

ProFrame Release Notes

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 Release Notes

발행일: 2014-05-15

소프트웨어 버전: ProFrame C v5.0 Fix#1

안내서 버전: v2.1.2

내용 목차

안내서에 대하여	vii
제1장 소개	1
1.1. 릴리즈 이력	1
제2장 ProFrame C 5.0 Fix#1	3
2.1. 신규 기능	3
2.2. 수정 기능	6
2.3. 삭제 기능	8
2.4. 버그패치	9
제3장 ProFrame C 5.0	13
3.1. 신규 기능	13
3.1.1. Multi Project	13
3.1.2. Release Image / Work Space	14
3.1.3. TCache	17
3.1.4. Runtime 환경 변수 추가	19
3.1.5. Tomcat 지원 기능	19
3.1.6. DBIO Tiberio 실행계획 보기 기능	19
3.2. 수정 기능	19
3.2.1. 스튜디오	19
3.2.2. 통합 서버	19
3.3. 삭제 기능	20
3.3.1. 스튜디오	20
3.4. 버그패치	20
3.4.1. 스튜디오	20
3.4.2. 통합 서버	21

그림 목차

[그림 3.1] 모듈 타겟 변경 방법	15
----------------------------	----

안내서에 대하여

안내서의 대상

본 안내서는 Tmax ProFrame[®](이하 ProFrame) 시스템의 새로운 기능과 이전 버전에 대한 변경 사항에 대해서 설명하고, ProFrame 이전 버전을 사용한 개발자와 시스템 운영자 및 관리자를 대상으로 기술한다.

안내서의 전제 조건

본 안내서를 이해하기 위해서는 이전 버전의 ProFrame에 대해서 충분히 이해하고 있어야 한다. ProFrame을 처음 접하는 사람은 본 안내서 대신에 ProFrame 관련 안내서를 먼저 읽어보길 권장한다.

본 안내서는 ProFrame의 변경된 기능과 추가된 기능에 대해 간단히 언급하며, 자세한 내용은 해당 안내서를 참고하기 바란다.

안내서 규약

표기	의미
<AaBbCc123>	프로그램 소스 코드의 파일명, 디렉터리
<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
[그림 1.1]	그림 이름
[표 1.1]	표 이름

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

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 소개

본 안내서는 릴리즈 대상 제품의 새로운 기능과 이전 버전과 다른 변경된 사항을 정리한 안내서이다. ProFrame v5.0 C는 기존 버전에 비해 다양한 실사용 환경에 적용하기 위한 기능이 추가되었다. 그러나 본 안내서는 ProFrame의 전체 내용을 다루기 어려우므로 자세한 내용은 해당 안내서를 참고한다.

1.1. 릴리즈 이력

일자	버전
2013-07-31	ProFrameC v5.0
2014-05-15	ProFrame C v5.0 Fix#1

제2장 ProFrame C 5.0 Fix#1

본 장에서는 ProFrame C 5.0 Fix#1에서 추가된 새로운 기능에 대해 간략히 설명한다.

2.1. 신규 기능

- 프로젝트별 권한 및 Role 관리 기능

각각의 프로젝트별로 리소스 권한 및 Role을 부여 할 수 있는 기능을 제공한다.

- 어드민 내비게이터

Admin Role을 가지고 있는 관리자 계정으로 스튜디오에 접속하는 경우 전체 프로젝트의 리소스를 확인할 수 있는 어드민 내비게이터를 제공한다. 단, 읽기 전용 상태로만 리소스 열기가 가능하다.

- DBIO IN/OUT 컬럼을 변경하는 경우 Alert 표시 기능

DBIO 편집 중 쿼리 변경에 의하여 IN/OUT 구조체의 변수 이름, 순서, 타입, 컬럼 길이가 변경되었을 경우 개발자에게 변경 정보를 알려주는 기능을 제공한다.

- EMB 모듈의 현재 수정 위치 알려주는 기능

EMB 에디터 [소스] 탭에서 모듈을 수정하는 경우 아웃라인 뷰에 현재 수정되고 있는 모듈의 위치가 표시되는 기능을 제공한다. 단, 성능 저하를 최소화하기 위하여 코드 수정, 블록지정의 액션이 발생할 때만 위치가 갱신된다.

- 프로젝트별 리소스 통계자료를 조회할 수 있는 기능

스튜디오에서 프로젝트별로 전체 리소스의 개발 및 완료 개수를 확인할 수 있는 뷰를 제공한다. 단, Admin Role을 가지고 있는 관리자 계정으로 접속하는 경우에만 모든 프로젝트의 리소스 통계자료가 나타난다.

- WebAdmin RI/WS 리소스 체크인 일괄 취소 기능

프로젝트 관리용 기능으로 WebAdmin에서 현재 프로젝트의 체크아웃되어 있는 리소스를 전체의 체크아웃을 취소하는 기능을 제공한다.

- DBIO 로깅 기능 고도화

기존 ProFrame C 4.0의 C 소스젠에 있던 DBIO 쿼리 입력 구조체 출력 기능을 제공한다.

- DBIO 맵 리스트 기능

스튜디오에서 DBIO를 컴파일하는 경우 현재 컴파일할 때 SO에 묶이는 DBIO의 목록을 조회할 수 있는 기능을 제공한다. WebAdmin 옵션(DBIO - CHECK_DBIO_LINKING_LIST)으로 DBIO 맵 리스트 기능을 ON/OFF할 수 있다.

- 스튜디오 RI/WS 자동 ON/OFF 기능

통합 서버의 RI/WS ON/OFF 옵션(USE_RIWS)에 따라 스튜디오에서 RI/WS 기능이 자동으로 ON/OFF 되는 기능을 제공한다.

● 메타 리소스 마이그레이션 기능 (5.0 to 5.0 Fix#1)

메타를 일원화하는 기능을 사용하기 위해 이전 버전(ProFrame C 5.0)의 메타 리소스를 마이그레이션하는 기능을 제공한다. 만약 ProFrame C 4.0 메타 리소스를 마이그레이션하기 위해서는 먼저 ProFrame C 4.0 DB 마이그레이션 기능을 사용해 ProFrame C 5.0 Fix#1으로 마이그레이션 후 본 기능을 이용해야 한다. (Oracle만 지원가능)

– 사용법

1. pfm5mig.properties에 마이그레이션 정보를 입력한다.

```
source.driver=oracle.jdbc.driver.OracleDriver
source.url=jdbc:oracle:thin:@192.168.1.155:1521:orcl
source.user=proframe5
source.password= proframe5

destination.driver=oracle.jdbc.driver.OracleDriver
destination.url=jdbc:oracle:thin:@192.168.1.156:1521:orcl
destination.user=proframe5
destination.password=proframe5

# default:5
migsize=5
# 새롭게 사용할 project_id를 입력
project_id=MDS
```

2. com.tmax.proframe.mig.meta4to5.Pfm5MetaMigMain을 실행하여 마이그레이션을 진행한다. (ojdbc driver는 제공하지 않으며, 직접 사이트를 통해 다운받으면 된다.)

```
java -classpath pfmdevsvr.jar:$JEUS_HOME/lib/datasource/ojdbc14.jar
com.tmax.proframe.mig.meta4to5.Pfm5MetaMigMain
```

● ProFrame DB 마이그레이션 기능 (4.0 to 5.0 Fix#1)

ProFrame C 4.0 DB 데이터를 5.0 포맷으로 변경하여 마이그레이션하는 기능을 제공한다. 마이그레이션은 테이블 단위로 수행되어 commit 또는 rollback되며, 문제가 발생하는 경우 에러가 난 테이블만 체크하여 개별적으로 이관할 수 있다. (Oracle만 지원 가능하며, 리소스 중복으로 인한 테이블은 관리자가 확인 후 수동으로 처리해야 한다.)

– 사용법

1. pfm5mig.properties에 마이그레이션 정보를 입력한다.

```
source.driver=oracle.jdbc.driver.OracleDriver
source.url=jdbc:oracle:thin:@192.168.1.155:1521:orcl
source.user=proframe4
source.password= proframe4
```

```
destination.driver=oracle.jdbc.driver.OracleDriver
destination.url=jdbc:oracle:thin:@192.168.1.156:1521:orcl
destination.user=proframe5
destination.password=proframe5

# default:5
migsiz=5
# 새롭게 사용할 project_id를 입력
project_id=MDS
```

2. mds_legacy_tables.txt(project_id가 포함되지 않은 테이블 목록)와 mds_mp_tables.txt(project_id가 포함된 테이블 목록)을 작성한다.
3. com.tmax.proframe.mig.proframe4to5.Pfm5MigMain을 실행하여 마이그레이션을 진행한다. (ojdbc driver는 제공하지 않으며, 직접 사이트를 통해 다운받으면 된다.)

```
java -classpath pfmdevsvr.jar:$JEUS_HOME/lib/datasource/ojdbc14.jar
com.tmax.proframe.mig.proframe4to5.Pfm5MigMain
```

• 빌드 서버(소스 생성 전용)

기존의 로그 서버(LogServer)의 성능 개선을 위해 파일 관련된 작업을 전담하는 빌드 서버(PfmBuildSvr)를 제공한다. 빌드 서버는 서버 관련 파일과 서버 포트 설정을 해주어야 사용이 가능하다. 해당 파일과 설정 정보가 없을 경우 기존의 로그 서버가 동작한다.

참고

로그 서버는 멀티 프로젝트 기능을 위해 기본적으로 설치되어야 한다. 하지만 빌드 서버는 파일 핸들링 성능향상을 위한 확장 모듈로 선택적 설치가 가능하다.

– 로그 서버가 설치되어 있는 경로에 빌드 서버를 복사한다.

- pfmbuildserver.sh에 TCP_PORT 추가

```
#!/bin/sh

export LOG_EXEC_FILE=$PFMROOT/./logserver/logserver.jar
export TCP_EXEC_FILE=$PFMROOT/./logserver/pfmbuildsvr.jar
export TCP_LOG_FILE=$PFMROOT/./logserver/pfmbuildsvr.log
export TCP_ERR_FILE=$PFMROOT/./logserver/pfmbuildsvr.err
export CLASSPATH=${CLASSPATH}:$LOG_EXEC_FILE
export CLASSPATH=${CLASSPATH}:$TCP_EXEC_FILE
export TCP_PORT_NO=9718

echo " "
echo " -----"
echo " PFM BUILD SERVER BOOT START"
echo " -----"
echo " "
```

- PfmDevSvr.xml에 BLD_PORT 추가

```
<configField id="MDS" type="subConfigField" value="true" xmlns="" >
  <subConfigField id="DATA_SOURCE" value="pfm" type="string" xmlns="" />
  <subConfigField id="LOG_IP" type="string" value="local" xmlns="" />
  <subConfigField id="LOG_PORT" type="string" value="9948" xmlns="" />
  <subConfigField id="BLD_PORT" type="string" value="9718" xmlns="" />
</configField>
```

- 빌드 서버 시작

```
pfmbuildsvr.sh
```

- 빌드 서버 중지

```
pfmbuilddown.sh
```

• 파일 버전 정보 열람 기능

통합 서버의 각 jar 파일들의 버전 정보를 열람할 수 있는 기능을 제공한다. 파일별 메인 패키지의 VersionConfig를 실행하면 해당 파일의 이름과 버전, 빌드 시간이 출력된다.

- 사용법

```
java -cp pfmdevsvr.jar com.tmax.proframe.VersionConfig
```

- VersionConfig 패키지

```
pfmdevsvr.jar : com.tmax.proframe
logserver.jar, logviewer.jar : com.tmax.proframe.logviewer
pfmwebadmin.jar : com.tmax.proframe.devsvr
pfm_jaxb_model.jar : com.tmax.proframe.core.model
smd-c-codegen.jar : com.tmax.proframe.core.codegen
meta-schemas.jar : kr.co.tmaxsoft.proframe
pfmbuildsvr.jar : com.tmax.proframe.pfmbuildsvr
```

2.2. 수정 기능

• 프로젝트별 메타관리 일원화 기능

프로젝트별로 중복 관리되던 메타 데이터 문제를 해결하기 위해 기존의 일반 리소스들과 함께 관리되던 메타 데이터를 별도의 테이블(DEV_META)로 관리하도록 변경되었다. 단, Fix#1 패치 이전의 메타 데이터는 메타 리소스 마이그레이션 기능을 이용하여 메타 데이터를 이관해야 한다.

- RI/WS 기능을 이용하기 위해서는 WS 영역의 DB 스키마에 synonym을 생성해야 한다.
- 메타 전용 테이블(DEV_META) 생성 쿼리

```
CREATE TABLE DEV_META
(
  META_ID          VARCHAR2(32 BYTE)    NOT NULL,
```



```

    META_TYPE      VARCHAR2(16 BYTE),
    FIELD_TYPE      VARCHAR2(16 BYTE),
    PHYSICAL_NAME   VARCHAR2(128 BYTE),
    LOGICAL_NAME    VARCHAR2(128 BYTE),
    META_GROUP      VARCHAR2(128 BYTE),
    LENGTH          VARCHAR2(5 BYTE),
    DEFAULT_VALUE   VARCHAR2(128 BYTE),
    COMMENTS        VARCHAR2(512 BYTE),
    POINT           VARCHAR2(3 BYTE),
    IS_KEY          VARCHAR2(1 BYTE),
    IS_INDEX        VARCHAR2(1 BYTE),
    ALLOW_NULL      VARCHAR2(1 BYTE),
    DB_TYPE         VARCHAR2(16 BYTE),
    PRT_ORDER       VARCHAR2(5 BYTE)
)
ALTER TABLE DEV_META ADD (PRIMARY KEY (META_ID));

```

● 통합 서버 로그 기능 강화

리소스를 CRUD하는 경우 통합 서버 로그에 물리명이 표현되도록 기능이 개선되었다.

● 리소스 풀에서 검색한 리소스 삭제 기능

스튜디오 리소스 풀에서 검색한 리소스를 삭제할 수 있는 기능을 제공한다. 단, 검색 옵션 중 '**의존성 선택-독립 리소스**'가 선택된 상태에서 검색된 리소스만 가능하다.

● 내비게이터 리소스 다중 오픈 기능

내비게이터에서 리소스를 다중으로 선택해서 한꺼번에 열 수 있도록 개선되었다.

- SiteConfig.xml에 NAVIGATOR_MULTI_SELECT_MAX_COUNT 옵션을 추가해 한꺼번에 열 수 있는 리소스 개수를 설정한다.

```

<!-- Multi Open ON/OFF -->
<siteElement id="NAVIGATOR_MULTI_SELECT_MAX_COUNT" type="integer" value="5"
xmlns=" " />

```

● 형상관리 기능 성능 개선

멀티 프로젝트의 형상관리를 요청하는 경우 처리 속도가 개선되었다.

● WebAdmin Explorer 지원

WebAdmin이 Explorer 9부터 11까지에서 사용할 수 있도록 개선되었다.

● Multi Node, Multi Gateway 환경에서 TCacheInvalidate하는 경우 Handler를 이용한 콘솔에 로그가 출력 기능

멀티 노드, 멀티 게이트웨이 환경에서 TCacheInvalidate하는 경우 콘솔로 출력되던 로그를 Handler를 이용하여 ON/OFF할 수 있도록 개선되었다.

● TCacheInvalidate 처리 속도 향상

pfmTCacheInvalidate 처리 속도가 향상되었다.

- **RIWS ON 상태에서 리소스를 복원하는 경우 WS의 리소스 중복 여부 체크 기능**

RIWS ON 상태에서 리소스를 복원하는 경우 WS에 중복되는 리소스가 있는지 확인하고 있다면 복원 및 취소를 선택할 수 있는 알림창을 추가하여, 사용자가 확인할 수 있도록 개선되었다.

- **EMB 리소스의 소스 저장 및 컴파일 방식 개선**

EMB 리소스의 소스 저장 및 컴파일 방식이 큐를 사용하여 순차적으로 처리될 수 있도록 개선되었다.

- **Persist 타입 DBIO의 컬럼 추가/삭제 기능 개선**

Persist 타입 DBIO의 컬럼을 버튼 이외의 마우스 컬럼명 더블클릭으로 추가/삭제가 가능하도록 개선되었다.

- **컨텍스트, 로컬변수, 상수 정의 시 기능 개선**

컨텍스트, 로컬변수, 상수 정의 시 tab 키를 이용해 에디트 영역을 이동할 수 있도록 개선되었다.

- **DUAL 테이블 DBIO 생성 지원**

시스템 테이블인 DUAL 테이블을 DBIO로 생성 가능하도록 개선되었다. (단, persist 타입만 가능)

2.3. 삭제 기능

- **MULTI_PROJECT 옵션 삭제**

ProFrame C 5.0은 기본적으로 멀티 프로젝트를 지원하므로 PfmDevSvr.xml에서 MULTI_PROJECT 옵션이 삭제되었다.

- **메타 싱크 기능에서 ResourceID 자동 설정하는 기능 삭제**

메타 싱크 기능을 사용하는 경우 ResourceID를 설정하지 않으면 물리명으로 자동 설정되는 기능이 삭제되었다.

- **스튜디오 뷰 삭제**

스튜디오 Project Pool, Project Resource Pool, Project Export Manager, Project Import Manager, Test Case Export Manager, QA 뷰가 삭제되었다.

- **Template 리소스 삭제**

EMB 위자드 및 기본 리소스에서 Template 리소스가 삭제되었다.

- **Template 및 UcsOnlineServer 검색 조건 삭제**

리소스 풀, 리소스 히스토리 뷰 등과 같은 리소스 검색 조건에서 지원하지 않는 Template과 UcsOnline Server 리소스가 삭제되었다.

- **뷰 삭제**

스튜디오 Project Pool, Project Resource Pool, Project Export Manager, Project Import Manager, Test Case Export Manager, QA 뷰가 삭제되었다.

- **EMB XML 에디팅 기능 삭제**

데이터 정합성을 위해 EMB 리소스의 XML 에디팅 기능이 삭제되었다.

2.4. 버그패치

- **tdlupdate가 안된 함수를 tdlcall하면 함수명이 이상하게 출력되는 현상**

tdlupdate가 되지 않은 함수를 사용하면 에러 로그에 함수명이 이상하게 출력되는 현상이 수정되었다.

- **구조체 관련 소스에 개행문자가 생성되는 현상**

구조체 관련 소스 생성시 개행문자(^M)가 생성되지 않도록 수정되었다.

- **DBIO 쿼리의 주석에 특수 문자를 입력하는 경우 소스 생성이 되지 않는 현상**

DBIO 쿼리의 주석에 콜론(:), 작은따옴표(')를 입력하는 경우 소스젠에 영향을 주지 않도록 수정되었다.

- **형상관리할 때 프로젝트별 중복되는 리소스가 있는 경우 여러 건이 출력되는 현상**

형상관리를 통해 리소스를 검색하는 경우 프로젝트별 중복되는 리소스가 있는 경우 여러 개가 출력되지 않도록 수정되었다.

- **DBIO의 배열크기 변경 후 모듈 재구성하는 경우 배열 크기가 반영되지 않는 현상**

DBIO의 배열크기를 변경한 후 모듈을 재구성하면 컨텍스트 정의 및 소스에서 해당 DBIO의 구조체 배열크기가 자동으로 변경되도록 수정되었다.

- **Dynamic DBIO에서 메타 값을 가져오지 못하는 현상**

Dynamic DBIO에서 메타 값을 가져오지 못하는 현상이 수정되었다.

- **SQL_Logging Script 생성 오류**

SQL_Logging 기능을 사용하는 경우 입력 DBIO의 첫 번째 값만 출력되는 현상이 수정되었다.

- **스튜디오 Batch Kind Disable 현상**

스튜디오 Batch 특성 화면에서 Kind 값이 Disable되는 현상이 수정되었다.

- **Windows 7 호환성 패치**

Windows 7에서 스튜디오의 화면이 깨지고 멈추는 등의 현상이 수정되었다.

- **한글 사용자의 경우 단위테스트 화면 로그인 실패 현상**

한글 사용자가 로그인 후 단위테스트할 때 화면이 정상적으로 열리지 않던 현상이 수정되었다.

- **리소스를 복원할 때 History_id 중복 현상**

리소스를 복원하는 경우 History_id가 중복되지 않도록 History_id 채번 방식이 변경되었다.

- **스튜디오 저장 후 서버에 생성된 소스 불일치 현상**

간헐적으로 스튜디오에서 소스를 저장 후 스튜디오의 소스와 서버의 소스가 불일치한 현상이 발생하여 서로 싱크가 맞도록 수정되었다.

- **EMB 소스 에디터에서 커맨드를 입력하는 경우 지속적으로 공백이 들어가는 현상**

EMB 소스 에디터에서 커맨드를 입력하는 경우 일정 크기 만큼 공백이 추가되어 소스가 생성될 때 반복적으로 공백이 들어 가는 현상이 수정되었다.

- **DBIO 쿼리에 큰따옴표가 들어가면 SQL_Logging이 되지 않는 현상**

SQL Logging을 사용하는 경우 DBIO 쿼리에 큰따옴표(" ")가 들어가면 컴파일되지 않는 현상이 수정되었다.

- **DBIO 쿼리 입력 창에서 최초 <Ctrl>+Z(실행취소)를 누르는 경우 쿼리가 삭제되는 현상**

쿼리를 직접 입력할 수 있는 타입의 DBIO(View, Execsql, Dynamicsql)를 오픈한 후 최초 <Ctrl>+Z(실행취소)를 누르는 경우 쿼리가 삭제되는 현상이 수정되었다.

- **TCacheInvalidate를 할 때 Invalidate를 2번하는 현상**

TCacheInvalidate를 할 때 Invalidate를 이중으로 하지 않도록 수정되었다.

- **Tmax 4.0, Tmax 5.0 Fix#2 이상에서 pfmTCacheItem이 안되는 현상**

Tmax 4.0, Tmax 5.0 Fix#2 이상에서 pfmTCacheItem이 안되는 현상이 수정되었다.

- **멀티 노드 환경에서 tpinit을 하지 않으면 Invalidate가 되지 않는 현상**

멀티 노드 환경에서 별도의 tpinit을 하지 않더라도 Invalidate가 가능하도록 수정되었다.

- **멀티 노드 환경에서 상대 노드가 에러일 때 TCacheInvalidate 성공으로 나타나는 현상**

멀티 노드 환경에서 상대 노드 중 1개 이상이 에러일 때 에러 값이 리턴되도록 수정되었다.

- **쿼리문에 예약어가 존재하는 경우 소스가 생성 되지 않는 현상**

키워드와 동일한 컬럼이 존재하는 테이블의 경우 소스가 생성되지 않는 현상이 수정되었다.

- **RIWS를 OFF하는 경우 리소스 복원이 정상적으로 되지 않는 현상**

RIWS를 OFF하는 경우 리소스 복원을 수행하는 과정에서 데이터가 중복되어 들어가던 현상이 수정되었다.

- **내비게이터 루트 타입이 그룹일 경우 리소스 변화 상태에 따라 리프레쉬가 잘못되는 현상**

루트 타입이 그룹일 경우 리소스 컴파일, 저장 후 다른 그룹의 리소스 리스트로 잘못 바뀌는 현상이 수정되었다.

- **암호화 기능을 사용하는 경우 스튜디오 로그인에 되지 않는 현상**

스튜디오 암호화 사용 옵션 값이 잘못 설정되어 로그인이 되지 않던 현상이 수정되었다.

- **암호화 기능을 사용하는 경우 스튜디오에서 변경한 암호가 암호화 되지 않는 현상**

스튜디오 환경설정에서 사용자 암호 변경 시 변경한 암호가 평문(plaintext) 그대로 저장되는 현상이 수정되었다.

- **모듈 IN/OUT을 맵핑하는 경우 오류가 발생하는 현상**

맵핑 기능을 사용하는 경우 읽기 전용으로 수정이 되지 않거나 모듈이 disable되는 현상이 수정되었다.

- **Array count 특성 창에 변수를 사용할 때 에러가 발생하는 현상**

DBIO 및 FileIO Array count 값에 사용자가 선언한 변수를 사용할 수 있도록 수정되었다.

- **DBIO 에디터에서 메타 등록을 할 때 정보가 누락되는 현상**

DBIO 에디터에서 미등록된 메타 등록 시 type과 length가 누락되는 현상이 수정되었다.

- **리소스 삭제, 체크아웃, 체크인할 때 리소스 그룹 정보가 삭제되는 현상**

RIWS ON 시 리소스 삭제, 체크아웃, 체크인할 때 리소스 그룹 정보가 리소스가 RIWS에 모두 없는 경우에만 삭제되도록 수정되었다.

- **기타 버그**

기타 가벼운 버그들이 수정되었다.

제3장 ProFrame C 5.0

본 장에서는 ProFrame C 5.0 릴리스에서 추가된 새로운 기능에 대해 간략히 설명한다.

3.1. 신규 기능

3.1.1. Multi Project

ProFrame에서는 논리적인 구분을 위해 메타데이터의 프로젝트 구분자(PROJECT_ID)를 사용하여 동시에 여러 프로젝트를 개발할 수 있는 기능을 추가하였다.

- **프로젝트별 Runtime 설정 기능**

프로젝트별로 Runtime을 설정하여 각 프로젝트별의 독립적인 실행환경을 제공한다.

- **이기종 DB 지원 및 프로젝트별 DBMS 독립적 사용을 위한 멀티 커넥션 처리 기능**

Oracle과 Tiberio DB를 지원하며, 프로젝트별로 DBMS 커넥션을 별도로 설정할 수 있는 기능을 제공한다.

- **WebAdmin의 Multi Project 지원 기능**

WebAdmin 시작 페이지에서 프로젝트 ID를 선택할 수 있으며, 프로젝트별로 WebAdmin 페이지를 별도로 제공한다.

- **프로젝트별 리모트 소스 젠 및 컴파일 기능**

프로젝트별로 Logserver를 설정하여 원격으로 소스 생성과 컴파일을 하는 기능을 제공한다. 생성된 소스와 컴파일 파일들은 Logserver가 있는 로컬의 설정한 디렉토리에 생성된다.

```
<configField id="PRJ1" type="subConfigField" value="true" xmlns="" >
  <subConfigField id="LOG_IP" type="string" value="192.168.1.123" xmlns="" />
  <subConfigField id="LOG_PORT" type="string" value="9998" xmlns="" />
</configField>
<configField id="PRJ2" type="subConfigField" value="true" xmlns="" >
  <subConfigField id="LOG_IP" type="string" value="192.168.1.124" xmlns="" />
  <subConfigField id="LOG_PORT" type="string" value="9999" xmlns="" />
</configField>
```

- **SitePlug을 통해 커스터마이징 형상관리를 할 수 있는 기능**

사이트 커스터마이징 기능이 추가되었다.

– REQUEST_PUBLISH 메시지 : 형상 관리 Agent에서 프로젝트별 리소스의 형상관리 파일 생성 요청

- REQUEST_APPROVAL 메시지 : 형상 관리 Agent에서 가져 갈 수 있는 tar를 프로젝트별 로그서버에 생성 승인 요청

- **ServerCMDStream 멀티프로젝트 지원 기능**

기존 송신 프로그램의 별도 수정 없이 멀티프로젝트를 지원하는 서버에서 작성된 xml 리소스를 지원한다.

- **ServerCMDStream을 통한 메타 싱크 기능 추가**

ServerCMDStream을 이용해 메타를 전송할 경우 모든 프로젝트에 메타가 등록되는 기능을 제공한다.

xml의 attribute로 projectId를 설정한다면 해당 프로젝트에만 등록이 되며, 해당 모드(C(최초 생성), U(업데이트), D(삭제))를 지원한다.

- **DBIO 사용 시 데이터소스 존재 여부에 따라 런타임으로 자동 Switching 되는 기능**

DBIO를 사용하는 경우 데이터소스가 존재 하지 않을 경우 런타임 데이터소스로 자동으로 스위칭 되는 기능을 제공한다.

- **프로젝트별 Logging On/Off 기능**

통합 서버의 PfmDevSvr.xml의 USE_PROJECT_LOG 옵션을 이용하여 사용자가 필요한 프로젝트만 logging할 수 있는 기능을 제공한다.

해당 기능을 사용하기 위해서는 <configField id="USE_PROJECT_LOG" value="TRUE" type="String" xmlns=""/>가 먼저 설정 되어 있어야만 사용 가능하다.

```
<configField id="MULTI_PROJECT" value="TRUE" type="String" xmlns="" />
<configField id="PROJECT_ID" value="PRJ1:PRJ2" type="String" xmlns="" />
<configField id="USE_PROJECT_LOG" value="TRUE" type="String" xmlns="" />
<configField id="PRJ1" type="subConfigField" value="true" xmlns="" >
  <!-- Logging ON -->
  <subConfigField id="USE_PROJECT_LOG" type="string" value="TRUE" xmlns="" />
</configField>
<configField id="PRJ2" type="subConfigField" value="true" xmlns="" >
  <!-- Logging OFF -->
  <subConfigField id="USE_PROJECT_LOG" type="string" value="FALSE" xmlns="" />
</configField>
```

- **프로젝트를 선택하여 스튜디오에 로그인할 수 있는 기능**

스튜디오에 로그인하는 경우 프로젝트를 선택하여 로그인할 수 있는 기능을 제공한다.

3.1.2. Release Image / Work Space

Release Image(RI)라는 운영영역과 Work Space(WS)라는 개발영역의 형상을 분리하여 관리할 수 있는 기능을 추가하였다. 체크인/체크아웃을 통해 개발과 운영 형상의 리소스를 이동할 수 있으며, 리소스의 생성 및 수정은 오직 WS 영역에서만 가능하다. RI/WS의 자세한 내용은 매뉴얼을 참고한다.

- **체크인 기능**

리소스를 WS 영역에서 RI 영역으로 이동하는 기능을 제공한다. 체크인 기능을 사용할 때는 체크인 히스토리 내역을 입력해야 하며, 체크인 된 리소스를 WS 영역에서 삭제된다.

● 체크아웃 기능

리소스를 RI 영역에서 WS 영역으로 복사하는 기능을 제공한다.

● 체크아웃 취소 기능

리소스를 체크아웃하였으나 체크아웃된 WS 영역의 리소스를 RI로 반영하지 않고 체크아웃을 취소하는 기능을 제공한다. 체크아웃이 취소된 리소스를 WS 영역에서 삭제된다.

● RI/WS 영역별 리소스 관리(리소스 열기, 삭제, 수정 등) 기능

RI 영역과 WS 영역별로 리소스가 각각 존재하며 이를 각각 관리할 수 있는 기능을 제공한다. RI와 WS 영역의 리소스는 관리 기능에 차이가 있다.

- RI 영역 : 열기, 삭제
- WS 영역: 생성, 저장, 열기, 삭제, 수정

● RI/WS 영역별 단위 테스트 기능

RI 영역과 WS 영역의 리소스를 각각 단위 테스트할 수 있는 기능을 제공한다.

● WS 영역의 EMB에서 피 호출 모듈의 타겟을 변경하여 소스 생성 및 단위 테스트할 수 있는 기능

WS영역의 EMB에서 호출된 피 모듈의 타겟(RI/WS)을 변경하여 리소스를 생성할 수 있으며, 생성된 리소스로 단위 테스트할 수 있는 기능을 제공한다.

호출된 모듈의 타겟 변경 방법은 해당 모듈의 특성창에서 변경할 수 있다.

[그림 3.1] 모듈 타겟 변경 방법



● RI/WS 기능 ON/OFF 기능

통합 서버의 PfmDevSvr.xml의 USE_RIWS 옵션을 이용하여 RI/WS 기능을 ON/OFF 할 수 있는 기능을 제공한다.

```
<!-- RI/WS의 ON/OFF 기능-->
<configField id="USE_RIWS" value="TRUE" type="String" xmlns="" />
```

● 리소스 체크인할 때 WS의 히스토리 데이터 삭제 여부를 옵션으로 설정하는 기능

리소스 체크인할 때 WS의 리소스 히스토리를 삭제하는 기능을 제공하며, 통합 서버의 PfmDevSvr.xml의 DELETE_WS_HISTORIES 옵션을 이용하여 ON/OFF할 수 있다.

```
<!-- RI/WS의 check-in 시 WorkspaceHistory 삭제 여부 -->
<configField id="DELETE_WS_HISTORIES" value="TRUE" type="string" xmlns="" />
```

- **RI, WS, Runtime, DBIO의 데이터소스를 각각 설정할 수 있는 기능**

RI, WS, Runtime, DBIO의 데이터 소스를 각각 설정할 수 있는 기능을 제공한다. 데이터 소스는 JEUSMain.xml에서 설정할 수 있다.

```
<resource>
  <data-source>
    <database>
      <vendor>oracle</vendor>
      <export-name>[EXPORT_NAME]</export-name>
      <data-source-class-name>
        oracle.jdbc.pool.OracleConnectionPoolDataSource
      </data-source-class-name>
      <data-source-type>ConnectionPoolDataSource</data-source-type>
      <database-name>orcl</database-name>
      <data-source-name>
        oracle.jdbc.pool.OracleConnectionPoolDataSource
      </data-source-name>
      <port-number>[PORT]</port-number>
      <server-name>[IP]</server-name>
      <user>[USER]</user>
      <password>[PASSWORD]</password>
      <driver-type>thin</driver-type>
      <connection-pool>
        <pooling>
          <min>10</min>
          <max>20</max>
          <step>2</step>
          <period>600000</period>
        </pooling>
      </connection-pool>
    </database>
    <database>
      <vendor>tibero</vendor>
      <export-name>[EXPORT_NAME]</export-name>
      <data-source-class-name>
        com.tmax.tibero.jdbc.ext.TbConnectionPoolDataSource
      </data-source-class-name>
      <data-source-type>ConnectionPoolDataSource</data-source-type>
      <database-name>tibero</database-name>
      <data-source-name>
        com.tmax.tibero.jdbc.ext.TbConnectionPoolDataSource
      </data-source-name>
      <port-number>[PORT]</port-number>
      <server-name>[IP]</server-name>
      <user>[USER]</user>
      <password>[PASSWORD]</password>
      <driver-type>thin</driver-type>
```

```

        <connection-pool>
        <pooling>
            <min>10</min>
            <max>20</max>
            <step>2</step>
            <period>600000</period>
        </pooling>
        </connection-pool>
    </database>
</data-source>
</resource>

```

- **단일 트랜잭션에서 다중의 커넥션을 Concurrency Problem 없이 처리 가능한 기능**

단일 트랜잭션에서 다중의 커넥션을 사용할 때 Concurrency Problem이 없이 처리 가능한 기능을 제공한다.

- **WS영역의 리소스 삭제 시 WS영역 소스 삭제 기능**

WS 영역의 리소스를 삭제할 때 해당 리소스의 소스, 헤더, xml 파일 등이 로컬에서 삭제되는 기능을 제공한다.

- **배치 프로그램별 RI/WS 영역의 모듈 테스트 기능**

RI/WS CALL 옵션을 export하여 RI/WS 영역의 모듈을 각각 테스트 할 수 있는 기능을 제공한다.

```

### Release Image 영역에 있는 배치 모듈 Call할때
export PFM_WS_DLCALL_YN=N

### Work Space 영역에 있는 배치 모듈 Call할때
export PFM_WS_DLCALL_YN=Y

```

3.1.3. TCache

- **Backup 기능**

공유 메모리에 Load된 TCache Item 내역을 Backup하는 기능을 제공한다.

– 사용법

```
$ pfmtcachebackup -t [ALL | ITEM_NAME | PFM_*]
```

옵션	설명
-t ALL	Item 내역 전체를 Backup한다.
-t [ITEM_NAME]	특정 Item 내역을 Backup한다.
-t PFM_*	시작 글자를 기준으로 와일드카드('*만 제공) 문자에 맞는 Item 내역을 Backup한다.

● Restore 기능

Backup한 파일을 TCache Item 내역에 Load하는 기능을 제공한다.

– 사용법

```
$ pfmtcacherestore -d [Date : YYYYMMDD] -t [ALL | ITEM_NAME | PFM_*]
```

옵션	설명
-d [Date : YYYYMMDD]	Load 할 Backup 파일의 일자를 설정한다.
-t ALL	Backup 파일 Item 내역 전체를 Load한다.
-t [ITEM_NAME]	Backup 파일 중 특정 Item 내역을 Load한다.
-t PFM_*	Backup 파일 중 시작 글자를 기준으로 와일드카드('*만제공) 문자에 맞는 Item 내역을 Load한다.

● Avg List에 의한 튜닝 기능

적절한 TCache 성능 보장을 위해 Avg List 값을 확인할 수 있는 기능을 제공한다.

Avg List 값은 pfmtcacheadmin -s를 이용하여 확인할 수 있으며, SIZE_HASH 값으로 조절할 수 있다. SIZE_HASH와 Avg List는 반비례 관계이다.

● pfmTCacheItem 기능 추가

ProFrame 전용 API인 pfmTCacheItem을 제공한다. pfmTCacheItem은 pfmTCacheGet과 pfmTCachePut 이 하나의 기능으로 구현된 부분이다.

pfmTCacheItem은 \$TCACHECONF의 CALLBACK_NAME에 설정된 BusinessModule의 DB 커넥션을 사용한다. "CALLBACK_NAME=GET_PFM_BM"인 경우를 예를 든다면, pfmTCacheGet했을 때 찾지 못했을 경우 GET_PFM_BM에서에서 데이터를 가져와 pfmTCachePut하는 로직을 처리한다. 이때 DCall 되는 라이브러리는 libGET_PFM_BMC.so.*가 호출된다. BM은 "_WS"가 붙지 않은 라이브러리인 RI 영역의 BM 라이브러리만 Dcall된다. 따라서 WS/RI 환경일 때에는 RI 영역의 BM만 호출된다.

BM은 내부적인 처리 구조로 인해 IN/OUT을 가지는 ProFrame C의 BusinessModule로 제한한다. 해당 BM의 Input 구조체의 size는 SIZE_KEY와 대응되며, Output 구조체의 size는 SIZE_REC와 대응된다. CALLBACK_NAME은 BM의 물리명인 "CALLBACK_NAME=GET_PFM_BM"로 설정한다.

– pfmtcache.cfg 설정 예

```
# cache for PFM_SVC
CACHE_NAME=PFM_SVC
SIZE_MEM=6553          # the total cache memory size in kilo-bytes
SIZE_HASH=1024         # the number of hash key (MAX=6553)
SIZE_KEY=30            # the number of digits of the index column
SIZE_REC=2048          # the size of a single record in bytes
INV_TIMEOUT=10         # invalidation timeout in sec
CALLBACK_NAME=GET_SVC_BM # libGET_SVC_BM.so
```

3.1.4. Runtime 환경 변수 추가

- \$ENVHOME/profile.proframe

```
export PROJECT_ID=[PROJECT_ID] # Project ID
export PFMRELDIR=$PRJROOT/release
export PFMRELDIR_WS=$PRJROOT/ws/release
export PFMWSPREFIX=_WS
```

- \$ENVHOME/profile.[OS Vender]

```
export OS_SO_SUFFIX=.so # Os suffix
```

3.1.5. Tomcat 지원 기능

Tomcat AP서버를 지원 기능 추가되었다.

3.1.6. DBIO Tiberio 실행계획 보기 기능

기존에 Oracle만 지원하던 DBIO의 실행계획을 Tiberio 환경에서도 지원 가능하도록 기능이 추가되었다.

3.2. 수정 기능

3.2.1. 스튜디오

- FileIO가 Plugin으로 변경

FileIO가 기본 리소스 타입에서 제외되고, Plugin 형태로 추가할 수 있도록 변경되었다.

- Server가 Plugin으로 변경

Server가 기본 리소스 타입에서 제외되고, Plugin 형태로 추가할 수 있도록 변경되었다.

- DBIO JAVA 소스젠 타입에서 strcpy 함수 사용이 strncpy로 변경

DBIO JAVA 소스젠 타입에서 문자열 타입의 데이터를 복사할 때 strcpy를 strncpy로 변경되었다.

3.2.2. 통합 서버

- ServerCMD를 별도 스레드로 수행가능하도록 기능 개선

별도 스레드로 생성 할 경우 내부 통신 객체인 SimpleBuffer에서 필요한 정보가 모두 포함되어 통신할 때 사용하도록 변경되었다.

- **ServerCMDStream 멀티프로젝트 지원 기능**

기존 송신 프로그램의 별도 수정 없이 프로젝트 정보가 포함된 HTTP 통신을 이용하여 멀티프로젝트를 지원하는 서버에서 작성된 xml 리소스를 지원하도록 기능 개선되었다.

- **Server가 Plugin으로 변경**

Server가 기본 리소스 타입에서 제외되고, Plugin 형태로 추가할 수 있도록 변경되었다.

- **리소스 형상관리 다중 요청 처리 기능 개선**

형상관리의 다중 요청이 처리될 수 있도록 기능이 개선되었다.

3.3. 삭제 기능

3.3.1. 스튜디오

- **구조체 Tansformation 기능 삭제**

EMB 모듈에서 구조체간의 필드 항목이 다르거나 개발자가 직접 매핑하고자 할 때 사용하는 Tnsformation 기능이 삭제되었다.

- **DBIO C 소스젠 삭제**

DBIO 소스젠 방식 중 하나인 C 소스젠이 삭제되었다. 5.0에서는 Java 소스젠 방식만을 사용한다.

3.4. 버그패치

3.4.1. 스튜디오

- **기본 쿼리를 체크인/아웃할 때 오류 발생하는 현상**

기본 쿼리를 체크인/아웃할 때 관련된 DBIO 전부와 관련 구조체가 같이 체크인/아웃되도록 기능이 수정되었다.

- **Persist DBIO 체크인 중 멈추는 현상**

Persist DBIO 체크인 중 아무런 메시지 없이 동작이 멈추는 현상이 수정되었다.

- **DBIO 기본 쿼리 생성 중 신규 생성 마법사의 테이블명, 그룹명이 반영 안되는 현상**

DBIO 기본 쿼리를 생성하는 중 오류가 발생하여 신규 생성 마법사가 닫히지 않은 상태에서 다른 테이블 또는 다른 그룹으로 변경하여 재생성을 시도할 때 변경된 테이블명, 그룹명이 반영 안되는 버그가 수정되었다.

3.4.2. 통합 서버

- **DEV_HISTORY 테이블에 클라이언트 IP가 누락되는 현상**

History 테이블에 클라이언트 IP가 누락되는 현상이 수정되었다.

- **크기가 큰 리소스 저장 시 로그 서버에서 에러가 나는 현상**

일정 크기 이상의 리소스 파일을 로그 서버를 통해 넘기려 할 때 분할하여 여러 번에 나누어 전송가능하도록 수정되었다.

