

# 모바일 앱 개발 방식 비교: 네이티브, 웹, 하이브리드



## 내용

### 2 개요

#### 2 모바일 앱 개발 방식에 대하여

- 2 네이티브 앱
- 3 애플리케이션 프로그래밍 인터페이스(API)
- 4 모바일 웹 앱
- 4 하이브리드 앱
- 5 세 가지 개발 방식 비교

### 6 적합한 방식 선택

- 6 네이티브 방식 시나리오
- 6 웹 방식 시나리오
- 6 하이브리드 방식 시나리오

### 7 요약

## 개요

모바일을 우선과제로 선정하고 추진하는 기업들은 모바일 과제의 성패에 영향을 미칠 모바일 앱 개발 방식을 먼저 결정해야 합니다. 네이티브, 웹, 하이브리드로 대표되는 모바일 애플리케이션(이후 앱으로 지칭) 개발 방식 중 어떤 개발 방식을 선택할 것인가 하는 것은, 기업에 배정된 예산, 프로젝트 기간, 사용자 요건, 앱 기능 등을 종합적으로 고려해 정해져야 합니다.

이 백서에서는 3가지 개발 방식의 장단점을 살펴보고, 기업이 처한 상황에 따라 어떤 개발 방식이 더 적합한지 제언하도록 하겠습니다.

## 모바일 앱 개발 방식에 대하여

### 네이티브 앱

네이티브 앱은 모바일 기기에 직접 다운로드하여 로컬(기기)에 저장되는 바이너리 실행 파일입니다. 사용자가 직접 설치하거나 경우에 따라 기업의 관리자가 설치할 수 있습니다. 네이티브 앱은 애플사의 앱스토어(App Store), 안드로이드의 마켓플레이스(Marketplace), 블랙베리사의 앱 월드(App World) 등과 같은 공용 앱스토어에서 다운로드하면 되나, 모바일 공급업체에서 직접 제공하는 경우도 있습니다.

모바일 기기에 앱을 설치한 후에는 기기에서 제공하는 다른 서비스처럼 사용자가 앱을 실행합니다. 초기화할 경우 네이티브 앱 인터페이스는 중개 요소나 컨테이너 없이 모바일 운영체제와 직접 통신합니다. 네이티브 앱은 모바일 OS에서 제공하는 모든 API에 액세스할 수 있으며, 대부분 해당 모바일 OS의 고유 기능을 활용합니다.

네이티브 앱을 만들려면 개발자가 사람이 읽을 수 있는 형식으로 소스 코드를 작성하고, 이미지, 오디오, 다양한 OS별 선언 파일 등의 리소스를 만들어야 합니다. 이러한 리소스를 개발한 후 모바일 OS에서 제공하는 툴을 이용해 소스 코드를 컴파일하고 배포할 수 있는 바이너리 형태의 실행 파일을 만듭니다.

이러한 툴 및 기타 유틸리티/파일을 일반적으로 모바일 OS의 소프트웨어 개발 키트(SDK) 라고 합니다. 네이티브 앱 개발 과정은 모바일 운영체제 특성을 타지 않고 모두 비슷하지만, SDK는 OS별로 다르며, 모바일 OS별로 고유 툴을 제공합니다. [표 1]은 주요 모바일 OS에서 제공하는 툴, 언어, 패키징 형식, 앱 배포 방법을 보여줍니다.

모바일 OS에서 제공하는 언어, 툴 등이 다르다는 점이 네이티브 앱 개발 방식의 가장 큰 단점입니다. 예를 들어 애플 iOS용으로 작성된 코드를 안드로이드에서는 사용할 수가 없습니다. 그러므로 멀티 플랫폼을 지원하는 네이티브 앱을 개발하려면 개발 기간도 많이 걸리고 비용도 많이 듭니다.

이런 단점에도 불구하고 기업들이 네이티브 방식을 선호하는 이유는 애플리케이션 프로그래밍 인터페이스(이하 API) 때문입니다.

**애플리케이션 프로그래밍 인터페이스(API)**

모바일 기기에 네이티브 앱을 설치하고 실행하면, 앱은 모바일 운영체제에서 제공하는 전용 API를 호출해 모바일 운영체제와 통신합니다. 모바일 운영체제에서 제공하는 API는 로우 레벨(Low Level) API와 하이 레벨(High Level) API로 구분됩니다.

**로우 레벨 API**

앱은 로우 레벨 API를 호출해 터치 스크린이나 키보드와 직접 입출력을 하고, 그래픽을 렌더링하며, 네트워크를 연결합니다. 또한 마이크를 통해 수신한 오디오를 처리하고, 스피커나 헤드폰을 통해 사운드를 재생하며, 카메라로부터 이미지나 동영상 등을 수신해 저장합니다. GPS(Global Positioning System) 액세스, 방향 정보 수신, 반도체 디스크 또는 상용 하드웨어의 파일 읽기 및 쓰기 등도 로우 레벨 API 호출로 이뤄집니다.

**하이 레벨 API**

모바일 운영체제는 로우 레벨 하드웨어 액세스 서비스 이외에, 개인의 사용자 경험에 영향을 미치는 하이 레벨 서비스도 제공합니다. 하이 레벨 서비스는 웹 검색, 일정 및 연락처 관리, 사진 앨범 관리 및 전화 또는 문자 메시지 송수신 기능 등을 일컫습니다.

대부분의 모바일 OS는 이러한 서비스를 수행할 수 있는 일련의 애플리케이션을 내장하고 있지만, 네이티브 앱에서는 퍼블릭하게 공개된 하이 레벨 API를 불러 사용할 수도 있습니다. 다운로드가 가능한 앱은 다른 API를 통해 OS 업체가 제공하는 푸시 알림이나 인앱(In App) 구매와 같은 다양한 클라우드 기반의 서비스에 액세스할 수 있습니다.

**그래픽 사용자 인터페이스(GUI) 툴킷**

OS 업체에서 제공하는 또 다른 중요한 API 세트는 GUI 툴킷입니다. 주요 모바일 OS는 버튼, 입력 필드, 슬라이더, 메뉴, 탭 바, 대화 상자와 같은 OS 고유의 사용자 인터페이스 구성요소를 제공합니다. 이러한 구성요소들은 해당 모바일 OS의 기능을 그대로 전이 받으므로 사용자가 앱을 쉽게 사용하고 즐길 수 있습니다.

각각의 모바일 플랫폼은 고유의 사용자 인터페이스(UI) 구성요소를 가지고 있습니다. 따라서 여러 운영체제에서 작동하는 앱을 설계하려면 설계자가 각 OS의 서로 다른 UI 구성요소에 대해 잘 알고 있어야 합니다.

API가 OS별로 다르므로 여러 개의 네이티브 앱을 개발하려면 개발시간도 많이 걸리고 비용도 많이 들지만, 모바일 기기가 제공하는 모든 기능을 제대로 활용하는 모바일 앱을 개발하기 위해서는 API 활용은 반드시 필요합니다.

	애플 iOS	안드로이드	블랙베리 OS	윈도우 폰
언어	Objective-C, C, C++	Java(일부 C, C++)	Java	C#, VB.NET 등
툴	Xcode	Android SDK	BB Java Eclipse 플러그인	Visual Studio, 윈도우 폰 개발 툴
패키징 형식	.app	.apk	.cod	.xap
앱스토어	Apple App Store	Google Play	Blackberry App World	Windows Phone Marketplace

**모바일 웹 앱**

최신 모바일 기기는 HTML5의 최신 기능, Cascading Style Sheets 3(CSS3) 및 고급 JavaScript를 지원하는 브라우저로 구성되어 있습니다. 이러한 최신 기능과 함께 HTML5는 페이지 정의 언어에서 브라우저 기반의 풍부한 기능을 가진 애플리케이션을 개발하는 표준으로 자리잡고 있습니다.

HTML5는 유려한 UI 구성요소, 다양한 미디어 형식 액세스, 위치 정보 서비스, 네트워크 끊긴 상태에서도 오프라인으로 사용할 수 있는 다양한 기능을 지원합니다. 이러한 HTML5 기술과 현재 개발 중인 많은 기능을 사용하여 개발자들은 웹 기술만으로 고품질의 애플리케이션을 만들 수 있습니다.

웹 앱 방식은 두 가지로 구분할 수 있는데, 두 가지 방식의 차이점을 비교하면서 보면 이해가 빠릅니다. 모바일 검색 및 모바일용 웹 사이트는 누구나 익숙하게 사용합니다. 모바일 웹 사이트는 스마트폰에서 액세스하고 작은 크기의 화면에서 터치하기 편안하게 설계된 HTML 페이지를 제공합니다. 일부 기업에서는 네이티브 앱을 실행하는 것처럼 바로가기로 실행할 수 있는 네이티브 앱과 같은 모바일 웹 사이트를 만들어 사용자 환경을 더욱 개선하고 있습니다.

대부분의 웹 사이트에서는 각자 고유의 기능을 구현하기 때문에 두 가지 웹 방식 모두 광범위한 가능성을 보유하고 있습니다. 모바일 웹 앱은 특히나 가능성이 많습니다. 이러한 추세를 반영해 dojox.mobile, Sencha Touch, jQuery Mobile 등의 JavaScript 툴킷이 개발되었습니다. 이러한 툴킷은 네이티브 앱과 같은 사용자 인터페이스를 생성합니다. 이들 모두 모바일 디바이스의 브라우저에서 실행되고 모바일 브라우저에서 지원하는 JavaScript, CSS 및 HTML5 기능을 활용합니다.

웹 앱의 가장 큰 장점 중 하나는 멀티 플랫폼을 지원한다는 것과 개발 비용이 적게 든다는 점입니다. 대부분의 모바일 업체들은 브라우저에 WebKit이라는 동일한 렌더링 엔진을 사용합니다. WebKit은 Google과 Apple에서 주도한 오픈 소스 프로젝트로, 오늘날 사용 가능한 가장 포괄적인 HTML5 구현을 제공합니다. 애플리케이션 코드가 WebKit과 호환되는 표준 웹 언어로 작성되기 때문에 한 앱에서 다른 디바이스 및 운영체제를 지원할 수 있으며 기본적으로 멀티 플랫폼 지원이 됩니다.

모바일 분야에서의 잠재력과 웹 기술이 지속적으로 발전할 것이라는 밝은 전망에도 불구하고 모바일 웹 방식은 아직 한계가 있습니다. 웹 앱의 작동 방식을 이해하면 한계점이 무엇인지 금방 이해할 수 있습니다.

OS와 직접 연동하여 독립적으로 실행하는 네이티브 앱과 달리 웹 앱은 브라우저 내에서 실행합니다. 브라우저는 그 자체가 OS API에 직접 액세스하는 네이티브 앱이지만 브라우저 내에서 실행되는 웹 앱이 액세스할 수 있는 API의 수에는 제한이 있습니다. 네이티브 앱은 디바이스에 대해 전체 액세스 권한을 갖지만 많은 기능은 웹 앱에서 제한적으로 사용됩니다.

**하이브리드 앱**

하이브리드 방식은 네이티브 개발과 웹 기술을 혼합한 것입니다. 이 방식을 사용하면 개발자가 모바일 플랫폼과 상관없이 웹 기술로 애플리케이션을 개발할 수 있으며, 필요 시 네이티브 API에 직접 액세스할 수 있습니다.

애플리케이션의 네이티브 부분은 운영체제 API를 사용하여 브라우저와 디바이스 API 간에 연결고리 역할을 하는 임베디드 HTML 렌더링 엔진을 만듭니다. 이 연결고리를 통해 하이브리드 앱은 최신 디바이스가 제공하는 모든 기능을 활용할 수 있습니다.

특성	Pure 모바일 웹 앱	Pure 모바일 웹 사이트
도구 및 지식	HTML, CSS 및 JavaScript로 작성	HTML, CSS 및 JavaScript로 작성
실행	바로가기 설치 가능. 네이티브 앱처럼 실행	URL(Uniform Resource Locator)을 통해 웹 사이트를 탐색하여 액세스
사용자 환경	터치식의 상호 작용 UI	고정 데이터 기반 페이지 간 이동을 내비게이션 방식으로 하는 UI
성능	UI 로직이 로컬에 상주하므로 앱 응답이 빠르고 오프라인 액세스 가능	모든 코드가 서버에서 실행되므로 네트워크 접속 환경에 따라 성능 좌우됨

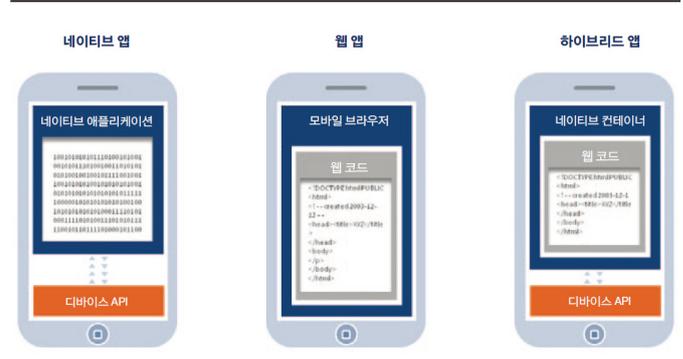
앱 개발자는 이 기능을 직접 코딩하거나 PhoneGap과 같은 기존 솔루션을 활용할 수 있습니다. PhoneGap은 모든 모바일 운영체제에 일관된 단일 JavaScript 인터페이스를 제공하는 오픈 소스 라이브러리입니다.

앱의 네이티브 부분은 독립적으로 개발할 수 있지만, 일부 솔루션들은 네이티브 컨테이너를 제품의 구성요소로 제공하기도 합니다. 따라서 개발자는 웹 언어만을 사용하여 모든 디바이스 기능을 활용하는 고급 애플리케이션을 개발할 수 있습니다. 경우에 따라 개발자가 기업의 고유한 요구사항에 따라 네이티브 컨테이너를 직접 지정할 수 있도록 허용하는 솔루션도 있습니다.

앱의 웹 부분은 서버에 상주하는 웹 페이지가 될 수도 있고, 애플리케이션 코드에 통합되어 디바이스에 로컬로 저장되는 일련의 HTML, JavaScript, CSS 및 미디어 파일이 될 수도 있습니다. 두 방식 모두 각각 장단점이 있습니다. 서버에서 호스팅되는 HTML 코드를 사용하면 앱스토어에서 규정하는 승인, 배포, 보안 규정에 상관없이 앱을 업데이트할 수 있습니다. 하지만 이 방식은 디바이스를 네트워크에 연결하지 않으면 콘텐츠에 액세스할 수 없기 때문에 네트워크 끊기면 앱 사용이 불가능합니다. 반면에 애플리케이션 자체에 웹 코드를 통합하면 성능과 액세스 가용성을 높일 수 있지만 원격 업데이트가 불가능합니다. 따라서 두 가지 방식을 결합하여 최선의 결과를 달성할 수 있습니다. 이러한 시스템은 유연성을 위해 웹 서버의 HTML 리소스를 호스팅하고, 성능을 위해 모바일 디바이스의 캐시에 로컬로 저장하도록 설계됩니다.

### 세 가지 개발 방식 비교

다음은 세 가지 개발 방식을 비교 요약한 것입니다.



네이티브 방식은 성능과 디바이스의 액세스 가용성을 높이지만 높은 비용과 업데이트 문제가 단점입니다. 웹 방식은 상대적으로 간단하고 비용이 적게 들며 업데이트가 쉽지만, 기능상 제한이 따르므로 네이티브 API 호출을 사용하여 제공할 수 있는 우수한 사용자 환경을 제공하기 어렵습니다. 하이브리드 방식은 이 두 방식의 절충안으로, 여러 기업들이 선택하고 있으며, 특히 여러 운영체제를 한 번에 지원해야 할 경우 선호되는 방식입니다.

위의 표를 보면 알 수 있듯이 한 가지 방식으로 모든 것을 만족시킬 수는 없습니다. 기업의 배정된 예산, 개발기간, 내부 개발 인력 현황, 사용자 층, 필요한 애플리케이션 기능, IT 인프라 등을 종합적으로 고려해 최선을 선택해야 합니다.

한 가지 분명한 것은 사용자 경험과 애플리케이션 기능을 한쪽에, 개발 비용과 출시 기간을 다른 한쪽에 놓고 적절한 균형을 찾아야 한다는 것입니다.

특성	네이티브 앱	하이브리드 앱	웹 앱
개발 언어	네이티브 전용	네이티브 및 웹 또는 웹 전용	웹 전용
코드 이식성 및 최적화	없음	높음	높음
디바이스 고유의 기능 액세스	높음	중간	낮음
기존 지식 활용도	낮음	높음	높음
유려한 UI	높음	중간	중간
업그레이드 유연성	낮음(항상 앱스토어 이용)	중간(일반적으로 앱스토어 이용)	높음
설치 경험 수준	높음(앱스토어에서)	높음(앱스토어에서)	중간(모바일 브라우저 이용)

## 적합한 방식 선택

다음으로 기업의 요건에 맞는 개발 방식을 선택할 때 참조할 수 있는 체크리스트를 정리해 보았습니다.

### 네이티브 방식 시나리오

**기존 네이티브 기술:** 네이티브 방식과 관련된 대표적인 논쟁 중 하나는 멀티 플랫폼 지원이 어렵다는 것입니다. 네이티브 방식으로 멀티 플랫폼을 지원하려면, 각 플랫폼별 앱을 별도로 만들어야 하고, 필요한 개발 인력도 모두 다릅니다. 당연히 비용이 많이 들고 모바일 개발 역량을 내재화하는 것도 어렵습니다.

**단일 모바일 OS 지원:** 어떤 기업은 단일 모바일 OS를 사용하는 특정 대상 사용자를 위한 모바일 애플리케이션을 필요로 하기도 합니다. 예를 들어 전사적으로 직원용 모바일 기기로 BlackBerry 기기를 제공하고 있다면, 사내 모바일 앱은 블랙베리용으로만 개발하면 됩니다. 제한된 기술과 도구만 있어도 단일 네이티브 애플리케이션 개발을 진행할 수 있기 때문에, 플랫폼을 1개만 지원할 것이라면 네이티브 앱이 적합합니다.

**네이티브 기능:** 일부 애플리케이션은 단일 기능이 핵심입니다. 예를 들어 Skype의 핵심 기능은 VoIP(Voice over Internet Protocol)와 사용자 연락처에 대한 액세스입니다. 그리고 이 기능은 네이티브 방식으로만 개발할 수 있습니다. 이런 경우는 다른 선택지가 없이 네이티브 방식을 선택해야 합니다.

**유리한 UI의 필요성:** 실시간 응답을 제공하는 유려한 UI가 필요한 게임과 같은 애플리케이션은 웹 기술로는 요건을 충족하기 어렵습니다. 이러한 요구사항을 가진 애플리케이션 개발은 네이티브 방식이 선호됩니다.

### 웹 방식 시나리오

**직접 배포:** 일부 기업에서는 앱을 내부적으로 제어할 수 있고 복잡하고 시간이 오래 걸리는 승인 절차 없이 배포하기를 원합니다. 이 경우 웹 언어만을 사용하여 앱의 배포와 정기 업데이트를 수행할 수 있습니다.

**파일럿 앱:** 네이티브 앱과 웹 앱 개발의 비용 및 출시 기간을 비교해 볼 때 웹 방식을 사용하여 앱의 파일럿 버전을 만들면 매우 효율적이고 경제적입니다. 앱을 기획한 후 처음부터 새로운 애플리케이션을 만들 것인지 아니면 하이브리드 애플리케이션의 기존 코드를 활용할 것인지를 선택할 수 있습니다.

**노출성:** 이외 웹 앱은 검색 엔진 노출도가 높습니다. 앱을스토어를 통해 제공되는 앱보다 더 많은 사용자에게 노출되는 장점이 있습니다.

### 하이브리드 방식 시나리오

**상호 절충:** 하이브리드 방식을 사용하면 웹 앱과 네이티브 방식의 장단점을 상호 보완하고 절충할 수 있습니다. 네이티브를 통해 개발자는 모바일 기기가 제공하는 다양한 기능을 모두 활용할 수 있으며, 한편으로는 웹 언어를 사용하여 작성한 코드를 멀티 플랫폼에서 공유하여 개발 및 지속적인 관리를 중앙에서 보다 빠르고 경제적으로 수행할 수 있습니다.

**자체 기술력:** 웹 개발 기술은 매우 일반적이므로 내부에 활용 가능한 개발 인력을 쉽게 찾을 수 있습니다. 하이브리드 방식을 사용하면 웹 개발자는 HTML, CSS, JavaScript 등과 같은 웹 기술만 가지고도 네이티브와 유사한 사용자 경험을 제공하는 애플리케이션을 만들 수 있습니다.

**미래를 위한 고려:** HTML5는 가용성이나 기능 모두에서 급속도로 성장하고 있습니다. 많은 분석가들이 HTML5가 클라이언트 애플리케이션 개발을 위한 기본 기술이 될 가능성이 높다고 예상합니다. 대부분의 앱을 HTML로 작성하고 필요한 부분에서만 네이티브 코드를 사용하면 기존 투자를 보호할 수 있습니다.

## 요약

모바일 앱이 비즈니스 성장과 새로운 기회 창출을 위한 핵심요소로 부상함에 따라, 기업들은 더 많은 핵심 기업 서비스를 모바일화하고 있습니다. 이 과정에서 기업의 목표를 달성하기 위한 최적의 개발 방법을 모색하지만, 한 가지 개발 방식으로는 기업의 복잡한 요건을 모두 충족시키기는 어렵습니다.

이 백서에서는 한 가지 개발 방식이 아니라, 각 방식이 제공하는 장점을 활용하고 개발 방식과 상관없이 지속적으로 개발하게 될 모바일 앱의 개발을 지원하는 유연한 솔루션을 해답으로 보여줍니다.

하이브리드, 네이티브, 웹 개발 방식의 선택은 한 방식이 주가 될 수 있는 있어도 한 가지 방식만을 선택하는 것은 아닙니다. 모바일 전략을 세우는 기업들은 다음 요소들을 고려해야 합니다.

- 모바일 앱 개발, 백엔드 서비스와 연동, 앱 관리 등 전체 수명주기 관리 관점의 전체 비용 및 복잡성을 증가시키는 모바일 기기 및 기술의 파편화
- 소비자 및 기업의 모바일 채택 가속화에 따른 보안, 확장성 및 제어에 대한 요구사항 증가
- 생각지도 못한 새로운 유형과 새로운 용도의 모바일 앱을 가져올 근거리 통신, 위치 정보, 증강 현실, 소셜 네트워크 등과 같은 새로운 기능과 보조 기술
- 앱 스토어의 제출 및 승인 과정 없이 사용자가 손쉽게 이용할 수 있도록 앱을 배포하고, 업데이트 배포 및 전체 앱 포트폴리오 관리를 신속하게 수행할 수 있는 새로운 앱 배포 채널(공용 및 개인 채널)

기업은 이러한 변수를 모두 고려하여 모든 유형의 앱을 지원하는 유연성과 백엔드 기간 업무 서비스와 완벽한 연동, 높은 수준의 보안 보장, 중앙의 단일 접점으로 전체 애플리케이션 포트폴리오 모니터링 및 관리 기능을 제공하는 솔루션을 선택해야 합니다.

자세한 내용을 보고 싶거나 무료 평가판을 다운로드하려면 다음 웹 사이트를 방문하시기 바랍니다.

[www.ibm.com/software/products/ko/worklight-foundation](http://www.ibm.com/software/products/ko/worklight-foundation)

## 추가 정보

모바일 앱 개발에 대한 IBM Worklight 자료를 원하시면 IBM SWG 마케팅 총괄본부(02-3781-7500) 또는 IBM 비즈니스 파트너사에 문의해 주십시오.

추가적으로, IBM Global Financing은 가장 비용 효율적 방법과 전략적 방법으로 비즈니스에서 필요로 하는 IT 솔루션을 취득할 수 있도록 도와줍니다. IBM은 귀사의 총소유 비용을 개선하는 맞춤형 IT 재무 솔루션을 제공합니다. IBM Global Financing은 중대한 IT 투자에 자본을 투입하고 귀사의 비즈니스를 발전시키는 가장 현명한 선택입니다. 자세한 정보는 다음 웹사이트를 참조하십시오. [ibm.com/kr/financing](http://ibm.com/kr/financing)



© Copyright IBM Corporation 2013

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589

Produced in the United States of America  
2013년 4월

IBM, IBM 로고, [ibm.com](http://ibm.com), WebSphere는 전 세계의 여러 관할 구역에 등록된 International Business Machines Corp.의 상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보" ([ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml))에 있습니다.

Java 및 모든 Java 기반 상표 및 로고는 Oracle 및/또는 해당 자회사의 상표 또는 등록상표입니다.

이 문서는 처음 발행될 당시의 날짜를 기준으로 업데이트되었으며 IBM은 언제든지 문서 내용을 변경할 수 있습니다. 일부 제품은 IBM 매장이 있는 국가에서도 제공되지 않습니다.

IBM 제품 및 프로그램과 함께 사용되는 기타 제품 또는 프로그램을 평가 및 검증하는 것은 사용자의 책임입니다. 이 문서의 정보는 상품성에 대한 보증, 특정 목적의 적합성 여부 및 저작권을 침해하지 않는다는 보증 또는 조건을 포함해 명시적 또는 암묵적 보증 없이 "있는 그대로" 제공됩니다. IBM 제품은 제공된 약정에 명시된 조항 및 조건에 따라 보증됩니다.



재활용하십시오