

DB 어댑터 사용자 안내서

AnyLink 7

TMAXSOFT

저작권 공지

Copyright 2025. TmaxSoft Co., Ltd. All Rights Reserved.

회사 정보

(주)티맥스소프트

주소 : 경기도 성남시 분당구 황새울로258번길 29, 티맥스수내타워 8-9층

기술 서비스 센터: 1544-8629

홈페이지: <https://www.tmaxsoft.com>

제한된 권리

이 소프트웨어(Tmax AnyLink®) 사용설명서와 프로그램은 저작권법과 국제 조약에 의해 보호됩니다. 사용설명서와 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 사용할 수 있으며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부를 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단으로 전송, 복제, 배포하거나 2차적 저작물을 작성할 수 없습니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 않으며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보 제공만을 목적으로 하며, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 않습니다. 또한 사용설명서 상의 내용이 법적 또는 상업적인 특정 조건을 만족시킬 것을 보장하지 않습니다. 사용설명서는 제품의 업그레이드나 수정에 따라 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 않습니다.

상표 공지

Tmax AnyLink®는 TmaxSoft Co., Ltd.의 등록 상표입니다. 본 사용설명서에 기재된 모든 제품과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용되며 반드시 상표 표시 (™, ®)를 하지는 않습니다.

오픈소스 소프트웨어 공지

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

관련 상세 정보는 제품의 다음 디렉터리에 기재된 사항을 참고하시기 바랍니다. : \${AnyLink_HOME}\AnyLink-licenses

유지 보수

구분	지원항목	서비스 내용
제품지원	패치 & 업그레이드	무상 패치 서비스 제공 메이저 버전 업그레이드 시 할인 혜택 웹 지원을 통한 패치 내역 제공
기술 지원 - 기본 서비스	장애 지원	장애 발생 시 원인 분석 및 조치 Service Desk팀 → 기술팀 → R&D의 3단계 장애 분석 및 조치
	일상 지원(온라인 지원)	E-mail, 전화, 원격, 웹 사이트 등 온라인 자원을 통한 질의 응답 서비스
	고객 맞춤 지원(방문 지원)	고객의 요청으로 수행하는 방문 지원 서비스
기술 지원 - 옵션 서비스	예방 지원	정기 점검을 통한 시스템 운영현황 보고 및 장애 예방 <ul style="list-style-type: none"> 관리자 또는 운영자의 요구사항 수렴 운영 현황(시스템, 엔진 운영) 보고서 제공 필요 시 시스템 개선 권장 사항 보고
유지 보수 비용 및 기간	계약 시 별도 협의	계약 시 EOL/EOS 문서 제공

안내서 이력

제품 버전	안내서 버전	발행일	비고
AnyLink 7	3.1.2	2025-01-20	-
AnyLink 7	3.1.1	2023-03-13	-
AnyLink 7	2.1.4	2019-08-23	-
AnyLink 7	2.1.1	2017-03-24	-

목차

1. 소개	1
1.1. 개요	1
1.2. 환경 구성	1
1.3. 어댑터 동작 방식	1
2. 환경설정	2
2.1. 어댑터 설정	2
2.2. 엔드포인트 설정	4
2.3. 아웃바운드 룰 설정	6
3. 예제	10
3.1. 개요	10
3.2. 어댑터 생성	11
3.2.1. 아웃바운드 어댑터 생성	11
3.3. JEUS 데이터소스 생성	12
3.4. 엔드포인트 생성	13
3.4.1. 아웃바운드 엔드포인트 생성	13
3.5. 스튜디오 리소스 생성	14
3.5.1. 거래그룹 생성	14
3.5.2. 거래 생성	15
3.5.3. 메시지 생성	16
3.5.4. 아웃바운드 룰 생성	16
3.5.5. 플로우 생성	18
3.6. 스튜디오 리소스 설정	19
3.6.1. 거래그룹 설정	20
3.6.2. 거래 설정	21
3.6.3. 아웃바운드 룰 설정	23
3.6.4. 플로우 설정	27
3.7. 배포	33
3.8. 거래 테스트	34
3.8.1. 잡 스케줄 생성	34
3.8.2. 테스트 수행	36
부록 A: Stored Procedure	38
A.1. 개요	38
A.2. 일반적인 Stored Procedure 호출	38
A.3. ResultSet을 반환하는 Stored Procedure 호출	39

1. 소개

본 장에서는 AnyLink DB 어댑터의 기본적인 동작방식에 대하여 설명한다.

1.1. 개요

DB 어댑터는 여러 DB 솔루션과의 연계를 지원한다. AnyLink에서 사용하는 DTO(Data Transfer Object)의 값과, SQL 파라미터를 매핑하여 동적으로 DB 솔루션과의 연계 업무를 처리할 수 있다.

1.2. 환경 구성

AnyLink에서 DB 어댑터를 구동하기 위해서는 다음과 같은 환경 구성이 필요하다.

- Java 버전 : 1.7(JDK 7.0)
- Vendor별 JDBC 드라이버 : 해당 라이브러리는 DB 솔루션 업체를 통해 제공되며, AnyLink 배포판에 포함되지 않는다.

필수 라이브러리는 다음의 폴더에 위치시킨다.

```
${DOMAIN_HOME}/lib/application
```

1.3. 어댑터 동작 방식

DB 어댑터는 DB 엔드포인트를 등록할 수 있으며, **DB 엔드포인트는 인바운드 엔드포인트가 존재하지 않고, 아웃바운드 엔드포인트만 사용한다.**

DB 아웃바운드 엔드포인트는 대상이 되는 DB의 데이터소스에 대한 JNDI 익스포트 네임(export name)을 통해 DB 연결을 가져오며, 해당 연결에 대하여 DB 아웃바운드 룰에서 정의한 쿼리 및 DTO와의 SQL 파라미터 매핑을 수행 및 실행시킨다.

2. 환경설정

본 장에서는 AnyLink DB 어댑터의 환경설정 방법과 아웃바운드 룰을 설정하는 방법에 대해서 설명한다.

2.1. 어댑터 설정

DB 어댑터는 WebAdmin을 이용하여 설정한다.



AnyLink WebAdmin의 접속 및 사용법에 관한 자세한 내용은 "AnyLink WebAdmin 안내서"를 참고한다.

웹 브라우저를 이용해 AnyLink WebAdmin에 로그인한 후 초기 화면에서 **[구성관리] > [어댑터]**를 선택하면 다음과 같이 어댑터 목록이 나타난다.

어댑터 목록 화면

어댑터 목록 화면의 [Add] 버튼을 클릭하면 어댑터를 등록할 수 있다. 어댑터 등록 화면에서 생성할 DB 어댑터에 대한 정보를 입력하고 [저장] 버튼을 클릭한다. 정상적으로 등록되었을 경우 “저장되었습니다.”라는 결과 메시지가 나타난다.

어댑터 등록

어댑터 목록

기본정보

어댑터 아이디*

DB_ADT

어댑터 이름*

DB어댑터

어댑터 종류*

DB

설명

DB 어댑터

업무시스템 이름*

ANL001

배포 사유

최초생성

상세설정

아웃바운드 스택드 폴 아이디

미사용

✓ 저장

✕ 취소

어댑터 등록

• 기본정보 (* : 필수 입력항목)

항목	설명
어댑터 아이디 *	<p>생성할 리소스 어댑터의 아이디를 입력한다.</p> <p>3~30자리 영문자 또는 숫자 조합, 특수문자(-, _) 입력이 가능하다. 해당 값은 동일 업무시스템에서 유일한 값이다.</p>
어댑터 이름 *	<p>생성할 리소스 어댑터의 이름을 입력한다.</p> <p>3~30자리 영문자, 한글 또는 숫자 조합, 특수문자(-, _) 입력이 가능하다.</p>
어댑터 종류 *	<p>어댑터의 종류를 선택한다. 리스트에서 DB를 선택한다.</p> <ul style="list-style-type: none"> • TCP • HTTP • Tmax • WebService • DB • FILE • FTP • MQ • SAP • UDP • SMTP • WEBDAV • Tuxedo • JMS • ebXML • ISO8583 • ProObject

항목	설명
설명	생성하는 어댑터에 대한 설명을 입력한다. 입력하지 않아도 무관하다.
업무시스템 이름 *	어댑터가 포함될 업무시스템을 리스트에서 선택한다.
배포 사유	배포 사유를 작성한다. 작성한 사유는 [배포관리] > [배포이력] 메뉴에서 조회하면 확인할 수 있다.

• [상세설정] 탭

항목	설명
아웃바운드 스레드 풀 아이디	아웃바운드 시에 사용될 스레드 풀 아이디를 설정한다. 엔진에서 수행된 스레드 풀과 다른 아이디가 지정될 경우 아웃바운드 시에 해당 스레드 풀로 스레드를 변경한다. 어댑터별로 스레드 풀을 분리하여 서로 다른 어댑터 간에 영향이 없게 하려는 경우에 설정한다.

2.2. 엔드포인트 설정

WebAdmin의 **[구성관리] > [어댑터]** 메뉴를 선택하면 왼쪽에 등록된 어댑터 트리가 조회된다. 트리에서 엔드포인트를 추가할 어댑터를 클릭하면 오른쪽에 어댑터 상세정보를 조회할 수 있다.

어댑터 상세정보 화면의 **[엔드포인트 목록]** 탭을 클릭하면 해당 어댑터 하위에 있는 엔드포인트 목록이 조회된다.

어댑터 상세정보

기본정보

어댑터 아이디*

DB_ADT

어댑터 이름*

DB어댑터

어댑터 종류*

DB

설명

DB 어댑터

배포 사유

업무시스템 이름*

ANL001

상세설정

엔드포인트 그룹 목록

엔드포인트 목록

No

엔드포인트 아이디

엔드포인트 이름

통신 방향

버전

상태

최종 변경일자

담당자

+ Add

- Del

▶ Start

■ Stop

엔드포인트 목록

[엔드포인트 목록] 탭 아래 **[Add]** 버튼을 클릭하면 **엔드포인트 등록 화면**이 나타난다. 각 항목의 정보를 입력하고 **[저장]** 버튼을 클릭하여 엔드포인트를 등록한다. 엔드포인트가 등록되면 WebAdmin의 왼쪽 트리에서 등록된 어댑터와 엔드포인트를 확인할 수 있다.

DB 엔드포인트 등록

엔드포인트 목록

기본정보

업무시스템 이름

ANL001

어댑터 종류

DB

어댑터 이름

DB어댑터

엔드포인트 아이디*

DB_OUT_EP

엔드포인트 이름*

DB_아웃바운드_엔드포인트

엔드포인트 그룹 이름

엔드포인트 상태

Running

부팅 초기값

Running

통신 방향

Outbound

설명

DB 아웃바운드 엔드포인트

배포 사유

최초 생성

연결정보

연결방식

익스포트 네임

LogAdapter_DataSource

기본 인코딩

엔드포인트 등록 화면

• 기본정보 (* : 필수 입력항목)

항목	설명
엔드포인트 아이디 *	엔드포인트 아이디를 입력한다. 3~30자리 영문자 또는 숫자 조합, 특수문자(-, _) 입력이 가능하다. 해당 값은 동일 어댑터 또는 엔드포인트 그룹에서 유일한 값이다.
엔드포인트 이름 *	엔드포인트 이름을 입력한다. 3~30자리 영문자, 한글 또는 숫자 조합, 특수문자(-, _) 입력이 가능하다.
엔드포인트 그룹 이름	엔드포인트의 그룹 이름이다.
엔드포인트 상태	생성할 때 상태 초기값을 선택한다. <ul style="list-style-type: none"> Running : Running 상태이면 엔드포인트가 시작된 상태가 된다. Stopped : Stopped 상태이면 엔드포인트가 중지된 상태가 된다.
부팅 초기값	부팅할 때 상태 초기값을 선택한다. <ul style="list-style-type: none"> Running : 부팅할 때 엔드포인트가 시작(Running)된 상태가 된다. Stopped : 부팅할 때 엔드포인트가 중지(Stopped)된 상태가 된다.
통신 방향	엔드포인트의 통신 방향을 설정한다. <ul style="list-style-type: none"> Outbound : 외부로 메시지를 전달하기 위한 엔드포인트이다. DB 엔드포인트는 Outbound만 생성할 수 있다.
설명	생성하는 엔드포인트에 대한 설명을 입력한다. 입력하지 않아도 무관하다.
배포 사유	배포 사유를 작성한다. 작성한 사유는 [배포관리] > [배포이력] 메뉴에서 조회하면 확인할 수 있다.

• [연결정보] 탭

항목	설명
익스포트 네임	엔드포인트에서 사용할 데이터소스의 JNDI 익스포트 네임을 입력한다. 해당 데이터소스는 JEUS에서 해당 엔드포인트가 배포될 업무시스템이 설치된 서버에서 사용 가능한 상태로 설정되어 있어야 한다.
기본 인코딩	해당 엔드포인트를 통해서 이동하는 문자 데이터의 기본 인코딩 값을 설정한다. 설정하지 않을 경우 서버 환경을 따른다.

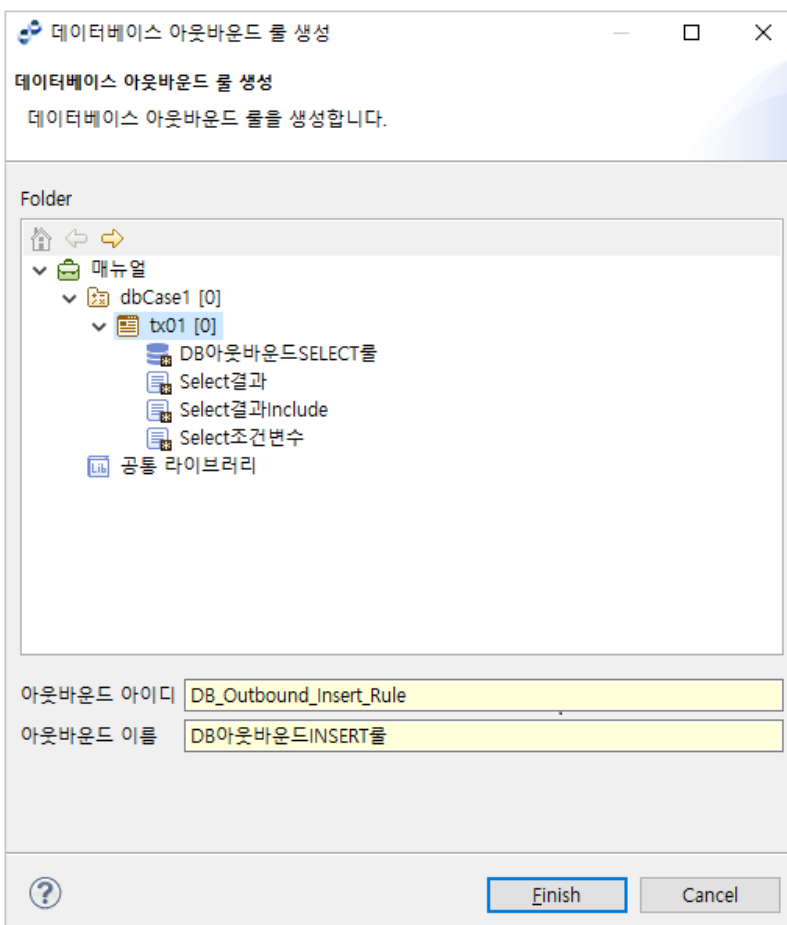
2.3. 아웃바운드 룰 설정

본 절에서는 DB 아웃바운드 룰 설정방법에 대해서 설명한다. DB 아웃바운드 룰은 AnyLink 스튜디오를 이용하여 설정한다.



AnyLink 스튜디오의 자세한 사용법은 AnyLink 스튜디오 사용법에 대한 자세한 내용은 "AnyLink 스튜디오 안내서"를 참고한다.

AnyLink 스튜디오 거래그룹 네비게이터의 **거래/거래그룹** 항목의 컨텍스트 메뉴에서 **[새로만들기] > [아웃바운드 룰] > [데이터베이스 아웃바운드 룰]**를 선택하면 다음과 같이 **데이터베이스 아웃바운드 룰 생성 화면**이 나타난다. 각 항목을 입력하고 **[Finish]** 버튼을 클릭한다.



아웃바운드 룰 생성 화면

항목	설명
아웃바운드 아이디	아웃바운드 룰의 아이디를 입력한다. 영어, 숫자, 특수문자(_) 입력이 가능하며 첫 글자는 영어만 가능하다. 해당 값은 동일 거래 노드에 유일한 값이다.
아웃바운드 이름	아웃바운드 룰의 이름을 입력한다. 한글, 영어, 숫자, 특수문자(-, _) 입력이 가능하다. 아웃바운드 룰 이름은 XML Naming Conversion을 따른다.

아웃바운드 룰을 생성하면 다음과 같은 상세설정 화면이 나타난다. 아래의 탭을 클릭하면 해당 설정 화면으로 이동할 수 있다.

• [DB 아웃바운드 룰] 탭

DB 아웃바운드 룰

아웃바운드 룰 정의

- 프로토콜 DB
- 아웃바운드 룰 ID * DB_Outbound_Select_Rule
- 아웃바운드 룰 이름 * DB아웃바운드SELECT룰
- 요청처리 타임아웃(ms) : 10,000
- Endpoint * (Group) DB_OUT_EP

Q 검색

• 설명

요청 메시지

이름	메시지 ID	배열 선택	
Select조건...	Select_Cond	설정 안함	

가져오기

+

추가

=

삭제

응답 메시지

이름	메시지 ID	배열 선택	
Select결과	Select_Result	설정 안함	

가져오기

+

추가

=

삭제

아웃바운드 룰

◦ 아웃바운드 룰 정의 (* : 필수 입력항목)

항목	설명
아웃바운드 룰 ID *	아웃바운드 룰 아이디를 입력한다. 영어와 숫자, 특수문자(_) 입력이 가능하다. 해당 값은 동일 거래 노드에 유일한 값이다.
아웃바운드 룰 이름 *	아웃바운드 룰 이름을 입력한다. 한글, 영어, 숫자, 특수문자(-, _) 입력이 가능하다.

항목	설명
요청처리 타임아웃(ms)	요청처리 타임아웃을 입력한다. 요청처리 타임아웃은 아웃바운드 룰이 호출된 시점에서부터 타임아웃 시간 동안 연결을 할당 받지 못하거나 외부로 요청을 보냈지만 응답을 받지 못했을 경우 발생한다. 아웃바운드 룰을 생성할 때 10000으로 설정되어 있고 양의 정수로 입력 가능하다.
Endpoint(Group) *	아웃바운드 룰이 등록될 엔드포인트 또는 엔드포인트 그룹을 선택한다. 해당 프로젝트가 업무시스템에 할당되고 서버에 접속한 상태여야만 선택이 가능하다.
설명	아웃바운드 룰의 설명을 입력한다. 설명은 입력하지 않아도 무방하다.

◦ 아웃바운드 룰 메시지 정의

아웃바운드 룰에서는 요청 메시지, 정상 응답 메시지 설정이 가능하다.

항목	설명
요청 메시지	아웃바운드 룰에서 사용될 요청 메시지를 선택한다.
응답 메시지	아웃바운드 룰에서 사용될 응답 메시지를 선택한다.

• [룰 SQL] 탭

데이터베이스 일반 룰

입력 스키마

> Body

SQL Statement 에디터

SQL 목록

No.	SQL 목록
1	SELECT COLA, COLB, TO_CHAR(COLC), UTL_RAW.CAST_TO_RAW(COLD) FROM DB_XA_TEST
2	{ call change_pkg.select_001(:[IN]/Body.colc, :[OUT]outVar) }

SQL 추가

SQL 삭제

SQL 에디터

> SQL 타임

프로시저 결과값 맵핑

출력 매핑 0 Copy(s).

☐ Continue On Error

☐ Continue on next record On Error

최대 결과값 :

룰 SQL

◦ 입력 스키마

아웃바운드 룰의 요청 메시지가 표시된다. 해당 요청 메시지의 필드를 이용해서 SQL에 적용할 수 있다.

◦ SQL Statement 에디터

▪ SQL 목록

룰에서 실행할 SQL의 목록이다.

항목	설명
[SQL 추가] 버튼	룰에서 실행할 SQL을 추가한다.
[SQL 삭제] 버튼	SQL 목록에서 선택한 SQL을 삭제한다.

▪ SQL 에디터

SQL 목록에서 선택한 SQL을 편집할 수 있다. 입력 스키마에 표시된 요청 메시지의 필드를 사용해서 SQL의 변수로 사용할 수 있다.

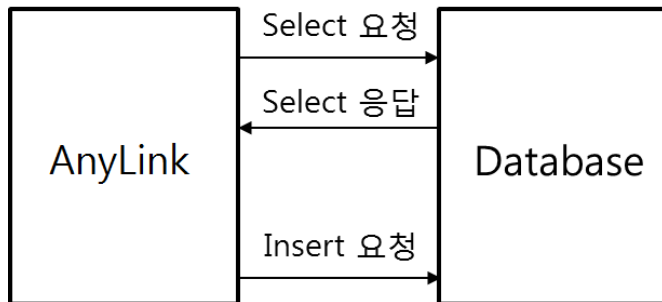
항목	설명
[SQL 타입] 버튼	입력한 SQL 문의 입출력 파라미터의 타입을 직접 지정할 수 있다. 매핑된 경우 자동으로 매핑 대상의 타입에 맞게 설정되므로 일반적으로는 수동으로 지정할 필요가 없다. 주로 Stored Procedure의 OUT 타입 파라미터의 타입을 지정하기 위해 사용된다.
[프로시저 결과값 매핑] 버튼	SQL이 프로시저를 호출하고, 프로시저에 변수를 사용했다면 해당 변수의 프로시저 결과값에 대한 매핑 설정을 한다. 변수가 Cursor 타입인 경우에는 해당 변수의 내부 필드에 대한 정의를 해야 한다. Stored Procedure에 관한 자세한 내용은 Stored Procedure 를 참고한다.
출력 매핑	SQL 에디터에서 사용한 변수를 아웃바운드 룰의 응답 메시지로 매핑한다.
Continue On Error	SQL을 실행할 때 오류가 발생해도 다음 SQL을 실행한다.
Continue on next record On Error	레코드별 값을 읽어오다가 오류가 발생해도 다음 레코드를 읽는다.
최대 결과값	ResultSet 값을 어디까지 읽을지 설정한다.

3. 예제

본 장에서는 AnyLink DB 어댑터를 사용한 예제를 설명한다.

3.1. 개요

다음은 DB 기본 거래 구성이다.



기본 거래 구성도

다음은 거래 순서를 수행하는 과정에 대한 설명이다.

1. 서비스 플로우를 수행한다.
2. DB 아웃바운드 룰을 이용해 Database로 select 쿼리문을 요청한다.
3. Database로부터 쿼리 결과 값을 응답 입력으로 전송받는다.
4. 서비스 플로우에서 Select 응답 입력 받은 결과를 Insert 쿼리문의 입력 변수로 매핑한다.
5. DB 아웃바운드 룰을 이용해 Database로 insert 쿼리문을 요청한다.

전문 구성

다음은 예제에서 사용할 전문 구성이다.

• Select조건변수

필드명	설명	타입	크기
field03	field03	string	20

• Select결과Include

필드명	설명	타입	크기
field01	field01	string	20
field02	field02	string	20
field03	field03	string	20

• Select결과

필드명	설명	타입	포함된 메시지	Array Size
arrayfield	arrayfield	include	Select결과Include	unbounded

3.2. 어댑터 생성

AnyLink WebAdmin에 로그인한 후 초기 화면에서 **[구성관리] > [어댑터]**를 선택한 후 **어댑터 목록 화면의 [Add]** 버튼을 클릭하면 **어댑터 등록 화면**이 나타난다. 어댑터 생성에 대한 자세한 내용은 [어댑터 설정](#)을 참고한다.



AnyLink WebAdmin의 접속 및 사용법에 관한 자세한 내용은 "AnyLink WebAdmin 안내서"를 참고한다.

3.2.1. 아웃바운드 어댑터 생성

다음은 DB 아웃바운드 어댑터를 생성하는 예제이다.

어댑터 등록 어댑터 목록

기본정보

어댑터 아이디*

DB_OUT_ADT

어댑터 이름*

DB_OUT_ADT

어댑터 종류*

DB

설명

DB 아웃바운드 어댑터

업무시스템 이름*

ANL001

백포 사유

상세설정

아웃바운드 스레드 풀 아이디

미사용

아웃바운드 어댑터 등록 화면

• 기본정보

항목	설정값
어댑터 아이디	DB_OUT_ADT
어댑터 이름	DB_OUT_ADT
어댑터 종류	DB
설명	DB 아웃바운드 어댑터
업무시스템 이름	AL001

• [상세설정] 탭

항목	설정값
아웃바운드 스레드 풀 아이디	사용 안함

3.3. JEUS 데이터소스 생성

JEUS WebAdmin에서 [LOCK & EDIT] 버튼을 클릭한 후 [Resources] > [DataSource]를 선택한 후 데이터베이스에서 [ADD] 버튼을 클릭하면 JEUS 데이터소스 설정 화면이 나타난다.

• JEUS 데이터소스 설정

anylink

Domain

Session

Clusters

Servers

Applications

Security

Resources

DataSource

Mail Source

URL Source

Message Bridge

Custom Resource

External Source

External Resource

Monitoring

Console

시스템 상태

0 Failed

0 Standby

2 Running

5 Shutdown

0 Suspended

0 Other

RUNTIME INFO

LOCK & EDIT

해당 Domain의 항목을 수정, 추가 또는 삭제하는 기능입니다.

운영자 매뉴얼

Domain

Server

더보기

Database

HISTORY

JDBC 드라이버의 데이터소스 인스턴스를 생성하고 Connection Pool을 구성하기 위한 특성들을 담고 있다.

Help

동적설정 필수항목

이 페이지의 설정을 수정하려면 왼쪽 메뉴에서 [LOCK & EDIT] 버튼을 클릭하세요.

Data Source Id

db_datasource_test

데이터소스의 ID를 설정한다. 하나의 도메인에서 데이터소스 ID는 데이터소스의 유일한 식별자로서 동작하도록 설정해야 한다.

Export Name

db_datasource_test

데이터소스의 JNDI 이름을 설정한다. 서로 다른 두 데이터소스가 서로 다른 서버에 JNDI 바인딩되는 것을 보장할 수 있으면 해당 데이터소스들은 서로 같은 JNDI 이름을 가질 수 있다. 이는 임의의 서버에서 동일한 JNDI 이름을 가지는 서로 다른 데이터소스를 허용하지 않음을 의미한다. 설정되지 않으면 데이터소스 ID를 JNDI 이름으로 사용한다.

Vendor

tibero

JDBC 드라이버 Vendor의 이름을 설정한다.

Data Source Class Name

com.tmax.tibero.jdbc.ext.TbConnectionPoolDataSource

oracle.jdbc.pool.OracleConnectionPoolDataSource

JDBC 드라이버 데이터소스 클래스의 이름을 설정한다. 패키지 이름을 포함하는 완전한 형태로 적는다.

Data Source Type

ConnectionPoolDataSource

데이터소스의 타입을 설정한다. DATA_SOURCE로 설정하면 Connection Pooling 서비스는 제공되지 않는다. CONNECTION_POOL_DATA_SOURCE로 설정해야 Connection Pooling 서비스가 제공되며 XA_DATA_SOURCE로 설정한 경우 Connection Pooling 서비스와 더불어 XA 연동이 지원된다.

Server Name

192.168.14.98

DB가 실행되는 호스트 이름 또는 IP를 설정한다.

Port Number

8629

DB Listener의 포트 번호를 설정한다.

Database Name

tibero

DB의 이름을 설정한다. Oracle의 경우에는 DB의 SID를 설정한다.

User

anylink

DB User의 ID를 설정한다. 트랜잭션 처리 등을 위해서는 충분한 권한을 가지고 있어야 한다.

Password

.....

(DES)FqrlbQ/D8O1DV571L28rw==

DB User의 Password를 설정한다. 암호화해서 저장할 때에는 '{algorithm}ciphertext'와 같은 형식으로 입력한다.

Support Xa Emulation

☐

[default: false] Connection Pool 데이터소스 타입의 데이터소스에만 유효한 설정으로 이 설정을 적용할 경우 Connection Pool 데이터소스의 커넥션이 글로벌 트랜잭션(XA)에 참여하도록 에뮬레이션한다. JEUS6까지의 LocalXADataSource의 대체 옵션으로 ConnectionPoolDataSource 타입의 Connection Pool에 사용한다. 하나의 트랜잭션에는 하나의 Connection Pool 데이터소스만 참여할 수 있다는 점에 유의해야 한다.

JEUS 데이터소스 설정

항목	설정값
Data Source Id	db_datasource_test
Export Name	db_datasource_test
Vendor	tibero
Data Source Class Name	com.tmax.tibero.jdbc.ext.TbConnectionPoolDataSource
Data Source Type	ConnectionPoolDataSource
Server Name	192.168.14.98
Port Number	8629

항목	설정값
Database Name	tibero
User	AnyLink

• JEUS 서버별 데이터소스 설정

JEUS WebAdmin에서 **[LOCK & EDIT]** 버튼을 클릭한 후 **[Servers]**를 선택한 후 엔드포인트가 구동되는 업무시스템의 서버를 선택한다.

Data Sources
 서버 혹은 클러스터에서 유효한 데이터소스를 지정한다.

Data Source	
<input checked="" type="checkbox"/>	tibero_pool1
<input checked="" type="checkbox"/>	jdbc/anylink_rm
<input checked="" type="checkbox"/>	db_datasource_test

서버 혹은 클러스터에서 유효한 데이터소스의 ID를 명시한다.

JEUS 서버별 데이터소스 설정

Data Source	설정값
db_datasource_test	체크함

3.4. 엔드포인트 생성

어댑터 생성이 완료되면 **[어댑터]** 탭에서 등록된 어댑터를 조회하고, 조회된 어댑터를 클릭하면 **어댑터 상세정보 화면**으로 이동한다.

[엔드포인트 목록] 탭 아래 **[Add]** 버튼을 클릭하면 **엔드포인트 등록 화면**이 나타난다. 엔드포인트 생성에 대한 자세한 설명은 [엔드포인트 설정](#)을 참고한다.

3.4.1. 아웃바운드 엔드포인트 생성

다음은 DB 아웃바운드 엔드포인트를 등록하는 예제이다.

• 기본정보

DB 엔드포인트 등록
DB 엔드포인트 목록

기본정보

업무시스템 이름	ANL001	어댑터 종류	DB	어댑터 이름	DB_OUT_ADT
엔드포인트 아이디*	DB_OUT_EP	엔드포인트 이름*	DB_OUT_EP	엔드포인트 그룹 이름	
엔드포인트 상태	Running	부팅 초기값	Running	통신 방향	Outbound
설명	DB				
백포 사유					

아웃바운드 엔드포인트 등록 화면 - 기본정보

항목	설정값
엔드포인트 아이디	DB_OUT_EP
엔드포인트 이름	DB_OUT_EP
엔드포인트 상태	Running
부팅 초기값	Running
통신 방향	Outbound
설명	DB 아웃바운드 엔드포인트

• [연결정보] 탭

연결정보

연결방식

익스포트 네임

db_datasource_test

기본 인코딩

EUC-KR

아웃바운드 엔드포인트 등록 화면 - [연결정보]

항목	설정값
익스포트 네임	db_datasource_test
기본인코딩	EUC-KR

3.5. 스튜디오 리소스 생성

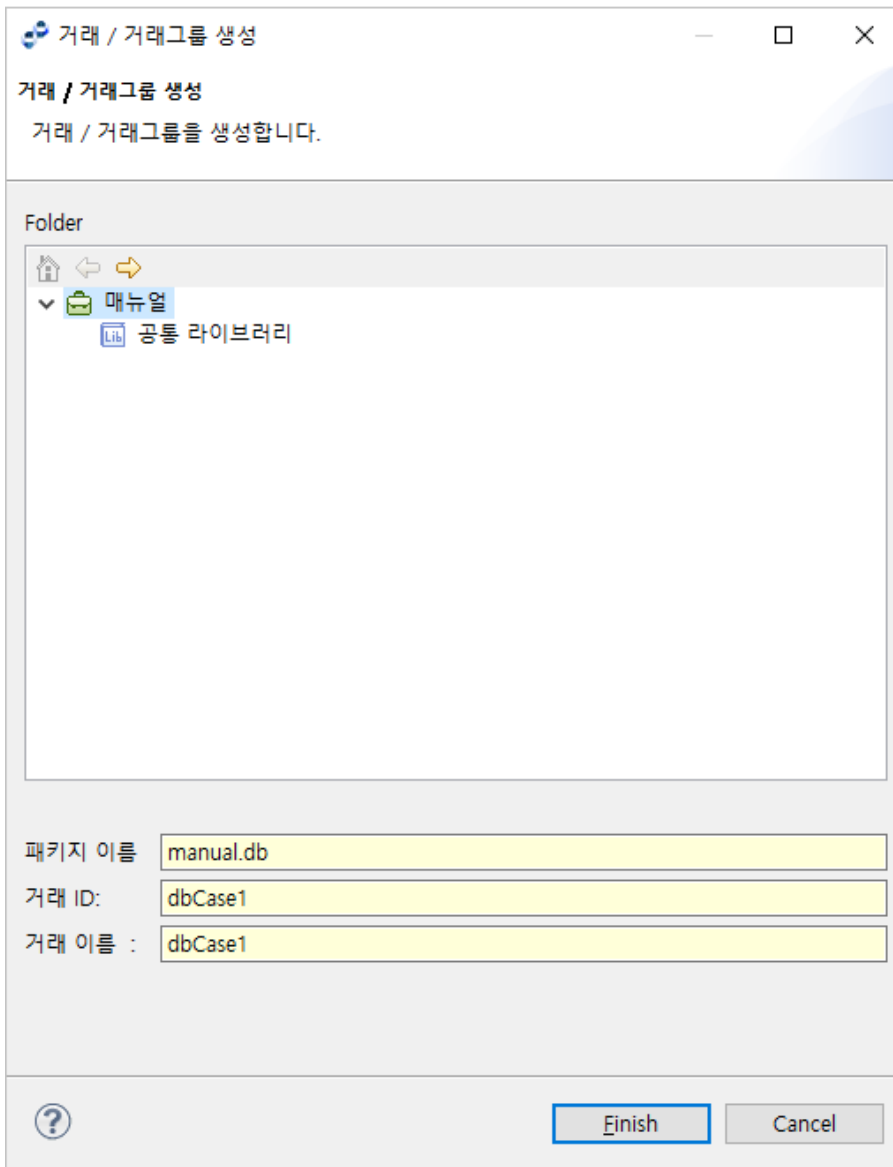
본 절에서는 스튜디오에서 리소스를 생성하는 방법에 대해서 설명한다.



AnyLink 스튜디오 사용법에 대한 자세한 내용은 "AnyLink 스튜디오 안내서"를 참고한다.

3.5.1. 거래그룹 생성

거래그룹 네비게이터에서 **프로젝트**를 선택한 다음 컨텍스트 메뉴에서 **[새로만들기] > [거래 / 거래그룹]**을 선택한다.



거래그룹 생성

항목	설정값
패키지 이름	manual.db
거래 ID	dbCase1
거래 이름	dbCase1

3.5.2. 거래 생성

거래그룹 네비게이터에서 **거래그룹**을 선택한 다음 컨텍스트 메뉴에서 **[새로만들기]** > **[거래 / 거래그룹]**을 선택한다.

거래 / 거래그룹 생성

거래 / 거래그룹 생성

거래 / 거래그룹을 생성합니다.

Folder

▼ 매뉴얼

dbCase1 [0]

공통 라이브러리

노드 타입: 거래

거래 ID: tx01

거래 이름: tx01

Finish Cancel

거래 생성

항목	설정값
노드 타입	거래
거래 ID	tx01
거래 이름	tx01

3.5.3. 메시지 생성

요청 입력 헤더/바디, 요청 출력 헤더/바디, 응답 입력 헤더/바디, 응답 출력 헤더/바디 메시지를 생성한다.



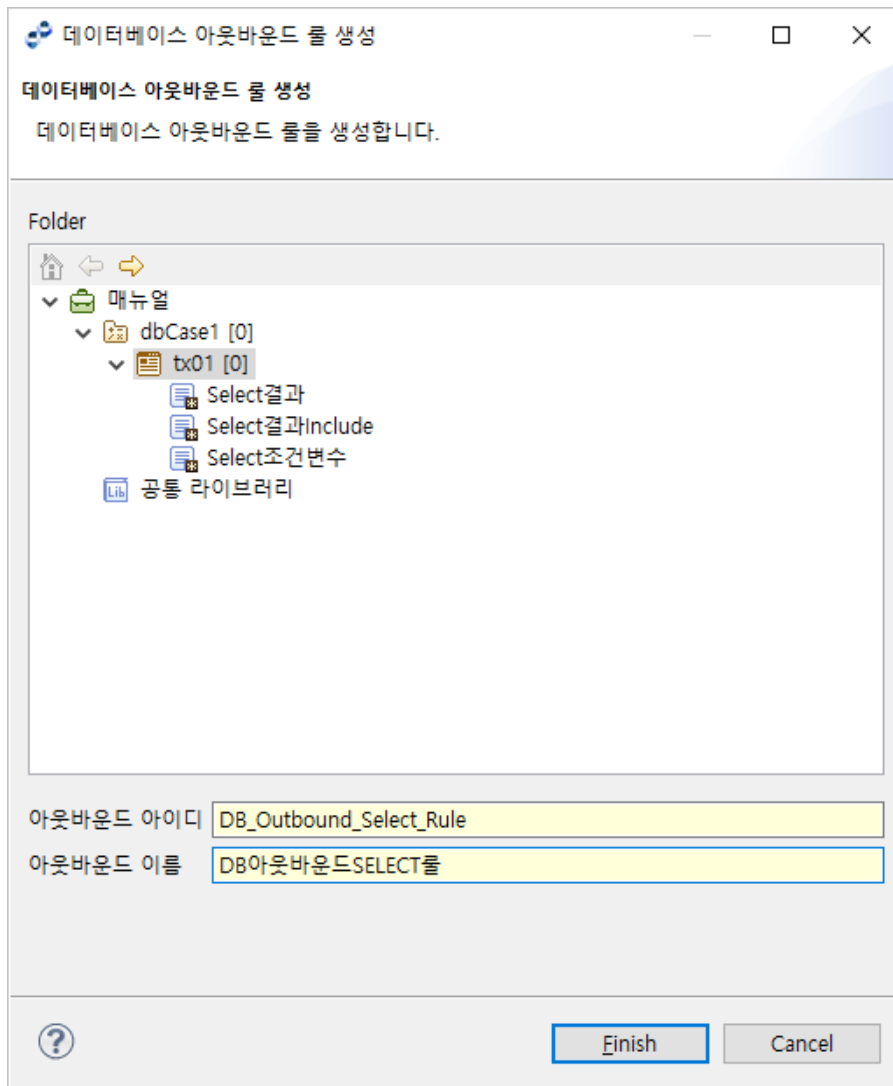
메시지 생성에 관한 자세한 내용은 "AnyLink 스튜디오 안내서"를 참고한다.

3.5.4. 아웃바운드 룰 생성

본 절에서는 DB 아웃바운드 룰(SELECT/INSERT)을 생성하는 예제에 대해서 설명한다.

아웃바운드 룰(SELECT) 생성

거래그룹 네비게이터에서 거래를 선택한 다음 컨텍스트 메뉴에서 [새로만들기] > [아웃바운드 룰] > [DB 아웃바운드 룰]을 선택한다.

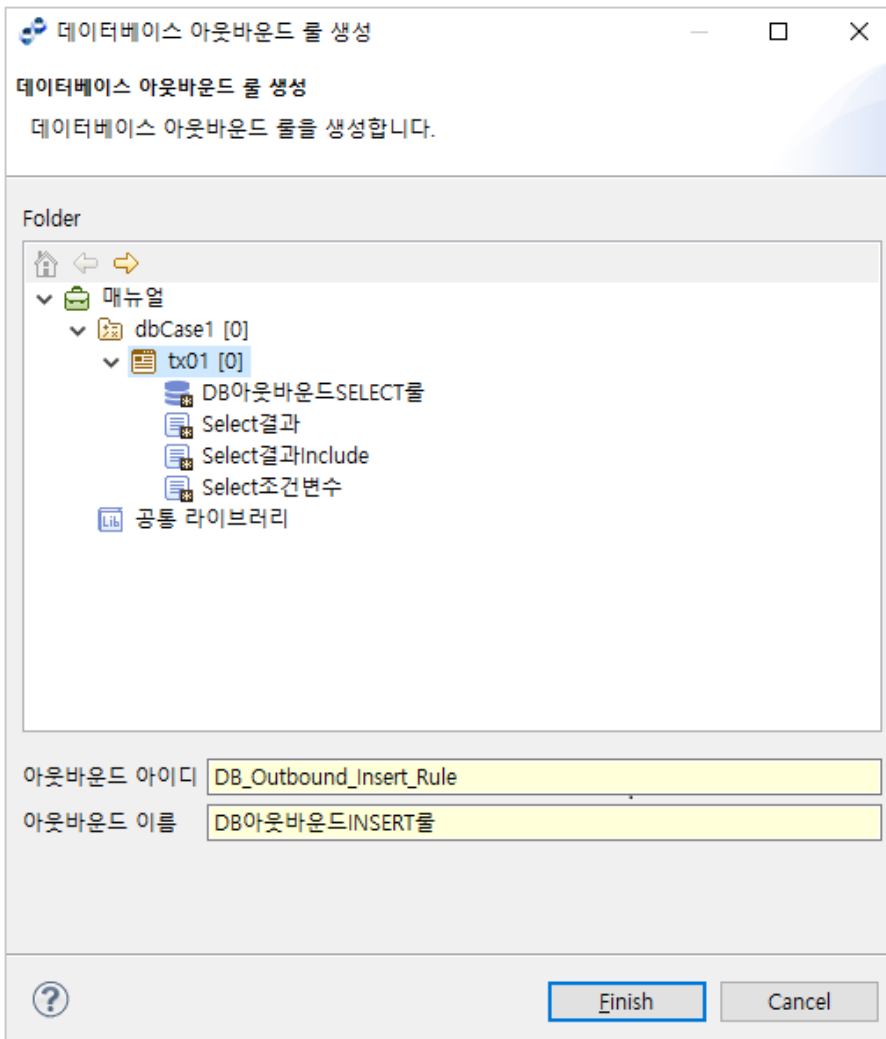


아웃바운드 룰 생성(SELECT)

항목	설정값
아웃바운드 아이디	DB_Outbound_Select_Rule
아웃바운드 이름	DB아웃바운드SELECT룰

아웃바운드 룰(INSERT) 생성

거래그룹 네비게이터에서 거래를 선택한 다음 컨텍스트 메뉴에서 [새로만들기] > [아웃바운드 룰] > [DB 아웃바운드 룰]을 선택한다.

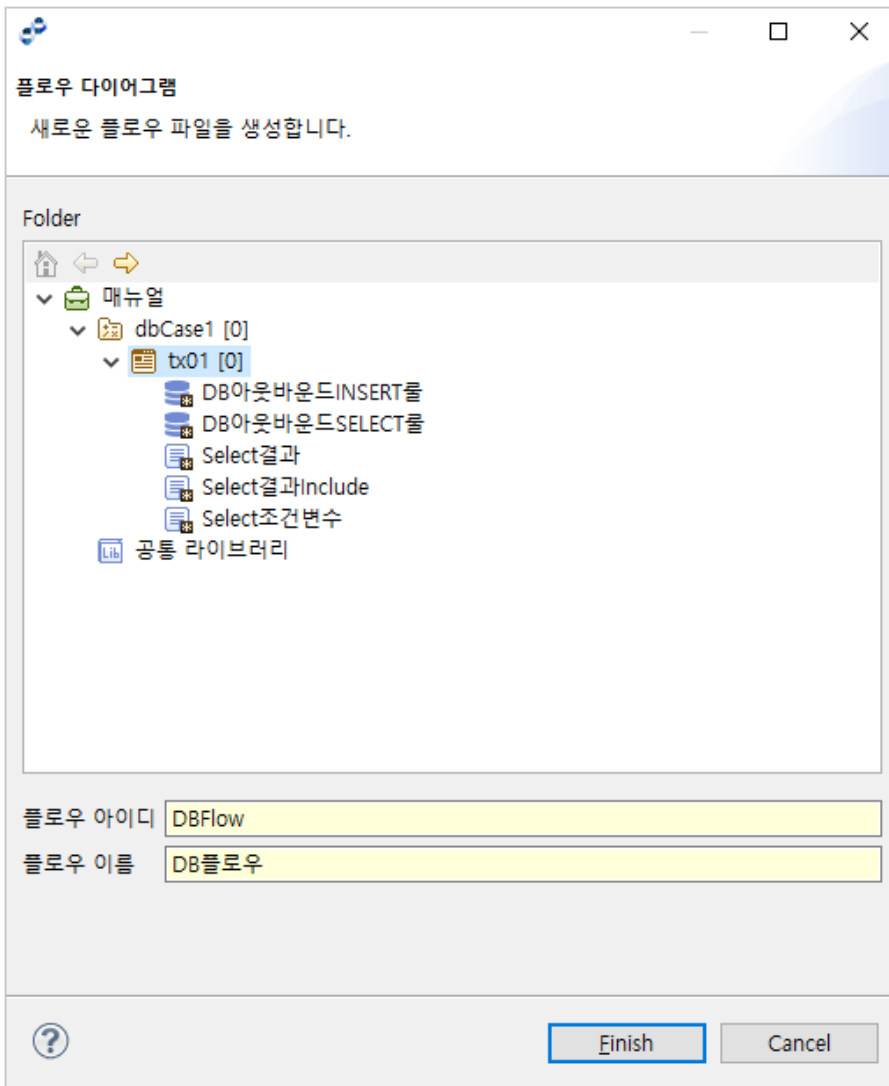


아웃바운드 룰 생성(INSERT)

항목	설정값
아웃바운드 아이디	DB_Outbound_Insert_Rule
아웃바운드 이름	DB아웃바운드INSERT룰

3.5.5. 플로우 생성

거래그룹 네비게이터에서 거래를 선택한 다음 컨텍스트 메뉴에서 [새로만들기] > [플로우]를 선택한다.



플로우 를 생성

항목	설정값
플로우 아이디	DBFlow
플로우 이름	DB플로우

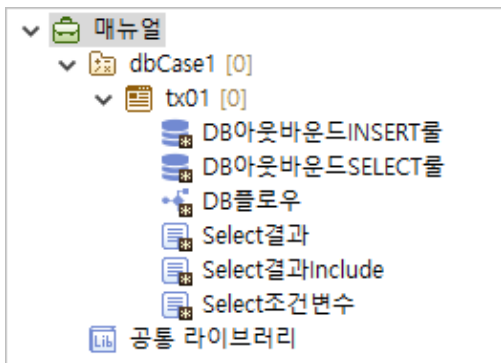
3.6. 스튜디오 리소스 설정

본 절에서는 생성한 리소스를 설정하는 방법에 대해서 설명한다.



AnyLink 스튜디오 사용법에 대한 자세한 내용은 "AnyLink 스튜디오 안내서"를 참고한다.

다음은 리소스를 모두 설정한 후 조회되는 거래그룹 네비게이터이다.



거래그룹 네비게이터

3.6.1. 거래그룹 설정

다음은 거래그룹을 설정하는 예제이다. **[파싱정보]** 탭과 **[파싱 옵션]** 탭은 설정하지 않는다. 전체 예제에서는 거래그룹 설정은 사용하지 않는다.

• [거래그룹 정보] 탭

거래그룹 설정 - [거래그룹 정보]

항목	설정값
Bypass 설정	NO
XA 설정	NO
오류 응답 방식	NONE

• [거래그룹 옵션] 탭

거래그룹 설정 - [거래그룹 옵션]

3.6.2. 거래 설정

- **[거래정보] 탭**

거래 정의

기본정보

거래 ID

tx01

패키지 이름

manual.db.dbCase1

거래 이름

tx01

버전

0

거래 타입

PARENT

Bypass 설정

NO

XA 설정

NO

설명

호출 서비스

서비스 타입

FLOW

서비스 이름

DB플로우 (DBFlow_Message_STRT_Event_611B08)

검색

매핑 설정

요청 메시지

이름	메시지 ID	타입 ID
Select조건변수	Select_Cond	Default Typ

추가

삭제

네임스페이스 URI

로컬 파트

그룹

NONE

그룹번호

정상 응답 메시지

이름	메시지 ID	타입 ID

추가

삭제

네임스페이스 URI

로컬 파트

그룹

NONE

그룹번호

업무오류 응답 메시지

이름	메시지 ID	타입 ID

추가

삭제

네임스페이스 URI

로컬 파트

그룹

NONE

그룹번호

오류 응답처리

오류 응답 방식

PARENT

거래 설정 - [거래정보]

항목	설정값
요청 메시지	Select조건변수
Bypass 설정	NO
XA 설정	NO
오류 응답 방식	PARENT
서비스 타입	FLOW
서비스 이름	DBFlow_Message_STRT_Event_611B08(랜덤으로 생성됨)

• [거래 옵션] 탭

22 | DB 어댑터 사용자 안내서

거래 설정 - [거래 옵션]

항목	설정값
타임아웃 설정 안함	체크
응답 유무	PARENT
거래 우선순위	MEDIUM

본 절에서는 아웃바운드 룰(SELECT/INSERT)을 설정하는 예제에 대해서 설명한다.

다음은 아웃바운드 룰과 [룰 SQL] 탭을 설정하는 예제이다.

DB 아웃바운드 룰

아웃바운드 룰 정의

- 프로토콜DB
- 아웃바운드 룰 ID *DB_Outbound_Select_Rule
- 아웃바운드 룰 이름 *DB아웃바운드SELECT룰
- 요청처리 타임아웃(ms) : 10,000
- Endpoint *(Group)DB_OUT_EP

요청 메시지

이름Select조건...

메시지 IDSelect_Cond

배열 선택설정 안함

응답 메시지

이름Select결과

메시지 IDSelect_Result

배열 선택설정 안함

아웃바운드 룰(SELECT) 설정

항목	설정값
요청처리 타임아웃(ms)	10000
Endpoint	DB_OUT_EP
요청 메시지	Select조건변수
정상 응답 메시지	Select결과

데이터베이스 일반 룰

입력 스키마

Select_Cond

field03

SQL Statement 에디터

SQL 목록

No.1

SELECT FIELD01, FIELD02, FIELD03 FROM SOURCE_TABLE WHERE FIELD03 = :/Select_Cond_0.field03

SQL 에디터

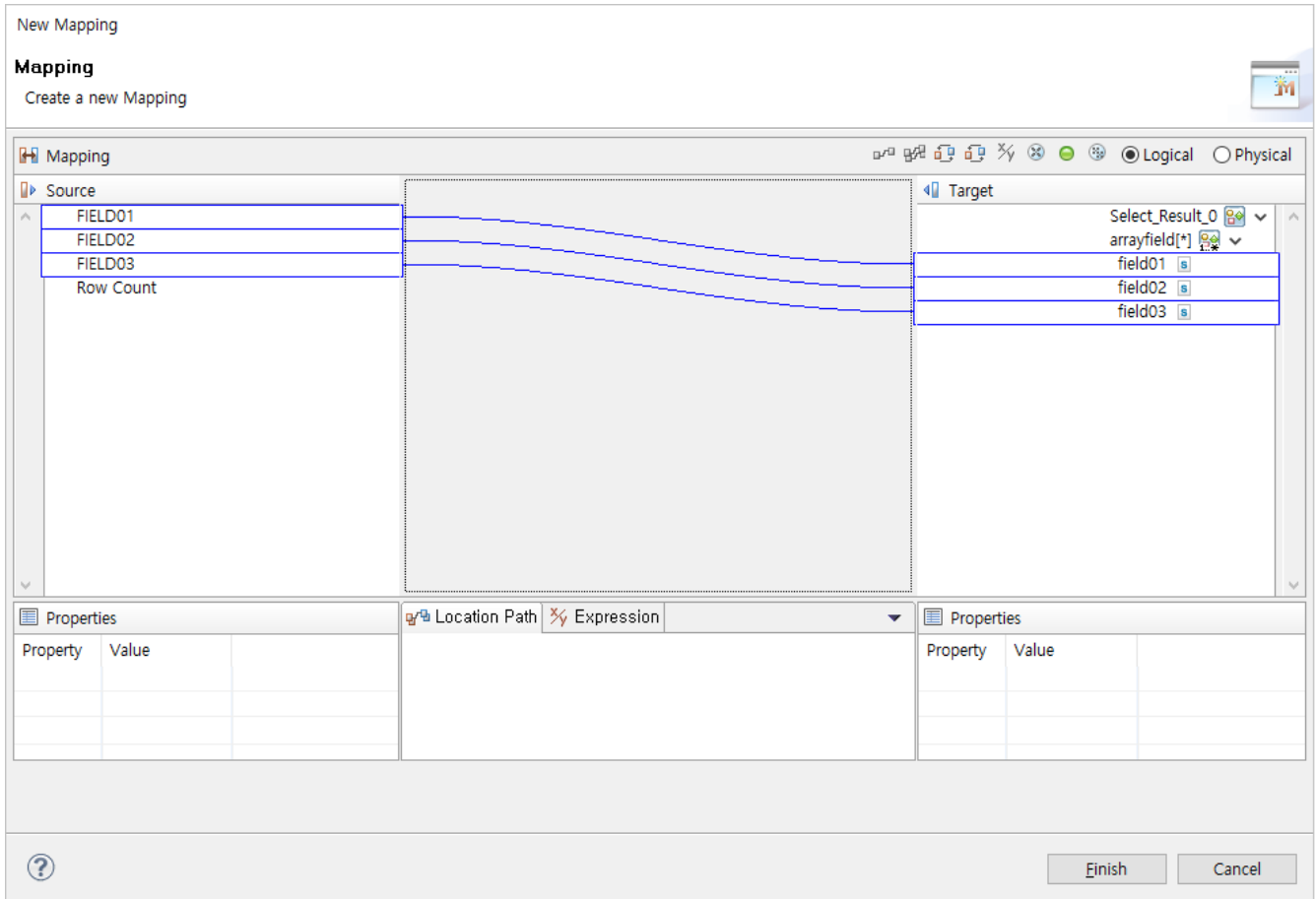
SELECT FIELD01, FIELD02, FIELD03 FROM SOURCE_TABLE WHERE FIELD03 = :/Select_Cond_0.field03

아웃바운드 룰(SELECT) - 룰 SQL 설정

24 | DB 어댑터 사용자 안내서

No	SQL 목록
1	SELECT FIELD01, FIELD02, FIELD03 FROM SOURCE_TABLE WHERE FIELD03 = :/Select_Cond_0.field03

출력 매핑을 위해 룰 SQL 설정 탭에서 **[출력 매핑]** 버튼을 클릭한다.



아웃바운드 룰(SELECT) - 출력 매핑

아웃바운드 룰(INSERT) 설정

다음은 아웃바운드 룰과 **[룰 SQL]** 탭을 설정하는 예제이다.

DB 아웃바운드 룰

아웃바운드 룰 정의

- 프로토콜DB
- 아웃바운드 룰 ID *DB_Outbound_Insert_Rule
- 아웃바운드 룰 이름 *DB아웃바운드INSERT룰
- 요청처리 타임아웃(ms) : 10,000
- Endpoint *(Group)DB_OUT_EP

검색

요청 메시지

이름	메시지 ID	배열 선택	
Select결과	Select_Result	설정 안함	

가져오기 + 추가 - 삭제

응답 메시지

이름	메시지 ID	배열 선택	

가져오기 + 추가 - 삭제

아웃바운드 룰(INSERT) 설정

항목	설정값
요청처리 타임아웃(ms)	10000
Endpoint	DB_OUT_EP
요청 메시지	Select결과

데이터베이스 일반 룰

입력 스키마

Select_Result

arrayfield[*]

field01

field02

field03

SQL Statement 에디터

SQL 목록

No.	SQL 목록
1	INSERT INTO TARGET_TABLE (FIELD01, FIELD02, FIELD03) VALUES (:/Select_Result_0.arrayfield*.field01, /Select_Res...

SQL 추가 SQL 삭제

SQL 에디터

INSERT INTO TARGET_TABLE (FIELD01, FIELD02, FIELD03) VALUES (:/Select_Result_0.arrayfield*.field01, /Select_Result_0.arrayfield*.field02, /Select_Result_0.arrayfield*.field03)

> SQL 타임

프로그래머 결과값 맵핑

출력 매핑 0 Copy(s)

Continue On Error

Continue on next record On Error

최대 결과값 :

아웃바운드 룰(INSERT) - 룰 SQL 설정

No	SQL 목록
1	INSERT INTO TARGET_TABLE (FIELD01, FIELD02, FIELD03) VALUES (:/Select_Result_0.arrayfield*.field01, :/Select_Result_0.arrayfield*.field02, :/Select_Result_0.arrayfield*.field03)

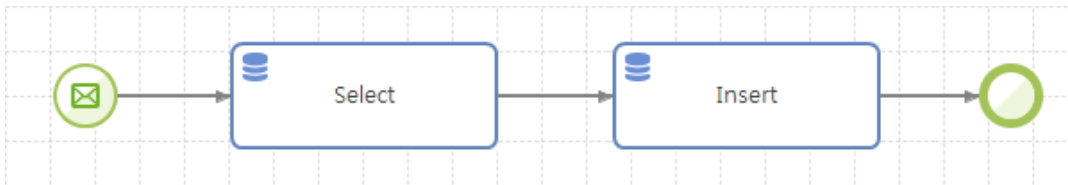
3.6.4. 플로우 설정

다음의 과정으로 플로우를 설정한다.

1. 플로우 그리기
2. 변수 설정
3. 메시지 이벤트 설정
4. 아웃바운드 호출(SELECT/INSERT) 설정
5. 응답 호출 설정

플로우 그리기

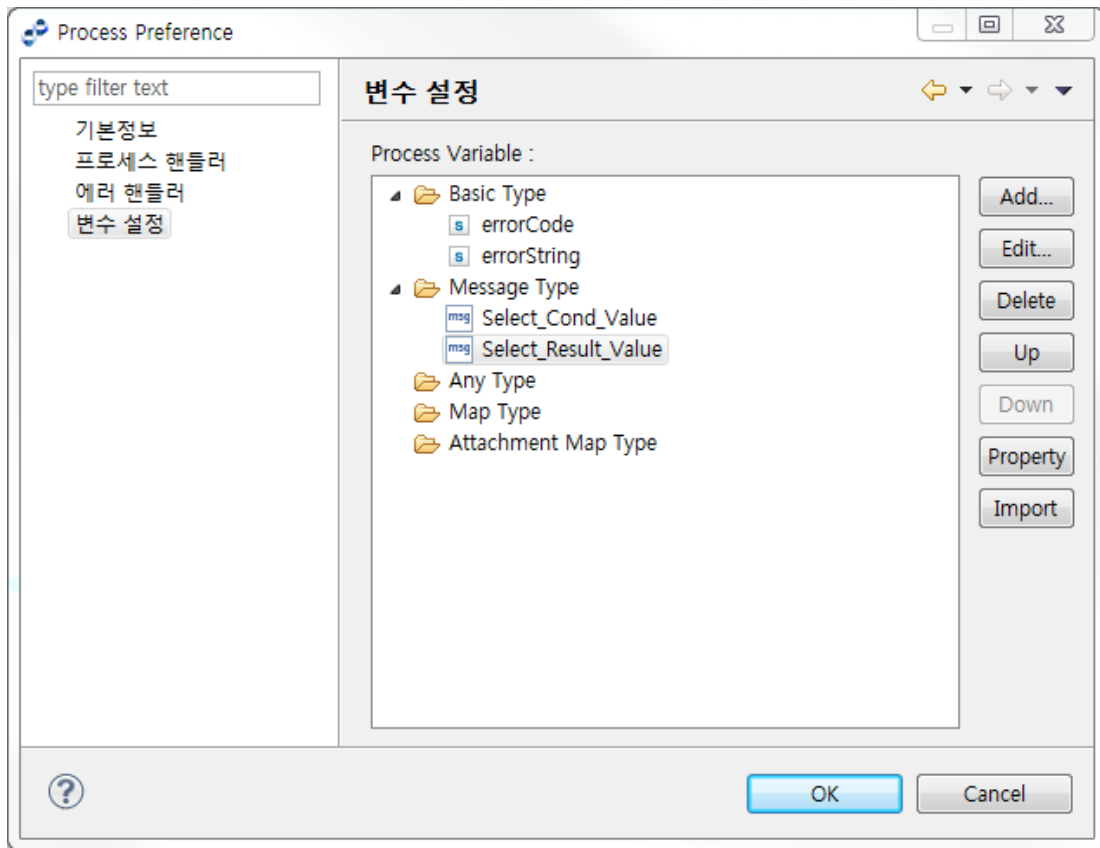
다음은 예제에 대한 서비스 플로우이다.



서비스 플로우 에디터 - 기본 그림

변수 설정

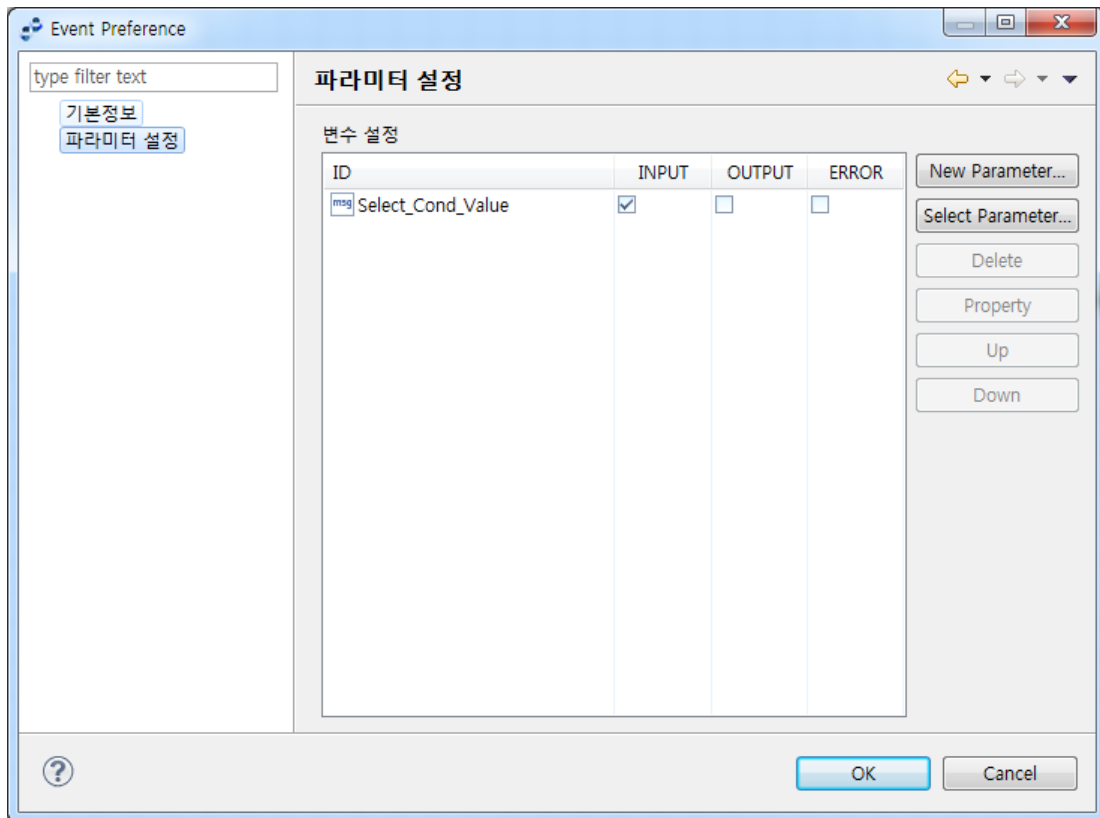
변수 설정을 하기 위해 플로우 에디터 바탕의 컨텍스트 메뉴에서 **[Property]**를 선택한다.



Property Preference - 변수 설정

메시지 이벤트 설정

플로우 에디터의 **[메시지 이벤트]**의 컨텍스트 메뉴에서 **[Property]**를 선택한다. **Event Preference 화면**에서 **[파라미터 설정]**을 선택한다.

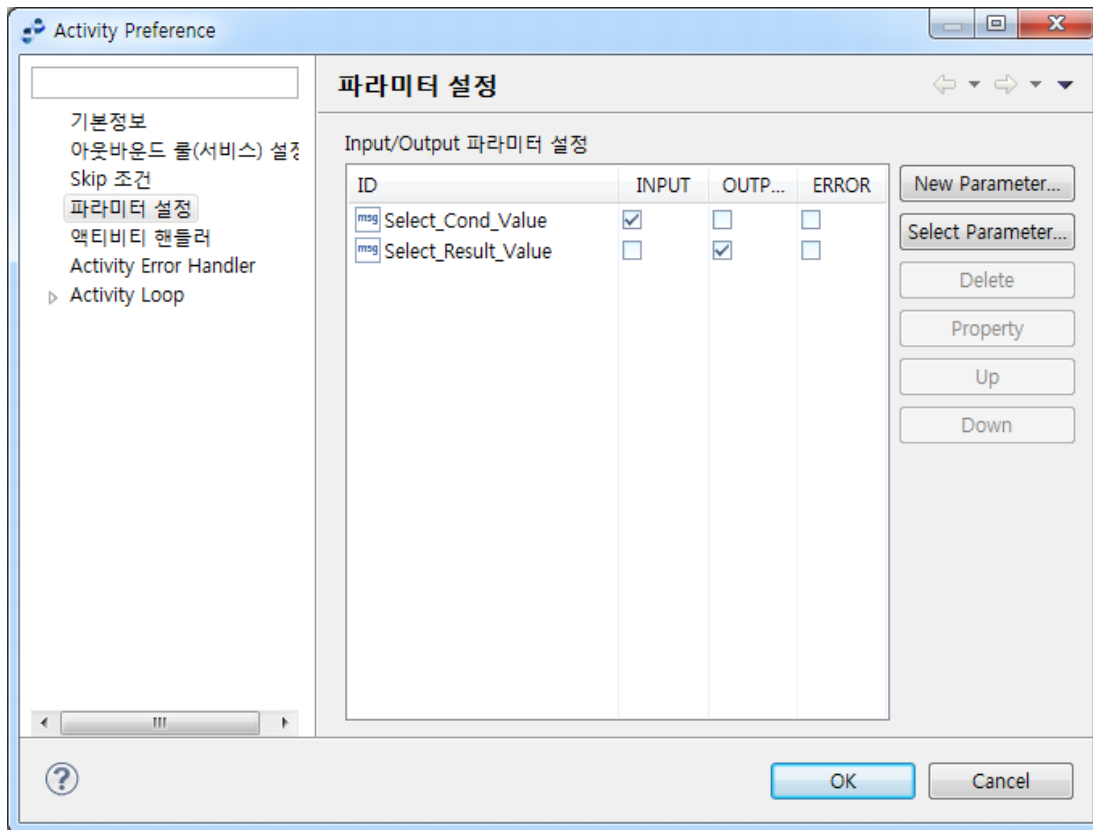


Event Preference - 파라미터 설정

아웃바운드(SELECT) 호출 설정

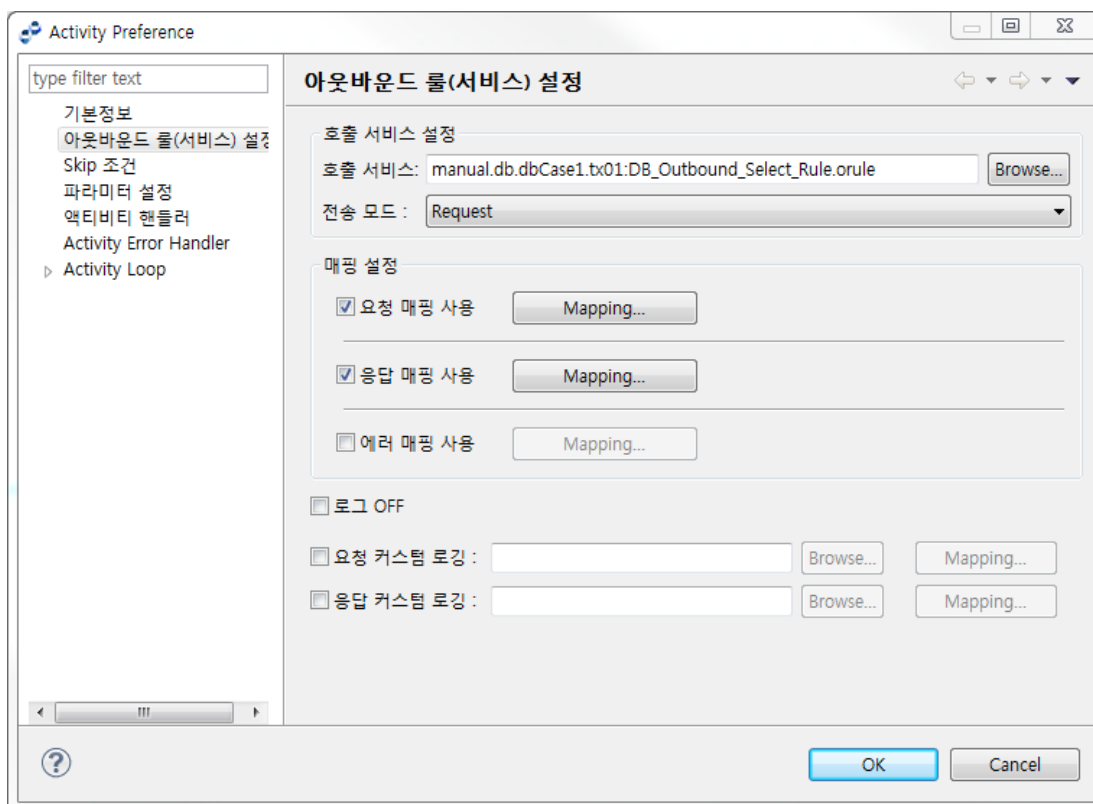
다음의 과정으로 아웃바운드(SELECT) 호출을 설정한다.

1. 플로우 에디터의 **[아웃바운드 호출] > [아웃바운드 룰]** 컨텍스트 메뉴에서 **[Property]**를 선택한다. **Activity Preference** 화면에서 **[파라미터 설정]**을 선택한다.



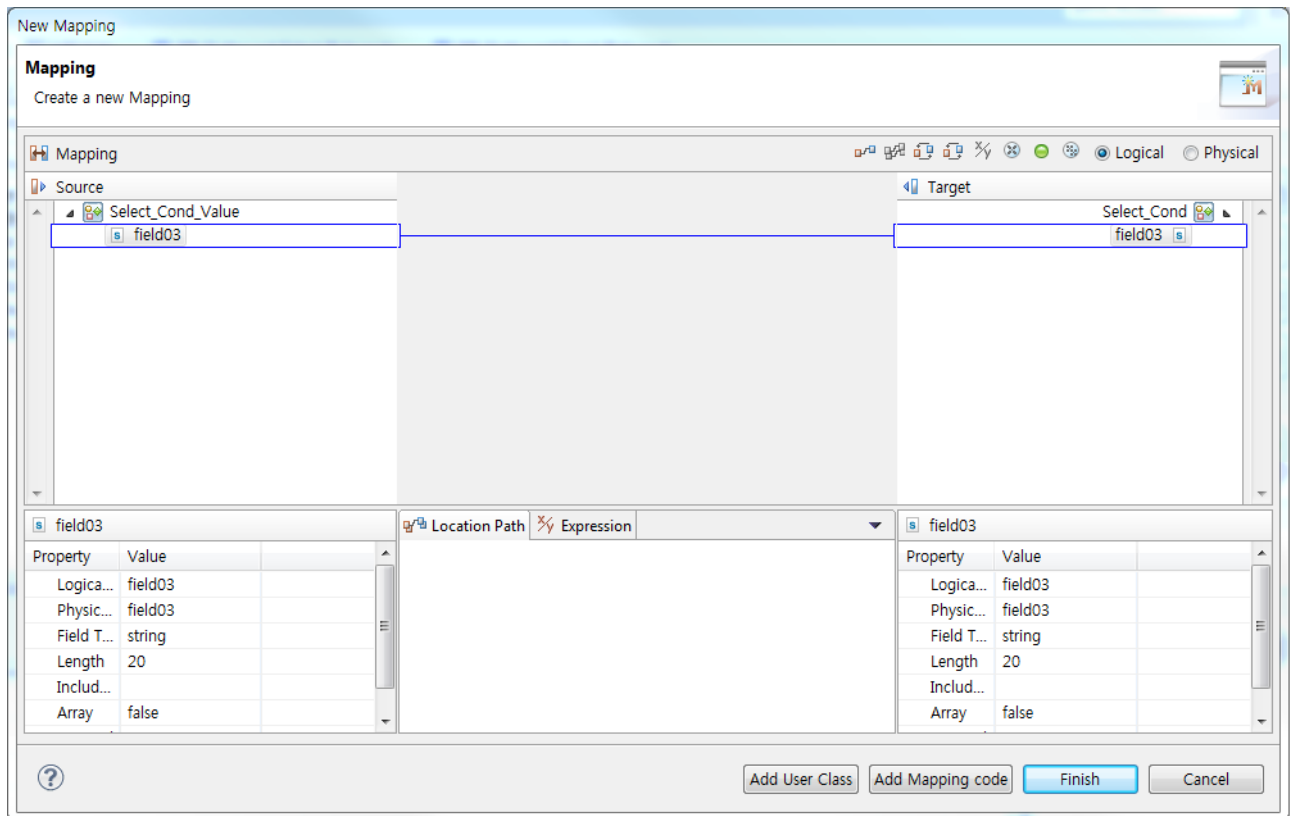
아웃바운드(SELECT) 호출 - 파라미터 설정

2. **Activity Preference** 화면에서 **[아웃바운드 룰(서비스) 설정]**을 선택한다.

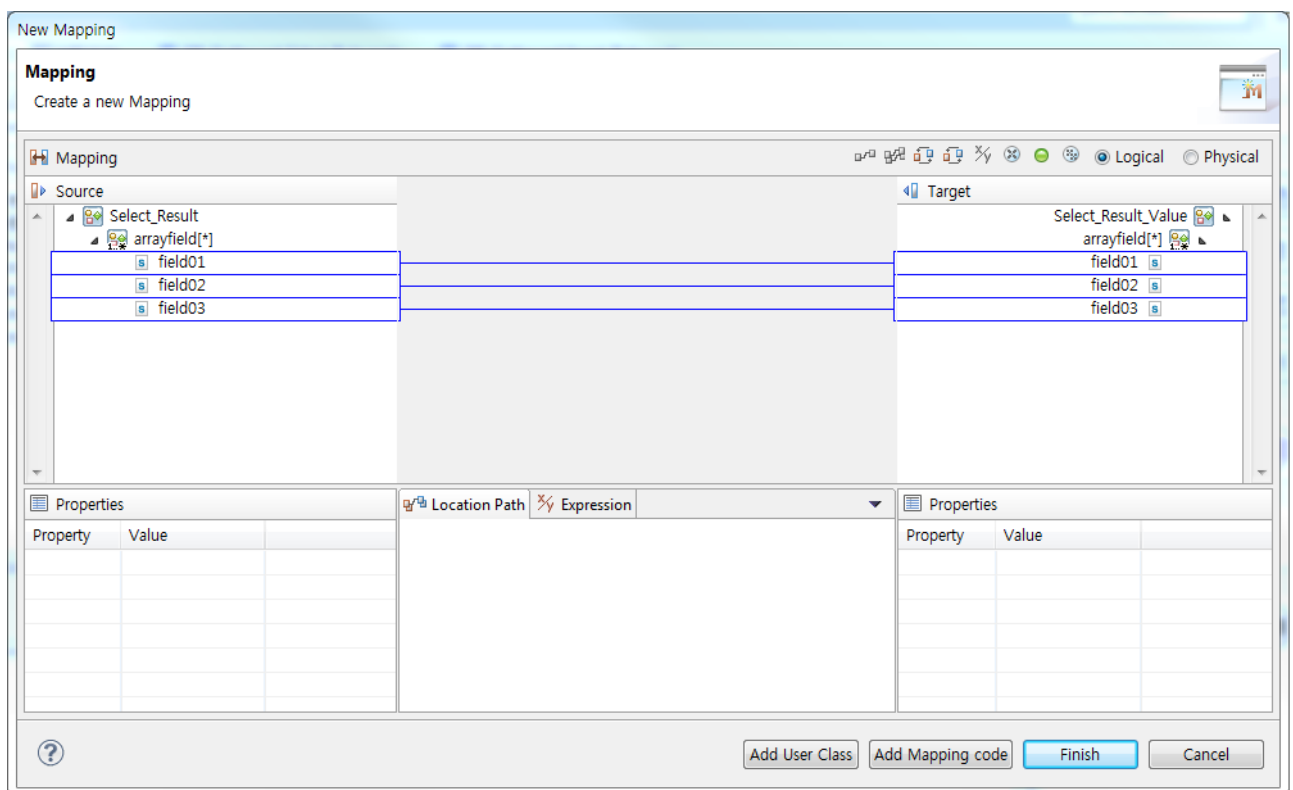


아웃바운드(SELECT) 호출 - 아웃바운드 룰(서비스) 설정

3. **Activity Preference** 화면에서 '요청 매핑 사용'과 '응답 매핑 사용'을 체크하고 **[Mapping]** 버튼을 클릭해서 각 매핑정보를 설정한다.



아웃바운드(SELECT) 호출 - 아웃바운드 룰(서비스) 설정 - 요청 매핑

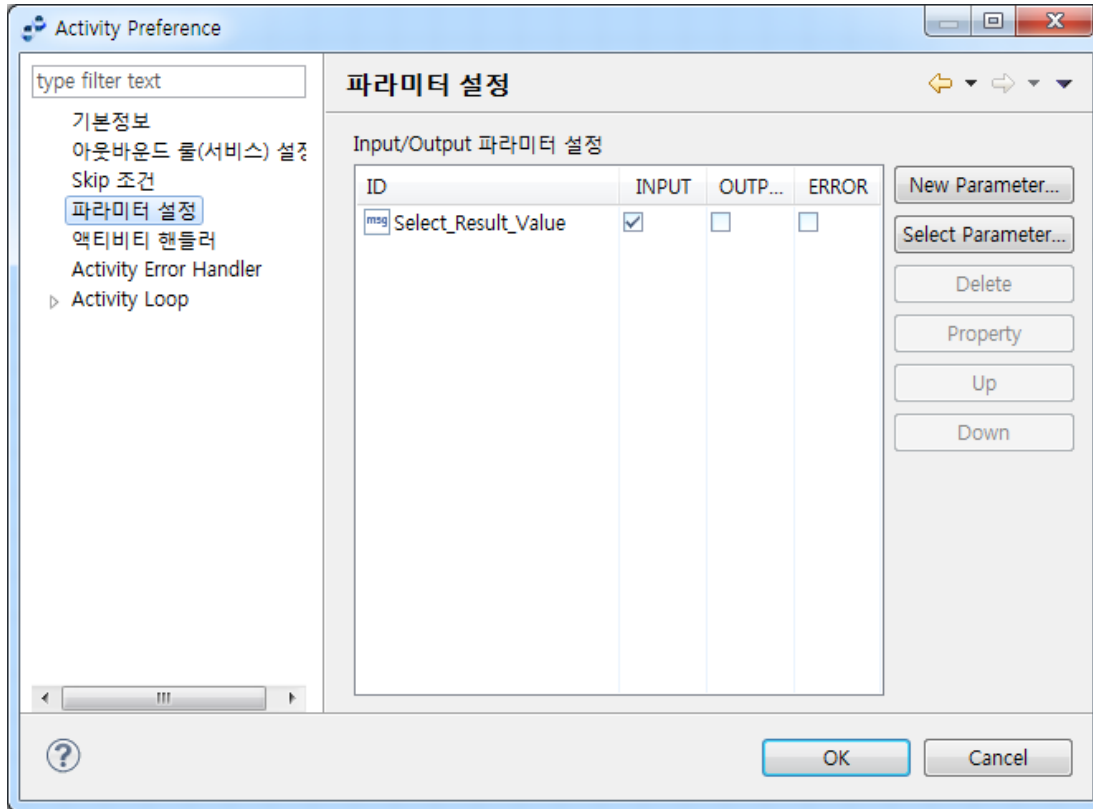


아웃바운드(SELECT) 호출 - 아웃바운드 룰(서비스) 설정 - 응답 매핑

아웃바운드(INSERT) 호출 설정

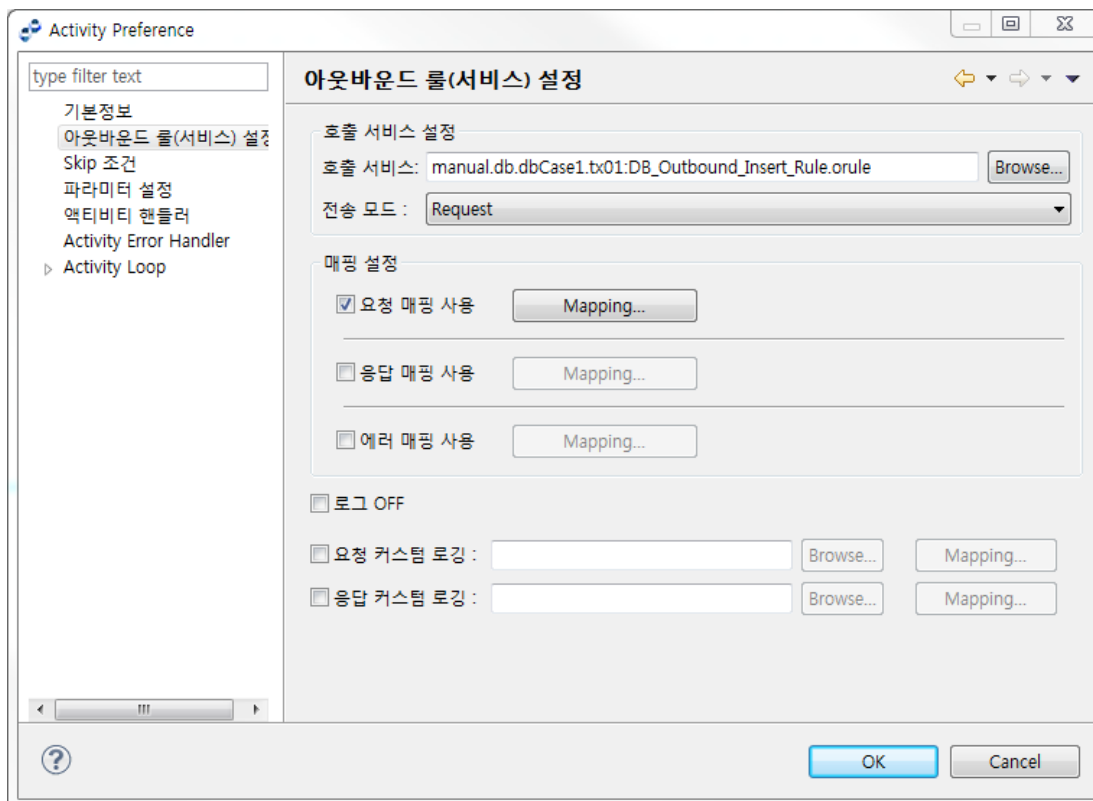
다음의 과정으로 아웃바운드(INSERT) 호출을 설정한다.

1. 플로우 에디터의 [아웃바운드 호출] > [아웃바운드 룰] 컨텍스트 메뉴에서 [Property]를 선택한다. Activity Preference 화면에서 [파라미터 설정]을 선택한다.



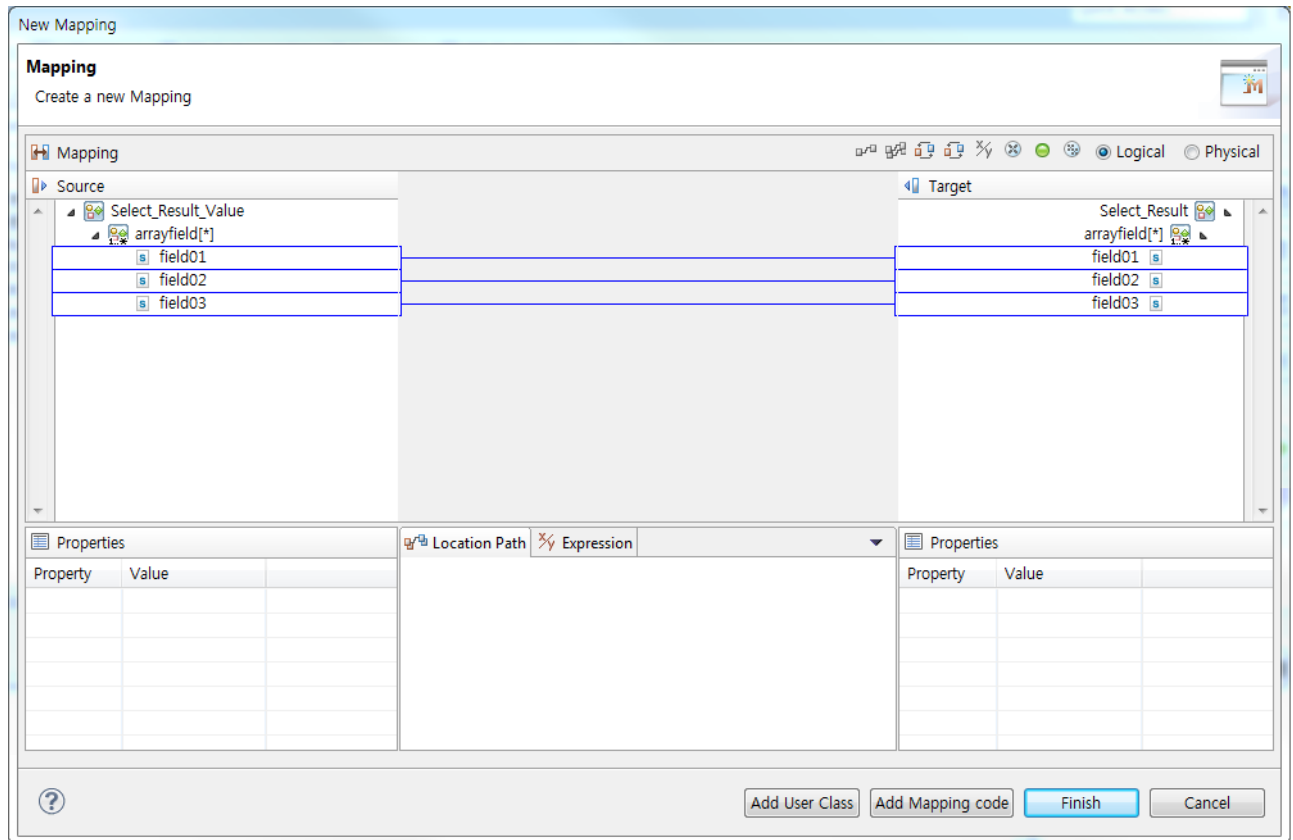
아웃바운드(INSERT) 호출 - 파라미터 설정

2. Activity Preference 화면에서 [아웃바운드 룰(서비스) 설정]을 선택한다.



아웃바운드(INSERT) 호출 - 아웃바운드 룰(서비스) 설정

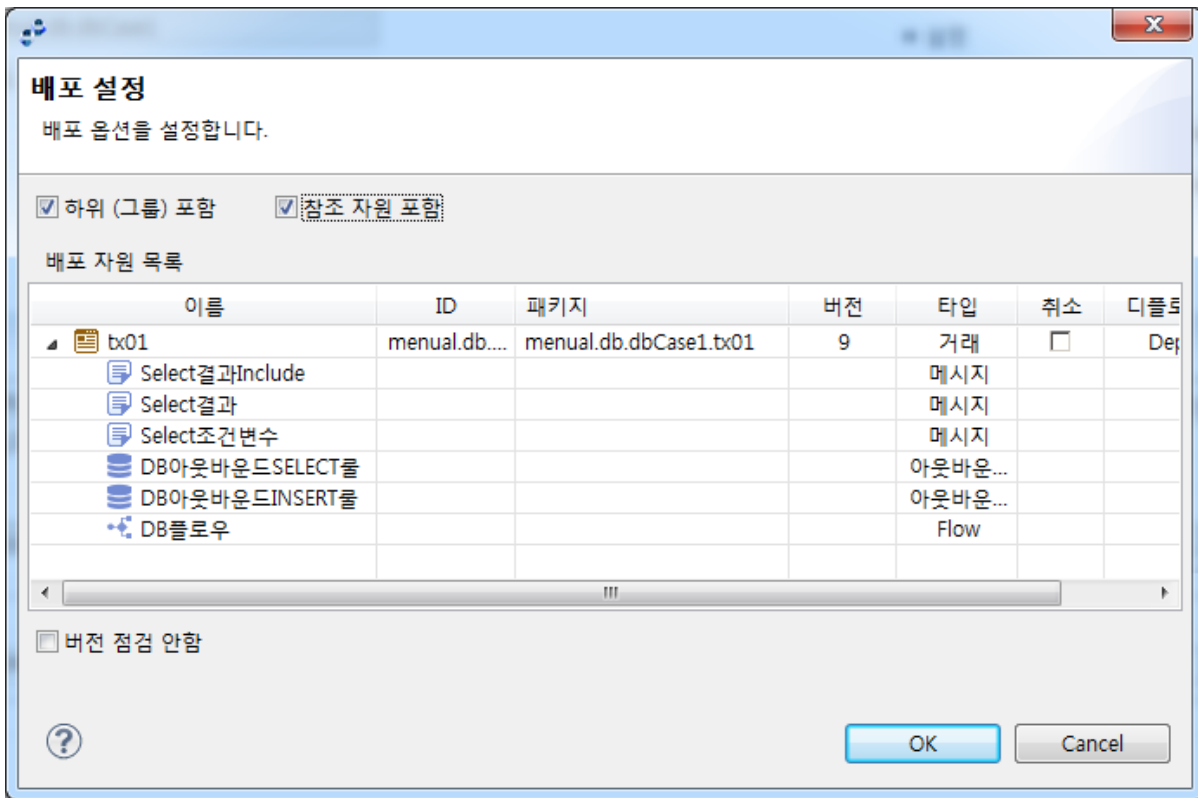
3. **Activity Preference 화면**에서 '요청 매핑 사용'을 체크하고 **[Mapping]** 버튼을 클릭해서 각 매핑정보를 설정한다.



아웃바운드(INSERT) 호출 - 아웃바운드 룰(서비스) 설정 - 요청 매핑

3.7. 배포

거래그룹/거래 배포정보를 설정한다.



거래그룹 / 거래 배포



거래 배포 설정화면에 대한 자세한 사용법은 "AnyLink 스튜디오 안내서"를 참고한다.

3.8. 거래 테스트

본 절에서는 잡스케줄을 생성하고 테스트 수행하는 방법을 설명한다.

3.8.1. 잡 스케줄 생성

본 예제를 구동하기 위해서 WebAdmin의 잡을 등록한다.

잡 등록

기본 정보

잡 아이디*

DBTestJob001

중복확인

잡 이름*

DBTestJob001

거래 아이디*

manual.db.dbCase1.tx01

거래 검색

거래 이름*

tx01

실행 모드

동기

중복처리 방식

이전 잡 종료 후

거래 응답 대기시간(ms)

1000

잡 응답 대기시간(ms)

1000

업무시스템 이름*

ANL001

잡 설정

필드 이름	아이디	타입	사용	값
field03	field03	string	<input checked="" type="checkbox"/>	YES

저장

취소

잡 등록

항목	설정값
잡 아이디	DBTestJob001
잡 이름	DBTestJob001
거래 아이디	manual.db.dbCase1.tx01
실행 모드	동기
중복처리 방식	이전 잡 종료 후
거래 응답 대기시간	1000
잡 응답 대기시간	1000
잡 설정 'field03' 필드	YES

잡 목록 화면에서 등록한 잡 스케줄을 선택한 후 '즉시발송' 항목에 [발송] 버튼을 클릭한다.

AnyLink

구성관리

운영관리

배포관리

모니터링

관리자

1 RUNNING

4 SHUTDOWN

에러로그

JEUS

admin (admin)

메포먼탈 | 메포이력 | 메포관리 | 리소스 식재 | 이관관리

잡 목록

스케줄 목록

스케줄 클러스터 목록

스케줄 실행 이력

캘린더

업무시스템 이름

전체

잡 아이디/이름

DBTestJob001

검색

No	업무시스템 이름	잡 아이디	잡 이름	거래 아이디	거래 이름	잡 상태	중복처리 방식	잡 응답 대기시	거래 응답 대기	실행 모드	즉시 발송
1	ANL001	DBTestJob001	DBTestJob001	manual.db.dbCase1.tx01	tx01	READY	이전 잡 종료 후	1000 (ms)	1000 (ms)	동기	발송

+ Add

- Del

Export

페이지당 목록 수: 10

잡 스케줄 즉시발송

3.8.2. 테스트 수행

다음은 거래 테스트를 수행한 결과 화면이다.

트랜잭션 트레이스

기본정보

GUID

7F000001ADGILGMKCCDLGKC00000030

PGUID

거래 아이디

manual.db.dbCase1.tx01

RGUID

거래 이름

tx01

서버

server1

상태

SUCCESS

요청시간

2016-09-07 15:49:36.714

경과시간(ms)

15

엔드포인트

default-scheduler.default-instant-s

트레이스

No	구분	실행 모듈	타임스탬프	GAP(ms)	메시지	오류
1		Scheduler Adapter (default-scheduler.default-instant-schedule)	2016-09-07 15:49:36.714	0		
2		Flow Start (DBFlow)	2016-09-07 15:49:36.715	1		
3		MESSAGE Event (DBFlow_Message_STRT_Event_611B08)	2016-09-07 15:49:36.716	1		
4		DB Activity (DBFlow_Database_613114) -> [DB_Outbound_Select_Rule.orule]	2016-09-07 15:49:36.716	0		
5		DB Adapter (DB_OUT_ADT) -> [DB_OUT_EP-> DB_OUT_ADT]	2016-09-07 15:49:36.717	1		
6		DB Adapter (DB_OUT_ADT) -> [DB_OUT_EP->]	2016-09-07 15:49:36.72	3		
7		DB Activity (DBFlow_Database_613114) -> [DB_Outbound_Select_Rule.orule]	2016-09-07 15:49:36.722	2		
8		DB Activity (DBFlow_Database_617536) -> [DB_Outbound_Insert_Rule.orule]	2016-09-07 15:49:36.722	0		
9		DB Adapter (DB_OUT_ADT) -> [DB_OUT_EP-> DB_OUT_ADT]	2016-09-07 15:49:36.723	1		
10		DB Activity (DBFlow_Database_617536) -> [DB_Outbound_Insert_Rule.orule]	2016-09-07 15:49:36.728	5		

확인

거래 테스트 - 트레이스 로그

- Source Table

SOURCE_TABLE				
anylink @192.168.14.98:8629:TIBERO_5X - ANYLINK.SOURCE_TABLE				
Row	FIELD01	FIELD02	FIELD03	
1	Test01	Test01	YES	
2	Test02	Test02	YES	
3	Test03	Test03	NO	

거래 테스트 - Source Table

- Target Table

TARGET_TABLE				
anylink@192.168.14.98:8629:TIBERO_5X - ANYLINK.TARGET_TABLE				
Row	FIELD01	FIELD02	FIELD03	
1	Test01	Test01	YES	
2	Test02	Test02	YES	

거래 테스트 - Target Table

부록 A: Stored Procedure

본 부록에서는 Stored Procedure를 사용하는 방법에 대해서 설명한다.

A.1. 개요

Stored procedure는 DBMS에서 DB 서버와 함께 저장되어 있는 연산(subroutine)을 말하며, 일반적으로 SQL로 작성된다. 특히, 복잡한 연산 수행을 Stored Procedure를 사용하여 작성하면 클라이언트 애플리케이션은 프러시저 호출을 통해 기능을 제공받을 수 있으며 프러시저가 수정되었을 때 동일한 변경사항을 적용받을 수 있다.

A.2. 일반적인 Stored Procedure 호출

Stored Procedure는 일반 쿼리를 이용하여 호출할 수 있으며 JDBC에서 정의된 프러시저 호출 문법을 사용한다.

프러시저 호출 문법의 예는 다음과 같다.

- 반환 파라미터가 있는 경우

```
{<return> = call <procedure-name>[<arg1>,<arg2>, ...]}
```

- 반환 파라미터가 없는 경우

```
{call <procedure-name>[<arg1>,<arg2>, ...]}
```

각 파라미터는 IN, IN OUT, OUT 타입 중 하나가 될 수 있으며, IN은 입력 매핑, IN OUT은 입출력 매핑, OUT은 출력 매핑이 가능하다. 각 타입은 [IN], [INOUT], [OUT] Annotation으로 지정된다. Annotation은 콜론(:)의 바로 뒤에 지정한다.

- IN 타입 변수의 경우

입력 메시지 DTO의 특정 필드와 매핑되어야 하며 일반적인 입력 매핑 방법과 동일하게 왼쪽의 DTO 트리에서 소스 필드를 드래그 앤드 드롭한 뒤 [IN] Annotation을 추가한다.

다음은 설정 예이다.

- 원래의 입력 매핑

```
{call myproc(:/path)} Annotation
```

- Annotation 추가 후

```
{call myproc(:[IN]/path)}
```

- IN OUT 타입의 경우

IN 타입과 동일하게 드래그 앤드 드롭 후 [INOUT] Annotation을 추가한다. IN OUT 타입은 IN 타입과 달리 출력 매핑 화면을 통한 출력 매핑도 가능하다.

다음은 설정 예이다.

- 원래의 입력 매핑

```
{call myproc(:/path)}
```

- Annotation 추가 후

```
{call myproc(:[INOUT]/path)}
```

- OUT 타입인 경우

입력 매핑이 없기 때문에 입력 DTO의 필드와 매핑을 할 수 없으므로 직접 전체 표현식(expression)을 입력한다. Annotation이 된 OUT 타입 변수는 출력 매핑 화면에서 지정된 레이블로 표현된다.

```
{call myproc(:[OUT]myOutputVar)}
```

```
{call :[OUT]myReturnVar := myproc()}
```

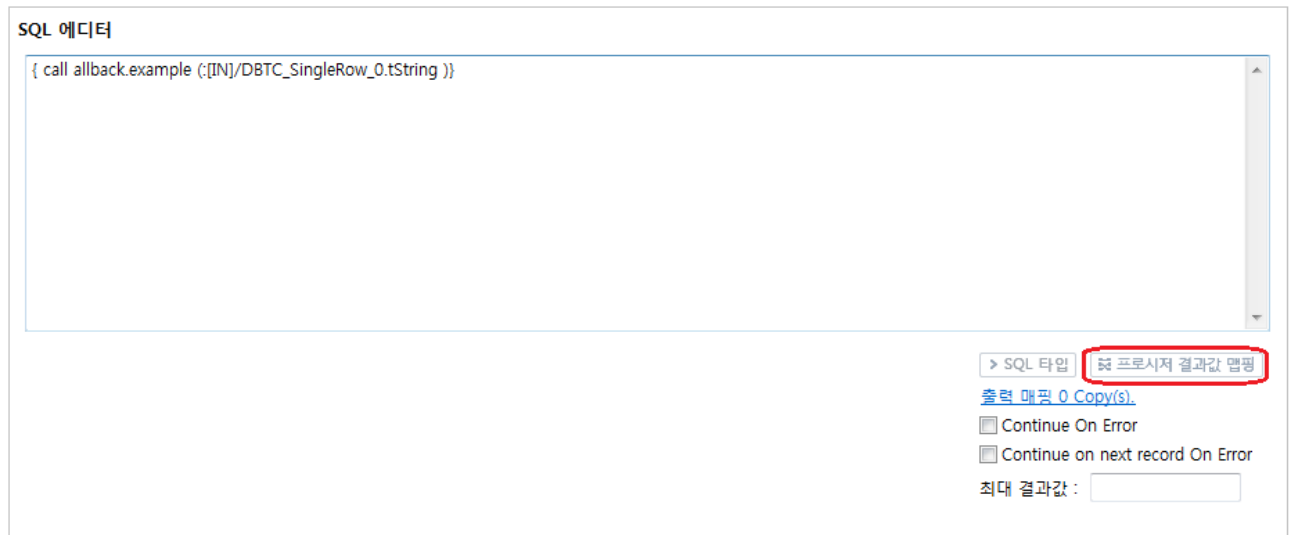


OUT 타입 파라미터 중에서 매핑되지 않은 파라미터는 **[SQL 타입]** 버튼을 클릭해서 수동으로 적절한 타입을 지정해야 프러시저 호출이 가능하다.

A.3. ResultSet을 반환하는 Stored Procedure 호출

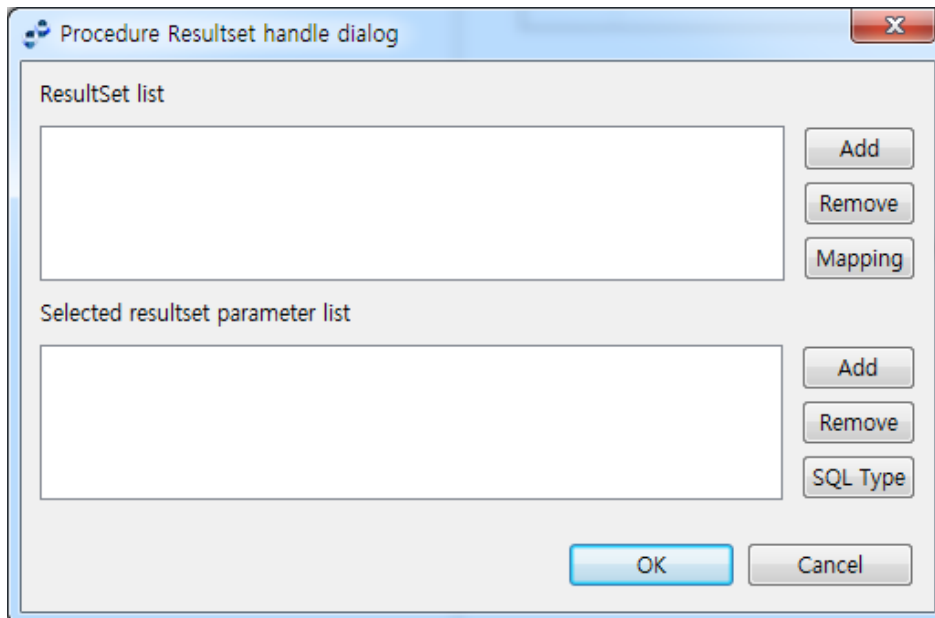
ResultSet을 반환하는 프러시저의 경우 다음과 같이 정의한다.

1. **[프러시저 결과값 매핑]** 버튼을 클릭해서 ReulstSet과 매핑을 정의한다.



프러시저 결과값 매핑 버튼

- 버튼을 클릭하면 나타나는 다음과 같은 대화상자에서 **[Add]** 버튼과 **[Remove]** 버튼을 클릭해서 ResultSet을 추가하고 제거할 수 있다.



ResultSet 정의 대화 상자

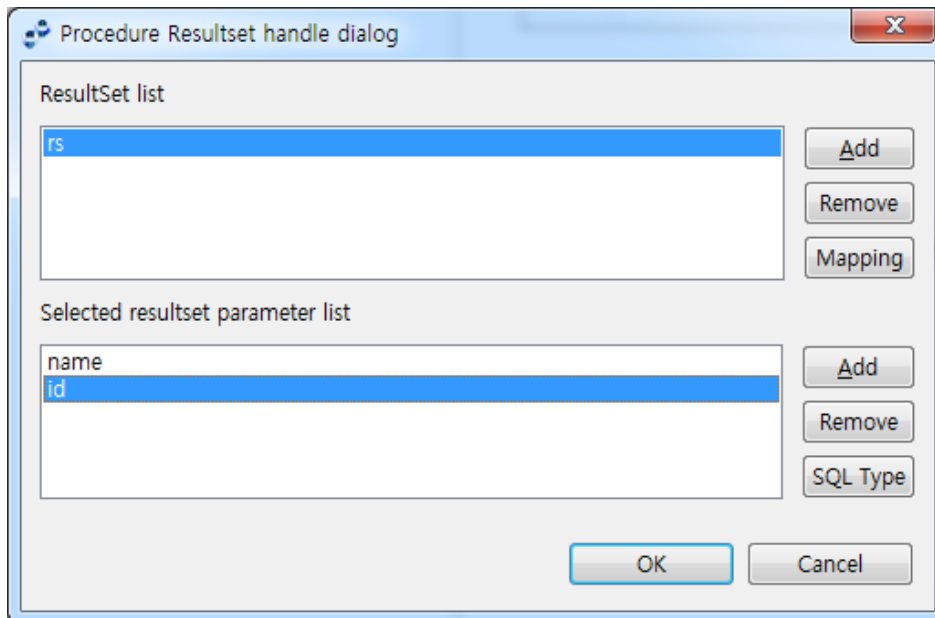
ResultSet 정의 방식은 DBMS의 지원 방식에 따라 크게 2가지 방법으로 나뉘어 진다. MS SQL 서버와 같이 프러시저가 곧바로 결과 집합을 반환하는 ResultSet을 직접 추가해야 한다. 하나의 프러시저가 복수의 ResultSet을 리턴하는 경우 해당 개수만큼 추가한다.



ResultSet의 순서는 룰의 동작에 영향을 미친다. 즉, ResultSet을 리턴되는 순서대로 정의해야 한다.

- '**ResultSet List**'에서 추가한 ResultSet을 선택한 뒤 '**Selected resultset parameter**' 항목의 **[Add]** 또는 **[Remove]** 버튼을 클릭해서 해당하는 ResultSet의 파라미터를 정의한다.

다음은 'name'과 'id'라는 2개의 파라미터를 가지는 하나의 ResultSet을 반환하는 예이다.

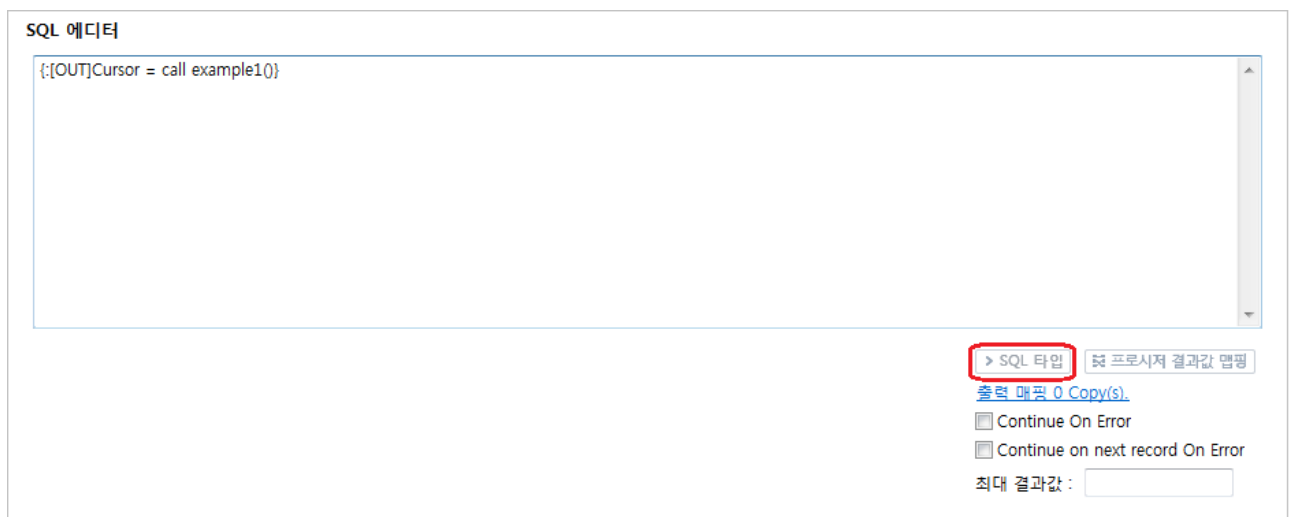


ResultSet을 추가한 예 주의



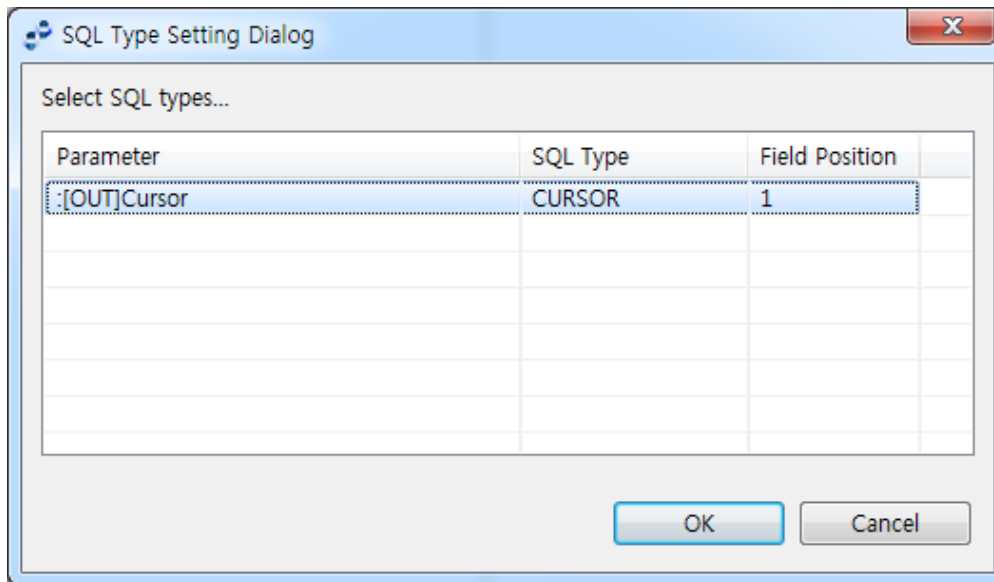
ResultSet과 마찬가지로 파라미터도 이름이 아닌 순서가 중요하다. 즉, ResultSet을 생성하는 SQL 문이 `select id, name from..` 과 같더라도, 이 화면에서 `name, id`의 순서로 정의하면 이후 매핑화면의 `name` 파라미터가 실제 Select 문의 `id` 파라미터에 대응되게 된다.

4. 'ResultSet List'의 **[Mapping]** 버튼을 클릭해서 매핑을 정의한다. 이때 매핑의 소스는 앞서 정의한 파라미터들이 되며 타겟은 룰의 출력 메시지가 된다. 매핑 정의 화면은 일반적인 매핑 화면과 동일하다. Oracle 및 Tmax Tibero는 ResultSet이 커서(Cursor) 타입의 OUT 파라미터로 넘겨지는 경우 직접 ResultSet을 추가하는 대신 **[SQL 타입]** 버튼을 통해 해당하는 OUT 파라미터를 CURSOR 타입으로 선언한다.

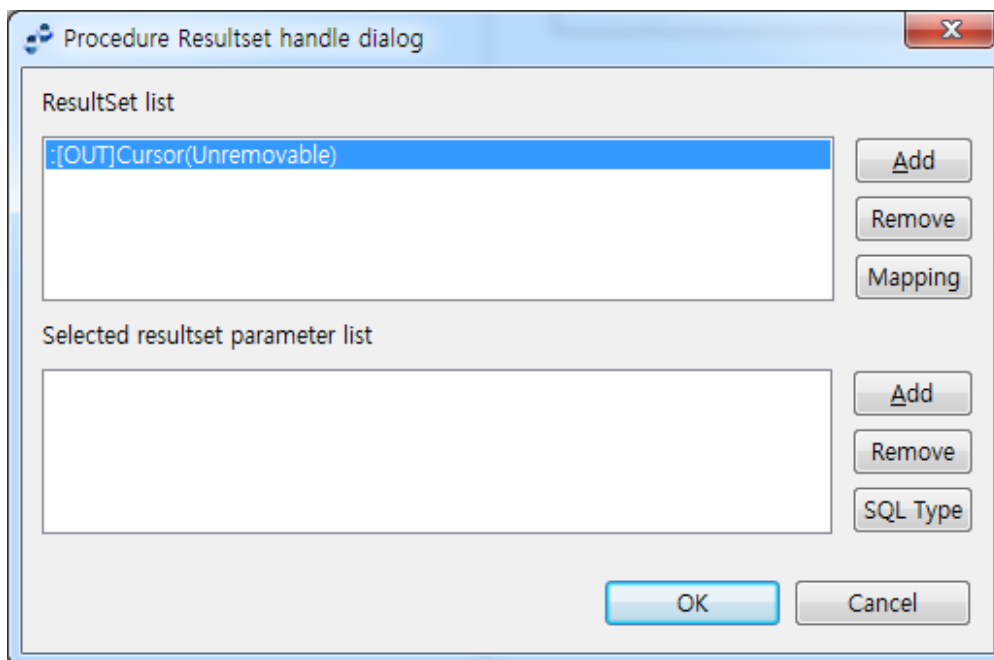


SQL 타입 버튼

5. 출력 파라미터를 CURSOR 타입으로 지정하면 다음과 같이 자동으로 ResultSet list에 추가된다. CURSOR 타입 파라미터의 경우 '(Unremovable)'이라고 표시되며 삭제가 불가능하다.



커서 타입으로 지정한 예



커서 타입 파라미터가 추가된 예

6. 이후 앞서 설명한 것과 동일하게 ResultSet의 파라미터를 선언한 뒤 매핑한다.