 가져옵니다.

-url
*urlname*은 새 데이터를 가져올 위치를 지정합니다.

예제: import -merge -drivetab -url FILE:///keymanager/data/export.table

list

config.keystore.file 등록 정보에서 이름을 지정한 키 저장소에 포함된 인증서를 나열합니다.

list [**-cert** |**-key**|**-keysym**][**-alias** *alias* **-verbose** |**-v**]

-cert
지정된 키 저장소에 있는 인증서를 나열합니다.

-key
지정된 키 저장소에 있는 모든 키를 나열합니다.

-keysym
지정된 키 저장소에 있는 대칭 키를 나열합니다.

-alias
*alias*는 특정 인증서를 나열하도록 지정합니다.

-verbose|-v
인증서에 대한 자세한 정보를 표시합니다.

예제:

`list -v`는 키 저장소의 모든 항목을 나열합니다.

`list -alias mycert -v`는 config.keystore.file 키 저장소에 인증서가 있는 경우 mycert 별명에서 사용 가능한 모든 데이터를 나열합니다.

listcerts

config.keystore.file 등록 정보에서 이름을 지정한 키 저장소에 포함된 인증서를 나열합니다.

listcerts [**-alias** *alias* **-verbose** |**-v**]

-alias

*alias*는 특정 인증서를 나열하도록 지정합니다.

-verbose|-v

인증서에 대한 자세한 정보를 표시합니다.

예제: listcerts -alias alias1 -v

listconfig

메모리에 있는 Encryption Key Manager 서버 구성 등록 정보를 나열합니다. 이는 KeyManagerConfig.properties 파일의 현재 내용 및 **modconfig** 명령을 사용하여 업데이트한 내용을 반영합니다.

listconfig

listdrives

드라이브 테이블에 있는 드라이브를 나열합니다.

listdrives [**-drivename** *drivename*]

-drivename

*drivename*은 나열할 테이프 드라이브의 일련 번호를 지정합니다.

-verbose|-v

테이프 드라이브에 대한 자세한 정보를 표시합니다.

예제: listdrives -drivename 000123456789

login

Encryption Key Manager 서버에서 CLI 클라이언트로 로그인합니다.

login -ekmuser *userID* **-ekmpassword** *password*

-ekmuser

사용한 인증 유형에 따라 *userID*에 EKMadmin 또는 localOS 사용자 ID 값을 지정합니다(5-6 페이지의 『CLI 클라이언트 사용자 인증』 참조).

-ekmpassword

사용자 ID의 올바른 암호입니다.

예제: login -ekmuser EKMAdmin -ekmpassword changeME

logout

현재 사용자를 로그오프합니다. 이와 같은 명령으로 **logoff**가 있습니다. 이 명령은 클라이언트 세션이 설정된 경우에만 사용 가능합니다.

예제: logout

modconfig

Encryption Key Manager 서버 구성 등록 정보 파일(KeyManagerConfig.properties)의 등록 정보를 수정합니다. 이와 같은 명령으로 **modifyconfig**가 있습니다.

modconfig {**-set** | **-unset**} **-property** *name* **-value** *value*

-set

지정된 등록 정보를 지정된 값으로 설정합니다.

-unset

지정된 등록 정보를 제거합니다.

-property

*name*은 대상 등록 정보 이름을 지정합니다.

-value

*value*은 **-set**가 지정된 경우 대상 등록 정보에 새 값을 지정합니다.

예제: modconfig -set -property sync.timeinhours -value 24

moddrive

드라이브 테이블에서 드라이브 정보를 수정합니다. 이와 같은 명령으로 **modifydrive**가 있습니다.

moddrive -drivename *drivename* {**-rec1** [*alias*] | **-rec2** [*alias*]| **-symrec** [*alias*]}

-drivename

*drivename*은 테이프 드라이브의 일련 번호를 지정합니다.

-rec1

드라이브 인증서의 *alias*(또는 키 레이블)를 지정합니다.

-rec2

드라이브 인증서의 두 번째 *alias*(또는 키 레이블)를 지정합니다.

-symrec

테이프 드라이브에서 대칭 키의 *alias* 또는 키 그룹 이름을 지정합니다.

예제: moddrive -drivename 000123456789 -rec1 newalias1

refresh

최신 구성 매개변수를 사용하여 디버그, 감사 및 드라이브 테이블 값을 새로 고치도록 Encryption Key Manager에 지시합니다.

예제: refresh

refreshks

키 저장소를 새로 고칩니다. 이 명령은 Encryption Key Manager 서버를 실행하는 동안 키 저장소가 수정된 경우 **config.keystore.file**에 지정된 키 저장소를 다시 로드하는 데 사용됩니다. 이 명령은 성능을 떨어뜨릴 수 있으므로 필요한 경우에만 사용합니다.

예제: refreshks

status

Key Manager 서버의 시작 또는 중지 여부를 표시합니다.

예제: status

stopekm

Encryption Key Manager 서버를 중지합니다.

예제: stopekm

sync

구성 파일 등록 정보나 드라이브 테이블 정보 또는 다른 Encryption Key Manager 서버에서 두 항목 모두를 명령을 실행한 Key Manager와 동기화합니다.

주: 두 동기화 방법 모두 키 저장소 또는 KeyGroups.xml 파일에서는 작동하지 않습니다. 따라서 수동으로 복사해야 합니다.

sync {**-all** | **-config** | **-drivetab**} **-ipaddr** *ip_addr* :*ssl*:*port* [**-merge** | **-rewrite**]

-all

구성 등록 정보 파일 및 드라이브 테이블 정보 모두를 **-ipaddr**에서 지정한 Encryption Key Manager 서버로 전송합니다.

-config

구성 등록 정보 파일만 **-ipaddr**에서 지정한 Encryption Key Manager 서버로 전송합니다.

-drivetab

드라이브 테이블 정보만 **-ipaddr**에서 지정한 Encryption Key Manager 서버로 전송합니다.

-ipaddr

*ip_addr:ssl:port*는 수신측 Encryption Key Manager 서버의 주소 및 SSL 포트를 지정합니다. *ssl:port*는 수신측 서버 KeyManagerConfig.properties 파일의 『TransportListener.ssl.port』에 지정된 값과 일치해야 합니다.

-merge

새 드라이브 테이블 데이터를 현재 데이터와 병합합니다. 구성 파일은 항상 다시 쓰기가 가능합니다. 기본값입니다.

-rewrite

현재 데이터를 새 데이터로 바꿉니다.

예제: sync -drivetab -ipaddr remoteekm.ibm.com:443 -merge

version

Encryption Key Manager 서버의 버전을 표시합니다.

예제: version

제 6 장 문제점 판별

개별 컴포넌트, 여러 개의 컴포넌트 또는 Encryption Key Manager의 모든 컴포넌트에서 디버깅을 설정할 수 있습니다.

다음의 중요 파일에서 Encryption Key Manager 서버 문제점 확인

Encryption Key Manager가 시작되지 않은 경우 다음 세 개 파일을 확인하여 문제점의 원인을 판별하십시오.

- **native_stdout.log** 및 **native_stderr.log**
  - Encryption Key Manager 서버는 백그라운드 프로세스로 실행되기 때문에 일반적인 정보 메시지와 오류 메시지를 표시할 콘솔이 없습니다. 이들 메시지는 native_stdout.log 및 native_stderr.log 파일에 기록됩니다.
  - Encryption Key Manager 서버 등록 정보 파일에 **debug.output.file** 등록 정보가 포함되어 있는 경우, native_stdout.log 및 native_stderr.log 파일은 디버그 로그와 동일한 디렉토리에 작성됩니다.
  - Encryption Key Manager 서버 등록 정보 파일에 **debug.output.file** 등록 정보가 포함되지 않은 경우, native_stdout.log 및 native_stderr.log 파일은 작업 디렉토리에 작성됩니다.
  - native_stdout.log 및 native_stderr.log 파일은 Encryption Key Manager 서버가 시작될 때마다 삭제 및 다시 작성됩니다.
- 감사 로그
  - 감사 로그에는 Encryption Key Manager가 처리될 때 기록된 레코드가 포함되어 있습니다.
  - 이 파일의 위치는 Encryption Key Manager 서버 구성 등록 정보 파일인 **KeyManagerConfig.properties**의 두 가지 등록 정보에 의해 지정됩니다.
    - Audit.handler.file.directory – 감사 로그가 위치해야 하는 디렉토리를 지정합니다.
    - Audit.handler.file.name – 감사 로그의 파일 이름을 지정합니다.
  - 감사에 대한 자세한 정보는 7-1 페이지의 제 7 장『감사 레코드』를 참조하십시오.

127자를 초과하는 키 스토어 암호 항목을 로그합니다.

Encryption Key Manager가 Windows 서비스로 설치되고 KeyManagerConfig.properties에 있는 키 저장소 암호의 길이가 128자 이상인 경우, 허용 가능한 암호 길이를 확인

할 방법이 없기 때문에 Encryption Key Manager가 시작에 실패합니다. 기본 Encryption Key Manager 로그에는 다음과 유사한 항목이 포함됩니다.

native_stdout.log

```
Server initialized
Default keystore failed to load
```

native_stderr.log

```
at com.ibm.keymanager.KeyManagerException: Default keystore failed to load
at com.ibm.keymanager.keygroups.KeyGroupManager.loadDefaultKeyStore
(KeyGroupManager.java:145)
at com.ibm.keymanager.keygroups.KeyGroupManager.init
(KeyGroupManager.java:605)
at com.ibm.keymanager.EKMServer.c(EKMServer.java:243)
at com.ibm.keymanager.EKMServer.<init>(EKMServer.java:753)
at com.ibm.keymanager.EKMServer.a(EKMServer.java:716)
at com.ibm.keymanager.EKMServer.main(EKMServer.java:129)
```

CLI 클라이언트 및 EKM 서버 간 통신 문제점 디버깅

EKM CLI 클라이언트 및 EKM 서버 간 통신은 서버 및 클라이언트 구성 등록 정보 파일 모두의 TransportListener.ssl.port 등록 정보에 지정된 포트를 통해 수행되며 SSL로 보호됩니다.

다음은 클라이언트를 EKM 서버에 연결할 수 없는 몇 가지 가능한 이유에 대한 목록입니다. 여기에는 문제점을 판별하고 이를 정정하는 방법에 대한 단계도 있습니다.

- EKM 서버가 실행 중이 아니므로 클라이언트와 통신할 대상이 없습니다.
  1. 명령 창에서 **netstat –an**을 실행하고 EKM 서버 등록 정보 파일의 TransportListener.ssl.port 및 TransportListener.tcp.port 등록 정보에 지정된 포트가 표시되는지 확인하십시오. 포트가 표시되지 않으면 서버가 실행 중이 아닌 것입니다.
- EKM CLI 클라이언트 등록 정보 파일의 TransportListener.ssl.host 등록 정보에서 EKM 서버가 실행 중인 올바른 호스트를 가리키지 않습니다.
  1. EKM CLI 클라이언트 등록 정보 파일의 TransportListener.ssl.host 등록 정보 값은 기본적으로 `localhost`입니다. 올바른 호스트를 가리키도록 이 등록 정보 값을 수정하십시오.
- EKM 서버 및 EKM CLI 클라이언트가 같은 포트에서 통신하지 않습니다.
  1. EKM 서버 및 EKM CLI 클라이언트 모두의 TransportListener.ssl.port 등록 정보가 같은 값으로 설정되었는지 확인하십시오.
- EKM 서버 및 EKM CLI 클라이언트에서 통신을 보안하는 데 사용하는 공통 인증서를 찾을 수 없습니다.

1. CLI 클라이언트 등록 정보의 TransportListener.ssl.keystore 및 TransportListener.ssl.truststore에 지정되어 있는 키 저장소에 서버 등록 정보의 Admin.ssl.keystore 및 Admin.ssl.truststore 키 저장소와 동일한 인증서가 포함되어 있는지 확인하십시오.
2. 클라이언트 등록 정보의 TransportListener.ssl.keystore.password에 올바른 암호가 있는지 확인하십시오.
3. 이 키 저장소의 인증서가 모두 만기되지 않았는지 확인하십시오. JSSE는 통신을 보안할 때 만기된 인증서는 사용하지 않습니다.

- EKM CLI 클라이언트 등록 정보 파일이 읽기 전용입니다.
  1. 파일에 대한 권한 또는 속성을 보고 EKM CLI 클라이언트를 실행하는 사용자가 파일에 대한 액세스 및 수정 권한이 있는지 확인하십시오.
- EKM 서버 등록 정보 파일에 Server.authMechanism = LocalOS이 있지만 EKMServicesAndSamples 패키지에서 요청한 파일이 설치되지 않았거나 잘못된 위치에 설치되었습니다.
  1. 인증에 대한 자세한 정보는 EKMServiceAndSamples 패키지에 포함된 readme를 참조하십시오.

Key Manager 서버 문제점 디버깅

Key Manager에서 발생하는 대부분의 문제점은 Key Manager 서버 시작 또는 구성과 관련이 있습니다. 디버그 등록 정보를 지정하는 방법에 대한 정보는 부록 B, 기본 구성 파일을 참조하십시오.

Encryption Key Manager가 시작되지 않으면 방화벽을 확인하십시오.

소프트웨어 방화벽 또는 하드웨어 방화벽에서 포트에 액세스하는 Encryption Key Manager을 차단할 수도 있습니다.

EKM server not started. EKM.properties config could not be loaded or found.

1. 이 오류는 등록 정보 파일이 기본 경로에 없는 경우, **KeyManagerConfig.properties**의 전체 경로를 지정하지 않고 KMSAdminCmd 또는 EKMLaunch를 시작할 때 발생합니다.

   Windows의 기본 경로는 **C:/Program Files/IBM/KeyManagerServer/**입니다.

   Linux 플랫폼의 기본 경로는 **/opt/ibm/KeyManagerServer/**입니다.
2. **KeyManagerConfig.properties** 파일의 전체 경로를 포함하고 KMSAdminCmd를 시작하는 명령을 다시 입력하십시오. 자세한 정보는 부록 B, 『Encryption Key Manager 구성 등록 정보 파일』을 참조하십시오.

EKM server is not started. File name for XML metadata file needs to be specified in the configuration file.

Audit.metadata.file.name 항목이 구성 파일에서 누락되었습니다.

이 문제점을 정정하려면 **KeyManagerConfig.properties** 구성 파일에 Audit.metadata.file.name 등록 정보를 추가하십시오.

Failed to start EKM.Mykeys. The system cannot find the specified file.

1. 이 오류 메시지는 **KeyManagerConfig.properties**의 키 저장소 항목이 기존 파일을 가리키지 않을 때 발생합니다.
2. 이 문제점을 정정하려면 **KeyManagerConfig.properties** 파일에서 다음 항목이 기존의 올바른 키 저장소 파일을 가리켜야 합니다.

   Admin.ssl.keystore.name

   TransportListener.ssl.truststore.name

   TransportListener.ssl.keystore.name

   Admin.ssl.truststore.name

   자세한 정보는 부록 B, 『Encryption Key Manager 구성 등록 정보 파일』을 참조하십시오.

Failed to start EKM. File does not exist = safkeyring://xxx/yyy

Encryption Key Manager 환경 쉘 스크립트의 IJO에 잘못된 제공업체를 지정하면 이 오류가 발생할 수 있습니다.

JCECCARACFKS 키 저장소의 경우 다음을 사용하십시오.

```
-Djava.protocol.handler.pkgs=com.ibm.crypto.hdwrCCA.provider
```

JCERACFKS 키 저장소의 경우 다음을 사용하십시오.

```
-Djava.protocol.handler.pkgs=com.ibm.crypto.provider
```

Failed to start EKM. keystore was tampered with, or password was incorrect.

1. 이 오류는 등록 정보 파일(부록 B, 『Encryption Key Manager 구성 등록 정보 파일』)에서 하나 이상의 항목에 잘못된 값이 설정된 경우 발생합니다.

   config.keystore.password(config.keystore.file에 해당)

   admin.keystore.password(admin.keystore.name에 해당)

   transportListener.keystore.password(transportListener.keystore.name에 해당)
2. 이 오류는 서버 시작 시 암호 프롬프트에 잘못된 암호를 입력한 경우 발생합니다.

3. 구성에 암호가 없는 경우 등록 정보 파일에 있는 3개 키 저장소 항목이 모두 고유하면 3회까지 암호를 묻는 프롬프트가 표시됩니다. 등록 정보의 항목 모두가 같으면 한 번만 프롬프트가 표시됩니다.

Failed to start EKM. Invalid keystore format.

1. 이 오류는 등록 정보 파일의 키 저장소 항목 중 하나에 잘못된 키 저장소 유형이 지정된 경우 발생합니다.
2. 등록 정보 파일의 모든 키 저장소 항목이 같은 파일을 가리키면 Encryption Key Manager는 모든 키 저장소의 키 저장소 유형으로 config.keystore.type 값을 사용합니다.
3. 등록 정보 파일에 특정 키 저장소에 대한 유형 항목이 없으면 Encryption Key Manager는 유형을 `jceks`로 가정합니다.

Failed to start the server. Listener thread is not up and running.

다음은 이 오류에 대한 몇 가지 가능한 이유입니다.

1. **KeyManagerConfig.properties** 파일에서 다음의 두 항목이 같은 포트를 가리킵니다.

   TransportListener.ssl.port

   TransportListener.tcp.port

   각 전송 리스너는 고유한 포트에서 청취하도록 구성되어야 합니다.
2. 하나 또는 두 항목 모두 Key Manager 서버와 같은 시스템에서 실행 중인 다른 서비스에서 이미 사용하고 있는 포트로 구성되었습니다. 다른 서비스에서 사용하고 있지 않은 포트를 찾아 이 포트를 사용하도록 Key Manager 서버를 구성하십시오.
3. Linux 운영 체제를 실행하는 시스템의 경우 하나 또는 두 포트 모두 1024보다 낮고 Key Manager 서버를 시작하는 사용자가 루트가 아니면 이 오류가 발생합니다. 1024가 넘는 포트를 사용하도록 **KeyManagerConfig.properties**의 전송 리스너 항목을 수정하십시오.

“[Fatal Error] :-1:-1: Premature end of file.” message in native_stderr.log.

Encryption Key Manager가 공백 키 그룹 파일을 로드하는 경우 이 메시지가 발생합니다. 이 메시지는 XML 구문 분석기에서 발행하는 것으로, 키 그룹을 사용하도록 구성되어 있지 않고 Encryption Key Manager 서버 등록 정보 파일, **KeyManagerConfig.properties**의 config.keygroup.xml.file 등록 정보에서 지정한 파일이 손상된 경우가 아니면 이로 인해 Encryption Key Manager을 시작할 수 없는 것은 아닙니다.

Error: Unable to find Secretkey in the config keystore with alias:MyKey.

등록 정보 파일의 symmetricKeySet 항목이 config.keystore.file에 없는 키 별명을 포함합니다.

이 문제점을 정정하려면 **KeyManagerConfig.properties**의 config.keystore.file 항목에 지정된 키 저장소 파일에 있는 별명만 포함하도록 구성 파일에서 symmetricKeySet 항목을 수정하십시오. 또는 키 저장소에 누락된 대칭 키를 추가하십시오. 자세한 정보는 부록 B, 『Encryption Key Manager 구성 등록 정보 파일』을 참조하십시오.

No symmetric keys in symmetricKeySet, LTO drives cannot be supported.

정보 메시지입니다. Encryption Key Manager 서버는 시작되지만 이 Encryption Key Manager 인스턴스에서 LTO 드라이브는 지원되지 않습니다. 이 Encryption Key Manager과 통신하도록 구성된 LTO 드라이브가 없는 경우 이는 문제점이 아닙니다.

Encryption Key Manager에서 보고하는 오류

이 섹션에서는 EKM(Encryption Key Manager)에서 보고하는 오류 메시지를 정의합니다. 또한 드라이브 감지 데이터로 리턴되는 메시지도 정의합니다. 일반적으로 결함 증상 코드 또는 FSC라고 합니다. 테이블에는 오류 번호, 실패에 대한 간략한 설명 및 정정 조치가 표시됩니다. 디버그 등록 정보를 지정하는 방법에 대한 정보는 부록 B, 기본 구성 파일을 참조하십시오.

표 6-1. Encryption Key Manager에서 보고된 오류

오류 번호	설명	조치
EE02	암호화 읽기 메시지 실패: DriverErrorNotifyParameterError: "잘못된 ASC 및 ASCQ가 수신되었습니다. ASC 및 ASCQ가 키 작성/키 변환/키 확보 조작과 일치하지 않습니다.	테이프 드라이브에 지원되지 않은 조치가 요청되었습니다. Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) 필요한 경우 드라이브 또는 프록시 서버 펌웨어 버전을 확인하고 최신 릴리스로 업데이트하십시오. Key Manager 서버에서 디버그 추적을 설정하십시오. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.

표 6-1. Encryption Key Manager에서 보고된 오류 (계속)

오류 번호	설명	조치
EE0F	암호화 논리 오류: 내부 오류: "예상하지 못한 오류. EKM의 내부 프로그래밍 오류."	Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) 필요한 경우 드라이브 또는 프록시 서버 펌웨어 버전을 확인하고 최신 릴리스로 업데이트하십시오. Key Manager 서버에서 디버그 추적을 설정하십시오. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
	오류: CSNDDSV 호출 시 하드웨어 오류 발생. returnCode 12 reasonCode 0.	하드웨어 암호를 사용하는 경우 ICSF를 시작했는지 확인하십시오.
EE23	암호화 읽기 메시지 실패: 내부 오류: "예상하지 못한 오류....."	일반적인 오류가 발생하여 드라이브 또는 프록시 서버에서 수신한 메시지의 구문을 분석할 수 없습니다. Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) Key Manager 서버에서 디버그를 설정합니다. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE25	암호화 구성 문제점: 드라이브 테이블과 관련된 오류가 발생했습니다.	config.drivetable.file.url이 제공된 경우 이 매개변수가 KeyManagerConfig.properties 파일에서 올바른지 확인하십시오. Encryption Key Manager 서버에서 `listdrives -drivename <drivename>` 명령을 실행하여 드라이브가 올바르게 구성되었는지 확인하십시오(예: 드라이브 일련 번호, 별명 및 인증서가 올바른지 확인). Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) 필요한 경우 드라이브 또는 프록시 서버 펌웨어 버전을 확인하고 최신 릴리스로 업데이트하십시오. 디버그 추적을 설정하고 조작을 다시 시도하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE29	암호화 읽기 메시지 실패: 올바르지 않은 서명	드라이브 또는 프록시 서버에서 수신한 메시지에서 해당 서명이 일치하지 않습니다. Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) Key Manager 서버에서 디버그를 설정합니다. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.

표 6-1. Encryption Key Manager에서 보고된 오류 (계속)

오류 번호	설명	조치
EE2B	암호화 읽기 메시지 실패: 내부 오류: ″DSK에 서명이 없거나 DSK의 서명을 확인할 수 없습니다.″	Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) 필요한 경우 드라이브 또는 프록시 서버 펌웨어 버전을 확인하고 최신 릴리스로 업데이트하십시오. Key Manager 서버에서 디버그 추적을 설정하십시오. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE2C	암호화 읽기 메시지 실패: QueryDSKParameterError: ″장치에서 QueryDSKMessage 구문 분석 중에 오류 발생. 예상치 못한 DSK 계수 또는 예상치 못한 과부하.″	테이프 드라이브에서 지원되지 않은 기능을 수행하도록 Encryption Key Manager에 요청했습니다. Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) 필요한 경우 드라이브 또는 프록시 서버 펌웨어 버전을 확인하고 최신 릴리스로 업데이트하십시오. Key Manager 서버에서 디버그 추적을 설정하십시오. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE2D	암호화 읽기 메시지 실패: 올바르지 않은 메시지 유형	Encryption Key Manager에서 순서가 맞지 않는 메시지를 수신하거나 처리 방법을 알지 못하는 메시지를 수신했습니다. Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) Key Manager 서버에서 디버그를 설정합니다. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE2E	암호화 읽기 메시지 실패: 내부 오류: 올바르지 않은 서명 유형	드라이브 또는 프록시 서버에서 수신한 메시지의 서명 유형이 올바르지 않습니다. Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) Key Manager 서버에서 디버그를 설정합니다. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE30	금지된 요청입니다.	지원되지 않는 조작을 테이프 드라이브에 요청했습니다. 대상 테이프 드라이브에 대해 올바른 지원되는 명령을 입력하십시오.

표 6-1. Encryption Key Manager에서 보고된 오류 (계속)

오류 번호	설명	조치
EE31	암호화 구성 문제점: 키 저장소와 관련된 오류가 발생했습니다.	기본값으로 구성되었거나 사용하려는 키 레이블을 확인하십시오. listcerts 명령을 사용하여 Encryption Key Manager에서 사용 가능한 인증서를 나열할 수 있습니다. 기본값을 사용한다는 사실을 알고 있는 경우 Encryption Key Manager 서버에서 listdrives -drivename *drivename* 명령을 실행하여 드라이브가 올바르게 구성되었는지 확인하십시오(예: 드라이브 일련 번호 및 이와 연관된 별명/키 레이블이 올바른지 확인). 문제가 되는 드라이브에 이와 연관된 별명/키 레이블이 없는 경우 default.drive.alias1 및 default.drive.alias2 값을 확인하십시오. 이 방법이 도움이 되지 않거나 별명/키 레이블이 있는 경우 디버그 로그를 수집하십시오. 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EE32	키 저장소 관련 문제점.	가장 큰 원인은 키가 다른 별개의 Encryption Key Manager를 사용하여 테이프를 암호화하였거나, 이 테이프를 암호화하는 데 사용한 키의 이름이 바뀌었거나 키 저장소에서 삭제되었을 수 있습니다. `list -keysym` 명령을 실행하고 요청 별명이 키 저장소에 있는지 확인하십시오.
EEE1	암호화 논리 오류: 내부 오류: "예상치 못한 오류: 서브 페이지에서 EK/EEDK 플래그 충돌."	Encryption Key Manager의 최신 버전을 실행하고 있는지 확인하십시오. (최신 버전을 판별하려면 3-1 페이지의 『최신 버전 Key Manager ISO 이미지 다운로드』를 참조하십시오.) 필요한 경우 드라이브 또는 프록시 서버 펌웨어 버전을 확인하고 최신 릴리스로 업데이트하십시오. Key Manager 서버에서 디버그를 설정합니다. 문제점을 재현하고 디버그 로그를 수집하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.
EF01	암호화 구성 문제점: "드라이브가 구성되지 않았습니다."	Encryption Key Manager과 통신하려는 드라이브가 드라이브 테이블에 없습니다. config.drivetable.file.url 이 제공된 경우 이 매개변수가 KeyManagerConfig.properties 파일에서 올바른지 확인하십시오. `listdrives` 명령을 사용하여 드라이브가 목록에 있는지 확인하십시오. 그렇지 않으면 올바른 드라이브 정보를 사용하여 `adddrive` 명령을 통해 드라이브를 수동으로 구성하거나 `modconfig` 명령을 통해 "drive.acceptUnknownDrives" 등록 정보를 true로 설정합니다. 디버그 추적을 설정하고 조작을 다시 시도하십시오. 문제점이 지속되면 이 책 서문에 나오는 『먼저 읽어야 할 사항』 섹션의 『Dell 연락처』를 참조하여 기술 지원을 문의하십시오.

메시지

다음 메시지는 Encryption Key Manager에서 생성하며 관리 콘솔에 표시할 수 있습니다.

구성 파일이 지정되지 않음

텍스트

```
Configuration file not specified:  KeyManager Configuration file not
specified when starting EKM.
```

설명

KMSAdmin 명령을 사용하려면 구성 파일을 명령행 매개변수로 전달해야 합니다.

시스템 응답

프로그램이 중지됩니다.

운영자 응답

구성 파일을 제공하고 명령을 다시 시도하십시오.

드라이브 추가 실패

텍스트

```
Failed to add drive. Drive already exists.
```

설명

이미 드라이브가 Encryption Key Manager에 구성되어 드라이브 테이블에 있으므로 **adddrive** 명령에 실패했습니다.

운영자 응답

listdrives 명령을 실행하여 드라이브가 Encryption Key Manager에 이미 구성되어 있는지 확인하십시오. 이미 드라이브가 있는 경우 **moddrive** 명령을 사용하여 드라이브 구성을 변경할 수 있습니다. 자세한 정보를 보려면 **help**를 실행하십시오.

로그 파일에 아카이브 실패

텍스트

```
Failed to archive the log file.
```

설명

로그 파일 이름을 바꿀 수 없습니다.

운영자 응답

해당 드라이브의 파일 권한 및 공간을 확인하십시오.

구성 삭제 실패

텍스트

```
"modconfig" command failed.
```

설명

modconfig 명령을 사용하여 Encryption Key Manager 구성을 삭제할 수 없습니다.

운영자 응답

help를 사용하여 올바른 매개변수를 제공했는지 명령 구문을 확인하십시오. 자세한 정보는 감사 로그를 확인하십시오.

드라이브 항목 삭제 실패

텍스트

```
"deldrive" command failed.
```

설명

deldrive 명령이 드라이브 테이블에서 드라이브 항목을 삭제하는 데 실패했습니다.

운영자 응답

help를 사용하여 명령 구문을 확인하고 올바른 매개변수를 제공했는지 확인하십시오. **listdrives** 명령을 사용하여 Encryption Key Manager에 드라이브가 구성되어 있는지 확인하십시오. 자세한 정보는 감사 로그를 확인하십시오.

가져오기 실패

텍스트

```
"import" command failed.
```

설명

드라이브 테이블 또는 구성 파일을 가져올 수 없습니다.

시스템 응답

Encryption Key Manager 서버가 시작되지 않습니다.

운영자 응답

지정된 URL이 있고 이에 대한 읽기 권한이 있는지 확인하십시오. **help**를 사용하여 명령 구문을 확인하십시오. 매개변수가 올바른지 확인하고 다시 시도하십시오.

구성 수정 실패

텍스트

```
"modconfig" command failed.
```

설명

modconfig 명령을 사용하여 Encryption Key Manager 구성을 수정할 수 없습니다.

운영자 응답

help를 사용하여 올바른 매개변수를 제공했는지 명령 구문을 확인하십시오. 자세한 정보는 감사 로그를 확인하십시오.

파일 이름은 널(null)일 수 없음

텍스트

```
File name was not supplied for audit log file.
```

설명

Encryption Key Manager 구성 등록 정보에서 감사 파일 이름을 제공하지 않았습니다. 이 매개변수는 필수 구성 매개변수입니다.

시스템 응답

프로그램이 중지됩니다.

운영자 응답

Encryption Key Manager에 제공되는 구성 등록 정보 파일에 `Audit.handler.file.name`이 정의되었는지 확인하고 EKM을 다시 시작하십시오.

파일 크기 한계는 음수일 수 없음

텍스트

```
Maximum file size for audit log can not be a negative number.
```

설명

Encryption Key Manager 구성 파일의 Audit.handler.file.size 등록 정보 값이 양수여야 합니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

Audit.handler.file.size에 올바른 숫자를 지정하고 Encryption Key Manager를 다시 시작하십시오.

동기화할 데이터 없음

텍스트

```
No data can be found to be synchronized with "sync".
```

설명

sync 명령에서 동기화할 데이터를 식별할 수 없습니다.

운영자 응답

제공된 구성 파일이 있는지와 *config.drivetable.file.url*을 사용하여 구성 파일에 드라이브 테이블이 제대로 구성되었는지 확인하십시오. **help**를 사용하여 구문을 확인하고 **sync** 명령을 다시 시도하십시오.

올바르지 않은 입력

텍스트

```
Invalid input parameters for the CLI.
```

설명

특정 명령 구문이 올바르지 않습니다.

운영자 응답

입력한 명령이 올바른지 확인하고 다시 시도하십시오. **help**를 사용하여 명령 구문을 확인하십시오. 매개변수가 올바른지 확인하고 다시 시도하십시오.

구성 파일의 SSL 포트 번호가 올바르지 않음

텍스트

```
Invalid SSL port number specified in the EKM configuration file.
```

설명

구성 파일에 제공된 SSL 포트 번호가 올바른 번호가 아닙니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

Encryption Key Manager를 시작할 때 구성 파일의 `TransportListener.ssl.port` 등록 정보에 올바른 포트 번호를 지정하고 다시 시작하십시오.

구성 파일의 TCP 포트 번호가 올바르지 않음

텍스트

```
Invalid TCP port number specified in the EKM configuration file.
```

설명

구성 파일에 제공된 TCP 포트 번호가 올바른 번호가 아닙니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

Encryption Key Manager를 시작할 때 구성 파일의 `TransportListener.tcp.port` 등록 정보에 올바른 포트 번호를 지정하고 다시 시작하십시오. 기본 TCP 포트 번호는 3801입니다.

구성 파일에서 SSL 포트 번호를 지정해야 함

텍스트

```
SSL port number is not configured in the properties file.
```

설명

SSL 포트 번호는 구성 등록 정보 파일에서 구성할 필수 등록 정보입니다. 이 항목은 여러 대의 서버 환경에서 Encryption Key Manager 서버 간 통신에 사용됩니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

TransportListener.ssl.port 등록 정보에 올바른 포트 번호를 지정하고 Encryption Key Manager를 다시 시작하십시오.

구성 파일에서 TCP 포트 번호를 지정해야 함

텍스트

```
TCP port number is not configured in the properties file.
```

설명

TCP 포트 번호는 구성 등록 정보 파일에서 구성할 필수 등록 정보입니다. 이 항목은 드라이브와 Encryption Key Manager 간 통신에 사용됩니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

TransportListener.tcp.port 등록 정보에 올바른 포트 번호를 지정하고 Encryption Key Manager를 다시 시작하십시오. 기본 TCP 포트 번호는 3801입니다.

서버 시작 실패

텍스트

```
EKM server failed to start.
```

설명

구성에 문제가 있어서 Encryption Key Manager 서버를 시작할 수 없습니다.

운영자 응답

구성 파일에서 제공된 매개변수를 확인하십시오. 자세한 정보는 로그를 확인하십시오.

동기화 실패

텍스트

```
"sync" command failed.
```

설명

두 Encryption Key Manager 서버 사이에서 데이터를 동기화하는 sync 조작에 실패했습니다.

운영자 응답

원격 Encryption Key Manager 서버에 지정된 IP 주소가 올바르며 컴퓨터에 액세스할 수 있는지 확인하십시오. 구성 파일이 있고 올바른 드라이브 테이블 정보를 포함하는지 확인하십시오. **help**를 사용하여 **sync** 명령 구문을 확인하십시오. 자세한 정보는 로그를 확인하십시오.

지정된 감사 로그 파일이 읽기 전용임

텍스트

```
The audit log file can not be opened for writing.
```

설명

Audit.handler.file.name 등록 정보에 지정된 Encryption Key Manager 구성의 감사 로그 파일을 쓰기 위해 열 수 없습니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

지정된 감사 파일 및 디렉토리에 대한 권한을 확인하고 Encryption Key Manager를 다시 시작하십시오.

관리 키 저장소를 로드할 수 없음

텍스트

```
Keystore for Admin cannot be loaded.
```

설명

Encryption Key Manager에 제공된 관리 키 저장소를 로드할 수 없습니다. 여러 대의 서버 환경에서 서버측과 통신하기 위해 Encryption Key Manager 서버 사이에서 관리 키 저장소가 사용됩니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

구성 파일 설정을 확인하십시오. Encryption Key Manager 구성 파일에 admin.keystore.file, admin.keystore.provider 및 admin.keystore.type 등록 정보가 올바른지(부록 B 참조)와 키 저장소 파일이 있고 이에 대한 읽기 권한이 있는지 확인하십시오. admin.keystore.password 등록 정보를 통해 관리 키 저장소에 제공되었거나 명령행에 입력한 암호가 올바른지 확인하십시오. Encryption Key Manager를 다시 시작하십시오.

키 저장소를 로드할 수 없음

텍스트

```
Keystore for EKM can not be loaded.
```

설명

Encryption Key Manager에 제공된 키 저장소를 로드할 수 없습니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

구성 파일 설정을 확인하십시오. Encryption Key Manager 구성 파일에 config.keystore.file, config.keystore.provider 및 config.keystore.type 등록 정보가 올바른지와 키 저장소 파일이 있고 이에 대한 읽기 권한이 있는지 확인하십시오. config.keystore.password 등록 정보를 통해 Encryption Key Manager 키 저장소에 제공되었거나 명령행에 입력한 암호가 올바른지 확인하십시오. Encryption Key Manager를 다시 시작하십시오.

전송 키 저장소를 로드할 수 없음

텍스트

```
Transport keystore cannot be loaded.
```

설명

Encryption Key Manager에 제공된 전송 키 저장소를 로드할 수 없습니다. 여러 대의 서버 환경에서 클라이언트측과 통신하기 위해 Encryption Key Manager 서버 사이에서 전송 키 저장소가 사용됩니다.

시스템 응답

Encryption Key Manager가 시작되지 않습니다.

운영자 응답

구성 파일 설정을 확인하십시오. Encryption Key Manager 구성 파일에 `transport.keystore.file`, `transport.keystore.provider` 및 `transport.keystore.type` 등록 정보가 올바른지와 키 저장소 파일이 있고 이에 대한 읽기 권한이 있는지 확인하십시오. `transport.keystore.password` 등록 정보를 통해 관리 키 저장소에 제공되었거나 명령행에 입력한 암호가 올바른지 확인하십시오. Encryption Key Manager를 다시 시작하십시오.

지원되지 않는 조치

텍스트

```
User entered action for the CLI which is not supported for EKM.
```

설명

sync 명령에 제공된 조치가 Encryption Key Manager에서 지원되지 않거나 인식되지 않습니다. 올바른 조치를 병합하거나 다시 작성합니다.

운영자 응답

help를 사용하여 명령 구문을 확인하고 다시 시도하십시오.

제 7 장 감사 레코드

주: 이 장에서 설명하는 감사 레코드 형식은 프로그래밍 인터페이스로 간주되지 않습니다. 이 레코드 형식은 릴리스마다 변경될 수 있습니다. 이 장에서는 일부 감사 레코드를 구문 분석하려는 경우에 대비하여 해당 형식을 설명합니다.

감사 개요

감사 서브 시스템은 Encryption Key Manager에서 요청을 처리하는 중 감사 가능한 다양한 이벤트가 발생하면 일련의 순차 파일에 텍스트 감사 레코드를 씁니다. 감사 서브 시스템은 파일에 항목을 씁니다. 이때 디렉토리 및 파일 이름을 사용자가 구성할 수 있습니다. 이 파일의 *크기*도 구성 가능합니다. 파일에 레코드를 쓰다가 파일 *크기*가 구성 가능한 *크기*에 도달하면 파일을 닫고 현재 시간소인에 따라 파일 이름을 바꾼 후 다른 파일을 열어 새로 작성한 이 파일에 레코드를 씁니다. 따라서 전체 감사 레코드 로그는 구성 가능한 *크기*로 설정된 파일로 구분되고 파일 *크기*가 구성 가능한 *크기*를 초과한 시점의 시간소인에 따라 파일 이름을 순차적으로 지정합니다.

작성된 모든 순차 파일을 포괄하는 전체 감사 로그의 정보 *크기*가 너무 커지지 않고 파일 시스템에서 사용 가능한 공간을 초과하지 않도록 하기 위해 구성된 감사 디렉토리/폴더/컨테이너에서 파일 세트를 모니터하는 스크립트 또는 프로그램 작성을 고려할 수도 있습니다. 파일을 닫고 시간소인에 따라 이름을 지정하면 파일 내용을 복사하여 원하는 장기간 연속되는 로그 위치에 추가한 후 지워야 합니다. 실행 중 Encryption Key Manager에서 레코드를 쓰고 있는 파일을 제거하거나 바꾸지 않도록 주의하십시오. 이 파일은 파일 이름에 시간소인을 포함할 수 없습니다.

감사 구성 매개변수

다음 매개변수는 Encryption Key Manager 구성 파일에서 감사 로그에 기록되는 이벤트, 감사 로그 파일을 쓰는 위치 및 감사 로그 파일의 최대 *크기*를 제어하는 데 사용됩니다.

Audit.event.types

구문

Audit.event.types={*type*[;*type*]}

사용법

감사 로그에 보내야 하는 감사 유형을 지정하는 데 사용됩니다. 구성 매개변수로 가능한 값은 다음과 같습니다.

all	모든 이벤트 유형
authentication	인증 이벤트
data_synchronization	Encryption Key Manager 서버 사이에서 정보를 동기화하는 중 발생하는 이벤트
runtime	Encryption Key Manager에 보내는 처리 작업 및 요청 중 발생하는 이벤트
configuration_management	구성을 변경하는 경우 발생하는 이벤트
resource_management	Encryption Key Manager에서 자원(테이프 드라이브) 설정을 변경하는 경우 발생하는 이벤트

예제

이 구성 값에 대한 예제 스펙은 다음과 같습니다.

```
Audit.event.types=all
```

다음은 또 다른 예제입니다.

```
Audit.event.types=authentication;runtime;resource_management
```

Audit.event.outcome

구문

Audit.event.outcome={*outcome*[;*outcome*]}

사용법

성공한 조작, 실패한 조작 또는 두 경우의 결과를 감사해야 할 때 이벤트 발생 여부를 표시하는 데 사용됩니다. 조작에 성공하면 발생하는 이벤트를 기록하려면 **success**로 지정하십시오. 조작에 실패하면 발생하는 이벤트를 기록하려면 **failure**로 지정하십시오.

예제

이 구성 값에 대한 예제 스펙은 다음과 같습니다.

```
Audit.event.outcome=failure
```

성공하거나 실패한 경우 모두에 대해 설정하려면 다음과 같이 설정합니다.

```
Audit.event.outcome=success;failure
```

Audit.eventQueue.max

구문

Audit.eventQueue.max=*number_events*

사용법

메모리 큐에서 보유할 최대 이벤트 오브젝트 수를 설정하는 데 사용됩니다. 이 매개변수는 선택적이지만 사용이 권장됩니다. 기본값은 0입니다.

예제

```
Audit.eventQueue.max=8
```

Audit.handler.file.directory

구문

Audit.handler.file.directory=*directoryName*

사용법

이 매개변수는 감사 레코드 파일을 써야 하는 디렉토리를 표시하는 데 사용됩니다. 이 디렉토리가 없으면 Encryption Key Manager에서 디렉토리 작성을 시도합니다. 그러나 성공하지 못하면 Encryption Key Manager가 시작되지 않습니다. 따라서 Encryption Key Manager를 실행하기 전에 먼저 디렉토리가 있는 것이 좋습니다. 또한 Encryption Key Manager를 실행하는 사용자 ID에 지정된 디렉토리에 대한 쓰기 액세스 권한이 있어야 합니다.

예제

디렉토리를 **/var/ekm/ekm1/audit**로 지정하려면 다음과 같이 설정합니다.

```
Audit.handler.file.directory=/var/ekm/ekm1/audit
```

Audit.handler.file.size

구문

Audit.handler.file.size=*sizeInKiloBytes*

사용법

이 매개변수는 감사 파일을 닫고 새 감사 파일을 쓰는 크기 제한을 표시하는 데 사용됩니다. 결과로 생성되는 감사 파일의 실제 크기는 크기 제한을 초과한 후 파일을 닫을 때 이 값보다 몇 바이트 더 클 수도 있습니다.

예제

최대 파일 크기를 약 2MB로 설정하려면 다음을 입력하십시오.

```
Audit.handler.file.size=2000
```

Audit.handler.file.name

구문

Audit.handler.file.name=*fileName*

사용법

이 매개변수를 사용하여 지정된 감사 디렉토리 내에서 감사 로그 파일을 작성할 때 기본 이름으로 사용할 기본 파일 이름을 지정합니다. 이 매개변수에는 기본 파일 이름만 포함하고 완전한 경로 이름은 포함하면 안됩니다. 감사 로그 파일의 전체 이름에는 파일 작성 시간에 해당하는 값이 이 이름에 추가됩니다.

이를 표시하기 위해 Audit.handler.file.name 값을 **ekm.log**로 설정하는 예제를 고려합니다. 이 경우 파일의 전체 이름은 ekm.log.2315003554와 유사한 형식을 사용합니다. 추가된 문자열은 감사 로그 파일을 작성한 순서를 판별하는 데 도움이 되기도 합니다. 숫자가 높을수록 최신 감사 로그 파일을 나타냅니다.

예제

기본 이름을 **ekm.log**로 설정하는 예제는 다음과 같습니다.

```
Audit.handler.file.name=ekm.log
```

Audit.handler.file.multithreads

구문

Audit.handler.file.multithreads={yes|true|no|false}

사용법

true로 지정하면 별도의 스레드를 사용하여 감사 로그에 이벤트 데이터를 씁니다. 따라서 감사 로그에 쓰는 작업이 완료되길 기다리지 않고도 현재 스레드 실행(조작)을 계속할 수 있습니다. 여러 개의 스레드 사용이 기본 동작입니다.

예제

기본 이름을 **true**로 설정하는 예제는 다음과 같습니다.

```
Audit.handler.file.multithreads=true
```

Audit.handler.file.threadlifespan

구문

Audit.handler.file.threadlifespan=*timeInSeconds*

사용법

이 매개변수는 스레드가 감사 로그 항목을 쓰는 데 필요한 최대 예상 시간을 지정하는 데 사용됩니다. 이 값은 스레드가 인터럽트되기 전에 작업을 완료할 수 있도록 정리 처리를 수행하는 중에 사용됩니다. 백그라운드 스레드가 threadlifespan 매개변수에 할당된 시간 내에 작업을 완료하지 못하면 정리 처리가 시작되고 스레드가 인터럽트됩니다.

예제

스레드가 감사 로그에 쓰는 데 걸리는 예상 시간을 10초로 설정하려면 다음을 지정합니다.

```
Audit.handler.file.threadlifespan=10
```

감사 레코드 형식

모든 감사 레코드는 여기서 설명하는 것과 유사한 출력 형식을 사용합니다. 모든 감사 레코드는 발생한 감사 이벤트에 특정한 정보와 함께 시간소인 및 레코드 유형을 비롯한 몇 가지 공통 정보를 포함합니다. 감사 레코드의 일반 형식은 다음과 같습니다.

```
AuditRecordType:[
  timestamp=timestamp
  Attribute Name=Attribute Value
  ...
  ]
```

각 레코드는 파일에서 여러 행에 걸쳐 입력됩니다. 레코드에서 첫 번째 행의 첫 번째 문자는 감사 레코드 유형으로 시작됩니다. 그 다음에 콜론(:)과 여는 왼쪽 대괄호([)가 나옵니다. 같은 감사 레코드와 연관된 후속 행은 로그 레코드를 쉽게 판독할 수 있도록 두 칸 들여씁니다. 단일 감사 레코드의 마지막 행은 두 칸 들여쓰고 닫는 오른쪽 대괄호(])를 포함합니다. 각 감사 레코드의 행 수는 감사 레코드 유형 및 감사 레코드에서 제공하는 추가 속성 정보에 따라 다릅니다.

감사 레코드의 시간소인은 Encryption Key Manager를 실행하는 시스템 시계에 기반합니다. 이 레코드가 다른 시스템에서 이벤트가 발생하는 시간 소인에 따라 해당 이벤트와 연관된 경우 일부 유형의 시간 동기화를 사용하여 환경의 다양한 시스템 시계를 허용되는 정확성 레벨에 맞춰 동기화해야 합니다.

Encryption Key Manager에서의 감사 지점

Encryption Key Manager는 구성에 기반하여 요청을 처리하는 중 발생하는 많은 이벤트에 대한 감사 레코드를 쓸 수 있습니다. 이 섹션에서는 감사 가능한 일련의 이벤트를 감사 레코드 구성 범주와 함께 설명합니다. 이러한 감사 레코드를 감사 파일에 쓰려면 이 범주를 설정해야 합니다(7-6 페이지의 표 7-1 참조).

표 7-1. Encryption Key Manager가 감사 파일에 쓰는 감사 레코드 유형

감사 레코드 유형	감사 유형	설명
인증	authentication	인증 이벤트를 기록하는 데 사용됩니다.
데이터 동기화	data_synchronization	데이터 동기화 처리를 기록하는 데 사용됩니다.
런타임	runtime	요청을 처리하는 중 Encryption Key Manager 서버에서 발생하는 여러 중요 처리 이벤트를 기록하는 데 사용됩니다.
자원 관리	resource_management	Encryption Key Manager에서 자원을 구성하는 방법에 대한 변경 사항을 기록하는 데 사용됩니다.
구성 관리	configuration_management	Encryption Key Manager 서버의 구성에 대한 변경 사항을 기록하는 데 사용됩니다.

감사 레코드 속성

다음 목록에서는 각 감사 레코드 유형에서 사용 가능한 속성을 보여줍니다.

인증 이벤트

이 레코드의 형식은 다음과 같습니다.

```
Authentication event:[
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_AUTHN
  message=message
  authentication type=type
  users=users
  ]
```

message 값은 이에 대한 정보가 사용 가능한 경우에만 나타납니다.

데이터 동기화 이벤트

이 레코드의 형식은 다음과 같습니다.

```
Data synchronization event:
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_DATA_SYNC
  message=message
  action=action
  resource=resource
  user=user
  ]
```

message 및 user 값은 이에 대한 정보가 사용 가능한 경우에만 나타납니다.

런타임 이벤트

이 레코드의 형식은 다음과 같습니다.

```
Runtime event:
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_RUNTIME
  message=message
  resource=resource
  action=action
  user=user
  ]
```

message 및 user 값은 이에 대한 정보가 사용 가능한 경우에만 나타납니다.

자원 관리 이벤트

이 레코드의 형식은 다음과 같습니다.

```
Resource management event:
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_MGMT_RESOURCE
  message=message
  action=action
  user=user
  resource=resource
  ]
```

message 값은 이에 대한 정보가 사용 가능한 경우에만 나타납니다.

구성 관리 이벤트

이 레코드의 형식은 다음과 같습니다.

```
Configuration management event:
  timestamp=timestamp
  event source=source
  outcome=outcome
  event type=SECURITY_MGMT_CONFIG
  message=message
  action=action
  command type=type
  user=user
  ]
```

message 값은 이에 대한 정보가 사용 가능한 경우에만 나타납니다.

감사된 이벤트

표 7-2에서는 감사 레코드를 작성하는 이벤트를 설명합니다. 테이블에는 이 이벤트가 발생할 때 기록되는 감사 레코드 유형이 나열되어 있습니다.

표 7-2. 감사된 이벤트별 감사 레코드 유형

감사된 이벤트	감사 레코드 유형
사용자 인증 성공	authentication
사용자 인증 실패	authentication
다른 EKM에 데이터를 전송하는 데 성공	data_synchronization
다른 EKM에 데이터를 전송하는 중에 오류 발생	data_synchronization
sync 명령 처리 성공	data_synchronization
sync 명령을 처리하는 중에 오류 발생	data_synchronization
명령행 처리 시작	runtime
exit 명령 수신	runtime
알 수 없는 명령 입력	runtime
드라이브에서 메시지 수신	runtime
드라이브에서 메시지를 처리하는 중에 오류 발생	runtime
드라이브에서 수신한 메시지에서 오류 발생	runtime
드라이브에서 수신한 정보로 드라이브 테이블을 업데이트하는 중에 오류 발생	runtime
드라이브 테이블에서 정보를 검색하는 중에 오류 발생	runtime
키 저장소에서 정보를 검색하는 중에 오류 발생	runtime
드라이브에서 인증서를 처리하는 중에 오류 발생	runtime
키 저장소에서 개인용 키를 찾는 중에 오류 발생	runtime
암호 값을 계산하는 중에 오류 발생	runtime
메시지 교환 처리 성공	runtime
메시지 처리 시작	runtime
명령행 처리 시작	runtime
암호 서비스를 사용하여 문제점 발견	runtime
새 드라이브 발견	runtime
드라이브 테이블로 드라이브를 구성하는 중에 오류 발생	runtime
드라이브에서 메시지 처리 시작	runtime
stopekm 명령 수신 및 처리	runtime
드라이브 테이블에서 드라이브 제거	resource_management
드라이브 테이블에서 드라이브를 제거하는 중에 오류 발생	resource_management
드라이브 테이블을 가져오는 데 성공	resource_management
드라이브 테이블을 가져오는 중에 오류 발생	resource_management
드라이브 테이블을 내보내는 데 성공	resource_management
드라이브 테이블을 내보내는 중에 오류 발생	resource_management

표 7-2. 감사된 이벤트별 감사 레코드 유형 (계속)

감사된 이벤트	감사 레코드 유형
listcerts 명령 성공	resource_management
드라이브를 드라이브 테이블에 추가하는 데 성공	resource_management
드라이브를 드라이브 테이블에 추가하는 중에 오류 발생	resource_management
listdrives 명령 성공	resource_management
listdrives 명령을 처리하는 중에 오류 발생	resource_management
드라이브 테이블 수정 성공	resource_management
드라이브 테이블을 수정하는 중에 오류 발생	resource_management
키 저장소를 여는 데 성공	resource_management
키 저장소를 여는 중에 오류 발생	resource_management
구성 등록 정보 변경	configuration_management
구성 등록 정보를 변경하는 중에 오류 발생	configuration_management
구성 등록 정보 삭제	configuration_management
구성 등록 정보를 삭제하는 중에 오류 발생	configuration_management
구성을 가져오는 데 성공	configuration_management
구성을 가져오는 중에 오류 발생	configuration_management
구성을 내보내는 데 성공	configuration_management
구성을 내보내는 중에 오류 발생	configuration_management
listconfig 명령 성공	configuration_management

제 8 장 메타데이터 사용

데이터를 암호화하고 테이프에 쓰는 경우 중요한 데이터를 캡처하는 XML 파일을 작성하도록 Encryption Key Manager를 구성해야 합니다. 이 파일은 볼륨 일련 번호로 조회하여 볼륨에서 사용되는 별명 또는 키 레이블을 표시할 수 있습니다. 반대로 별명으로 조회하여 해당 키 레이블/별명과 연관된 모든 볼륨을 표시할 수도 있습니다.

주: 메타데이터 파일을 구성하지 않으면 Encryption Key Manager가 시작되지 않습니다.

암호화 처리를 수행하면 Encryption Key Manager에서 다음 데이터를 수집합니다.

- 드라이브 일련 번호
- 드라이브의 WWN(WorldWideName)
- 제작 날짜
- 키 별명 1
- 키 별명 2
- DKi
- VolSer

수집한 데이터가 일정 한계에 도달하면 XML 파일에 기록됩니다. Encryption Key Manager 등록 정보 파일(KeyManagerConfig.properties)에서 설정 가능한 기본 한계는 레코드 100개입니다. 일단 파일이 작성되면 Encryption Key Manager를 실행하는 동안 계속 조회할 수 있습니다. 파일 *크기*가 너무 커지지 않도록 최대 파일 *크기*에 도달하면 새 파일에 자동으로 롤오버됩니다. 롤오버되는 기본 최대 파일 *크기*는 1MB로 이 값 역시 Encryption Key Manager 등록 정보 파일에서 설정할 수 있습니다. 현재 및 이전 파일 버전만 저장됩니다. Encryption Key Manager 구성 등록 정보 파일에 설정할 값은 다음과 같습니다.

Audit.metadata.file.name

메타데이터가 저장된 XML 파일 이름입니다. 필수입니다.

Audit.metadata.file.size

현재 버전에서 이전 버전으로 파일을 롤오버하기 전에 허용되는 최대 파일 *크기*(KB)입니다. 선택적입니다. 기본값은 1024(1MB)입니다.

Audit.metadata.file.cachecount

메타데이터 파일을 쓰기 전까지 캐시할 레코드 수입니다. 선택적입니다. 기본은 100입니다.

XML 파일 형식

파일에는 다음과 같은 형식의 레코드가 있습니다.

```
<KeyUsageEvent>
 <DriveSSN>FVTDRIVE0000</driveSSN>          -Drive Serial Number
 <VolSer>TESTER</volSer>                    -Volume Serial
 <DriveWWN>57574E414D453030</driveWWN>      -drive WWN
 <keyAlias2>cert2</keyAlias2>               -Key Alias1
 <keyAlias1>cert1</keyAlias1>               - keyAlias2
 <dateTime>Tue Feb 20 09:18:07 CST 2007</dateTime>    - creation date
</KeyUsageEvent>
```

참고: LTO 4 및 LTO 5 드라이브의 경우 <keyAlias1></keyAlias1> 레코드만 있으며 DKi가 기록됩니다.

메타데이터 XML 파일 조회

EKMDataParser 도구를 사용하여 메타데이터 파일을 조회합니다. 이 도구는 DOM(Document Object Model) 기술을 사용하여 XML 파일 구문을 분석하며 Encryption Key Manager 명령행 인터페이스에서는 실행할 수 없습니다. 다음과 같이 호출됩니다.

java com.ibm.keymanager.tools.EKMDataParser -filename
full_path_to_metadata_file {**-volser** *volser* | **-keyalias** *alias*}

metadata_path

KeyManagerConfig.properties 파일의 Audit.metadata.file.name에 메타데이터 파일의 디렉토리 경로로 지정한 값과 같습니다.

-filename

*filename*은 필수이며 XML 메타데이터 파일 이름이어야 합니다. 보통 **KeyManagerConfig.properties** 파일의 Audit.metadata.file.name 등록 정보에 지정된 이름과 일치합니다.

-volser

XML 파일에서 검색하는 테이프 카트리지의 볼륨 일련 번호입니다. **-volser** 또는 **-keyalias**를 지정해야 합니다.

-keyalias

XML 파일에서 검색하는 키 레이블 또는 별명입니다. **-volser** 또는 **-keyalias**를 지정해야 합니다.

예제

KeyManagerConfig.properties의 메타데이터 파일 이름 등록 정보 (Audit.metadata.file.name)가 `metadata` 값으로 설정되고 파일이 Encryption Key Manager를 실행하는 로컬 디렉토리에 있다고 가정했을 때 다음 명령은 volser 72448과 관련된 XML 레코드만 필터링하여 표시합니다.

```
<jvm_path>/bin/java com.ibm.keymanager.tools.EKMDataParser -filename metadata -volser 72448
```

출력은 다음과 같이 형식화됩니다.

표 8-1. 메타데이터 조회 출력 형식

keyalias1	keyalias2	volSer	dateTime	driveSSN	dki
cert1	cert2	72448	Wed Mar 14 10:31:32 CDT 2007	FVTDRIVE0004	

손상된 메타데이터 파일 복구

Encryption Key Manager 시스템을 올바르지 않게 시스템 종료하거나 Encryption Key Manager를 실행하는 시스템이 손상되는 경우 Encryption Key Manager 메타데이터 파일이 손상될 수 있습니다. 메타데이터 파일을 올바르지 않게 편집 또는 수정하는 경우에도 파일이 손상될 수 있습니다. EKMDataParser가 메타데이터 파일을 구문 분석할 때까지는 손상이 통지되지 않습니다. 다음과 유사한 오류로 EKMDataParser가 실패할 수 있습니다.

```
[Fatal Error] EKMData.xml:290:16: The end-tag for element type "KeyUsageEvent" must
 end with a '>' delimiter.
org.xml.sax.SAXParseException: The end-tag for element type "KeyUsageEvent" must
 end with a '>' delimiter.
at org.apache.xerces.parsers.DOMParser.parse(Unknown Source)
at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown Source)
at javax.xml.parsers.DocumentBuilder.parse(Unknown Source)
at com.ibm.keymanager.tools.EKMDataParser.a(EKMDataParser.java:136)
at com.ibm.keymanager.tools.EKMDataParser.a(EKMDataParser.java:26)
at com.ibm.keymanager.tools.EKMDataParser.main(EKMDataParser.java:93)
```

요소의 XML 끝 태그가 누락되었기 때문에 이 오류가 발생합니다. EKMDataParser에서 파일을 다시 구문 분석하도록 Encryption Key Manager 메타데이터 파일을 복구할 수 있습니다.

1. Encryption Key Manager 메타데이터 파일의 백업 사본을 작성하십시오.
2. Encryption Key Manager 메타데이터 파일을 편집하십시오.
3. XML에서는 각 데이터나 이벤트에 대해 앞 태그와 이에 상응하는 끝 태그가 있어야 합니다.
   - 앞 태그의 예:
     - <KeyUsageEvent>
     - <driveSSN>
     - <keyAlias1>
   - 끝 태그의 예:
     - </KeyUsageEvent>

- </driveSSN>
- </keyAlias1>

4. 파일을 스캔하여 일치하지 않는 태그가 있는지 확인하십시오. EKMDataParser의 오류 메시지에 끝 태그가 누락된 태그가 나열되므로 쉽게 검색할 수 있습니다.
5. 일치하지 않는 태그가 발견되면 임시로 이벤트를 삭제하거나 필요한 태그를 추가하여 이벤트를 완성하십시오.

   • Encryption Key Manager 메타데이터 파일의 다음 예를 보면 첫 번째 KeyUsageEvent에 끝 태그가 없습니다.

   ```
   <KeyUsageEvent>
   <driveSSN>001310000109</driveSSN>
   <volSer>      </volSer>
   <driveWWN>5005076312418B07</driveWWN>
   <keyAlias1>key0000000000000000F</keyAlias1>
   <dki>6B65790000000000000000F</dki>
   <dateTime>Thu Aug 30 09:50:53 MDT 2007</dateTime>
   <KeyUsageEvent>
   <driveSSN>001310000100</driveSSN>
   <volSer>      </volSer>
   <driveWWN>5005076312418ABB</driveWWN>
   <keyAlias1>key00000000000000000</keyAlias1>
   <dki>6B6579000000000000000000</dki>
   <dateTime>Thu Sep 06 16:49:39 MDT 2007</dateTime>
   </KeyUsageEvent>
   ```

   <dateTime>Thu Aug 30 09:50:53 MDT 2007</dateTime> 행과 <KeyUsageEvent> 행 사이에 </KeyUsageEvent>를 추가하면 첫 번째 <KeyUsageEvent>가 완성됩니다.

파일 손상이 복구되면 EKMDataParser가 데이터를 올바로 구문 분석할 수 있습니다.

부록 A. 샘플 파일

샘플 시작 디먼 스크립트

경고: 키 저장소 데이터를 보존하는 작업은 매우 중요합니다. 키 저장소에 액세스하지 않으면 암호화된 테이프의 암호를 해독할 수 없습니다. 따라서 키 저장소 및 암호 정보를 저장해 두십시오.

Linux 플랫폼

다음은 입증된 방식으로 백그라운드에서 EKM을 시작할 수 있는 샘플 스크립트입니다. 이 스크립트는 EKM을 시작하고 키 저장소 암호, *keystore_password*를 스크립트를 통해 전달합니다. 이 경우 키 저장소 암호는 EKM 구성 파일에 포함될 수 없습니다. 아래 참고를 참조하십시오. 스크립트 파일에는 다음이 포함되어야 합니다.

```
java com.ibm.keymanager.KMSAdminCmd KeyManagerConfig.properties <<EOF
startekm
keystore_password
status
EOF
```

주: 스크립트를 통해 키 저장소 암호를 EKM에 입력하는 경우(즉, EKM 구성 파일에 키 저장소 암호가 포함되지 않음) EKM을 백업하면 파일(구성 파일, 드라이브 테이블 및 키 저장소 백업 파일)을 비밀리에 처리할 필요가 없습니다. 하지만 키 저장소 암호를 포함하는 스크립트는 **반드시** 안전하게 저장되어야 합니다(예: 여러 위치에 여러 사본 존재). 키 저장소 암호는 기밀 정보이므로 기밀 정보로 처리해야 합니다. 스크립트 파일을 안전하게 백업하는 경우에도 키 저장소 암호를 포함하는 구성 파일을 백업할 때와 같은 옵션이 있습니다. 하지만 EKM 백업 파일에서 스크립트를 비밀리에 별도로 백업하고 저장/전송할 수 있습니다. 그러면 간접적인 레벨의 보안을 추가할 수 있습니다. 마지막으로 키 저장소 암호를 스크립트 또는 EKM 구성 파일에 저장하지만 키 저장소 암호를 항상 복구할 수 있도록 반드시 안전하게 저장해야 합니다. **키 저장소 암호의 모든 사본을 유실하면 키 저장소의 모든 키를 유실하는 것이며 이 경우 복구 방법은 없습니다.**

샘플 구성 파일

다음은 모든 키 저장소 항목이 같은 소프트웨어 키 저장소를 가리키는 샘플 EKM 등록 정보 파일입니다.

```
Admin.ssl.keystore.name = /keymanager/testkeys
Admin.ssl.keystore.type = jceks
Admin.ssl.truststore.name = /keymanager/testkeys
```

```
Admin.ssl.truststore.type = jceks
Audit.event.outcome = success,failure
Audit.event.types = all
Audit.eventQueue.max = 0
Audit.handler.file.directory = /keymanager/audit
Audit.handler.file.name = kms_audit.log
Audit.handler.file.size = 10000
Audit.metadata.file.name = /keymanager/metafile.xml
config.drivetable.file.url = FILE:///keymanager/drivetable
config.keystore.file = /keymanager/testkeys
config.keystore.provider = IBMJCE
config.keystore.type = jceks
fips = Off
TransportListener.ssl.ciphersuites = JSSE_ALL
TransportListener.ssl.clientauthentication = 0
TransportListener.ssl.keystore.name = /keymanager/testkeys
TransportListener.ssl.keystore.type = jceks
TransportListener.ssl.port = 443
TransportListener.ssl.protocols = SSL_TLS
TransportListener.ssl.truststore.name = /keymanager/testkeys
TransportListener.ssl.truststore.type = jceks
TransportListener.tcp.port = 3801
```

다음은 모든 키 저장소 항목이 다른 소프트웨어 키 저장소를 가리키는 샘플 EKM 등록 정보 파일입니다. 굵은체로 표시된 항목은 위의 첫 번째 샘플 등록 정보 파일과 다른 부분입니다.

```
Admin.ssl.keystore.name = /keymanager/adminkeys.jceks
Admin.ssl.keystore.type = jceks
Admin.ssl.truststore.name = /keymanager/admintrustkeys
Admin.ssl.truststore.type = jceks
Audit.event.outcome = success,failure
Audit.event.types = all
Audit.eventQueue.max = 0
Audit.handler.file.directory = /keymanager/audit
Audit.handler.file.name = kms_audit.log
Audit.handler.file.size = 10000
Audit.metadata.file.name = /keymanager/metafile.xml
config.drivetable.file.url = FILE:///keymanager/drivetable
config.keystore.file = /keymanager/drive.keys
config.keystore.provider = IBMJCE
config.keystore.type = jceks
fips = Off
TransportListener.ssl.ciphersuites = JSSE_ALL
TransportListener.ssl.clientauthentication = 0
TransportListener.ssl.keystore.name = /keymanager/sslkeys
TransportListener.ssl.keystore.type = jceks
TransportListener.ssl.port = 443
TransportListener.ssl.protocols = SSL_TLS
TransportListener.ssl.truststore.name = /keymanager/ssltrustkeys
TransportListener.ssl.truststore.type = jceks
TransportListener.tcp.port = 3801
```

부록 B. Encryption Key Manager 구성 등록 정보 파일

Encryption Key Manager에는 Encryption Key Manager 서버 및 CLI 클라이언트에 하나씩 2개의 구성 등록 정보 파일이 필요합니다. 이러한 각 파일은 Java.util.Properties 로드 파일로 처리되고 구문 분석되어 등록 정보의 형식 및 스펙에 다음과 같은 특정 제한사항을 부과합니다.

- 구성 등록 정보는 한 줄에 하나씩 기록됩니다. 특정 등록 정보 값의 범위는 줄 끝까지 입니다.
- 공백이 포함된 암호와 같은 등록 정보 값은 인용 부호로 묶을 필요가 없습니다.
- 키 저장소 암호의 길이는 127자 이하여야 합니다.
- 줄 끝에 잘못된 공백이 있는 경우 이 공백은 등록 정보 값의 일부로 해석됩니다.

샘플 구성 등록 정보 파일은 다운로드할 수 있도록 **http://support.dell.com**에서 EKMServicesandSamples 파일로 제공됩니다.

Encryption Key Manager 서버 구성 등록 정보 파일

다음은 Encryption Key Manager 서버 구성 파일(KeyManagerConfig.properties)의 전체 등록 정보 세트를 구성합니다. 파일에서 등록 정보 설정의 순서는 중요하지 않습니다. 파일에 설명이 나타날 수도 있습니다. 설명을 추가하려면 행의 첫 번째 열에서 『#』 기호를 사용합니다.

주: KeyManagerConfig.properties 파일에서 변경한 사항은 시스템 종료 시 유실될 수 있습니다. 따라서 구성 등록 정보를 편집하기 전에 Encryption Key Manager 서버를 실행하지 않도록 주의하십시오. Encryption Key Manager 서버를 중지하려면 CLI 클라이언트에서 **stopekm** 명령을 실행하십시오. Encryption Key Manager 서버를 다시 시작하면 변경 사항이 활성화됩니다.

Admin.ssl.ciphersuites = *value*

Encryption Key Manager 서버 간 통신에서 사용할 암호 스위트를 지정합니다. 암호 스위트는 데이터 전송 시 데이터 교환 프로토콜 TLS(Transport Layer Security) 및 보안 소켓 계층(SSL: Secure Sockets Layer) 사용과 암호화 알고리즘을 설명합니다.

필수 선택적입니다.

값 가능한 값은 IBMJSSE2에서 지원되는 암호 스위트입니다.

기본값 JSSE_ALL

Admin.ssl.keystore.name = *value*

이 항목은 Encryption Key Manager 서버 간 보안 소켓 계층(SSL) 클라이언트 작업(예: **sync** 명령)에 사용되는 키 쌍 및 인증서 데이터베이스의 이름입니다. sync 작업에서 보안 소켓 클라이언트가 보안 소켓 서버로 제공하는 인증서는 이 키 저장소에서 제공됩니다.

필수 선택적입니다. **sync** 명령에서만 사용합니다. **config.keystore.file** 등록 정보 값이 기본값입니다.

Admin.ssl.keystore.password = password

Admin.ssl.keystore.name에 액세스할 때 사용하는 암호입니다.

필수 선택적입니다. 제공하지 않으면 Encryption Key Manager 시작 시 암호를 묻는 프롬프트가 표시됩니다. 지정하는 경우 추가 보안을 위해 이 등록 정보 값을 인식하기 어렵게 만들며 등록 정보 파일의 스탠자 이름 자체는 'Admin.ssl.keystore.password.obfuscated'라는 이름의 새 스탠자로 바뀝니다.

Admin.ssl.keystore.type = *value*

사용되는 키 저장소 유형입니다.

필수 선택적입니다.

기본값 jceks

Admin.ssl.protocols = *value*

보안 프로토콜입니다.

필수 선택적입니다.

값 SSL_TLS | SSL | TLS

기본값 SSL_TLS

Admin.ssl.timeout = *value*

SocketTimeoutException이 발생하기 전에 소켓에서 읽기까지 대기하는 시간을 지정합니다.

필수 선택적입니다.

값 단위는 분입니다. 0은 시간 종료되지 않음을 나타냅니다.

기본값 1

Admin.ssl.truststore.name = *value*

이 항목은 보안 소켓 서버가 보안 소켓 클라이언트로 제공하는 보안 소켓 서버 인증서를 신뢰하는 데 사용되는 데이터베이스 파일 이름입니다.

필수 선택적입니다. **sync** 명령에서만 사용합니다. **config.keystore.file** 등록 정보 값이 기본값입니다.

Admin.ssl.truststore.type = *value*

사용되는 키 저장소 유형입니다.

필수 선택적입니다.

기본값 jceks

Audit.event.outcome = *value*

지정된 출력을 생성하는 감사 이벤트만 기록됩니다.

필수 예

값 success | failure. 두 경우 모두 쉼표 또는 세미콜론으로 구분하여 지정할 수 있습니다.

기본값 success

Audit.event.Queue.max = 0

파일에서 비워지기 전에 감사 메모리 큐에서 보유하는 최대 이벤트 오브젝트 수입니다.

필수 선택적입니다. 권장됩니다.

값 0 - ? 0은 즉시 비움을 나타냅니다.

기본값 0

Audit.event.types = *value*

지정된 출력을 생성하는 감사 이벤트만 기록됩니다.

필수 예

값 all | authentication | authorization | data synchronization | runtime | audit management | authorization terminate | configuration management | resource management | none. 여러 값을 쉼표 또는 세미콜론으로 구분하여 지정할 수 있습니다.

기본값 all

Audit.handler.file.directory = ../audit

Audit.handler.file.name이 있는 디렉토리입니다.

필수 선택적입니다. 권장됩니다.

Audit.handler.file.multithreads = *value*

감사 핸들러에서 별도의 스레드를 디스패치하여 감사 레코드를 처리해야 하는지 여부를 지정합니다.

필수 선택적입니다.

값 true | false

기본값 true

Audit.handler.file.name = kms_audit.log

감사 항목이 기록되는 파일 이름입니다.

필수 예

Audit.handler.file.size = 100

Audit.Handler.file.name이 겹쳐쓰기 전까지 증가할 수 있는 크기입니다.

필수 선택적입니다. 권장됩니다.

값 0 - ? 단위는 KB입니다.

기본값 100

Audit.handler.file.threadlifespan = *value*

감사 레코드 처리 스레드의 수명을 제한합니다. audit.handler.file.multithreads= true인 경우에만 사용 가능합니다.

필수 선택적입니다.

값 단위는 밀리초입니다.

기본값 10000

Audit.metadata.file.cachecount = 100

메타데이터 파일을 쓰기 전까지 메모리에 저장할 수 있는 레코드 수를 지정합니다.

필수 아니오

기본값 100

Audit.metadata.file.name = *value*

메타데이터 레코드를 저장할 XML 파일 이름을 지정합니다.

필수 예

Audit.metadata.file.size = 1024

파일을 닫고 새 파일을 시작하기 전까지 XML 메타데이터 파일에서 보관할 수 있는 최대 파일 크기(단위: KB)를 지정합니다. 파일의 현재 버전과 이전 버전만 저장됩니다.

필수 아니오

기본값 1024

config.drivetable.file.url = FILE:../filedrive.table

일련 번호, 인증서 등과 같은 테이프 드라이브와 관련된 정보를 포함하는 파일입니다.

필수 예

config.keygroup.xml.file = *value*

개별 별명을 키 그룹으로 저장하는 XML 파일 이름을 지정합니다.

필수 선택적입니다.

config.keystore.file = *value*

사용할 키 저장소를 지정합니다.

필수 예

config.keystore.password = password

config.keystore.file에 액세스할 때 사용하는 암호입니다. 지정하는 경우 추가 보안을 위해 이 등록 정보 값을 인식하기 어렵게 만들며 등록 정보 파일의 스탠자 이름 자체는 'config.keystore.password.obfuscated'라는 이름의 새 스탠자로 바뀝니다.

필수 선택적입니다. 제공하지 않으면 Encryption Key Manager 시작 시 암호를 묻는 프롬프트가 표시됩니다.

config.keystore.provider = IBMJCE

필수 선택적입니다.

config.keystore.type = jceks

필수 선택적입니다. 권장됩니다.

기본값 jceks

debug = *value*

지정된 Encryption Key Manager 컴포넌트에서 디버그를 설정합니다.

필수 선택적입니다.

값 all | audit | server | drivetable | config | admin | transport | logic | keystore | console | none. 여러 값을 쉼표로 분리하여 사용할 수 있습니다.

기본값 none

debug.output = *value*

지정된 위치로 디버그 출력을 라우트합니다.

필수 선택적입니다.

값 simple_file | console(권장되지 않음)

debug.output.file = debug

디버그 출력을 쓰는 파일 이름 및 해당 경로입니다.

필수 선택적입니다. debug.output = simple_file인 경우 필수입니다. 파일에 대한 경로도 있어야 합니다.

drive.acceptUnknownDrives = *value*

Encryption Key Manager를 드라이브 테이블에 연결하는 새 드라이브를 자동으로 추가합니다.

필수 예

값 true | false

기본값 false

보안 참고 - 올바른 drive.default.alias1 설정과 함께 조합하여 이 설정을 사용하면 Encryption Key Manager에 연결되는 테이프 드라이브를 관리자가 해당 장치 추가의 유효성을 검증하지 않고도 추가 및 작동시킬 수 있습니다. 자세한 정보는 3장, 『테이프 드라이브 테이블 자동 업데이트』를 참조하십시오.

fips = *value*

FIPS(Federal Information Processing Standard). 자세한 정보는 2장, 『FIPS(Federal Information Processing Standard) 140-2 고려사항』을 참조하십시오.

필수 선택적입니다.

값 on | off

기본값 off

maximum.threads = 200

Encryption Key Manager에서 작성할 수 있는 최대 스레드 수입니다.

필수 선택적입니다.

Server.authMechanism = *value*

로컬/원격 클라이언트에서 사용할 인증 메커니즘을 지정합니다. 값이 EKM으로 설정되면 CLI 클라이언트 사용자는 사용자 이름/암호로 EKMAdmin/changeME를 사용하여 서버에 로그인해야 합니다. 이 암호는 chgpasswd 명령으로 변경할 수 있습니다. 값이 LocalOS로 지정되면 클라이언트 인증은 로컬 운영 체제 레지스트리에서 수행됩니다. (KeyManagerConfig.properties 파일을 변경하기 전에 Encryption Key Manager 서버를 작동 중지하십시오.) 이 경우 CLI 클라이언트 사용자는 OS 사용자 이름/암호로 로그인해야 합니다. Linux 플랫폼에서의 로컬 운영 체제에 기반한 인증 시 다음과 같은 추가 단계를 수행하십시오.

1. **http://support.dell.com**에서 Dell 릴리스 R175158(EKMServicesAndSamples)을 다운로드하여 원하는 디렉토리에 추출하십시오.
2. EKMServiceAndSamples.jar(Dell 제품 매체에 포함되어 있으며 **http://support.dell.com**에서 사용 가능함) 내용을 임시 디렉토리에 추출하십시오.
3. 플랫폼에 해당하는 LocalOS-setup의 libjaasauth.so 파일을 *java_home*/jre/bin으로 복사하십시오.

- 32비트 Intel Linux 환경에서, LocalOS-setup/linux_ia32/libjaasauth.so 파일을 *java_home*/jre/bin/ 디렉토리로 복사하십시오. 여기서 *java_home*은 보통 1.4.2 JVM을 실행하는 32비트 Intel Linux 커널의 경우 *java_install_path*/IBMJava2-i386-142입니다.
- 64비트 AMD64 Linux 환경에서, LocalOS-setup/linux-x86_64/libjaasauth.so 파일을 *java_home*/jre/bin/ 디렉토리로 복사하십시오. 여기서 *java_home*은 보통 1.4.2 JVM을 실행하는 64비트 AMD Linux 커널의 경우 *java_install_path*/IBMJava2-amd64-142입니다.

Windows 플랫폼의 경우 이 파일은 필요하지 않습니다.

설치가 완료되면 Encryption Key Manager 서버를 시작할 수 있습니다. 이제 Encryption Key Manager 클라이언트는 운영 체제 기반 사용자/암호를 사용하여 로그인할 수 있습니다. 서버에 로그인하여 명령을 제출할 수 있도록 허용된 사용자 ID만이 서버를 실행하고 superuser/루트 권한을 갖습니다.

Dell 제품 매체에 포함되어 있으며 **http://support.dell.com**에서 제공하는 readme 파일에서는 추가 설치 세부사항을 제공합니다.

필수 선택적입니다.

값 EKM | LocalOS

기본값 EKM

Server.password = *value*

내부 등록 정보입니다. 편집하지 마십시오.

symmetricKeySet = {GroupID | keyAliasList [, keyAliasList,]}

LTO 4 및 LTO 5 테이프 드라이브에서 사용할 대칭 키 별명 및 키 그룹을 지정합니다.

필수 선택적입니다. LTO 4 및 LTO 5 테이프 카트리지에만 적용됩니다.

값

*GroupID*에 대한 하나의 값 또는 *keyAliasList*에 대한 하나 이상의 값을 지정합니다.

*GroupID*는 대칭 키 목록을 제공하는 키 그룹 이름을 지정하며 테이프 드라이브에 지정된 별명이 없을 때 기본값으로 사용됩니다. *GroupID*는 KeyGroups.xml 파일의 기존 키 그룹 ID와 일치해야 합니다. 그렇지 않으면 KeyManageException이 리턴됩니다. 둘 이상의 *GroupID*가 지정되면 KeyManagerException이 리턴됩니다. 올바른 *GroupID*를 지정하면 키 그룹 XML에서 사용된 마지막 키가 추적되고 KeyGroups.xml에서 getKey를 호출할 때마다 대칭 키 목록에

서 다음 키를 무작위로 선택하여 사용됩니다. *keyAliasList*의 각 스펙은 *keyAlias* 또는 *keyAliasRange*에 대한 값 중 하나를 포함합니다.

*keyAlias*는 키 저장소에서 대칭 키 이름 또는 별명으로 BNF(Backus-Naur Form)를 최대 12자로 지정하거나 sequentialKeyID를 정확히 21자로 지정합니다.

*keyAliasRange*는 sequentialKeyID 및 16진 숫자를 최대 18자(하이픈(-)으로 구분)로 지정합니다. 18자를 지정하는 경우 처음 2자는 00이어야 합니다. 또한 한 행에 지정해야 하며 cr-lf를 포함할 수 없습니다.

*GroupID*는 별명 그룹 이름을 지정합니다.

예제 `symmetricKeySet = KMA0238ab34,KMB0000034acd2345678a,THZ001-FF` 이 항목은 LTO 4 및 LTO 5 테이프 드라이브에 키를 제공할 때 Encryption Key Manager에 별명으로 KMA0238ab34, KMB0000034acd2345678a를 사용하고 별명 범위로 THZ000000000000000001 - THZ00000000000000FF를 사용하도록 지시합니다. 이 키는 등록 정보 파일의 **config.keystore.file**에서 지정한 키 저장소에 있어야 합니다.

sync.action = *value*

자동 동기화 중 데이터에서 수행할 작업을 지정합니다.

필수 선택적입니다.

값 rewrite | merge

기본값 merge

주: 병합 구성 정보는 다시 작성하는 정보와 동일합니다.

sync.ipaddress = *ip_addr:ssl*

자동으로 동기화하도록 원격 Encryption Key Manager의 IP 주소 및 포트를 지정합니다.

필수 선택적입니다. 이 등록 정보를 지정하지 않거나 잘못 지정하면 sync 기능을 사용할 수 없습니다.

값 원격 서버 IP 주소:SSL 포트 번호

sync.timeinhours = *value*

원격 Encryption Key Manager와 자동으로 동기화하기 전에 대기해야 하는 시간을 지정합니다.

필수 선택적입니다.

값 단위는 시간입니다.

기본값 24

sync.type = *value*

자동으로 동기화할 데이터를 지정합니다.

필수 선택적입니다.

값 config | drivetab | all

기본값 drivetab

TransportListener.ssl.ciphersuites = JSSE_ALL

Encryption Key Manager 서버 간 통신에서 사용할 암호 스위트입니다. 암호 스위트는 데이터 전송 시 데이터 교환 프로토콜 TLS(Transport Layer Security) 및 보안 소켓 계층(SSL: Secure Sockets Layer) 사용과 암호화 알고리즘을 설명합니다.

필수 선택적입니다.

값 값 – IBMJSSE2에서 지원하는 모든 암호 스위트가 가능합니다.

TransportListener.ssl.clientauthentication = 0

Encryption Key Manager서버 간 통신에 필요한 SSL 인증입니다.

필수 선택적입니다.

값 0 - 클라이언트를 인증하지 않음(기본값)
1 - 서버가 클라이언트에서 클라이언트 인증을 수행하려고 함
2 - 서버가 클라이언트에서 클라이언트 인증을 반드시 수행해야 함

TransportListener.ssl.keystore.name = *value*

Encryption Key Manager 서버에서 보안 소켓 서버의 인증서 및 개인용 키를 보유할 때 사용하는 데이터베이스 이름입니다. 이 인증서는 인증 및 신뢰 확인 시 보안 소켓 클라이언트에 제공됩니다. 이 키 저장소는 Encryption Key Manager 클라이언트가 Encryption Key Manager 서버와 통신하고 보안 소켓 클라이언트 역할을 수행할 때 사용되기도 합니다.

필수 예

TransportListener.ssl.keystore.password = password

TransportListener.ssl.keystore.name에 액세스할 때 사용하는 암호입니다. 지정하는 경우 추가 보안을 위해 이 등록 정보 값을 인식하기 어렵게 만들며 등록 정보 파일의 스탠자 이름 자체는
'TransportListener.ssl.keystore.password.obfuscated'라는 이름의 새 스탠자로 바뀝니다.

필수 선택적입니다.

TransportListener.ssl.keystore.type = jceks

필수 선택적입니다. 권장됩니다.

값 JCEKS

TransportListener.ssl.port = *value*

Encryption Key Manager 서버가 다른 Encryption Key Manager 서버 또는 Encryption Key Manager CLI 클라이언트의 요청을 청취하는 포트입니다.

필수 예

값 예를 들어 포트 번호는 443입니다. 이 값은 CLI 클라이언트 구성 등록 정보 파일의 TransportListener.ssl.port 등록 정보와 일치해야 합니다.

TransportListener.ssl.protocols = SSL_TLS

보안 프로토콜입니다.

필수 선택적입니다.

값 SSL_TLS (기본값) | SSL | TLS

TransportListener.ssl.timeout = 10

SocketTimeoutException이 발생하기 전에 소켓에서 읽기까지 대기하는 시간을 지정합니다.

필수 선택적입니다.

값 단위는 분입니다.

기본값 1

TransportListener.ssl.truststore.name = *value*

다른 클라이언트 및 서버의 정체를 확인하는 데 사용되는 서명된 인증서 및 공용 키를 포함하는 데이터베이스 이름입니다.

TransportListener.ssl.clientauthentication 등록 정보의 기본값은 0(클라이언트를 인증하지 않음)으로 설정되지 **않습니다**. 따라서 보안 소켓 서버 역할을 하는 Encryption Key Manager 서버가 이 파일을 사용하여 클라이언트를 인증해야 합니다. 이 신뢰 저장소는 Encryption Key Manager 클라이언트가 Encryption Key Manager 서버와 통신하고 보안 소켓 클라이언트 역할을 수행할 때 사용되기도 합니다.

필수 예

TransportListener.ssl.truststore.type = jceks

필수 선택적입니다. 권장됩니다.

값 JCEKS

TransportListener.tcp.port = *value*

Encryption Key Manager 서버가 테이프 드라이브의 요청을 청취하는 포트입니다. 기본 TCP 포트 번호는 3801입니다.

필수 예

값 예를 들어 포트 번호는 10입니다.

TransportListener.tcp.timeout = *value*

SocketTimeoutException이 발생하기 전에 소켓에서 읽기까지 대기하는 시간을 지정합니다.

필수 선택적입니다.

값 단위는 분입니다. 0은 시간 종료되지 않음을 나타냅니다.

기본값 10

CLI 클라이언트 구성 등록 정보 파일

이 ClientKeyManagerConfig.properties 파일에는 KeyManagerConfig.properties 파일에 포함된 등록 정보 서브세트가 있습니다. 이 서브세트는 다음 등록 정보를 포함합니다.

TransportListener.ssl.ciphersuites = JSSE_ALL

Encryption Key Manager 서버 및 CLI 클라이언트 간 통신에서 사용할 암호 스위트입니다. 암호 스위트는 데이터 전송 시 데이터 교환 프로토콜 TLS(Transport Layer Security) 및 보안 소켓 계층(SSL: Secure Sockets Layer) 사용과 암호화 알고리즘을 설명합니다.

필수 선택적입니다.

값 이 값은 Encryption Key Manager 서버 등록 정보 파일 (KeyManagerConfig.properties)의 TransportListener.ssl.ciphersuites에 지정된 값과 일치해야 합니다.

TransportListener.ssl.host = *value*

Encryption Key Manager CLI 클라이언트에 대해 Encryption Key Manager 서버를 식별합니다.

필수 선택적입니다.

값 IP 주소 또는 호스트 이름

기본값 localhost

예제
```
TransportListener.ssl.host = 9.24.136.444
TransportListener.ssl.host = ekmsvr02
```

주: KeyManagerConfig.properties 파일에서 사용되지 않습니다.

TransportListener.ssl.keystore.name = *value*

이 키 저장소는 Encryption Key Manager 클라이언트가 Encryption Key Manager 서버와 통신하고 보안 소켓 클라이언트 역할을 수행할 때 사용되기도 합니다.

필수 예

TransportListener.ssl.keystore.type = jceks

입력 유형입니다.

필수 선택적입니다. 권장됩니다.

기본값 jceks

TransportListener.ssl.port = *value*

Encryption Key Manager 서버와 통신할 때 CLI 클라이언트가 사용하는 포트입니다.

필수 예

값 이 값은 Encryption Key Manager 서버 등록 정보 파일 (KeyManagerConfig.properties)의 TransportListener.ssl.port에 지정된 값과 일치해야 합니다.

TransportListener.ssl.protocols = SSL_TLS

보안 프로토콜입니다.

필수 선택적입니다.

값 이 값은 Encryption Key Manager 서버 등록 정보 파일 (KeyManagerConfig.properties)의 TransportListener.ssl.protocols에 지정된 값과 일치해야 합니다.

TransportListener.ssl.truststore.name = *value*

다른 클라이언트 및 서버의 정체를 확인하는 데 사용되는 서명된 인증서 및 공용 키를 포함하는 데이터베이스 이름입니다.

필수 예

TransportListener.ssl.truststore.type = jceks

신뢰 저장소 유형입니다.

필수 선택적입니다. 권장됩니다.

기본값 jceks

샘플 구성 등록 정보 파일은 다운로드할 수 있도록 **http://support.dell.com**에서 EKMServicesAndSamples 파일로 제공됩니다.

부록 C. 자주 묻는 질문(FAQ)

응용프로그램 기반 키 관리 및 라이브러리에서 관리하는 암호화를 조합하여 사용할 수 있습니까?

아니오. 응용프로그램에서 관리하는 암호화를 사용하는 경우 암호화는 라이브러리 계층에서 투명합니다. 마찬가지로 라이브러리에서 관리하는 암호화를 사용하는 경우 해당 프로세스는 다른 계층에서 투명합니다. 각 암호화 관리 방법은 다른 방법에 대해 배타적이므로 함께 사용할 수 없습니다. 라이브러리에서 관리하는 암호화의 경우 응용프로그램을 변경하지 않아도 됩니다.

테이프를 암호화하거나 암호 해독하는 요청을 생성할 수 있는 모든 시스템에 Encryption Key Manager를 설치하여 실행해야 합니까?

라이브러리에서 관리하는 암호화의 경우 테이프 드라이브 쓰기 요청이 생성되는 시스템이 Encryption Key Manager을 실행하는 시스템일 필요는 없습니다. 또한 Encryption Key Manager 인스턴스를 암호화한 테이프 드라이브에 액세스하는 모든 시스템에서 실행할 필요는 없습니다.

″drive.acceptUnknownDrives = True″ 매개변수를 포함하는 경우 구성 파일에 ″config.drivetable.file.url = FILE:/filename″ 매개변수를 계속 포함해야 합니까?

config.drivetable.file.url은 항상 지정되어야 합니다. 이 위치에 드라이브 정보가 있습니다. `drive.acceptUnknownDrives = True`로 설정하면 올바른 인증서 별명/키 레이블로 `drive.default.alias1` 및 `drive.default.alias2` 변수를 지정해야 합니다.

`config.drivetable.file.url` 등록 정보의 올바른 구문이 FILE:/filename입니까? 샘플 파일에서는 FILE:///filename으로 설명에서는 FILE:../로 나타납니다.

예제가 올바릅니다. 이 항목은 URL 스펙으로 디렉토리 구조 스펙의 형식과는 다릅니다.

Windows에서 실행되는 Encryption Key Manager 인스턴스의 KeyManagerConfig.properties 파일에 완전한 경로를 지정할 때 슬래시 또는 백슬래시를 사용해야 합니까?

KeyManagerConfig.properties는 Java 등록 정보 파일이므로 Windows에서도 슬래시만 경로 이름에서 인식됩니다. KeyManagerConfig.properties 파일에서 백슬래시를 사용하면 오류가 발생합니다.

Encryption Key Manager에서 CRL(Certificate Revocation List) 검사를 수행합니까?

아니오. Encryption Key Manager에서는 CRL 검사를 수행하지 않습니다.

테이프를 암호화하는 데 사용하는 인증서가 만기되면 어떻게 됩니까? Encryption Key Manager에서 이전에 암호화된 테이프를 읽습니까?

인증서 만기 여부는 Encryption Key Manager에서 중요하지 않습니다. 이 인증서를 계속 사용하여 이전에 암호화된 테이프를 읽습니다. 그러나 이전에 암호화된 테이프를 읽거나 추가하려면 만기된 인증서를 계속 키 저장소에 두어야 합니다.

Encryption Key Manager는 인증서를 갱신할 때 이름을 바꿉니까?

Encryption Key Manager는 기본적으로 인증서 만기 시 새 키를 요청하도록 구성되어 있습니다. Encryption Key Manager가 이렇게 구성된 경우 인증서 갱신이 필요하지 않습니다. 이 기능이 사용 불가능하고 이 개인용 키/인증서 쌍이 새 키 요청에도 사용되는 경우에는 인증서를 갱신해야 합니다. 인증서만(유효 날짜) 갱신되고 연관된 키는 갱신되지 않습니다.

Encryption Key Manager의 나중 버전에서 소프트웨어의 이전 버전에서 작성한 암호화된 테이프를 계속 읽을 수 있습니까?

예. Encryption Key Manager는 릴리스에 상관없이 인증서를 계속 사용합니다.

주의사항

상표

이 본문에 사용되는 상표인 Dell, Dell 로고 및 PowerVault는 Dell Inc.의 상표이며, Microsoft와 Windows는 Microsoft Corporation의 등록상표입니다. 기타 상표 및 상호를 사용하는 법인 또는 이들 법인의 제품을 언급하기 위해 타사의 상표 및 상호가 본 문서에서 사용될 수 있습니다. Dell Inc.는 자사의 것이 아닌 상표 또는 상호에 대해 어떠한 소유권도 갖지 않습니다.

용어

이 용어집에는 본 서적 및 관련 서적에 사용되는 특수 용어, 약어 및 두문자어가 정의되어 있습니다.

개인용 키(Private key). 일반적으로 암호 해독에 사용되는 하나의 비대칭 키 쌍에 있는 한 키입니다. Encryption Key Manager는 개인용 키를 사용하여 암호 해독 이전에 보호되는 AES 데이터 키 랩핑을 해제합니다.

공용 키(Public key). 일반적으로 암호화에 사용되는 하나의 비대칭 키 쌍에 있는 한 키입니다. Encryption Key Manager는 공용 키를 사용하여 테이프 카트리지에 저장하기 전에 AES 데이터 키를 랩핑(보호)합니다.

별명(Alias). 키 레이블을 참조하십시오.

암호화(encryption). 데이터를 암호로 변환하는 작업입니다. 데이터를 암호화하고 암호 해독하는 경우 키가 필요합니다. 암호화를 통해 키 없이도 데이터에 액세스하려는 소프트웨어 및 사용자를 보호합니다.

인증서 레이블(Certificate label). 키 레이블을 참조하십시오.

인증서 스토어(Certificate store). 키 저장소를 참조하십시오.

인증서(Certificate). 인증서 소유자 ID에 대한 공용 키에 바인드한 디지털 문서로 이를 통해 인증서 소유자를 인증할 수 있습니다.

키 레이블(Key label). 보호되는 대칭 데이터 키 랩핑을 해제하는 데 필요한 개인용 키(KEK)를 EEDK와 일치시킬 때 사용되는 고유 ID입니다. 사용하는 키 저장소에 따라 인증서 레이블 또는 별명이라고도 합니다.

키 링(Key ring). 키 저장소를 참조하십시오.

키 저장소(Keystore). 개인용 키 및 이와 연관된 X.509 디지털 인증서 체인으로 구성된 데이터베이스로 이에 대응하는 공용 키를 인증하는 데 사용됩니다. 일부 환경에서는 인증서 스토어 또는 키 링이라고도 합니다.

AES. Advanced Encryption Standard. 미국 정부에서 암호화 표준으로 채택한 블록 암호입니다.

DK. 데이터 키. 데이터를 암호화하는 데 사용되는 영숫자 문자열입니다.

EEDK. Externally Encrypted Data Key. 데이터 카트리지에 저장하기 전에 키 암호화 키에서 암호화(랩핑)되는 데이터 키입니다. KEK를 참조하십시오.

KEK. 키 암호화 키(Key Encrypting Key). 데이터 키를 암호화하는 데 사용되는 영숫자 비대칭 키입니다. EEDK를 참조하십시오.

PKDS. 공용 키 데이터 세트(Public Key Data Set). PKA 암호화 키 데이터 세트라고도 합니다.

rekey. 다른 엔티티에서 데이터에 액세스할 수 있도록 이미 암호화된 테이프에 저장된 데이터 키(DK)를 보호하는 비대칭 키 암호화 키(KEK)를 변경하는 프로세스입니다.

RSA. Rivest-Shamir-Adleman 알고리즘. 암호화 및 인증 시 사용되는 비대칭 공용 키 암호에 대한 시스템입니다. 1977년 Ron Rivest, Adi Shamir, Leonard Adleman이 개발했습니다. 이 시스템 보안은 아주 큰 2개의 소수의 제품을 인수 분해하는 경우의 난이도에 따라 다릅니다.

색인

[가]

[다]

[라]

[마]

[사]

[아]