

ProFrame

시작하기 안내서

ProFrame C v5.0 Fix#1



Copyright © 2014 TmaxSoft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2014 TmaxSoft Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 황새울로 329번길 5 티맥스빌딩 우) 463-824

Restricted Rights Legend

All TmaxSoft Software (Tmax ProFrame®) and documents are protected by copyright laws and international convention. TmaxSoft software and documents are made available under the terms of the TmaxSoft License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxSoft Co., Ltd.

이 소프트웨어(Tmax ProFrame®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

Trademarks

Tmax ProFrame® is a registered trademark of TmaxSoft Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tmax ProFrame®은 TmaxSoft Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : APACHE2.0, BSD, CDDL1.0, IBM Common Public License 1.0, INRIA, France Telecom, LGPL 2.1

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/licenses

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : APACHE2.0, BSD, CDDL1.0, IBM Common Public License 1.0, INRIA, France Telecom, LGPL 2.1

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고해 주십시오. : \${INSTALL_PATH}/licenses

안내서 정보

안내서 제목: ProFrame 시작하기 안내서

발행일: 2014-05-15

소프트웨어 버전: ProFrame C v5.0 Fix#1

안내서 버전: v2.1.2

내용 목차

안내서에 대하여	vii
제1장 비즈니스 프레임워크	1
1.1. 개요	1
1.2. 구성요소	3
1.2.1. 애플리케이션 프레임워크	3
1.2.2. 비즈니스 룰	4
1.2.3. 상품 팩토리	5
1.3. 도입 효과	6
제2장 ProFrame 소개	7
2.1. 개요	7
2.2. 아키텍처	11
2.2.1. Channel Tier	12
2.2.2. Business Tier	13
2.2.3. Data Tier	14
2.3. 구성요소	16
제3장 안내서 구성	19
3.1. 소개	19
3.2. 안내서 구성과 내용	19
용어해설	25
색인	29

그림 목차

[그림 1.1]	애플리케이션 아키텍처 변화	1
[그림 1.2]	애플리케이션 개발 생명 주기	4
[그림 2.1]	ProFrame 아키텍처	11
[그림 2.2]	ProMapper 아키텍처	12
[그림 2.3]	업무처리 플로우	13
[그림 2.4]	DB 접근 모듈과 AP 프로그램 분리	14
[그림 2.5]	DB 접근의 일관성	15
[그림 2.6]	EMB기반의 모듈 조합 및 재구성	17

안내서에 대하여

안내서의 대상

본 안내서는 Tmax ProFrame[®](C 기반의 Tmax ProFrame 이하 ProFrame)이 제공하는 기능과 아키텍처에 대한 이해가 필요하거나 ProFrame를 사용하여 시스템을 개발하는 개발자를 대상으로 한다.

본 안내서에는 ProFrame의 기능 및 시스템 아키텍처에 대한 기본적인 내용을 설명하여 사용자의 이해를 돕고자 설계되었다.

안내서의 전제 조건

본 안내서는 TmaxSoft의 비즈니스 프레임워크 솔루션인 ProFrame의 개괄적인 설명과 본 솔루션을 적용하는데 필요한 내용에 대해 기술한 입문서이다.

ProFrame의 개념을 이해하기 위해서는 UNIX 시스템에 대한 기본 지식을 필요로 한다.

안내서의 제한 조건

본 안내서는 ProFrame의 시스템 아키텍처와 구성요소 등 개념적인 내용만을 포함하고, 실제 ProFrame를 실무에 적용하거나 운용하는데 필요한 사항은 포함하고 있지 않다. 따라서 ProFrame설치, 환경설정 등 운용 및 관리에 대한 사항과 모듈 별 사용법 및 기능에 대한 자세한 사항은 각 제품 안내서를 참고하기 바란다.

참고

ProFrame에서 제공하는 안내서에 대한 내용은 “[제3장 안내서 구성](#)”을 참고한다.

안내서 구성

ProFrame시작하기 안내서는 총 3개의 장으로 구성되어 있다.

각 장의 주요 내용은 다음과 같다.

- 제1장: 비즈니스 프레임워크

비즈니스 프레임워크의 개념과 모델에 대해 기술하며, 비즈니스 프레임워크모델 중 애플리케이션 프레임워크에 대해 간략히 소개한다.

- 제2장: ProFrame 소개

애플리케이션 프레임워크를 기반으로 한 ProFrame의 특징 및 도입 효과와 아키텍처 및 구성요소에 대해 설명한다.

- 제3장: 안내서 구성

ProFrame안내서에 대한 내용을 간략히 소개한다.

- 용어해설

ProFrame에서 자주 사용되거나 중요한 용어에 대해 기술 한다.

안내서 규약

표기	의미
<<AaBbCc123>>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
<u>참고</u>	참고 또는 주의사항
<u>주의</u>	주의할 사항
[그림 1.1]	그림 이름
[표 1.1]	표 이름
AaBbCc123	Java 코드, XML 문서
[<i>command argument</i>]	옵션 파라미터
< xyz >	'<'와 '>' 사이의 내용이 실제 값으로 변경됨
	선택 사항. 예) A B: A나 B 중 하나
...	파라미터 등이 반복되어서 나옴

시스템 사용 환경

	요구 사항
Platform	IBM AIX 5.x
	HP-UX 11.xx
	Solaris 9 (SunOS 5.9)
	Linux kernel 2.4 이상
Hardware	최소 120MB 하드디스크 공간
	256MB 이상 메모리 공간
	1GB 이상 하드디스크와 512MB 이상 메모리 공간 권장
Database	Oracle 9i 또는 10g
	Tibero 3.0 sp2 이상
TP-Monitor	Tmax 4.0 이상
	Tuxedo 6.5 이상

관련 안내서

안내서	설명
ProFrame 설치 안내서	ProFrame설치를 위한 시스템 요구사항 및 설치/제거 방법에 대한 안내서이다.
ProFrame 스튜디오 안내서	GUI 기반에서 각종 리소스 모듈을 생성하기 위해 스튜디오를 사용하는 방법에 대한 안내서이다.
ProFrame 개발 안내서	ProFrame 시스템을 이용하여 업무를 개발하기 전에 개발의 개념적인 이해와 개발 절차에 대한 안내서이다.
ProFrame ProMapper 개발안내서	GUI 기반에서 ProMapper 모듈을 생성하는 방법에 대한 안내서이다.
ProFrame FileIO 개발 안내서	GUI 기반에서 FileIO 모듈을 생성하는 방법에 대한 안내서이다.
ProFrame DBIO 개발 안내서	GUI 기반에서 DBIO 모듈을 생성하는 방법에 대한 안내서이다.
ProFrame EMB 개발 안내서	GUI 기반에서 EMB 모듈을 생성하는 방법에 대한 안내서이다.
ProFrame 단위 테스트 안내서	ProFrame 시스템을 이용하여 개발이 완료된 서비스 또는 업무 모듈을 단위 테스트 하는 방법에 대한 안내서이다.
ProFrame 온라인 프로그래밍 안내서	ProFrame 시스템을 이용하여 온라인 프로그램을 작성하는 방법에 대한 안내서이다.
ProFrame 배치 프로그래밍 안내서	ProFrame 시스템을 이용하여 배치 프로그램을 작성하는 방법에 대한 안내서이다.
ProFrame 관리자 안내서	ProFrame 시스템을 관리하는 방법에 대한 안내서이다.
ProFrame 유틸리티 안내서	ProFrame이 제공하는 유틸리티에 관한 안내서이다.
ProFrame TCache 안내서	ProFrame이 제공하는 TCache에 관한 안내서이다.

연락처

Korea

TmaxSoft Co., Ltd
5, Hwangsaеul-ro 329beon-gil, Bundang-gu,
Seongnam-si, Gyeonggi-do, 463-824
South Korea
Tel: +82-31-8018-1000
Fax: +82-31-8018-1115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmax.co.kr>
기술지원: <http://technet.tmaxsoft.com>

USA

TmaxSoft, Inc.
560 Sylvan Avenue Englewood Cliffs, NJ 07632
U.S.A
Tel: +1-201-567-8266
Fax: +1-201-567-7339
Email: info@tmaxsoft.com
Web (English): <http://www.tmaxsoft.com>

Japan

TmaxSoft Japan Co., Ltd.
5F Sanko Bldg, 3-12-16 Mita, Minato-Ku, Tokyo, 108-0073
Japan
Tel: +81-3-5765-2550
Fax: +81-3-5765-2567
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

TmaxSoft China Co., Ltd.

Beijing Silver Tower, RM 1508, 2# North Rd Dong San Huan,
Chaoyang District, Beijing, China, 100027

China

Tel: +86-10-6410-6145~8

Fax: +86-10-6410-6144

Email: info.cn@tmaxsoft.com

Web (Chinese): <http://www.tmaxsoft.com.cn>

제1장 비즈니스 프레임워크

본 장에서는 비즈니스 프레임워크의 개념과 모델에 대해 기술하며, 비즈니스 프레임워크 모델 중 애플리케이션 프레임워크에 대해 간략히 소개한다.

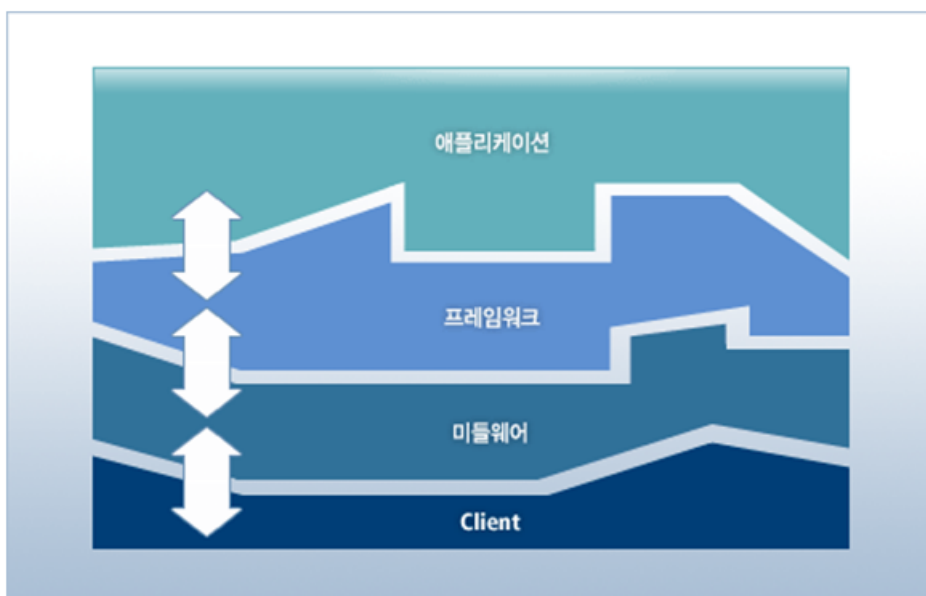
1.1. 개요

현대의 기업은 급격하게 변화하는 비즈니스 및 컴퓨팅 환경에 빠르게 순응하기 위해 적절한 변화가 필요하게 되었다. 특히 기업은 무한 경쟁 속에서 생존하기 위한 전략으로 IT 경쟁력을 키워야 한다는 인식을 하게 되었다. IT 경쟁력은 곧 비즈니스의 경쟁력과 연관된다. 따라서 기업은 프로세스 개선과 인터페이스의 표준화, 비즈니스의 모듈화가 필요하게 되었고 IT 환경에 대한 변화가 필요했다. IT 환경에 대한 변화의 대표적인 경우는 애플리케이션 아키텍처의 변화이다.

애플리케이션 아키텍처 변화

애플리케이션 아키텍처는 애플리케이션에서 시스템과 관련된 처리와 모든 부분을 직접 개발하는 구조에서 순수 업무 로직만을 구현하는 구조인 N-Tier 구조로 변화하고 있다. 즉, 애플리케이션은 순수 업무 로직 처리만 하고 그 이외의 기능은 미들웨어나 프레임워크가 담당하게 되었다.

[그림 1.1] 애플리케이션 아키텍처 변화



이와 같은 애플리케이션 아키텍처의 변화로 인해 애플리케이션 개발, 실행, 운영에 대한 최적의 아키텍처를 보장하게 되었다. 또한 다른 시스템과의 인터페이스를 표준화하고 빠른 응용 프로그램 개발 및 디버깅도 쉬워졌다.

프레임워크

특히 [그림 1.1] 애플리케이션 아키텍처 변화에서 프레임워크는 잘 정의된 애플리케이션 아키텍처와 표준화된 컴포넌트와 툴의 집합으로 이루어져 있다.

예를 들면 ProFrame가 제공하는 애플리케이션 아키텍처와 컴포넌트 및 툴은 다음과 같다.

- RunTime Engine
 - 기간제 아키텍처, 공통 컴포넌트
- 통합 서버
 - 메타 관리
 - 개발 툴 및 지원용 서버 API
- 통합 개발 GUI 툴
 - 스튜디오, ProMapper, DBIO, EMB Designer 등

이 툴은 서비스 또는 업무 개발 생산성 및 유지보수성 향상에 도움을 준다.

기업이 다양한 비즈니스 요구사항에 대응하거나 개발 생산성 향상 등을 위해 프레임워크를 도입해야 하는 필요성은 다음과 같다.

기준	필요성
비즈니스 요구사항	프레임워크를 통한 개발로 다양한 고객의 요구사항에 대해 유연하게 대응할 수 있으며 프레임워크 패키지를 보유한 Best Practice 를 제공한다.
비용	시스템 및 업무 공통 등의 기능 모듈만을 도입할 수 있고 순수 업무 모듈만을 자체적으로 개발하여 개발인력 및 비용을 최소화할 수 있다. 또한 강제화된 표준과 유연한 아키텍처 플랫폼을 기반으로 유지보수할 때 추가 비용 부담이 감소된다.
개발 생산성	개발 프레임워크를 통한 재사용성 및 업무 구현의 용이성 등이 개발 생산성을 극대화할 수 있다.
신기술 수용성	시스템 및 업무 로직에 대한 안정성을 보장하고 3-Tier 구조를 바탕으로 SOA 등 IT 기술환경 변화에 유연하게 대응할 수 있다.
위험	프레임워크 모듈의 선택적인 도입과 그 외 자체개발을 도입함으로써 소프트웨어를 구축할 때 전체적으로 위험 부담을 감소시켜 최적의 커스터마이징을 수행할 수 있다.

기업이 프레임워크를 도입하면 SI 자체의 개발 방식이나 패키지 도입 방식에 비해 자사 업무에 최적화되고 효과적인 기능 구현이 가능할 뿐만 아니라 고성능의 안정적이고 유연한 기반 플랫폼을 제공받을 수 있다.

다시 말해 검증된 프레임워크를 도입함으로써 새롭고 다양한 고객의 요구사항에도 유연하게 대처하며 안정적이고 고성능을 보장하는 아키텍처가 필요하다.

TmaxSoft 비즈니스 프레임워크

TmaxSoft는 기업의 비즈니스 경쟁력을 위해 기업의 전체 시스템을 SOA 사상을 반영하여 4개의 프레임워크 영역으로 제시한다.

4개의 프레임워크 영역은 UI 프레임워크, 인터페이스 프레임워크, 비즈니스 프레임워크, 데이터 프레임워크이다. 그 중 비즈니스 프레임워크는 기업의 업무를 위한 비즈니스 로직 처리를 담당하는 영역으로 애플리케이션 프레임워크와 비즈니스 룰, 상품 팩토리가 통합 및 연계되어 기업의 업무 시스템에 고성능, 유연성, 안정성을 보장한다. 비즈니스 프레임워크는 “서비스 기반의 통합 시스템” 구현의 핵심을 이루는 영역이다.

1.2. 구성요소

본 절에서는 비즈니스 프레임워크에 대해 간략히 소개하도록 한다.

TmaxSoft의 비즈니스 프레임워크는 다음과 같이 구성된다.

- 애플리케이션 프레임워크인 ProFrame
- 비즈니스 룰인 Tmax ProRule[®](이하 ProRule)
- 상품 팩토리 솔루션인 Tmax ProFactory[®](이하 ProFactory)

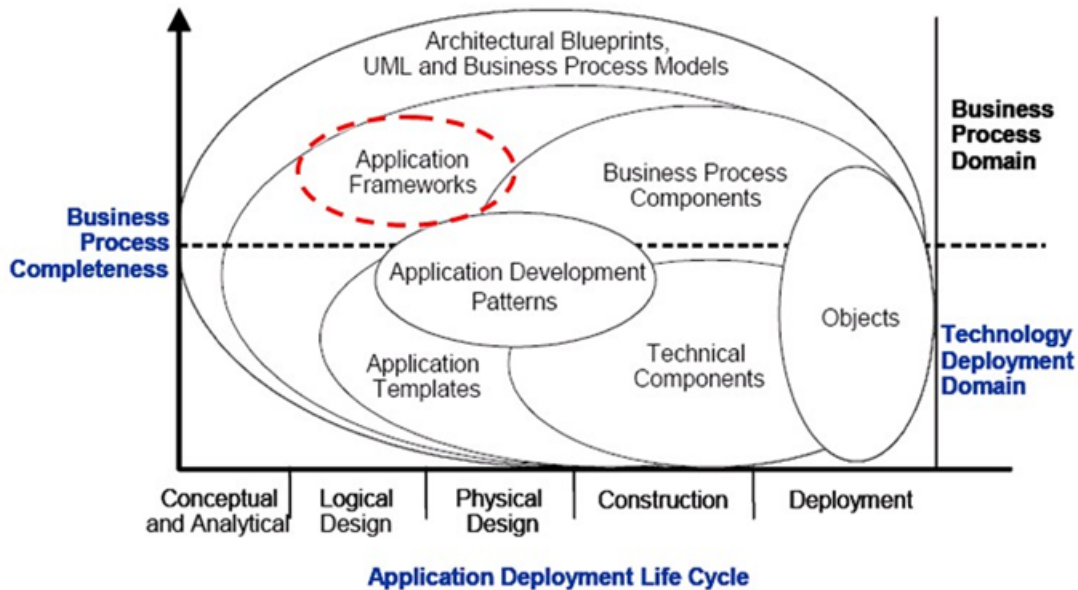
1.2.1. 애플리케이션 프레임워크

애플리케이션 프레임워크는 특정 애플리케이션 소프트웨어를 만들 때 사용하는 기반 구조로 반쯤 완성된 애플리케이션(semi-complete application)이다.

애플리케이션 프레임워크는 IT와 기업의 비즈니스 환경이 급속히 변화하면서 증가하는 채널과 서비스에 대해서 일관되고 민첩한 개발 요구와 동시에 여러 종류의 소프트웨어(WAS, WS, TP-Monitor, MCI, DBMS 등)와 연계하면서 발생하는 문제를 해결하기 위해 등장하게 되었다.

또한 개발의 생산성 향상뿐만 아니라 안정된 서비스를 보장하고 운영 관리의 효율성, 그리고 컴퓨팅 환경 등의 변화에 쉽고 빠르게 대응할 수 있는 IT 인프라의 요구를 만족하기 위해 등장하게 되었다.

[그림 1.2] 애플리케이션 개발 생명 주기



TmaxSoft 애플리케이션 프레임워크

TmaxSoft의 애플리케이션인 프레임워크인 ProFrame은 EMB(Enterprise Module Bus) 기술을 기반으로 개발되었다. ProFrame은 업무를 처리하는 서비스 또는 업무 모듈을 만들어 모듈 Pool에 등록한 후 그 모듈들을 조합하여 다양한 업무 플로우 및 신규 서비스를 재구성할 수 있다. 기존 서비스 또는 업무 모듈을 재사용하여 코딩 없이 서비스를 개발함으로써 서비스 기반의 아키텍처 구현이 가능하다.

TmaxSoft의 애플리케이션 프레임워크는 대부분의 기업들이 고민하는 통합성, 확장성, 개발 생산성의 기반이 되는 시스템 등을 제공한다. 또한 업무 영역의 애플리케이션 개발만을 고려하여 업무 시스템을 완성할 수 있다. 완성된 기업의 업무 시스템은 일관된 애플리케이션 구조로 이루어져 있어서 관리가 용이하다.

1.2.2. 비즈니스 룰

기업의 IT 환경은 외부에 의해 비즈니스 변경이 증가되고 있다. 이러한 환경에서 애플리케이션은 IT 환경의 민첩성을 키우고 유지보수의 편의성을 위해 N-Layer화 되어 가고 있으며, 그 한 부분으로 BRMS(Business Rule Management System)가 등장하게 되었다. BRMS는 비즈니스 룰을 자유롭게 표현할 수 있도록 개선한 방식이다.

비즈니스 룰은 일을 수행하거나 처리하는 지식이나 규정, 규칙, 업무처리 방식이다. 예를 들어 “어떤 경우에는 이렇게 하는 것이 좋다.”, “이것은 무엇과 무엇의 값으로 계산한다.” 등과 같이 표현된다.

BRMS의 사용으로 기업은 더 이상 비즈니스 변경이 빈번하고 신속한 업무 반영이 필요한 영역에 대해 하드 코딩을 하지 않고 룰 기반으로 시스템을 구축하게 되었다. 룰 기반의 시스템은 비즈니스와 업무의 변경에 유연한 시스템을 제공하고 신규 서비스 또는 상품 개발기간을 단축할 수 있게 되었다.

하지만 여전히 상품의 구성요소는 파라미터 방식으로 관리하고 있기 때문에 데이터베이스 I/O의 증가와 새로운 상품에 대한 수용력이 떨어지는 단점이 남아 있다. 또한, 상품의 구성요소와 룰을 모델링하는 툴이 통합되지 않아 별도로 모델링해야 하는 불편함도 있다.

개발 방식	비교 설명
일반 개발 방식	<ul style="list-style-type: none"> - 업무를 변경할 때 개발이 필요하다. - 업무 변경에 유연하지 않다. - 실행 시점에 변경이 불가하다. - 업무 처리를 위해 하드 코딩이 필요하다.
BRMS 개발 방식	<ul style="list-style-type: none"> - 업무를 변경할 때 개발이 필요하지 않다. - 업무 변경에 유연하다. - 실행 시점에 변경이 가능하다.

TmaxSoft 비즈니스 룰

TmaxSoft의 ProRule은 BRMS 기반의 개발 방식으로 설계 및 개발 단계에서 GUI 툴을 이용하여 쉽게 개발 생산성을 향상시킬 수 있다. 또한 실제 운영 단계에서는 룰 모듈 호출구조를 제공하여 시스템 성능을 보장할 수 있다.

ProRule은 신뢰성, 성능, 가시성, 편의성 등의 특징점을 기반으로 기업의 비즈니스 환경에 민첩하게 대응할 수 있는 정책 기반의 시스템을 구현한다.

1.2.3. 상품 팩토리

비즈니스와 IT의 관계가 밀접해지고 고객들의 요구가 다양해지면서 기업은 이러한 변화에 민첩하고 유연하게 대응할 수 있는 상품 개발 시스템이 필요하게 되었다. 이러한 고객의 요구를 충족시킬 수 있는 하이브리드 상품 개발과 상품 요구에 민첩한 IT의 획기적인 개선에 의한 Time to market(TTM)의 실현이 중요하게 되었다. Time to market(TTM)이란 한 제품을 만들어 시장에 내놓는데 걸리 시간 또는 기간을 의미한다. 이러한 기업의 요구와 환경에 부합하기 위해 상품 팩토리라는 새로운 상품 개발 시스템이 나타나게 되었다.

상품 팩토리는 상품 기획, 개발, 운영, 평가의 전사 비즈니스 영역에서 시스템적인 기능을 지원하고 상품 형상을 개발하고 업무 시스템에 상품 정보를 제공하는 시스템을 의미한다.

지금까지의 상품 개발시스템은 상품의 정형적 속성이나 비정형적 규칙을 각각 파라미터나 룰로 구현하는 형태로 구축되어 왔다. 그러나 상품 팩토리 솔루션은 상품의 정형 및 비정형 정보를 함께 관리하여 형상을 정확히 정의하고 시스템에 반영한다. 그 결과, 상품 팩토리 시스템은 통합 상품관리 솔루션 방식으로 융•복합상품 및 다양한 상품의 유형을 표현할 수 있으며 상품의 Time to market을 위한 상품개발이 가능하게 한다.

TmaxSoft 상품 팩토리

TmaxSoft의 상품 팩토리 즉, ProFactory는 확장성, 유연성, 고성능의 상품을 정의할 수 있고 상품관리 및 운용의 용이성을 제공한다. 다시 말해, ProFactory는 향후에 시장에 출시될 어떤 유형의 다양한 복합상품도 예외 없이 빠르고 쉽게 개발할 수 있고 시스템의 속도를 저하시키지 않으며 업무량까지 줄일 수 있다. 또한 ProFactory는 객체지향 관점으로 상품과 서비스의 구조/속성/룰을 관리하고 이를 조합하여 다양한 신상품 정의를 지원하는 선진 상품 팩토리이다.

1.3. 도입 효과

비즈니스 프레임워크는 앞서 설명한 애플리케이션 프레임워크, 비즈니스 룰, 상품 팩토리의 구성 요소를 가지며, 비즈니스 구현과 운영에 필요한 솔루션이다.

- 애플리케이션 프레임워크

애플리케이션 프레임워크는 기업의 애플리케이션 아키텍처를 구성하고 지원하며 애플리케이션의 표준화를 통해 개발 편의성과 운영 및 유지보수 편의성을 제공한다. 또한 코어 아키텍처 영역을 지원함으로써 기업에서 설계하기 힘든 영역에 대해 **Best Practice**를 지원한다.

- 비즈니스 룰

비즈니스 룰은 기업의 비즈니스에서 변경이 잦은 업무에 대해 복잡한 로직을 손쉽게 적용 및 반영이 가능하게 해주며, 기업의 관여도가 높은 업무에 대해서는 기업이 직접 관리 및 적용하게 하므로 신속한 업무 적용이 가능하다.

- 상품 팩토리

상품 팩토리는 현재 기업에서 제공하는 많은 상품과 상품에 대한 서비스를 제공하기 위해 소요되는 시간과 상품 출시에 대한 검증을 용이하게 한다 또한 상품과 서비스를 민첩하게 고객에게 제공하여 상품 개발에 필요한 과정(기획, 개발 및 운영, 평가)을 상품관리 시스템을 통해 개발하고 운영할 수 있도록 지원한다.

각각의 애플리케이션 프레임워크의 구성요소는 비즈니스 프레임워크로 연계 및 통합하여 관리되어 기업이 필요로 하는 비즈니스 업무 구현에 신속하고 민첩한 개발과 적용, 운영이 가능하도록 지원한다.

제2장 ProFrame 소개

애플리케이션 프레임워크를 기반으로 한 ProFrame의 특징 및 도입효과와 아키텍처 및 구성요소에 대해 설명한다.

2.1. 개요

ProFrame는 RTE(Real-Time Enterprise)라는 기업 목표와 Virtual Enterprise라는 시장 환경변화를 수용하기 위해 비즈니스의 업무 프로세스를 개선시키고 즉각적인 IT 구현과 통합 관리 방안을 제시함으로써 기업 전략을 빠르게 실현할 수 있는 통합 애플리케이션 프레임워크이다.

RTE는 실시간 기업을 의미하는 말로 기업의 차세대 정보기술전략이라는 화두로 급부상하고 있는 개념이다. 즉, 기업 내부 또는 외부의 프로세스를 실시간으로 처리할 수 있도록 함으로써 경쟁력을 극대화한 기업을 말한다.

ProFrame은 다수의 구축 경험과 소프트웨어 공학적 관점에서 정보시스템의 성격과 역할을 재정의하고 IT 아키텍처를 보다 유연하고 민첩하게 설계한다. 또한 개발 생산성 극대화를 위해 룰 기반의 소프트웨어 팩토리를 제공하여 분산 데이터의 일관성을 보장하며 시스템 모니터링과 관리의 효율화, 배치관리의 혁신으로 기업의 효과적인 IT 환경구현과 실시간 변경을 가능하게 한다.

설계 사상

ProFrame은 대규모 IT 환경에 필수적으로 요구되는 안정성, 고성능 유연성을 보장하는 강력한 프레임워크를 제공한다.

- 안정성

- 검증된 TP-Monitor 기반의 아키텍처

- TP-Monitor 장애가 발생했을 때 Take-Over 기능에 의한 안정성을 확보한다.
- 기능별로 계층화된 애플리케이션 아키텍처를 적용하고 트랜잭션 흐름 제어로 시스템 운영의 안정성을 강화한다.
- 다양한 기업의 IT 환경에서 검증된 TP-Monitor 기반의 아키텍처를 사용한다.

- 안정적인 업무 프레임워크 제공

- 장애 예방을 위한 프레임워크 서비스

- 메모리 및 데이터베이스 커서 등의 자원 관리 기능으로 안정적으로 애플리케이션을 개발한다.
- 업무 모듈에서 포인터 사용을 배제하여 메모리 오류를 방지한다.

- 금지함수를 사용하는 것을 검출한다.
- 고성능
 - 효율적인 데이터베이스 접근 보장
 - DBA를 위한 다양한 관리기능을 DBIO에서 제공하여 성능에 영향을 주는 SQL 사전을 검출할 수 있다.
 - DBIO는 DBMS 벤더에서 제공하는 Embedded SQL를 이용하여 소스를 생성하므로 데이터베이스를 처리할 때 빠른 성능을 보장한다.
 - 리소스 사용이 적은 간결한 구조
 - 네트워크 및 디스크 등의 I/O를 최소화하는 각종 기능을 내장한다.
 - 전체 프로그램을 Shared Object로 빌드하여 메모리 사용량을 최소화한다. 여기서 각각의 Shared Object는 Dcall 기능으로 호출되므로 서비스 수준의 관리 기능을 제공한다.
 - 표준화된 CommBuff 사용으로 반복적인 데이터베이스 접근을 제거한다.
 - 효율적인 일괄처리 구조
 - 온라인 배치에 대한 멀티 노드를 처리할 수 있다.
 - 온라인 트랜잭션의 빈도에 따라 후행처리의 유량제어를 통해서 온라인 서비스의 신속한 응답을 보장한다.
 - 효율적인 온라인 배치처리 구조를 제공함으로써 일괄처리 거래 성능을 향상시킨다.
 - Non-Blocking 인터페이스 구조
 - 대내외 시스템의 인터페이스를 Non-Blocking 구조로 구현하여 적은 자원으로 고성능을 발휘하는 아키텍처이다.
 - 비동기식 대외 연동으로 코어 서버의 Blocking를 예방한다.
- 유연성
 - EMB 기반 플로우 중심 업무 개발
 - 업무처리 플로우와 비즈니스 로직을 분리한다.
 - 플로우 기반의 설계를 통해 비즈니스 가시성을 높인다.
 - 하드 코딩을 최소화한다.
 - 모듈의 재사용을 극대화한다.
 - 유연한 운영 및 관리 기능 제공
 - 공통 업무기능을 모듈로 설계하여 업무요건 변화에 대해 유연하게 대응한다.
 - 복합 거래 제어 기능으로 관리 유연성을 증대한다.
 - EDW, 24x365 무중단 서비스, 트랜잭션 제어와 관련된 기능 등을 프레임워크에서 처리하여 애플리케이션 프로그램의 하부 인프라에 대한 독립성을 확보한다.

- 룰 기반의 유연하고 신속한 업무 로직 개발
 - 룰에 등록된 비즈니스 로직은 하부 플랫폼의 변화에 독립적으로 유지보수할 수 있다.
 - 룰에 의해서 자동으로 생성된 소스 코드는 버그가 없는 안전성을 보장한다.
 - 순수 비즈니스 로직만을 개발하므로 개발 생산성이 향상된다.
- 인프라에 독립적인 업무 로직 구조
 - 다양한 채널에 대한 입출력을 파라미터로 관리함으로써 룰 기반으로 처리하는 ProMapper를 이용하여 신규 트랜잭션 채널을 추가할 때 기존 프로그램을 수정하지 않고 입출력에 대한 재정의만 수행함으로써 채널 추가를 할 수 있다.
 - DBIO를 통해서 Data-Tier 로직을 자동으로 추가하거나 제거하여 애플리케이션 프로그램에 영향을 주지 않는다.
 - 다양한 제어 기능을 프레임워크 레벨에서 제공하여 애플리케이션 프로그램의 독립성을 강화한다.
- 멀티 프로젝트 기능을 통한 프로젝트 병렬 진행
 - ProFrame에서는 논리적인 구분을 위해 메타데이터의 프로젝트 구분자를 사용하여 동시에 여러 프로젝트의 개발을 진행할 수 있다.
 - 프로젝트별로 RunTime을 설정할 수 있으므로 실행환경의 독립성을 제공한다.
- 개발 영역과 운영 이미지를 분리 관리하는 구조 제공
 - Work-space라는 개발 영역과 Release-image라는 운영 영역의 형상을 분리하여 관리할 수 있는 구조를 제공한다.
 - 체크인/체크아웃을 통해 개발과 운영 형상의 리소스를 이동할 수 있다.
 - 개발 중인 리소스의 호출관계에 있는 운영 버전의 리소스를 호출할 수 있는 유연한 시뮬레이션을 제공한다.

특징

ProFrame의 특징은 다음과 같다.

• 고성능 및 안정성 보장

ProFrame은 고성능, 안정성, 가용성을 고려한 효율적인 자원사용 및 아키텍처 확장성을 지원하는 시스템 구축을 위한 아키텍처를 제공한다.

ProFrame은 고성능을 보장하는 다양한 기술들을 제공한다. 모듈 호출 방식에 있어서 Dllcall (Dynamic Library Call) 방식의 동적 호출 기법을 사용하여 성능의 우수성 보장하며 ProMapper를 이용한 Source-Gen 방식의 전문 구조체 변환으로 오버헤드를 최소화한다. TCache 아키텍처는 시스템을 운영할 때 빈번하게 사용되는 데이터의 빠른 접근을 지원하여 성능을 향상시키고 정보의 동기화를 가능하게 한다. 또한 Memory Manager로 과도한 메모리 접근 및 누수 현상을 원천적으로 방지하여 메모리 과다 사용으로 인한 속도가 느려지는 현상 및 Memory leak을 예방할 수 있다.

ProFrame에서 지원하는 Hot Deploy 기술은 서버의 중단 없이 변경된 해당 모듈을 실시간 적용 가능하여 24x365 무중단 서비스를 가능하게 한다. 전사적 애플리케이션의 중앙 관제 기능은 시스템의 장애 예방 및 안정적인 서비스 환경을 제공할 뿐 아니라 자원의 효율적인 관리를 지원한다.

- **Intra-SOA를 위한 EMB**

ProFrame은 개발환경에서 EMB(Enterprise Module Bus)아키텍처를 제공하여 다양한 형태의 모듈을 코딩하지 않고 디자인만으로 프로그래밍할 수 있다. 이는 단위 모듈의 재사용이 가능하게 하고 모듈 간의 연계 및 분해 조립을 통해 새로운 서비스 개발이 가능하여 SOA 사상을 그대로 반영하고 있다. 또한 개발의 표준화와 생산성으로 시스템 구축 및 추가 개발의 효율성을 대폭 향상 시킨다.

- **비즈니스 지향 아키텍처**

차세대 IT 시스템은 자원의 중심을 시스템이 아닌 비즈니스 생산성에 집중하여 비즈니스 경쟁력을 목표로 한다. ProFrame은 TP-Monitor 기능 및 트랜잭션 처리, 인터페이스 처리, 자동화된 배치처리 등 시스템적인 기능은 공통 모듈로 제공되어 개발자는 순수하게 비즈니스에 특화된 업무 로직만 집중하여 구현하도록 하였다. 또한 비즈니스 모듈을 개발하는 경우에도 기업이 이해할 수 있는 UI 환경에서 분해 조립만으로 자유롭게 구현 또는 변경이 가능하여 시스템 내 기업의 요건을 효과적으로 반영 시킬 수 있다. ProFrame은 비즈니스의 신속한 시스템 반영과 출시 그리고 변화에 민첩한 대응을 통하여 기업의 비즈니스 역량을 높여 준다.

- **차세대 통합 플랫폼 실현**

기업은 변화하는 비즈니스 환경에 유연하게 대응하고 더 나아가 비즈니스 경쟁력을 강화하기 위한 수단으로 차세대 시스템을 구축한다. ProFrame의 민첩성, 유연성, 확장성 등의 개발 사상은 이러한 차세대 핵심 요건을 배경으로 한다. ProFrame은 개발과 운영의 통합환경을 통해 다양하고 복잡한 정보시스템의 요소 기술들을 단일 애플리케이션 내에서 구축이 가능하게 하고 이질적인 요소 기술들은 상호 끊임이 없는 연계를 보장하여 차세대 통합 플랫폼을 제공한다.

도입 효과

ProFrame를 도입하여 얻을 수 있는 도입 효과는 다음과 같다.

- **내·외부 환경 및 기술 변화에 대해 탄력적 대응**

ProFrame은 차세대 논리 아키텍처 사상을 수용한 미래지향적 아키텍처로 IT 기술의 패러다임 변화에 유연하게 대응할 수 있다.

- **개발, 유지보수 및 관리의 용이성 및 비용 절감**

ProFrame의 소프트웨어 구축 환경은 효율적인 생산뿐만 아니라 한눈에 관리할 수 있게 하고 효과적인 변경, 추가 개발을 지원한다. 이는 IT 유지관리에 대한 비용을 기존에 고정비에서 변동비로 전환하고 그 비용을 획기적으로 절감할 수 있다.

- **검증된 프레임워크 기반으로 업무 애플리케이션의 안정성 확보**

다수의 기간제 시스템 구축으로 기 검증된 안정성을 보장한다.

- **IT 총소유비용(TCO) 절감**

새로운 프레임워크를 적용하면서 업무 간 중복 코딩 및 시스템 수준의 코딩을 줄여 프로그램 본수를 획기적으로 줄일 수 있으며, 표준화된 운영환경으로 관리의 효율성이 증대된다.

- 신속한 IT 접목으로 기업의 민첩성 확보

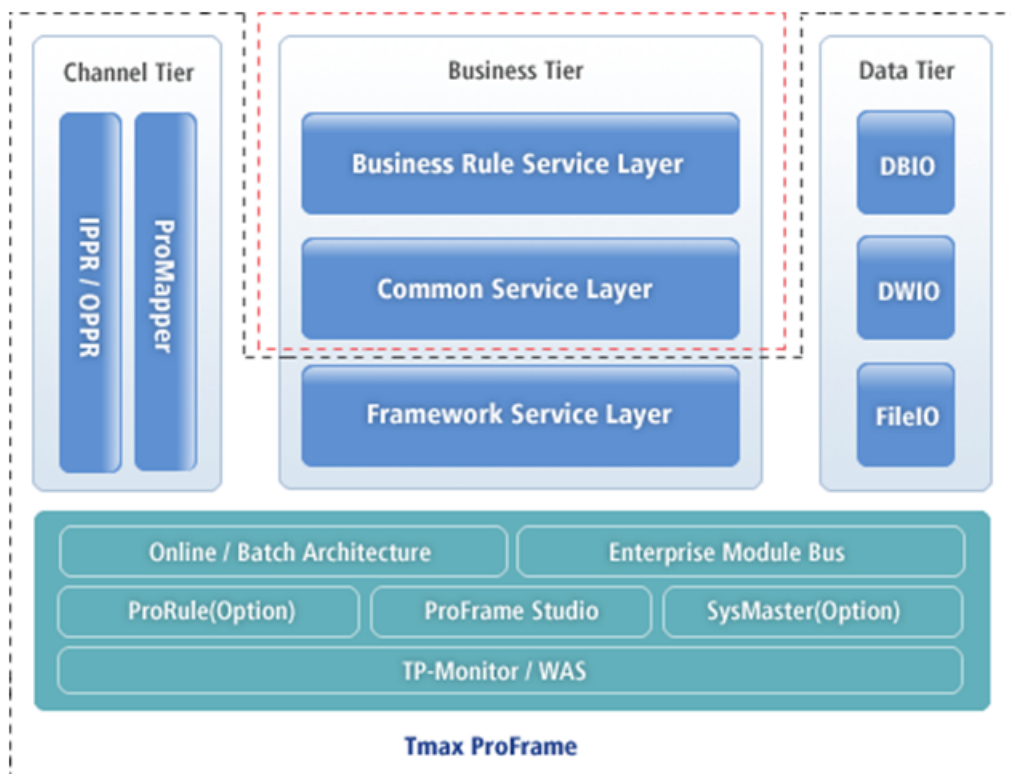
기업의 신상품 또는 서비스의 개발 속도는 기업의 경쟁력과 매출에 직결되며 이와 더불어 제품의 성공률을 높이는 전략적인 요소이다. ProFrame은 개발 생산 속도는 소프트웨어 개발이 아닌 고객과 시장에 더 많은 집중과 시간 투자를 할 수 있게 한다.

2.2. 아키텍처

ProFrame은 입출력 데이터 처리를 위한 Channel Tier, 데이터 저장 및 관리를 위한 Data Tier, 순수 비즈니스 로직 처리를 위한 Business Tier의 3-Tier 구조로 구성된다.

- Channel Tier
- Data Tier
- Business Tier

[그림 2.1] ProFrame 아키텍처



이러한 3-Tier 구조는 업무 Tier(Business Tier 중 Business Tier를 Business Rule Service Layer, Common Service Layer)를 Channel과 데이터 처리를 분리하므로 업무를 개발할 때에는 비즈니스 로직만 고려한 시스템 구축이 가능하다.

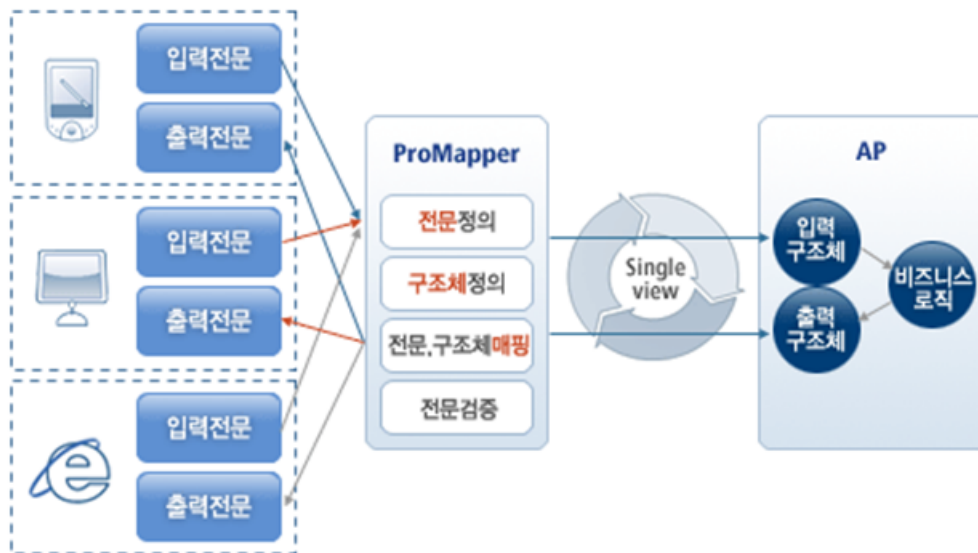
또한, Business Tier를 Business Rule Service Layer, Common Service Layer, Framework Service Layer로 분리하여 개발의 표준화 및 단순화를 지원할 수 있다.

2.2.1. Channel Tier

Channel Tier는 다양한 입력 데이터를 Business Tier에서 사용될 구조체로 변환하거나 Business Tier 내부에서 처리된 결과를 출력 데이터로 변환 및 매핑을 담당한다. GUI 기반의 Mapper 즉, ProMapper를 통하여 데이터 처리의 가시성을 제공하고 유연한 변경, 삭제를 지원한다.

Channel Tier의 ProMapper는 다양한 경로를 통하여 입력된 정보를 서비스에서 사용 가능한 형태로 변환한다. 또한 서비스의 처리 결과를 다양한 채널에서 출력이 가능한 형태로 변환하고 매핑한다.

[그림 2.2] ProMapper 아키텍처



위 그림과 같이 ProMapper는 입출력을 분리하여 구조체를 생성함으로써 프로그램 소스 상에 ProMapper 헤더를 Include하고 그 안에 정의된 입출력 구조체를 매크로로 정의하여 쉽게 서비스 또는 업무 모듈에 사용할 수 있다.

ProMapper 아키텍처에서는 다음과 같은 세부 기능을 제공한다.

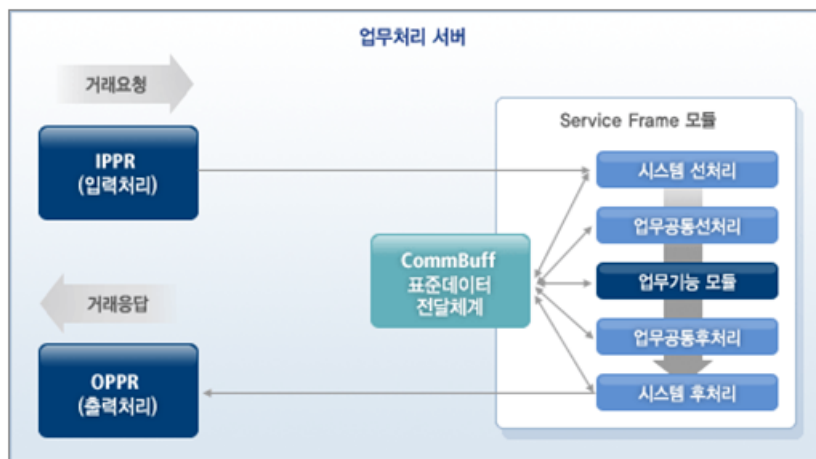
- 비즈니스 로직 단에 Single View를 제공한다.
- 업무 Tier에서 입력 값 변환, 트랜잭션 로깅 등의 기능을 제공한다.
- 입출력 데이터 포매팅 및 변환 기능을 제공한다.
- GUI 기반의 스튜디오에 내장된 ProMapper 편집기를 제공한다.

2.2.2. Business Tier

Business Tier는 시스템 및 업무계층 간의 공통기능 중심으로 부품화된 모듈구조를 제공한다. 개발자는 비즈니스에 특화된 모듈만 추가 개발하여 애플리케이션 프로그램을 완성할 수 있다. **Business Tier**는 개발의 환경 및 절차 표준 지침, 응용 프로그래밍 지침 등을 포함한다.

ProFrame은 개발자로 하여금 순수 비즈니스 로직만을 구현하도록 개발 프레임워크를 제공한다. 시스템 선후처리와 업무 공통 선후처리는 공통의 기준을 통해 설계하고 적용할 수 있다.

[그림 2.3] 업무처리 플로우



다음은 업무처리 플로우 중 **Service Frame** 모듈의 각 영역에 대한 설명이다.

- 시스템 선후처리

시스템 선후처리는 업무의 트랜잭션 흐름을 제어하는 영역으로서 다음과 같은 역할을 수행한다.

- 트랜잭션의 유효성 검사를 한다. 이 검사를 통해 트랜잭션에 대한 활성화 또는 비활성화 여부를 결정한다.
- 트랜잭션 전반에 사용되는 정보를 초기화하고 로딩한다.
- 업무처리 결과를 트랜잭션 통계 정보 수집 파일에 기록한다.

- 업무 선후처리

CommBuff 할당을 통해 업무 공통 데이터를 공유하는 영역으로서 다음과 같은 역할을 수행한다.

- 업무적으로 공유해야 하는 데이터를 업무팀별로 할당한다.
- 업무 유형별로 유효성 검사를 한다.
- 업무 그룹의 시작 및 종료할 때 공통적인 업무를 처리한다.

- 업무 기능 모듈

개발자가 온라인 또는 배치 등의 서비스를 쉽게 작성할 수 있도록 유틸리티를 제공하는 영역이다. 이 영역은 다음과 같은 역할을 수행한다.

- ProMapper, DBIO에서 자동 생성된 헤더와 라이브러리를 제공한다.

- 공통 유틸리티를 제공한다. 예를 들어 날짜 또는 문자열을 처리하는 유틸리티를 제공한다.
- OPPER 출력처리 데이터를 자동으로 큐에 누적하여 처리한다.
- 고객정보 보안항목 표시 제한처리를 위한 작업을 최소화한다.

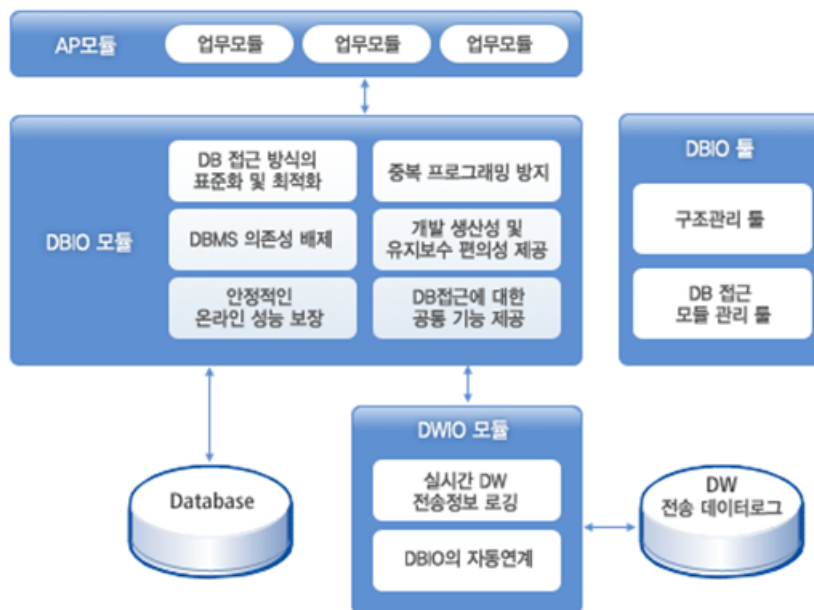
2.2.3. Data Tier

Data Tier는 데이터베이스 접근 모듈과 애플리케이션 프로그램을 분리하여 비즈니스 로직의 성능을 도모하고 데이터베이스 접근의 일관성을 보장한다. 데이터베이스 접근의 공통 기능 수행 및 권한관리, 에러처리, 프로그래밍 중복 방지, 성능 보장, 실시간 로깅 등 데이터베이스 입출력에 관련된 로직의 생산성을 보장한다.

Data Tier는 DBIO를 통하여 데이터베이스 접근 모듈과 애플리케이션 프로그램의 분리한다. 데이터베이스 접근을 위한 공통의 API를 제공함으로써 데이터베이스의 종류에 상관없이 데이터베이스 접근의 일관성을 확보하고 EDW와의 연동 기능을 제공한다.

- DB 접근 모듈과 AP 프로그램의 분리

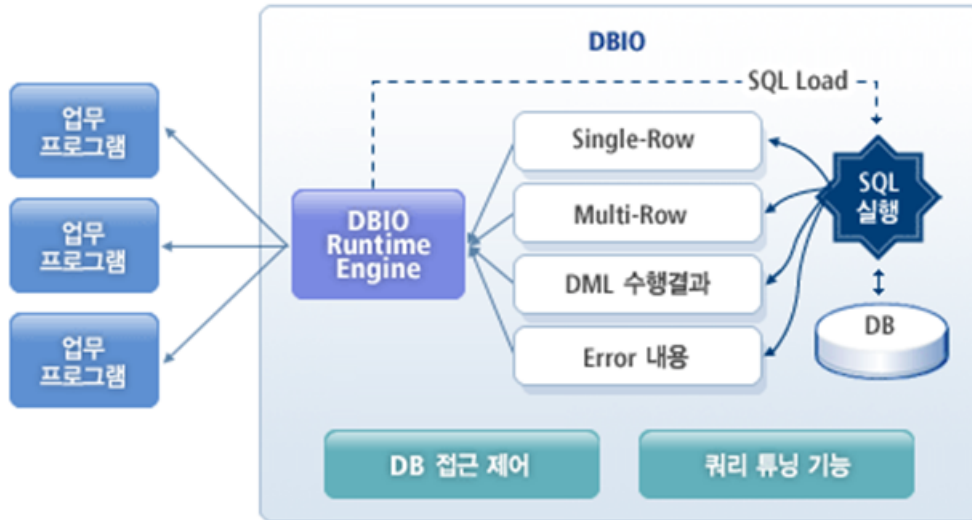
[그림 2.4] DB 접근 모듈과 AP 프로그램 분리



- 애플리케이션은 DBIO의 API를 통해 데이터베이스에 접근한다.
- 애플리케이션에서 사용되는 데이터베이스 접근은 공유 라이브러리 형태로 제공한다.
- 데이터베이스 접근 로깅, 정보계 전달 등 데이터베이스 관련 공통 기능을 제공한다.

- 데이터베이스 접근의 일관성 확보

[그림 2.5] DB 접근의 일관성



- 데이터베이스 접근의 공통 기능 수행 및 데이터베이스 접근권한 제어
- DWIO와의 연동 및 데이터베이스를 접근할 때 일관된 에러처리
- Single-Row, Multi-Row, DML 문을 처리한다.
- 실행 계획을 통한 SQL 구문을 분석한다.

ProFrame은 N-Tier/Layered 아키텍처 구조를 기반으로 애플리케이션을 기능에 따라 분리할 수 있다. 이때 각 기능별 모듈들은 유연하게 서로 결합될 수 있다.

ProFrame 애플리케이션 구조에 대한 특징은 다음과 같다.

- Multi-Tier Architecture

- 입출력 정보, 비즈니스 로직, 데이터 접근 로직을 분리한다.
- 입출력 전문을 독립적으로 관리한다.
- SQL을 업무 영역과 독립적으로 구현할 수 있다.
- Tier간에 Hot Deploy가 가능하다.

- Layered Architecture

- 시스템 공통 Layer, 업무 공통 Layer, 업무 기능 Layer를 분리한다.
- 시스템에 의존되는 로직을 공통화한다.
- 중복되는 업무 로직을 공통화한다.
- 개발자는 업무 기능 부분만을 구현한다.

- Parameter Driven/Rule Based

- 파라미터 기반의 환경 구축을 통해 중복 로직을 제거한다.
- 룰 기반의 비즈니스 로직 개발을 통해 업무 로직 관리가 용이하다.

2.3. 구성요소

ProFrame를 구성하는 요소는 다음과 같다.

- ProFrame/RunTime Engine

ProFrame가 제공하는 운영환경은 **Source-Generation** 방식의 입출력 처리, **Dlcall** 방식의 모듈 호출 방식, **Cache** 아키텍처를 제공하여 시스템 성능을 보장하고 **Memory Manager**등을 통하여 리소스의 효율적인 관리를 지원한다. 또한 대용량 온라인 배치처리 및 다중 거래를 제어, 다단계 로깅 기능은 애플리케이션의 효율적인 서비스 운영을 지원한다. 이밖에 **Before Image** 로깅 및 **Hot Deploy**를 통한 무중단 시스템을 보장하고 애플리케이션의 전사적 트랜잭션 추적 및 분석 등 안정적인 시스템 운용을 보장한다.

- ProFrame/Development Master

ProFrame의 개발 아키텍처는 개발 엔진과 스튜디오로 구성된다. ProFrame은 소프트웨어 구축을 통한 업무, 서비스, 상품 개발의 소스 코딩을 최소화하고 블록 형태의 조립을 지향합니다. 개발 및 운영환경의 **EMB(Enterprise Module Bus)** 아키텍처는 서비스/업무 모듈을 만들어 Pool에 등록한 후 그 모듈들을 조합하여 다양한 업무 플로우 및 신규 서비스를 재구성할 수 있으며 단품화된 형태로 저장하여 모듈의 재사용성을 높인다.

- ProFrame/스튜디오

스튜디오는 통합 개발환경으로 전문 매핑개발, **SQL**개발, 서비스 플로우 디자인, 업무 모듈의 작성 모든 개발 작업을 수행한다. 또한 개발자 권한 관리, 소스 버전 관리, 소스 자동생성, 이력관리, 영향도 분석 및 **Deploy** 기능을 제공한다. 스튜디오는 **EMB Designer**, **ProMapper**, **DBIO** 등으로 구성된다.

- ProFrame/ProMapper

ProMapper는 생성한 구조체를 기반으로 **C** 언어에서 사용 가능한 라이브러리를 생성하여 프레임워크의 매핑을 담당하는 매핑 기반 엔진이다. **ProMapper**는 다양한 전문 타입(**Fixed Length**, **Delimiter** 등)을 구조체와 매핑한다. 그리고 개발자는 오직 구조체만 바라보고 애플리케이션을 개발할 수 있게 한다.

- ProFrame/FileIO

FileIO는 ProFrame에서 제공하는 파일의 **Read/Write**에 대한 표준 접근방법이다. **FileIO**는 정형화된 포맷을 가지고 파일에 대한 처리를 하며, 파일 접근을 모듈화하여 **EMB** 모듈에서 파일 접근을 용이하게 한다.

- ProFrame/DBIO

DBIO는 ProFrame에서 제공하는 데이터베이스 접근에 관한 표준적인 방법을 제공한다. **DBIO**를 사용하면 데이터베이스 접근하는 프로그램을 업무 로직에서 분리하여 개발할 때 손쉽게 호출하여 사용할 수 있다. 이는 개발 생산성을 높여주고 표준화된 방법을 제공하여 오류를 최소화할 수 있게 한다.

- ProFrame/EMB

개발 및 운영환경의 EMB 아키텍처는 서비스 또는 업무 모듈을 만들어 Pool에 등록한 후 그 모듈들을 조합하여 다양한 업무 플로우 및 신규 서비스를 재구성할 수 있으며 단품화된 형태로 저장을 통한 재사용성을 높인다. 자세한 내용은 "EMB"를 참고한다.

- ProFrame/배치

온라인 서비스와 같이 일회성 거래 또는 즉시 응답성 거래가 아니라 다중 트랜잭션 또는 지연되는 응답성 트랜잭션이 있을 경우 일반적으로 배치 프로그램을 작성하여 업무를 수행한다. ProFrame의 배치 프레임워크는 업무에서 사용되는 배치 프로그램을 표준화하고 일관되게 작성하기 위해 관련 툴 및 RunTime 바이너리를 제공하여 개발자의 배치 프로그램 작성을 하는데 있어 개발 생산성 향상을 할 수 있도록 지원한다.

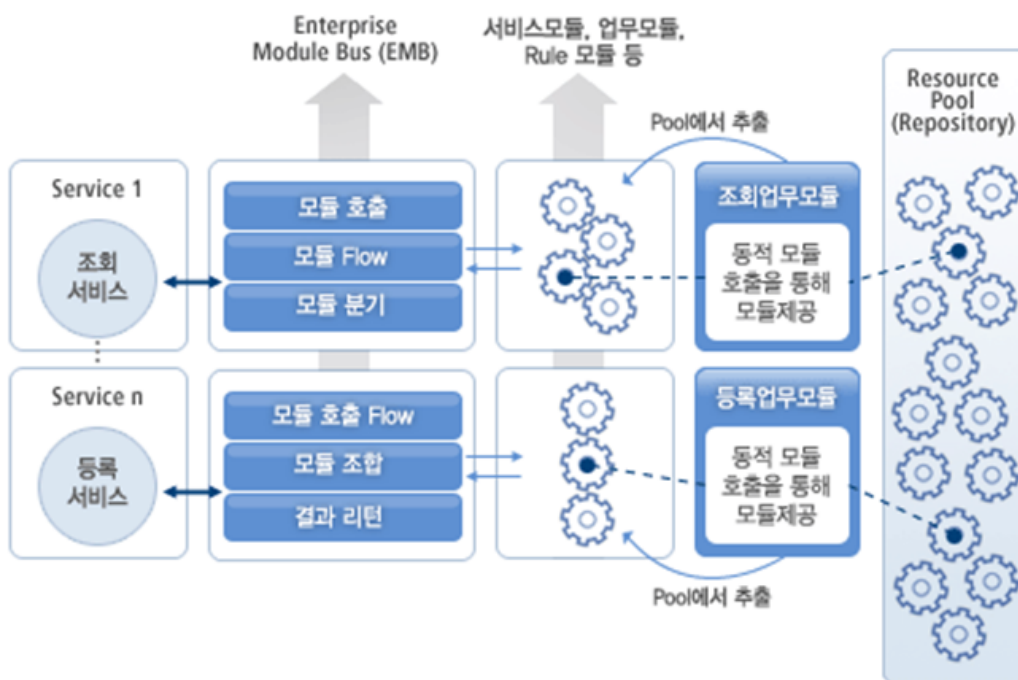
EMB

개발 및 운영환경의 EMB 아키텍처는 서비스 또는 업무 모듈을 만들어 Pool에 등록한 후 그 모듈들을 조합하여 다양한 업무 플로우 및 신규 서비스를 재구성할 수 있으며 단품화된 형태로 저장을 통한 재사용성을 높인다.

모듈의 기능을 최소단위로 분할하여(decomposition) 제공될 때 개발자는 레고 블록을 조립하듯이 프로그램을 구성할 수 있다. 이로 인해 업무흐름 파악을 쉽게 할 수 있으며 개발 및 유지보수의 비용절감이 실현된다.

EMB 기술을 사용하면 Loosely Coupled된 시스템을 구현함으로써 각 컴포넌트 사이의 독립성을 보장하여 유연한 시스템 구조를 제공한다. 업무 모듈 사이의 의존성을 최소화함으로써 업무 모듈의 재사용성을 높이고 유지보수를 용이하게 한다.

[그림 2.6] EMB기반의 모듈 조합 및 재구성



EMB는 소스 코딩의 최소화, 설계 단계의 표준 강제화, 업무 모듈의 재사용성 극대화라는 설계사상을 바탕으로 한다. 이를 통하여 EMB는 서비스 처리흐름을 가시화하고 업무 모듈 간의 의존성을 최소화한다.

- 소스 코딩의 최소화
 - Not Coding, But Design
 - 서비스는 기능 중심의 업무 모듈을 조합하는 플로우 기반 로직을 구현한다.
 - 입출력에 대한 전문 설계는 전문 설계 툴에 의해 정의한다.
 - 데이터베이스 접근 로직은 전문 GUI 기반 툴로 정의한다.
- 설계단계의 표준 강제화
 - EMB Designer에 의한 설계 표준의 강제화
 - 설계단계에서 정의된 스펙이 개발단계에 그대로 반영되는 기반 구조
- 업무 로직의 재사용성 극대화
 - 업무 모듈 Pool 기반의 모듈 관리
 - 자유롭게 모듈을 EMB Designer에서 가져다 쓸 수 있는 구조
 - 서비스 모듈, 업무 모듈, 공통 플로우 모듈의 재사용
 - 다양한 형태의 업무 모듈 재사용
- 분해 및 조립을 통합 업무 서비스 구현
 - 현업 수준의 가시성 제공
 - 서비스 플로우 상에서 자유롭게 업무 모듈을 추가/변경/삭제 가능
 - 유연한 Composite 서비스의 개발

EMB 기반 시스템에서는 여러 단위 서비스를 결합하는 연동 서비스 개발의 경우 서비스 모듈 디자이너에서 연동에 참여하는 서비스 모듈을 드래그 앤드 드롭하는 방식으로 조립하여 구현이 가능하다. 또한 개발자가 기존 소스를 열어 확인해야 했던 서비스 수행 플로우를 GUI 툴을 이용하여 쉽게 확인할 수 있어, 서비스와 관련된 전반적인 사항을 지식화, 자산화하여 사내의 구성원 및 시스템에 제공할 수 있다.

EMB는 서비스 정보를 자산화함으로써 다음과 같은 이점을 얻을 수 있다.

- 서비스 플로우의 가시화
- 직관적인 업무 이해
- 처리 단계별 프로세스 이해
- 업무 로직 확인
- 업무 관련 지식 공유 촉진

제3장 안내서 구성

3.1. 소개

제품에 대한 안내서가 있어도 제품에 대해 여러 가지 안내서가 존재하다 보니 제품을 처음 접하거나 제품에 익숙하지 않은 사용자는 원하는 안내서를 찾는데 문제를 겪을 수 있다.

예를 들어, ProFrame에서 DBIO 모듈을 생성하기 위한 방법에 대한 정보를 얻고자 할 때 정확히 원하는 내용이 ProFrame DBIO 개발 안내서에 기술되어 있는지 아니면 함께 참고해야 할 안내서가 있는지를 파악하고 있어야 한다.

본 장에서는 안내서 사용에 있어 사용자의 편의를 돕기 위해 ProFrame 안내서를 어떻게 활용할 수 있는지 기술한다. 즉, ProFrame 안내서에는 어떤 것들이 있고, 각 안내서에서 어떤 내용을 기술하고 있으며, 각 안내서가 다른 안내서와 어떻게 연관을 맺고 있는지에 대해 기술한다.

ProFrame 안내서를 처음 접하는 사용자는 본 장을 주의 깊게 읽어볼 것을 권장한다. 비록 본 장이 실제 제품 사용법과 직접적인 관련이 있는 것은 아니지만, 안내서 구성을 전체적으로 이해하는데 도움이 되도록 작성되었다.

3.2. 안내서 구성과 내용

다음은 ProFrame의 전체 안내서 목록이다.

ProFrame	ProFrame 안내서
공통	시작하기 안내서
	설치 안내서
	스튜디오 안내서
	개발 안내서
모듈	ProMapper 개발 안내서
	FileIO 개발 안내서
	DBIO 개발 안내서
	EMB 개발 안내서
기타	단위 테스트 안내서
	유틸리티 안내서
	관리자 안내서
	온라인 프로그래밍 안내서
	배치 프로그래밍 안내서

ProFrame	ProFrame 안내서
	TCache 안내서

ProFrame은 총 13권의 안내서를 제공한다. 각 안내서에 대한 내용은 본 장에 설명되어 있으니 특정한 내용에 대해 빨리 찾기를 원한다면 아래의 내용을 먼저 확인하기 바란다.

- ProFrame 시작하기 안내서

본 안내서이다.

- ProFrame 설치 안내서

ProFrame 설치는 ProFrame 서버 및 스튜디오의 설치 과정에 대해 기술한다.

설치 안내서의 구성은 다음과 같다.

- ProFrame 서버 및 스튜디오 설치를 위한 시스템 요구사항
- 추가 소프트웨어 확인
- UNIX에서의 서버 설치 및 설치 확인, 제거
- Windows에서의 스튜디오 설치 및 설치 확인, 제거
- 설치 문제 해결

- ProFrame 스튜디오 안내서

스튜디오 안내서는 ProFrame 스튜디오 툴을 사용하여 시스템을 개발하는 개발자를 대상으로 하며 사용하기 전 숙지해야 할 사항을 소개하기 위해 작성 되었다.

스튜디오 안내서의 구성은 다음과 같다.

- 스튜디오 소개 및 화면 구성
- 스튜디오 실행 방법
- 스튜디오 환경설정
- 단축키

- ProFrame 개발 안내서

ProFrame를 이용하여 온라인 서비스 프로그램 작업에 투입되는 개발자들에게 통일된 프로그램 작성법에 대해 기술한다.

개발 안내서의 구성은 다음과 같다.

- 개발 개요, 개발 기술, 개발 구성요소
- 개발 관련 도구(스튜디오 및 테스트 프레임워크)
- 개발 절차
- 개발과 관련된 안내서 소개

- ProFrame ProMapper 개발 안내서

스튜디오에 내장된 **ProMapper** 편집기를 이용하여 애플리케이션 프로그램의 입출력 설계 및 변환 규칙을 생성하는 방법에 대해 기술한다.

ProMapper 개발 안내서의 구성은 다음과 같다.

- **ProMapper** 소개 및 환경설정
- 리소스 타입별 기본 정보 등록
- 리소스 타입별 **ProMapper** 편집기 사용법
 - 구조체 편집기
 - 전문 편집기
 - 맵 편집기
- 소스생성, 컴파일 및 **Dlupdate**
- 생성된 **ProMapper** 리소스 사용법
- 가변배열 구조체 사용법 및 사용예제
- **ProMapper** 관련 API
- **ProMapper** 관련 에러코드

- **ProFrame FileIO** 개발 안내서

스튜디오에 내장된 **FileIO** 편집기를 이용하여 **FileIO** 모듈을 생성하는 방법에 대해 기술한다.

FileIO 개발 안내서의 구성은 다음과 같다.

- **FileIO** 소개 및 환경설정
- **FileIO** 기본 정보 등록
- **FileIO** 편집기 사용법
- 컴파일 및 **Dlupdate**
- 생성된 **FileIO** 모듈 사용법 및 사용예제
- **FileIO** 관련 API

- **ProFrame DBIO** 개발 안내서

스튜디오에 내장된 **DBIO** 편집기를 이용하여 **DBIO** 모듈을 생성하는 방법에 대해 기술한다.

DBIO 개발 안내서의 구성은 다음과 같다.

- **DBIO** 소개 및 환경설정
- **DBIO** 기본 정보 등록
- **DBIO** 편집기 사용법
- 소스 생성, 컴파일 및 **Dlupdate**
- **DBIO** 부가 기능

- DWIO 기능 연계

- ProFrame EMB 개발 안내서

스튜디오에 내장된 **EMB Designer**를 이용하여 **EMB** 모듈을 생성하는 방법에 대해 기술한다.

EMB 개발 안내서의 구성은 다음과 같다.

- EMB 소개

- EMB Designer 소개

- 환경설정

- 화면 구성

- EMB 모듈 생성 절차

- EMB 모듈 기본 정보 등록

- 플로우 작성

- 플로우 편집

- 컴파일 및 Dlupdate, 단위 테스트

- EMB 기타 기능 및 소스 바로가기 버튼 소개

- ProFrame 단위 테스트 안내서

스튜디오에서 개발된 서비스 또는 업무 모듈을 단위 테스트하는 방법에 대해 기술한다.

단위 테스트 안내서의 구성은 다음과 같다.

- 테스트 프레임워크 소개

- 단위 테스트 기능 및 화면 구성

- 단위 테스트 기능별 테스트 실행방법

- 스튜디오에서 테스트 로그 보는 방법

- 로그 보기 환경설정 방법

- 단위 테스트 중 문제 해결

- ProFrame 유틸리티 안내서

ProFrame가 제공하는 유틸리티에 대하여 기술한다.

유틸리티 안내서의 구성은 다음과 같다.

- ProFrame API 소개

- Date 유틸리티 소개 및 관련 API 설명

- Number 유틸리티 소개 및 관련 API 설명

- String 유틸리티 소개 및 관련 API 설명

- Long 유틸리티 소개 및 관련 API 설명
- 에러처리 및 디버그 유틸리티 소개 및 관련 API 설명
- 기타 유틸리티 소개 및 관련 API 설명
- PfmNumber 활용 방법
- ProFrame 관리자 안내서

ProFrame 시스템을 관리해야 하는 구성 중 Runtime Engine, 통합 서버, WebAdmin에 대해 상세히 기술한다.

관리자 안내서의 구성은 다음과 같다.

 - 런타임 환경 관리
 - 런타임 운영
 - 통합 서버 환경 관리
 - 개발환경 운영
 - WebAdmin 관리
- ProFrame 온라인 프로그래밍 안내서

ProFrame를 이용하여 온라인 프로그램을 작성하는 방법에 대해 기술한다.

온라인 프로그래밍 안내서의 구성은 다음과 같다.

 - 온라인 서비스 프로그램 소개
 - 온라인 서비스 프로그램 작성 절차
 - 온라인 서비스 프로그램 작성 방법
 - 서비스 연동 방법
 - 서비스 연동할 때 주의 사항 소개
- ProFrame 배치 프로그래밍 안내서

ProFrame를 이용하여 배치 프로그램을 작성하는 방법에 대해 기술한다.

배치 프로그래밍 안내서의 구성은 다음과 같다.

 - 배치 프레임워크 소개
 - 배치 프레임워크의 종류 소개
 - 일반 배치
 - POD 배치
 - 상주 배치
 - 배치 Job Information
 - 배치 생성 절차 소개

- 배치 프레임워크의 종류별 배치 생성방법
- 배치 파라미터 소개
- ProFrame TCache 안내서

ProFrame를 이용하여 TCache 환경설정 및 사용방법에 대해 기술한다.

ProFrame TCache 안내서의 구성은 다음과 같다.

 - TCache 개요
 - TCache구성
 - pfmtcacheadmin
 - pfmtcache.cfg
 - pfmtcachebackup
 - pfmtcacherestore
 - TCache의 기동 및 관리
 - pfmtcacheadmin 사용법
 - TCacheAPI
 - pfmTCacheGet
 - pfmTCacheGetAll
 - pfmTCachePut
 - pfmTCacheInvalidate
 - pfmTCacheInvalidateAll
 - pfmTCacheItem
 - TCache API 사용예제
 - pfmTCacheGet, pfmTCachePut 사용 예제
 - pfmTCacheInvalidate 사용 예제
 - TCache 동기화
 - Multi Domain Invalidation
 - Multi Node Invalidation(COUSIN)

용어해설

프레임워크

애플리케이션 아키텍처와 표준화된 컴포넌트와 툴들의 조합이다.

비즈니스 프레임워크

기업의 업무를 위한 비즈니스 로직 처리를 담당하는 영역이다.

애플리케이션 프레임워크

특정 애플리케이션 소프트웨어를 만들 때 사용하는 기반 구조로 반쯤 완성된 애플리케이션(semi-complete application)이다.

비즈니스 룰

일을 수행하거나 처리하는 지식이나 규정, 규칙, 업무처리 방식이다.

상품 팩토리

상품 기획, 개발, 운영, 평가의 전사 비즈니스 영역에서 시스템적인 기능을 지원하고 상품형상을 개발하고 업무시스템에 상품정보를 제공하는 시스템이다.

구조체

서비스 모듈 내부에서 사용되는 표준 자료구조이다.

전문

입출력 데이터에 대한 형태, 길이, 전문 형태 등의 정보를 전달하는 것으로 입력 또는 출력전문을 정의할 수 있다.

맵

전문과 구조체간의 매핑 규칙을 정의한다.

서비스 모듈

서비스 기능을 수행하는 최소 단위 모듈이다.

비즈니스 모듈

업무 처리 로직을 포함하고 있는 기능의 블록화된 모듈이다.

스튜디오

서버 프로그램을 개발하기 위한 통합 개발 도구로서 전문 개발, SQL 개발, 서비스 Flow 디자인, 업무 모듈의 작성 모든 개발 작업을 수행한다.

통합서버

ProFrame 기반의 리소스를 통합 관리하는 서버이다.

테스트 프레임워크

개발된 서비스 모듈, 비즈니스 모듈을 테스트할 수 있는 툴이다.

메타

IT용어, 업무용어, DB 테이블 명칭, 컬럼 명칭 등 다양한 형태의 용어들을 등록하여 관리하는 데이터 사전이다.

미들웨어

여러 운영 체제(UNIX, Windows 등)에서 애플리케이션 프로그램들 사이에 위치한 소프트웨어이다.

Channel-Tier

ProFrame 아키텍처에서 제시하고 있는 3-Tier 중의 하나로 입출력 데이터 처리를 위한 계층이다.

Business-Tier

ProFrame 아키텍처에서 제시하고 있는 3-Tier 중의 하나로 시스템 공통 기능 및 업무 계층간의 공통기능 중심으로 부품화 된 모듈구조를 제공하는 계층이다.

CommBuff

업무 모듈간의 데이터 공유를 하기 위해 ProFrame에서 제공하는 인터페이스 모듈이다.

Data-Tier

ProFrame 아키텍처에서 제시하고 있는 3-Tier 중의 하나로 데이터베이스 접근 모듈과 애플리케이션 프로그램을 분리하여 비즈니스 로직의 성능을 높이고 데이터베이스 접근의 일관성을 보장하는 계층이다.

Dynamic Call

시스템 운영 중에 동적으로 라이브러리를 교체하기 위해 ProFrame에서 제공하는 인터페이스 모듈이다.

EMB

가트너(Gartner, Inc.)에서 제시한 Microflows의 개념을 실체화해 Intra-SOA를 구현하는 TmaxSoft의 독창적인 기술이다. 그리고 서비스 처리 흐름을 가시화하고 업무 간의 의존성을 최소화하는 아키텍처이다.

ProMapper

생성한 구조체를 기반으로 C언어에서 사용 가능한 라이브러리를 생성하여 프레임워크의 매핑을 담당하는 매핑 기반 엔진이다.

DBIO

데이터베이스 접근 방식을 표준화한 방법이다.

Hot Deploy

ProFrame는 서비스 및 모듈 내부 로직이 변경될 때 서버 중단 없이 변경된 해당 모듈만 실시간 동적 교체가 가능하여 안정적인 무중단 서비스를 제공하는 기능이다.

Memory Manager

애플리케이션 프로그램이나 프레임워크에 메모리 누수 현상이 있을 경우 전체 시스템이 느려지는 현상을 겪게 된다. ProFrame은 Memory Manager라는 모듈을 통해 메모리 제어를 하기 때문에 메모리 누수를 원천적으로 방지할 수 있다.

EMB Designer

스튜디오에 내장되어 있으며 서비스 또는 업무 처리 플로우를 디자인하는 툴이다.

Deploy

개발된 소스 및 바이너리에 대해 개발 운영 시스템으로 배포하는 기능이다.

Dlupdate

컴파일 된 소스를 하나의 라이브러리로 묶어 Shared Memory에 올려주는 작업이다.

T-Cache 아키텍처

시스템 운영 시 빈번하게 사용되는 데이터를 빠른 접근을 지원하여 성능을 향상시키고 정보의 동기화를 가능하게 하는 구조이다.

TP-Monitor

각종 프로토콜에서 동작하는 세션과 시스템과 데이터베이스 사이의 최소 처리단위인 트랜잭션을 감시하여 일관성 있게 보관 및 유지하는 역할을 수행하는 트랜잭션 관리 미들웨어이다. 가트너(Gartner, Inc.)에서는 TPM으로 표시한다.

색인

B

BRMS, 4
Business Tier, 11, 13

C

Channel Tier, 11, 12
CommBuff, 13

D

Data Tier, 11, 14
DBIO, 14, 16
Dlcall, 9

E

EMB, 17

F

FileIO, 16

H

Hot Deploy, 10, 15

N

N-Tier, 1

P

ProFactory, 6
ProFrame, 4, 7
ProMapper, 12, 16

R

RTE, 7

T

TCache 아키텍처, 9
TTM, 5

V

Virtual Enterprise, 7

비즈니스

비즈니스 룰, 4
비즈니스 프레임워크, 3, 6

상품

상품 팩토리, 5

애플리케이션

애플리케이션 아키텍처, 1
애플리케이션 프레임워크, 3
일반 개발 방식과 BRMS 기반 개발방식 비교, 5

통합

통합 애플리케이션 프레임워크, 7

프레임워크

프레임워크, 2

