

# JavaServer Faces, Ajax, Flash: 차세대 사용자 인터페이스

오라클 기술 백서  
2006년 10월

Ajax는 Asynchronous JavaScript, Dynamic HTML(DHTML), XML 등을 조합한 테크놀로지 환경으로, 웹 애플리케이션이 웹 페이지 전체가 아닌 일부분을 다이내믹한 방식으로 업데이트할 수 있게 합니다.

Document Object Model(DOM)은 구조형 문서의 표현을 위한 플랫폼/언어 독립적 접근법입니다.

## 서론

Web 2.0은 Gmail™, NetFlix™처럼 과거에는 현실적으로 불가능하다고 여겨졌던 새로운 형태의 애플리케이션, 데스크탑 애플리케이션을 대체할 수 있는 웹 기반 도구들을 양산하고 있습니다. Ajax와 같은 개발 테크닉은 새로운 컴포넌트를 이용한 웹 인터페이스와 시각적 효과를 제공하며 JavaScript와 DHTML의 새로운 한계를 시험할 수 있게 합니다. JavaScript와 DOM 기반 사용자 인터페이스는 프리젠테이션 계층 오브젝트를 위한 이상적인 대안을 제공하지만 리치 인터넷 애플리케이션(RIA)의 구현을 위한 필수적인 요소인 비즈니스 로직을 구현하지는 못합니다. 또, 브라우저의 종류별로 JavaScript와 DHTML 지원 기능에서 차이를 보이는 점 때문에 안정적인 개발 플랫폼이 확보되지 못했으며 브라우저별로 별도로 코드를 유지해야 하는 문제가 있었습니다.

그 해답으로, Ajax와 Ajax의 복잡성을 추상화하고 보다 표준적인 테크놀로지로 통합할 수 있는 기술을 조합하여 사용하는 방안이 제시되고 있습니다. JavaServer Faces(JSF)는 바로 이러한 기반을 제공합니다. Ajax와 JSF의 조합은 JavaScript의 복잡성 문제를 해결하고 사전 정의된 컴포넌트, 비주얼 툴 및 Java EE 플랫폼에 대한 액세스를 제공함으로써 Web 2.0 인터페이스의 구현 작업을 단순화할 수 있게 합니다.

## 진화하는 요구 사항

Web 2.0의 시대가 도래하면서 그저 "클릭하고 기다리는" 식의 고전적인 웹 애플리케이션의 인터랙션 방식은 더 이상 용인되지 않습니다. 사용자는 이제 실시간 업데이트, 데스크탑 환경과 유사한 기능을 기대하고 있습니다. 개발자들은 기존 웹 애플리케이션 모델에서 데스크탑의 기능을 흉내내기 위한 새로운 대안을 모색하고 있습니다.

개발자들은 Web 2.0 인터페이스를 구현하기 위해 다양한 테크놀로지를 활용할 수 있습니다. 그 중에서도 유력한 후보로 거론되는 것이 Ajax, Flash, SVG, XUL, XAML, Java 애플릿입니다. 이처럼 대안은 다양하지만 플랫폼과 벤더로부터 모두 독립적인 환경을 제공하는 것은 Ajax 뿐입니다. Ajax는 네이티브 클라이언트-사이드 브라우저 기술(Javascript와 DHTML)을 사용하기 때문에 별도의 브라우저 플러그인이나 벤더 종속적인 브라우저 기능을 요구하지 않습니다. 하지만 Ajax는 멀티미디어를 네이티브한 형태로 지원하지 않는다는 약점이 있습니다. 미디어 중심적인 애플리케이션 환경에서 Ajax를 보완하기 위해 SVG, Flash가 사용되는 이유가 여기에 있습니다. SVG는 뛰어난 그래픽 지원 기능을 제공하며, Flash는 네이티브 멀티미디어 지원 기능(예: 비디오 스트리밍, 벡터 그래픽 등)을 구현하고 있습니다. 두 가지 솔루션 모두 브라우저 플러그인의 다운로드를 요구하지만, 거의 모든 브라우저를 지원한다는 장점이 있습니다. 마지막으로, JavaScript와 DHTML은 EMCA와 W3C가 지원하는 표준 기반 솔루션을 제공합니다. 하지만 제품마다 언어 구현 방식의 편차가 크다는 문제가 있습니다.

Ajax는 공통적인 개발 언어를 제외하고는 별다른 표준을 제공하지 않고 있습니다. 때문에 개발자는 클라이언트, 서버 간의 데이터 마샬링(marshaling), 언마샬링(unmarshaling)과 같은 까다로운 작업을 혼자 힘으로 처리해야 합니다. 데이터 교환을 위한 타입 또는 포맷(WS-\* SOAP, REST, JSON)을 규정하거나 데이터 교환을 통해 프리젠테이션 레이어의 DOM 표현이 어떻게 업데이트되는지를 정의하는 표준은 존재하지 않습니다. 게다가 Ajax 컴포넌트를 구현하는 표준적인 접근법도 없는 상태에서 컴포넌트 프레임워크 간의 호환성을 보장할 수 있는 방법은 거의 없다고 해도 과언이 아닙니다.

푸어 JavaScript 솔루션(Dojo, Prototype, RICO, Scripttolicious)을 제공하는 요즘의 컴포넌트 프레임워크들은 다양한 Ajax 기반 컴포넌트와 시각 효과를 제공하고 JavaScript의 복잡성을 부분적으로 숨길 수 있게 합니다. 하지만 이러한 프레임워크들은 한 가지 아키텍처 상의 결정적인 문제를 안고 있습니다. 바로 로직/프리젠테이션 계층이 엄격히 구분되지 않는다는 점이 그것입니다. 사용자의 환경을 풍요롭게 하기 위해서는, 서버-사이드 데이터와 비즈니스 로직을 복제하여 클라이언트-티어에 미리 전달(pre-fetch)하고 브라우저 내에서 처리될 수 있게 함으로써 전체 페이지를 다시 로드할 필요가 없도록 만들어야 합니다. 이러한 로직의 마이그레이션 과정은 웹 애플리케이션 개발 작업이 Model-View-Control(MVC) 패턴의 Model 2 구현 방식에서 (로직과 프리젠테이션이 구분되지 않는) Model 1 구현 방식으로 역행하게 만드는 결과를 초래합니다. Java 개발의 관점에서 볼 때, 이것은 JSF와 같은 UI 프레임워크에서 (비즈니스 로직을 페이지의 Scriptlet에 구현하는) 고전적인 JSP 개발 환경으로의 극적인 퇴보라 할 수 있습니다. 또 JavaScript를 이용하여 비즈니스 로직을 실행함으로써, 개발자는 JavaScript 언어의 한계를 시험하게 됩니다. JavaScript는 브라우저 내에 제한된 언어로, 현재는 단일 쓰레드 기반의 런타임 환경만을 제공하고 있습니다. 또 브라우저 플랫폼의 호환성, 안정성을 보장할 수 있는 방법도 없습니다. 따라서 보다 안정적인 솔루션, 검증된 엔터프라이즈급 테크놀로지와 Ajax의 조합이 필요합니다.

Java EE는 서버-사이드 개발을 위한 표준 기반 테크놀로지입니다. Java EE는 지난 수 년간 eBay와 같은 엔터프라이즈 애플리케이션 환경에 적용되어 검증을 받아왔습니다. 하지만 서버-사이드에서 거둔 성공과 비교할 때 Java EE의 클라이언트-사이드 기능은 다소 초라해 보입니다. 따라서 Java EE에는 다이내믹 프리젠테이션 계층을, Ajax에는 엔터프라이즈 비즈니스 트랜잭션처리 기능을 새로이 제공함으로써 두 기술에 모두 이득이 될 수 있는 "최고의 대안"을 조합한 접근법이 가능합니다. 이러한 조합이 성공을 거두기 위해서는 미들 티어의 애플리케이션 코드와 클라이언트 티어의 비주얼 컴포넌트 간에 존재하는 간극을 해소할 수 있어야 합니다.

그 대안이 브라우저의 DOM UI 표현을 미들-티어에 복제하여 Java 코드에 의해 처리될 수 있게 하는 방법입니다. 이와 같이 함으로써 애플리케이션의 상태(state)가 서버-사이드에서 유지, 통제되도록 할 수 있습니다. 따라서 상태의 변경 정보는 비동기식 업데이트를 통해 충분히 브라우저에 배포되며, 그 결과로 브라우저의 DOM UI 표현을 변경할 수 있습니다. 이 솔루션은 리치 인터넷 테크놀로지와 Java EE의 결합을 통해 Flash, SVG와 같은 다른 테크놀로지로 쉽게 변환될 수 있게 하는 일반화된 접근법을 제공합니다. 하지만 단일 컴포넌트 모델을 제공하고 클라이언트/서버 인터랙션 관리의 복잡성을 추상화하기 위해서는 표준화된 프레임워크가 필요합니다.

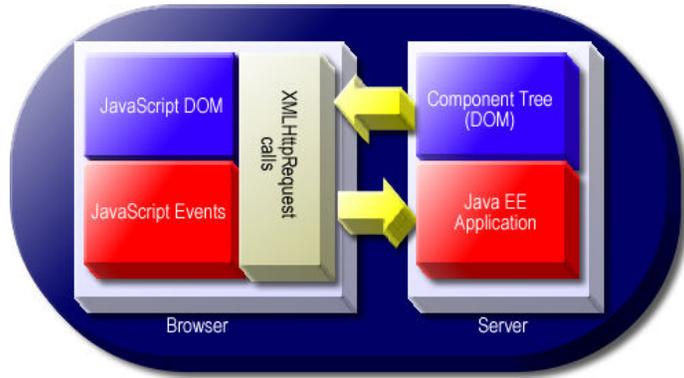


그림 1: Java EE 서버-사이드와 Ajax 사용자 인터페이스

### 표준

Java Community Process(JCP)는 Java 플랫폼의 미래 버전과 기능을 규정하는 공식화된 프로세스입니다.

Java Community Process(JCP)가 개발한 JavaServer Faces(JSF)는 웹 기반 사용자 인터페이스 구현 표준을 정의하는 서버-사이드 애플리케이션 프레임워크를 제공합니다. JSF는 컴포넌트 중심적인 개발 모델과 함께 고도로 유연한 렌더링 아키텍처를 제공하고 컴포넌트의 동작(behavior)와 표현(presentation) 간의 느슨한 결합(loose coupling)을 정의합니다. 따라서 개발자는 단순히 애플리케이션에 렌더 키트(render kit)이라 불리는 라이브러리들을 플러그인 처리함으로써 클라이언트 독립적인 UI를 구현하고 다양한 출력 디바이스를 지원할 수 있습니다. JavaServer Faces 프리젠테이션 계층의 유연성 덕분에, 개발자들은 테크놀로지에 대한 세부적인 지식을 갖지 않은 상태에서도 HTML, WML 등의 마크업 언어 뿐 아니라 telnet과 같은 보다 고전적인 형태의 프로토콜을 활용하는 것이 가능합니다. 따라서, 개발자들은 하나의 간단한 프로그래밍 모델만을 이해하고 있으면 충분하며, 이 지식을 이용하여 모바일 디바이스 또는 데스크탑 브라우저에 렌더링하는 단일 애플리케이션을 구현할 수 있습니다.

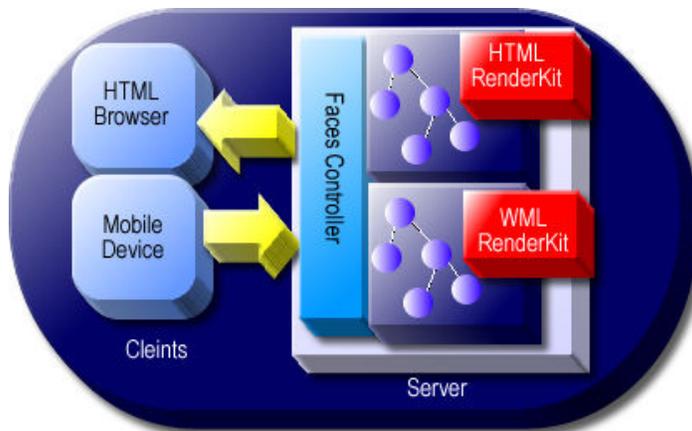


그림 2: JSF 렌더링 아키텍처

JSF 렌더링 아키텍처는 재사용 가능한 Ajax 기반 Java EE 웹 애플리케이션을 생성하는데 필요한 핵심 요소로 활용됩니다. Ajax 렌더 키트(render kit)을 구현함으로써, Ajax에 수반되는 여러 가지 복잡한 문제들을 추상화하고 전체 Java 개발자 커뮤니티가 RIA 개발 환경을 쉽게 활용할 수 있게 만들 수 있습니다. 그 뿐 아니라, Ajax 렌

더 낫은 표준 기반의 포괄적인 컴포넌트 기반 애플리케이션 모델과 Ajax를 조합하고 Java EE를 이용한 통합을 지원합니다.

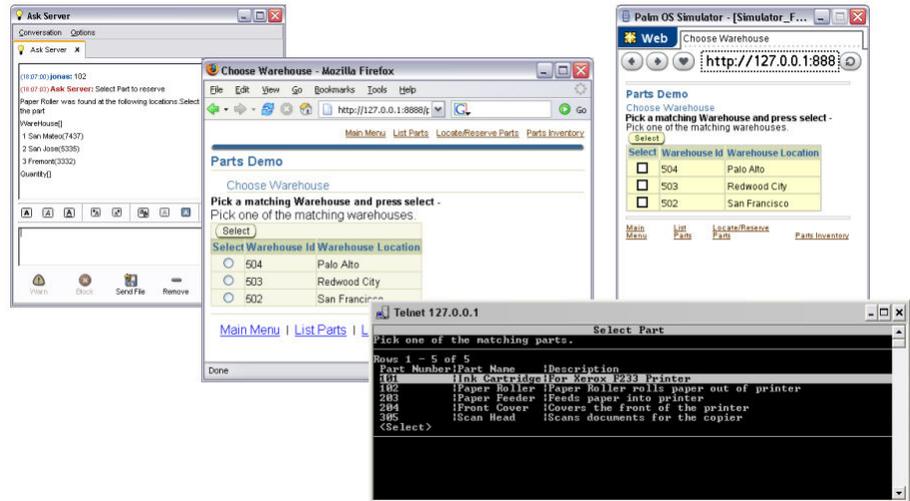


그림 3: 멀티-채널 클라이언트 전송

JSF는 RIA 개발 작업을 단순화해 줄 뿐 아니라, 다양한 개발 플랫폼을 선택할 수 있게 합니다. 커스텀 컴포넌트 셋, 렌더 키트, JSF 임플리멘테이션 등은 서로 다른 개발 툴과 런타임 환경 간에 호환이 가능합니다. 이 모든 것이 JSF의 배경을 이루는 표준들이 제공하는 혜택입니다. 현재도 리치 서버-사이드 사용자 인터페이스를 구현하기 위한 여러 가지 옵션이 제공되고 있습니다. 그 중 대부분은 특정 벤더의 솔루션으로 제공되며 한 가지 개발 툴, 한 가지 런타임 환경만을 지원합니다. JSF는 개발자들의 벤더 종속을 방지할 수 있는 개방적인 대안을 제공합니다. 그 뿐 아니라 JSF의 기반을 이루는 표준 덕분에, 오늘 구현한 애플리케이션이 미래에 출시되는 툴과도 호환됨을 보장할 수 있습니다.

OpenAjax Alliance는 개방적이고  
호환 가능한 Ajax 기반 웹  
테크놀로지의 성공적인 도입을  
목표로 하는 벤더들의 연합입니다.

오라클은 RIA 개발 환경의 발전 및 지원을 위해 노력하고 있습니다. 오라클은 핵심 웹 UI 테크놀로지로서 JSF와 Ajax를 선택하고, 이 기술들을 Oracle Fusion Middleware에 구현하였습니다. 오라클은 JSF 및 Ajax의 지원을 위한 노력의 일환으로 JSF 전문가 그룹에 참여하고 있으며, JSR252를 포함하는 다양한 표준 정의 작업에 협력하고 있습니다. 오라클은 또 Ajax의 미래를 설계하기 위한 Open Ajax Alliance의 노력에 적극적으로 동참하고 있습니다.

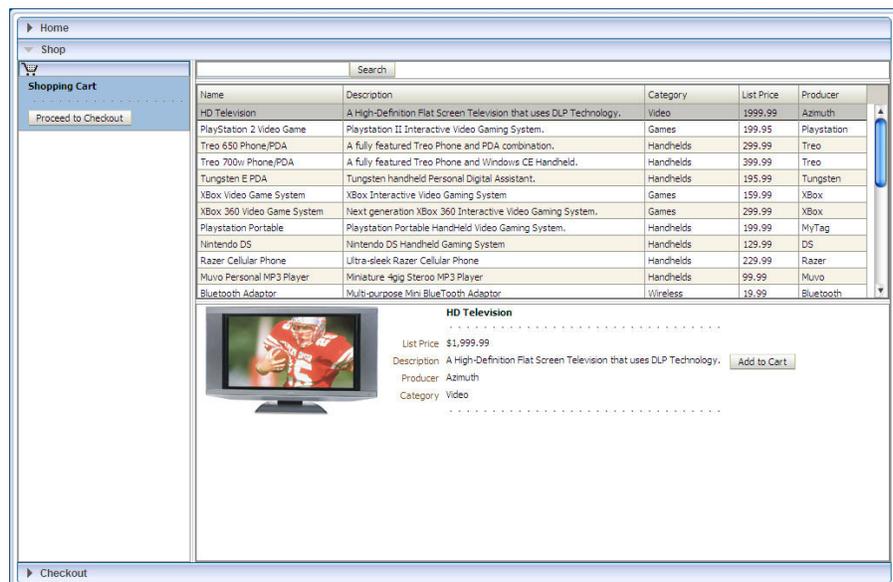
ADF Faces는 새로운 버전에서 가장 돋보이는 기능의 하나입니다. 새로운 ADF Faces는 학습 과정을 간단하게 만들어 주며, 특히 JDeveloper 환경으로 전환하고자 하는 Forms, Designer 개발자들에게 많은 도움이 됩니다. 또, JDeveloper에 새로 추가된 4GL 언어 기능의 이벤트 기반 모델과 새로운 컴포넌트들을 이용하여, 드래그-앤-드롭 작업만으로 대부분의 작업을 완료할 수 있습니다!

Eric Marcoux, Technical Architect  
Fujitsu Consulting

## 오라클의 제품 전략

Oracle Application Development Framework(ADF)는 MVC 패턴과 같은 베스트 프랙티스를 기반으로 잘 정의된 아키텍처를 제공함으로써 Java EE 개발 작업을 단순화하고 개발자들의 기술 선택의 폭을 넓혀 줍니다. ADF Faces는 ADF 프레임워크에서 제공되는 다양한 프리젠테이션 계층 테크놀로지 중 하나로, 유연한 JavaServer Faces 렌더링 아키텍처를 이용하여 Ajax, JSF의 강력한 기능을 조합하고 100 가지가 넘는 RIA 컴포넌트를 제공합니다. 컴포넌트 옵션으로는 계층형 데이터 테이블, 트리 메뉴, 인-페이지 대화상자, 어코디언, 디바이더, 정렬 가능한 테이블 등이 있습니다.

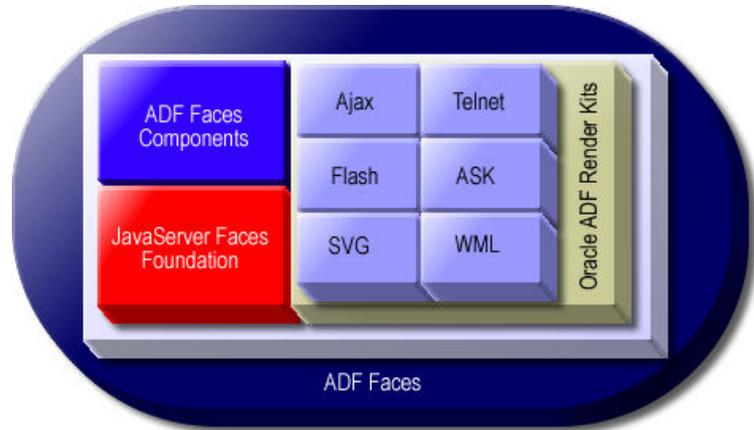
그림 4: ADF Faces 리치 클라이언트의 예



각 컴포넌트는 다국어 환경 지원을 위한 완벽한 커스터마이징, 스킨, 지원 기능을 제공합니다. Oracle ADF Faces는 Ajax 뿐 아니라 Flash, SVG 컴포넌트를 통해 다이내믹 차트, 그래프, 게이지 등의 렌더링 및 실시간 업데이트 기능을 제공합니다. 이러한 기능들은 비즈니스 액티비티 모니터링 (BAM)을 위한 핵심 컴포넌트로 활용됩니다. 개발자는 Oracle ADF Faces 컴포넌트의 Ajax, SVG, Flash 지원 기능을 이용하여 (하부 테크놀로지에 대한 지식이 없는 상태에서도) RIA를 쉽게 구현할 수 있습니다.

Oracle ADF Faces는 하부 RIA 테크놀로지를 추상화하는 Ajax 렌더링 킷을 제공하고, 개발자들에게 친숙한 JSP/JSF API를 이용하여 프로그램을 수행할 수 있게 합니다. Oracle Ajax 렌더 킷은 프리젠테이션 계층에 어떤 기술(DHTML, JavaScript, SVG, Flash)이 사용되든 관계 없이 컴포넌트 디스플레이에 필요한 마크업 및 지원 코드를 생성합니다. Ajax 렌더 킷은 오라클의 차세대 ADF Faces 컴포넌트를 통해 지원될 예정입니다.

## 그림: ADF Faces 렌더 키트



Oracle ADF Faces는 사용자 인터페이스 개발 환경 뿐 아니라, 기존 비즈니스 서비스의 애플리케이션 로직을 UI 컴포넌트에 연결하는데 필요한 코드를 제공해 줍니다. Oracle ADF Faces는 JSR-227 호환 컴포넌트를 제공함으로써, 개발자들이 UI를 설계 시점에 서비스에 쉽게 바인딩할 수 있게 합니다.

ADF Faces는 JavaServer Faces 표준을 준수하는 컴포넌트들을 구현하고 있으므로 다양한 개발 툴과 런타임 환경을 선택하는 것이 가능합니다. Oracle ADF Faces 컴포넌트는 오라클 개발 툴 및 Java EE 컨테이너가 아닌 다른 벤더 솔루션에서도 동일한 방식으로 동작, 실행됩니다.

"우리가 JDeveloper 10.1.3으로 전환하게 된 것은, EJB 3, JSF와 같은 최신 기술의 지원, ADF Faces 컴포넌트의 강력한 기능, 그리고 툴의 사용 편의성 때문이었습니다. JDeveloper는 우리가 필요로 하는 툴을 제공해주었으며, 그 덕분에 최신 기술들을 효과적으로 활용하고 애플리케이션의 지원 범위를 극대화할 수 있었습니다."

Gary Connell, Architect Grid Consulting

컴포넌트만으로는 개발 프로세스를 혁신적으로 단순화하기 어렵습니다. 하지만 개발 도구와 컴포넌트가 조합된다면 목표를 달성할 수 있습니다. RIA 컴포넌트의 다양한 솔루션과 별개로, 오라클은 코딩 전문가와 비즈니스 애플리케이션 개발자 모드를 만족시킬 수 있는 성숙된 개발 환경을 제공합니다. JDeveloper는 BPEL, EJB 3.0, Oracle ADF Business Components, 웹 서비스 등을 포함하는 기술을 지원하는 포괄적인 SOA 개발 툴을 제공합니다. 따라서 RIA를 구현하는 개발 팀은 비즈니스 서비스의 구현을 위해 별도의 툴에 의존할 필요가 없습니다. JDeveloper는 다양한 기능과 뛰어난 응답성의 인터페이스를 조합한 단일 플랫폼을 제공하고, 비즈니스 서비스를 구현하며, 이 서비스를 UI 컴포넌트에 연결해 줍니다.

오라클은 자체적인 솔루션과 별개로, 오픈 소스 커뮤니티에 적극적으로 참여하고 있습니다. 현재 오라클은 WTP(Web Tools Platform)의 Eclipse JSF 디자인-타임 프로젝트를 주도하고 있습니다. 오라클은 또 Apache Foundation의 Apache Trinidad 프로젝트에 1 세대 ADF Faces 컴포넌트를 기증하였으며, 현재 Trinidad 프로젝트를 개선하기 위한 컴포넌트의 구현 작업을 진행 중입니다.

## 결론

Web 2.0 애플리케이션은 최신 기술을 기반으로 하는 보다 정교한 수준의 웹 인터페이스를 요구하고 있습니다. 오라클은 RIA 개발 환경의 구현을 위해 적극적으로 노력하고 있으며, 앞으로도 개발자들이 개발 플랫폼과 컴포넌트를 자유롭게 선택할 수 있게 하는 기술을 계속적으로 권장, 지원할 것입니다. Oracle JDeveloper와 Oracle ADF Faces는 오라클의 이러한 노력을 반영한 결과로, 개발

자들에게 엔터프라이즈 환경을 위한 RIA를 구현하는데 필요한 생산성 개선 효과와 표준 프레임워크를 제공합니다. Web 2.0과 엔터프라이즈 인터페이스 테크놀로지가 수렴하는 오늘날의 환경에서, 오라클은 미래의 표준을 정의하는 작업을 지원하고, 오픈 소스 커뮤니티를 통한 표준의 도입을 후원함으로써 차세대 UI 개발 작업을 주도해 나갈 것입니다.



Web 2.0 User Interfaces for the Enterprise: JavaServer Faces & Ajax

October 2006

Author: Ric Smith

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2006, Oracle. All rights reserved. 본 문서는 정보 제공만을 목적으로 제공되며, 문서의 내용은 별도 공지 없이 변경될 수 있습니다. 본 문서에는 오류가 포함되어 있을 수 있으며, 상업성 또는 특정 목적의 부합성에 대한 명시적, 암시적인 일체의 보장을 제시하지 않고 있습니다. 오라클은 본 문서에 관련하여 직접적/간접적으로 발생하는 일체의 법적 책임 또는 계약상의 의무를 거부합니다. 본 문서는 오라클의 사전 서면 승인 없이 어떤 목적, 어떤 방법으로도 전자적/기계적인 형태로 복제, 전송될 수 없습니다. Oracle, JD Edwards, PeopleSoft, Siebel은 Oracle Corporation 및 계열사의 공식등록상표입니다. 다른 이름은 해당 업체의 공식등록상표일 수 있습니다.