

# Tmax 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"> <li>관리자 또는 운영자의 요구사항 수렴</li> <li>운영 현황(시스템, 엔진 운영) 보고서 제공</li> <li>필요 시 시스템 개선 권장 사항 보고</li> </ul>
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
2.2.1. 서버 모드 엔드포인트	6
2.2.2. 클라이언트 모드 엔드포인트	8
2.3. 아웃바운드 룰 설정	10
3. 예제	14
3.1. 개요	14
3.2. 어댑터 생성	16
3.2.1. 인바운드 어댑터 생성	16
3.2.2. 아웃바운드 어댑터 생성	17
3.3. 엔드포인트 생성	18
3.3.1. 인바운드 엔드포인트 생성	18
3.3.2. 아웃바운드 엔드포인트 생성	19
3.4. 스튜디오 리소스 생성	21
3.4.1. 거래그룹 생성	21
3.4.2. 거래 생성	22
3.4.3. 메시지 생성	23
3.4.4. 아웃바운드 룰 생성	23
3.4.5. 플로우 생성	24
3.5. 스튜디오 리소스 설정	25
3.5.1. 거래그룹 설정	26
3.5.2. 거래 설정	27
3.5.3. 아웃바운드 룰 설정	30
3.5.4. 플로우 설정	30
3.6. 배포	36
3.7. 거래 테스트	37
3.7.1. 클라이언트 프로그램	37
3.7.2. Tmax 환경 설정 및 서버 프로그램	38
3.7.3. 테스트 수행	39

# 1. 소개

본 장에서는 AnyLink Tmax 어댑터의 기본적인 동작방식에 대해서 설명한다.

## 1.1. 개요

Tmax 어댑터는 TmaxSoft의 미들웨어인 Tmax와 연계를 지원한다.

## 1.2. 환경 구성

Tmax 어댑터를 사용하기 위해서는 다음과 같은 환경 구성이 필요하다.

- Tmax 라이브러리 : webt.jar, webtasync.jar (해당 라이브러리는 AnyLink 배포판에 포함되어 있다.)

필수 라이브러리의 경우 다음의 폴더에 위치시킨다.

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

## 1.3. 어댑터 동작 방식

Tmax 어댑터는 Tmax 엔드포인트를 등록할 수 있으며, Tmax 엔드포인트는 연결 방식에 따라 서버 모드와 클라이언트 모드로 나눈다.

- 서버 모드

서버 모드는 선택된 포트로 Tmax로부터의 연결이 들어 오기를 대기한다.

- 클라이언트 모드

클라이언트 모드는 지정된 IP/호스트, 포트를 이용하여 Tmax Server에 연결을 한다. Tmax 클라이언트는 항상 연결을 유지하는 방식이다.

## 2. 환경설정

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

### 2.1. 어댑터 설정

Tmax 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.

AnyLink

구성관리

운영관리

배포관리

모니터링

관리자

2

3

에러로그

JEUS ▶

admin (admin)

업무시스템

리모트 에이전트

어댑터

어댑터 클러스터

메시지 큐

패치

search

이름

아이디

엔드포인트 열기

ALL

ANL001

DB

WEB\_ML

FILE

FTP

HTTP

IMS

LOG

MQ

SAP

SMTP

TCP

TMAX

TUXEDO

UDP

WEBDAV

WEB\_SERVICE

클러스터001

전체

어댑터 목록

엔드포인트 그룹 목록

엔드포인트 목록

어댑터 목록

업무시스템 이름

전체

어댑터 종류

전체

어댑터 아이디/이름

검색

	No	업무시스템 이름	어댑터 종류	어댑터 이름	버전	최종 변경일자	담당자
	1	ANL001	TCP	TCP_OUT_ADT	1	2018-12-14 14:53:33	admin(admin)
	2	ANL001	TCP	TCP_IN_ADT	1	2018-12-14 14:42:31	admin(admin)
	3	ANL001	SAP	SAP_ADT	1	2018-12-13 15:43:18	admin(admin)
	4	ANL001	HTTP	HTTP_ADT_Tutorial	1	2018-12-11 17:27:32	admin(admin)
	5	ANL001	Tmax	TMAX_OUT_ADT	1	2018-12-11 14:44:46	admin(admin)
	6	ANL001	Tmax	TMAX_IN_ADT	1	2018-12-11 14:43:47	admin(admin)
	7	ANL001	FTP	FTP_OUT_ADT	1	2018-12-11 10:34:35	admin(admin)
	8	ANL001	WebService	WS_ADT_Tutorial	1	2018-12-10 16:44:56	admin(admin)
	9	ANL001	FILE	FILE_OUT_ADT	1	2018-12-07 17:04:27	admin(admin)
	10	ANL001	DB	DB_OUT_ADT	1	2018-12-07 15:51:39	admin(admin)

+ Add

- Del

▶ Start

■ Stop

⬇ Export

◀◀

◀

1

2

3

4

5

6

7

▶

▶▶

페이지당 목록 수: 10

어댑터 목록 화면

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

#### 기본정보

어댑터 아이디*	TMAX_ADT	어댑터 이름*	TMAX_ADT	어댑터 종류*	Tmax
설명	TMAX 어댑터			업무시스템 이름*	ANL001
배포 사유					

#### 상세설정

아웃바운드 스트림 룰 아이디	미사용
-----------------	-----

어댑터 등록 화면

- 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 Tmax를 선택한다.</p> <ul style="list-style-type: none"> <li>• TCP</li> <li>• HTTP</li> <li>• Tmax</li> <li>• WebService</li> <li>• DB</li> <li>• FILE</li> <li>• FTP</li> <li>• MQ</li> <li>• SAP</li> <li>• UDP</li> <li>• SMTP</li> <li>• WEBDAV</li> <li>• Tuxedo</li> <li>• JMS</li> <li>• ebXML</li> <li>• ISO8583</li> <li>• ProObject</li> </ul>
설명	생성하는 어댑터에 대한 설명을 입력한다. 입력하지 않아도 무관하다.
업무시스템 이름 *	어댑터가 포함될 업무시스템을 리스트에서 선택한다.
배포 사유	배포 사유를 작성한다. 작성한 사유는 <b>[배포관리] &gt; [배포이력]</b> 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

어댑터 목록

기본정보

어댑터 아이디\*

TMAX\_ADT

어댑터 이름\*

TMAX\_ADT

어댑터 종류\*

Tmax

설명

TMAX 어댑터

업무시스템 이름\*

ANL001

배표 사유

상세설정

엔드포인트 그룹 목록

엔드포인트 목록

	No	엔드포인트 아이디	엔드포인트 이름	통신 방향	버전	상태	최종 변경일자	담당자
<div> <div>+ Add</div> <div>- Del</div> <div>▶ Start</div> <div>■ Stop</div> </div>								

어댑터 상세정보 - [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.

엔드포인트 등록 화면은 기본 정보와 [연결정보], [상세설정] 탭으로 구성된다. [연결정보] 탭의 '연결방식' 항목에 따라서 각 탭의 구성이 달라진다. 각 탭 화면에 대한 설명은 해당 절을 참고한다.



#### 기본정보

업무시스템 이름	ANL001	어댑터 종류	Tmax	어댑터 이름	TMAX_ADT
엔드포인트 아이디*	TMAX_INBOUND_EP	엔드포인트 이름*	TAX_INBOUND_EP	엔드포인트 그룹 이름	
엔드포인트 상태	Running	부팅 초기값	Running	통신 방향	Inbound
설명	TMAX 인바운드 엔드포인트				
배포 사유					

#### 연결정보

#### 상세설정

#### 연결방식

연결방식	<input checked="" type="radio"/> Server	<input type="radio"/> Client	
스레드 최대값*	10	기본 인코딩	

#### 주소

IP/호스트 (*)		포트*	31000
------------	--	-----	-------

#### 타임아웃/연결수

최대 연결 수*	10
----------	----

### 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"> <li>Running : Running 상태이면 엔드포인트가 시작된 상태가 된다.</li> <li>Stopped : Stopped 상태이면 엔드포인트가 중지된 상태가 된다.</li> </ul>
부팅 초기값	부팅할 때 상태 초기값을 선택한다.  <ul style="list-style-type: none"> <li>Running : 부팅할 때 엔드포인트가 시작(Running)된 상태가 된다.</li> <li>Stopped : 부팅할 때 엔드포인트가 중지(Stopped)된 상태가 된다.</li> </ul>
통신 방향	엔드포인트의 통신 방향을 설정한다.  <ul style="list-style-type: none"> <li>Inbound : 외부로부터 메시지를 받기위한 엔드포인트이다.</li> <li>Outbound : 외부로 메시지를 전달하기 위한 엔드포인트이다.</li> </ul>
설명	생성하는 엔드포인트에 대한 설명을 입력한다. 입력하지 않아도 무관하다.

항목	설명
배포 사유	배포 사유를 작성한다. 작성한 사유는 <b>[배포관리] &gt; [배포이력]</b> 메뉴에서 조회하면 확인할 수 있다.

## 2.2.1. 서버 모드 엔드포인트

다음은 **[연결정보]** 탭의 '**연결방식**' 항목을 'Server'로 선택한 경우 각 탭 화면에 대한 설명이다.

- **[연결정보] 탭** (\* : 필수 입력항목)

엔드포인트 등록 화면 - [연결정보] - Server

### ◦ 연결방식

항목	설명
연결 방식	엔드포인트가 동작하는 연결 방식을 설정한다. <ul style="list-style-type: none"> <li>◦ Server : 엔드포인트가 서버 모드로 동작한다.</li> <li>◦ Client : 엔드포인트가 클라이언트 모드로 동작한다.</li> </ul>
스레드 최대값 *	최대 작업 스레드(work thread) 개수를 지정한다.  아웃바운드 어댑터의 경우 큐를 통해 전달된 메시지를 각 어댑터에 할당된 작업 스레드가 처리하게 된다. 따라서 작업 스레드 개수는 동시에 처리되는 메시지의 수를 의미한다. 하지만 최대 스레드 수가 너무 크면 잦은 스레드 컨텍스트 교환에 의한 오버헤드가 증가하므로 적절한 값을 설정해야 한다.
기본 인코딩	엔드포인트에서 사용할 기본 인코딩을 설정한다.  메시지에 설정된 인코딩이 우선 적용되고 메시지에 설정되지 않았을 경우에 적용된다.  메시지와 엔드포인트 모두 설정되지 않은 경우에는 시스템 기본 인코딩으로 적용된다.

### ◦ 주소

항목	설명
IP/호스트 *	노드의 IP/호스트를 설정한다.
포트 *	Listen할 포트 번호를 설정한다.

◦ 타임아웃/연결수

항목	설명
Read 타임아웃 (ms) *	Tmax 서버에 요청 전문을 보내고 응답을 대기하는 최대 시간을 지정한다. 이 시간을 넘어서 응답 전문이 오지 않으면 어댑터는 해당 전문을 예외 처리한다. 이후에 온 요청 전문은 무시된다. Client 타입에서만 설정 가능하다. (기본값: 30초)
연결 타임아웃 (ms) *	Tmax 서버에 커넥션을 얻는 데 걸리는 최대 시간을 지정한다. Client 타입에서만 설정 가능하다.
최대 연결수 *	Tmax의 JEUS Gateway와 엔드포인트 간의 최대 연결수이다.  Tmax 서버 설정 파일 항목의 MAX_CLH * CPC 값보다 큰 값으로 설정하는 것이 바람직하다.
aliveCheck 기간 (ms)	Tmax 서버로 Ping 메시지를 보내는 주기이다. aliveCheck를 사용할 경우 aliveCheck 기간과 aliveCheck 타임아웃을 모두 설정해야 한다. 기본값은 없고 설정할 경우에만 동작한다. Client 타입에서만 설정 가능하다.
aliveCheck 타임아웃 (ms)	해당 커넥션에서 'Alive Timeout'에 설정된 시간까지 응답 전문이 다시 오지 않으면, Ping 메시지를 보낸다.  보낸 Ping 메시지에 대해 (AliveTimeout * 3)의 시간까지 응답이 없으면 해당 커넥션은 끊긴 것으로 판단한다.  기본값은 없고 설정할 경우에만 동작한다. Client 타입에서만 설정 가능하다.

• [상세설정] 탭

연결정보

상세설정

메시지 핸들러

🔍 핸들러

거래그룹/거래

com.tmax.ANL001.tmaxTutorial.txCrp\_Tmax\_01.tx\_Tmax\_01

🔍 거래그룹/거래 선택

엔드포인트 등록 화면 - [상세설정]

항목	설명
메시지 핸들러	<b>[핸들러]</b> 버튼을 클릭해서 <b>메시지 핸들러 선택 화면</b> 에서 메시지를 선택한다.
거래그룹/거래	메시지를 수신할 때 엔드포인트가 받아서 처리해야 할 메시지에 대한 거래를 리스트에서 선택한다.

등록할 핸들러를 선택한 후 **[선택]** 버튼을 클릭한다.



항목	설명
연결 방식	엔드포인트가 동작하는 연결 방식을 설정한다.  ◦ Server : 엔드포인트가 서버 모드로 동작한다. ◦ Client : 엔드포인트가 클라이언트 모드로 동작한다.
스레드 최대값 *	최대 작업 스레드(work thread) 개수를 지정한다.  아웃바운드 어댑터의 경우 큐를 통해 전달된 메시지를 각 어댑터에 할당된 작업 스레드가 처리하게 된다. 따라서 작업 스레드 개수는 동시에 처리되는 메시지의 수를 의미한다. 하지만 최대 스레드 수가 너무 크면 잦은 스레드 컨텍스트 교환에 의한 오버헤드가 증가하므로 적절한 값을 설정해야 한다.
기본 인코딩	엔드포인트에서 사용할 기본 인코딩을 설정한다.  메시지에 설정된 인코딩이 우선 적용되고 메시지에 설정되지 않았을 경우에 적용된다.  메시지와 엔드포인트 모두 설정되지 않은 경우에는 시스템 기본 인코딩으로 적용된다.

#### ◦ 주소

항목	설명
IP/호스트 *	연결할 IP/호스트를 설정한다.
포트 *	연결할 포트 번호를 설정한다.

#### ◦ 타임아웃/연결수

항목	설명
Read 타임아웃 (ms) *	Tmax 서버에 요청 전문을 보내고 응답을 대기하는 최대 시간을 지정한다. 이 시간을 넘어서 응답 전문이 오지 않으면 어댑터는 해당 전문을 예외 처리한다. 이후에 온 요청 전문은 무시된다. Client 타입에서만 설정 가능하다. (기본값: 30초)
연결 타임아웃 (ms) *	Tmax 서버에 커넥션을 얻는 데 걸리는 최대 시간을 지정한다. Client 타입에서만 설정 가능하다.
최대 연결수 *	Tmax의 JEUS Gateway와 엔드포인트 간의 최대 연결수이다.  Tmax 서버 설정 파일 항목의 MAX_CLH * CPC 값보다 큰 값으로 설정하는 것이 바람직하다.
aliveCheck 기간 (ms)	Tmax 서버로 Ping 메시지를 보내는 주기이다. aliveCheck를 사용할 경우 aliveCheck 기간과 aliveCheck 타임아웃을 모두 설정해야 한다. 기본값은 없고 설정할 경우에만 동작한다. Client 타입에서만 설정 가능하다.

항목	설명
aliveCheck 타임아웃 (ms)	<p>해당 커넥션에서 'Alive Timeout'에 설정된 시간까지 응답 전문이 다시 오지 않으면, Ping 메시지를 보낸다.</p> <p>보낸 Ping 메시지에 대해 (AliveTimeout * 3)의 시간까지 응답이 없으면 해당 커넥션은 끊긴 것으로 판단한다.</p> <p>기본값은 없고 설정할 경우에만 동작한다. Client 타입에서만 설정 가능하다.</p>

#### ◦ Failover Server

항목	설명
IP/호스트	연결에 실패하는 경우 사용할 Failover Server의 IP/호스트를 설정한다.
포트	연결에 실패하는 경우 사용할 Failover Server의 포트를 설정한다.
Failback 기간 (ms)	<p>연결에 실패하는 경우 Failover Server로 Fail over한 뒤 다시 기본 서버로 Failback하는 경우 설정한다. 기간이 지나면 기본 서버로 접속을 시도한다. Failback 기능을 사용하기 위해서는 Failback 기간과 Failback 타임아웃을 모두 설정해야만 한다.</p> <p>기본값은 없고 설정할 경우에만 동작한다.</p>
Failback 타임아웃 (ms)	<p>기본 서버로 접속을 시도하는 경우 타임아웃 시간을 설정한다.</p> <p>기본값은 없고 설정할 경우에만 동작한다.</p>

#### • [상세설정] 탭

[상세설정] 탭 화면에 대한 내용은 [서버 모드 엔드포인트](#)를 참고한다.

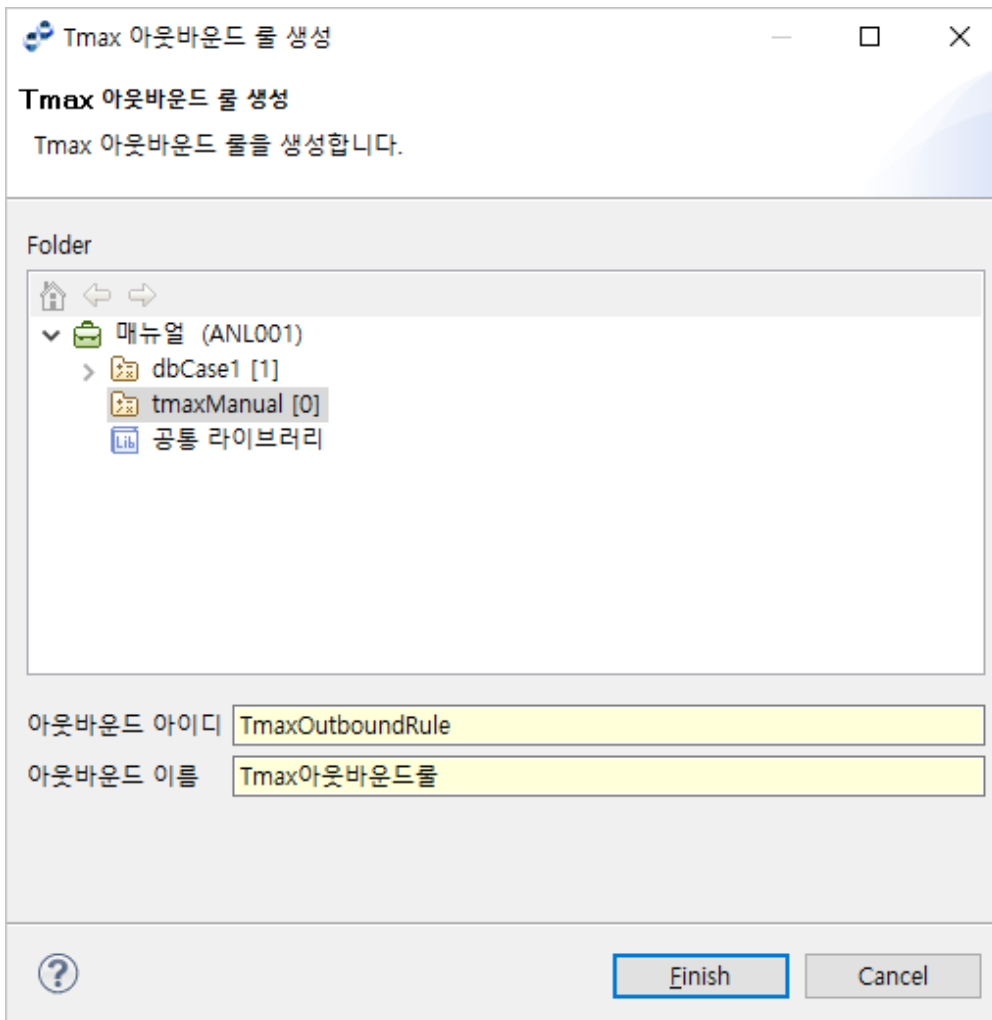
## 2.3. 아웃바운드 룰 설정

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



AnyLink 스튜디오의 자세한 사용법은 "AnyLink 스튜디오 안내서"를 참고한다.

AnyLink 스튜디오 거래그룹 네비게이터의 **거래/거래그룹** 항목의 컨텍스트 메뉴에서 **[새로만들기] > [아웃바운드 룰] > [Tmax 아웃바운드 룰]**를 선택하면 다음과 같이 **Tmax 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.

## 아웃바운드 룰 정의

- 프로토콜 TMAX
- 아웃바운드 룰 ID \* TmaxOutboundRule
- 아웃바운드 룰 이름 \* Tmax아웃바운드룰
- 요청처리 타임아웃(ms) : 10,000
- Endpoint \* (Group)
- 설명
- Tmax 응답 유무
- Tmax XA 트랜잭션
- Tmax 서비스 타임아웃(ms)
- Tmax 서비스 ID
- ByPass ☐

## 요청 메시지

이름	메시지 ID	타입 ID	배열 선택




## 응답 메시지

이름	메시지 ID	타입 ID	배열 선택




## 아웃바운드 룰 설정 화면

## • 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.
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. 서버에 접속한 상태여야만 선택이 가능하다.
설명	아웃바운드 룰에 대한 설명을 입력한다.
Tmax 응답 유무	Tmax 서비스를 호출한 후 Tmax에서 응답이 들어오는 경우 선택한다.
Tmax XA 트랜잭션	Tmax 서비스를 호출할 때 XA 트랜잭션 사용 유무를 선택한다.



항목	설명
Tmax 서비스 타임아웃 (ms)	<p>Tmax 서비스를 호출할 때 서비스 타임아웃을 설정한다. 요청처리 타임아웃 시간보다는 작은 시간을 사용할 것을 권장한다.</p> <p>-1, 0, 양의 정수 값을 설정할 수 있다. 아웃바운드 룰을 생성할 때 -1으로 설정되어 있다.</p> <ul style="list-style-type: none"> <li>• -1 : 엔드포인트 설정을 따른다.</li> <li>• 0 : 무한 대기한다.</li> <li>• 양의 정수 : 해당 설정 값만큼 대기한다.</li> </ul>
Tmax 서비스 ID	호출할 Tmax 서비스 아이디를 입력한다.

#### • 아웃바운드 룰 메시지 정의

아웃바운드 룰에서는 요청 메시지, 정상 응답 메시지를 설정할 수 있다.

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

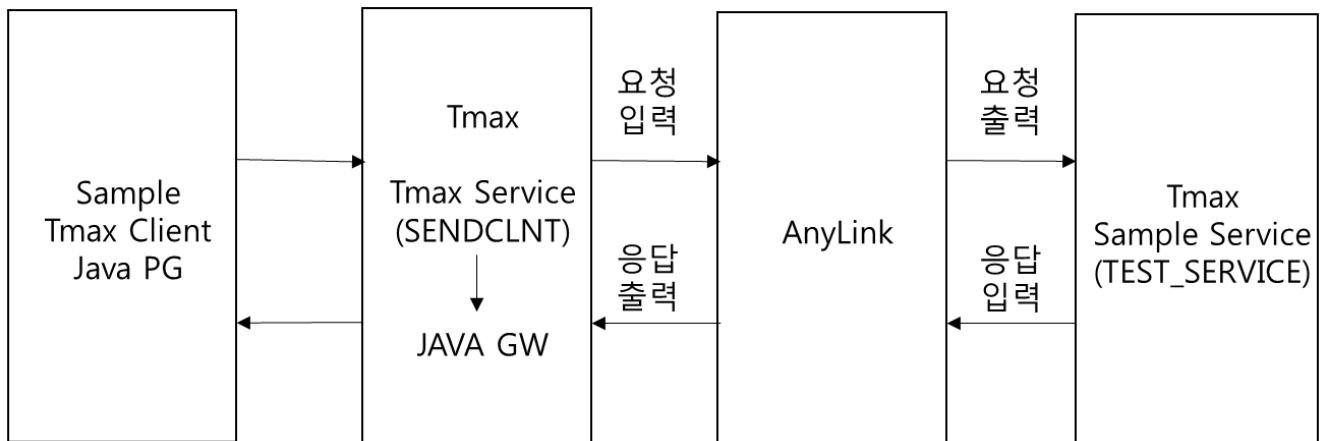
## 3. 예제

본 장에서는 AnyLink Tmax 어댑터를 사용한 예제를 설명한다.

### 3.1. 개요

사용된 전문은 모두 WebtStringBuffer 방식을 기본으로 한다. Tmax 및 WebT에 관한 자세한 내용은 Tmax 안내서를 참고한다.

다음은 Tmax 기본 거래 구성이다.



기본 거래 구성도

다음은 거래 순서를 수행하는 과정에 대한 설명이다.

1. Sample Tmax Client Java 프로그램을 통해 요청 입력 전문을 Tmax의 SENDCLNT 서비스로 전송한다.
2. Tmax에서 전문 수신 후 JAVA GW를 통해 AnyLink로 전문을 전송한다.
3. 인바운드 어댑터로 전문 수신 후 서비스 플로우를 수행한다.
4. 서비스 플로우에서 요청 입력 전문을 요청 출력 전문으로 매핑한다.
5. 아웃바운드 어댑터를 통해 요청 출력 전문을 Tmax Server의 TEST\_SERVICE 서비스를 요청한다.
6. Tmax Server에서 아웃바운드 어댑터로 응답 입력 전문을 전송한다.
7. 서비스 플로우에서 응답 입력 전문을 응답 출력 전문으로 매핑한다.
8. 인바운드 어댑터로 응답 출력 전문을 Tmax Java GW로 전송한다.
9. Tmax JAVA GW는 응답 출력 전문을 Sample Tmax Client로 전송한다.

#### 전문 구성

다음은 예제에서 사용할 전문 구성이다.

- 요청 입력
  - 헤더 전문

필드명	설명	타입	크기
kind_code	종별코드	Char	4
tx_code	거래코드	Char	4
date	날짜	Char	8
id	아이디	Char	10

◦ 바디 전문

필드명	설명	타입	크기
branch_number	지점번호	Number	3
teller_number	담당자번호	Char	5
name	이름	Char	10
account	계정	Char	12
data	데이터	Char	50

• 요청 출력

◦ 헤더 전문

필드명	설명	타입	크기
tx_code	거래코드	Char	4
id	아이디	Char	10

◦ 바디 전문

필드명	설명	타입	크기
name	이름	Char	10
account	계정	Char	12
data	데이터	Char	50

• 응답 입력

◦ 헤더 전문

필드명	설명	타입	크기
return_code	응답 코드	Char	4
id	아이디	Char	10

◦ 바디 전문

필드명	설명	타입	크기
transactionDate	응답일자	Char	8
transferType	관리정보	Char	4

#### • 응답 출력

##### ◦ 헤더 전문

필드명	설명	타입	크기
kind_code	종별코드	Char	4
tx_code	거래코드	Char	4
date	날짜	Char	8
id	아이디	Char	10

##### ◦ 바디 전문

필드명	설명	타입	크기
return_code	응답코드	Char	4
transferType	관리정보	Char	4

## 3.2. 어댑터 생성

AnyLink WebAdmin에 로그인한 후 초기 화면에서 **[구성관리] > [어댑터]**를 선택한 후 **어댑터 목록 화면**의 **[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. 인바운드 어댑터 생성

다음은 Tmax 인바운드 어댑터를 생성하는 예제이다.

#### 기본정보

어댑터 아이디*	TMAX_IN_ADT	어댑터 이름*	TMAX_IN_ADT	어댑터 종류*	Tmax
설명	TMAX 인바운드 어댑터			업무시스템 이름*	ANL001
배포 사유					

#### 상세설정

아웃바운드 스프레드 폴 아이디 미사용

### 인바운드 어댑터 등록 화면

#### • 기본정보

항목	설정
어댑터 아이디	TMAX_IN_ADT
어댑터 이름	TMAX_IN_ADT
어댑터 종류	Tmax
설명	TMAX 인바운드 어댑터
업무시스템	ANL001

#### • [상세설정] 탭

항목	설정
아웃바운드 스프레드 폴 아이디	사용 안함

### 3.2.2. 아웃바운드 어댑터 생성

다음은 Tmax 아웃바운드 어댑터를 생성하는 예제이다.

#### 기본정보

어댑터 아이디*	TMAX_OUT_ADT	어댑터 이름*	TMAX_OUT_ADT	어댑터 종류*	Tmax
설명	TMAX 아웃바운드 어댑터			업무시스템 이름*	ANL001
배포 사유					

#### 상세설정

아웃바운드 스프레드 폴 아이디 미사용

### 아웃바운드 어댑터 등록 화면

#### • 기본정보

항목	설정
어댑터 아이디	TMAX_OUT_ADT
어댑터 이름	TMAX_OUT_ADT
어댑터 종류	Tmax
설명	TMAX 아웃바운드 어댑터
업무시스템	ANL001

• [상세설정] 탭

항목	설정
아웃바운드 스레드 풀 아이디	사용 안함

### 3.3. 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.

#### 3.3.1. 인바운드 엔드포인트 생성

다음은 Tmax 인바운드 엔드포인트를 등록하는 예제이다.

• 기본정보

기본정보					
업무시스템 이름	ANL001	어댑터 종류	Tmax	어댑터 이름	TMAX_IN_ADT
엔드포인트 아이디*	TMAX_IN_EP	엔드포인트 이름*	TMAX_IN_EP	엔드포인트 그룹 이름	
엔드포인트 상태	Running	부팅 초기값	Running	통신 방향	Inbound
설명	Tmax 인바운드 엔드포인트				
배포 사유					

인바운드 엔드포인트 등록 화면 - 기본정보

항목	설정
엔드포인트 아이디	TMAX_IN_EP
엔드포인트 이름	TMAX_IN_EP
엔드포인트 상태	Running
부팅 초기값	Running

항목	설정
통신 방향	Inbound
설명	Tmax 인바운드 엔드포인트

#### • [연결정보] 탭

연결정보
상세설정

연결방식

연결방식
☒ Server
☐ Client

스레드 최대값\*
기본 인코딩

주소

IP/포트 (\*)
포트\*

타임아웃/연결수

최대 연결 수\*

인바운드 엔드포인트 등록 화면 - [연결정보]

항목	설정
연결 방식	Server
스레드 최대값	10
포트	31000
최대 연결 수	10

#### • [상세설정] 탭

연결정보
상세설정

메시지 핸들러
🔍 핸들러

거래그룹/거래
🔍 거래그룹/거래 선택

인바운드 엔드포인트 등록 화면 - [상세설정]

항목	설정
거래그룹/거래	manual.tmax.tmaxCase1.tx01 (거래그룹/거래 항목은 직접 입력해도 되고 차후 거래그룹/거래 생성 후 선택하여도 무방하다.)

### 3.3.2. 아웃바운드 엔드포인트 생성

다음은 Tmax 아웃바운드 엔드포인트를 등록하는 예제이다. [상세설정] 탭은 설정하지 않는다.

## • 기본정보

### 기본정보

업무시스템	IFC	어댑터 종류	Tmax	어댑터 이름	TMAX_OUT_ADT
엔드포인트 아이디*	TMAX_OUT_EP	엔드포인트 이름*	TMAX_OUT_EP	엔드포인트 그룹 이름	
엔드포인트 상태	Running	부팅 초기값	Running	통신 방향	Outbound
설명	Tmax 아웃바운드 엔드포인트				

### 아웃바운드 엔드포인트 등록 화면 - 기본정보

항목	설정
엔드포인트 아이디	TMAX_OUT_EP
엔드포인트 이름	TMAX_OUT_EP
엔드포인트 상태	Running
부팅 초기값	Running
통신 방향	Outbound
설명	Tmax 아웃바운드 엔드포인트

## • [연결정보] 탭

연결정보
상세설정

#### 연결방식

연결방식
☐ Server
☒ Client

스레드 최대값\*
기본 인코딩

#### 주소

IP/호스트 (\*)
포트\*

#### 타임아웃/연결수

Read 타임아웃\*
 ms
연결 타임아웃\*
 ms

최대 연결 수\*

aliveCheck 기간
 ms
aliveCheck 타임아웃
 ms

#### Failover Server

IP/호스트
포트

Failback 기간
 ms
Failback 타임아웃
 ms

### 아웃바운드 엔드포인트 등록 화면 - [연결정보]

항목	설정
연결 방식	Client
스레드 최대값	10



항목	설정
IP/호스트	192.168.15.75 (Sample Tmax 서버가 구동되어 있는 장비의 장비의 IP를 지정한다.)
포트	7282
Read 타임아웃	30000
연결 타임아웃	3000
최대 연결 수	10

## 3.4. 스튜디오 리소스 생성

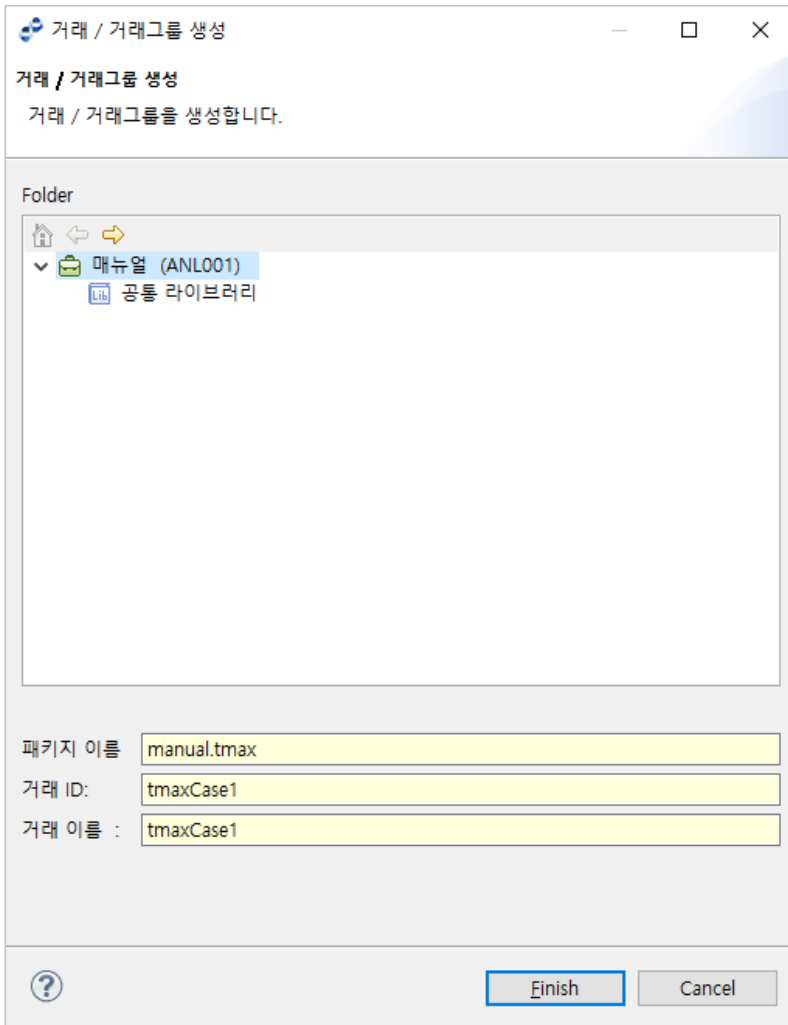
본 절에서는 스튜디오에서 리소스를 생성하는 방법에 대해서 설명한다.



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

### 3.4.1. 거래그룹 생성

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

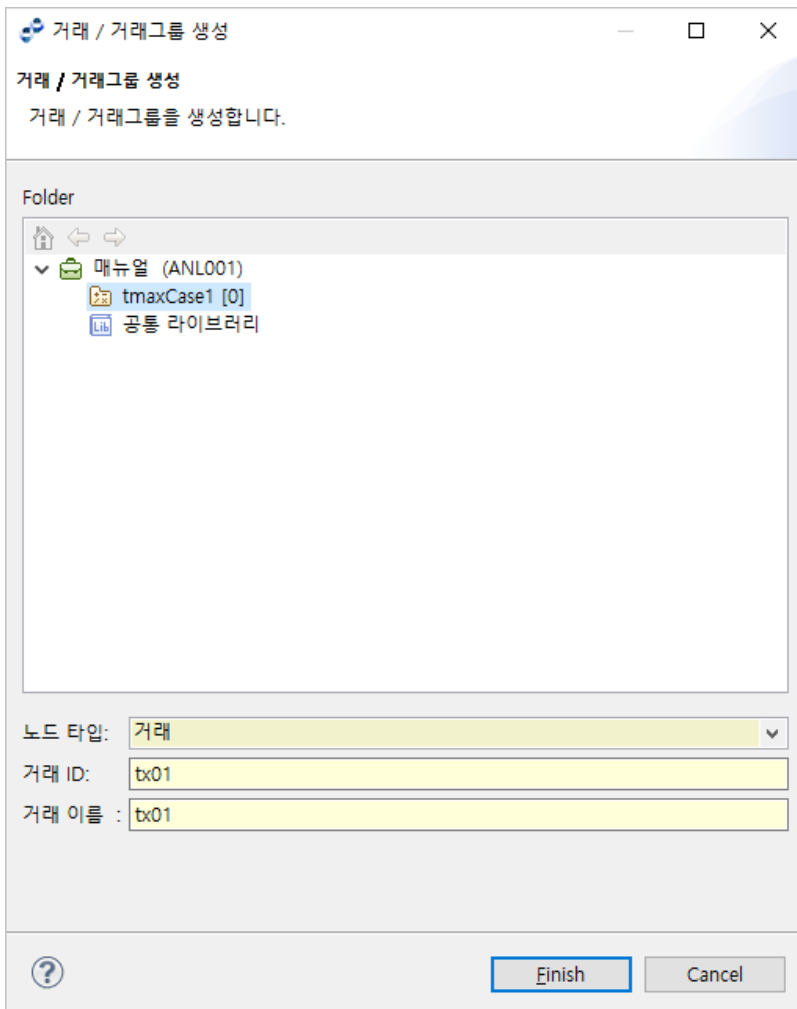


거래그룹 생성

항목	설정
패키지 이름	manual.tmax
거래 ID	tmaxCase1
거래 이름	tmaxCase1

### 3.4.2. 거래 생성

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



거래 생성

항목	설정
노드 타입	거래
거래 ID	tx01
거래 이름	tx01

### 3.4.3. 메시지 생성

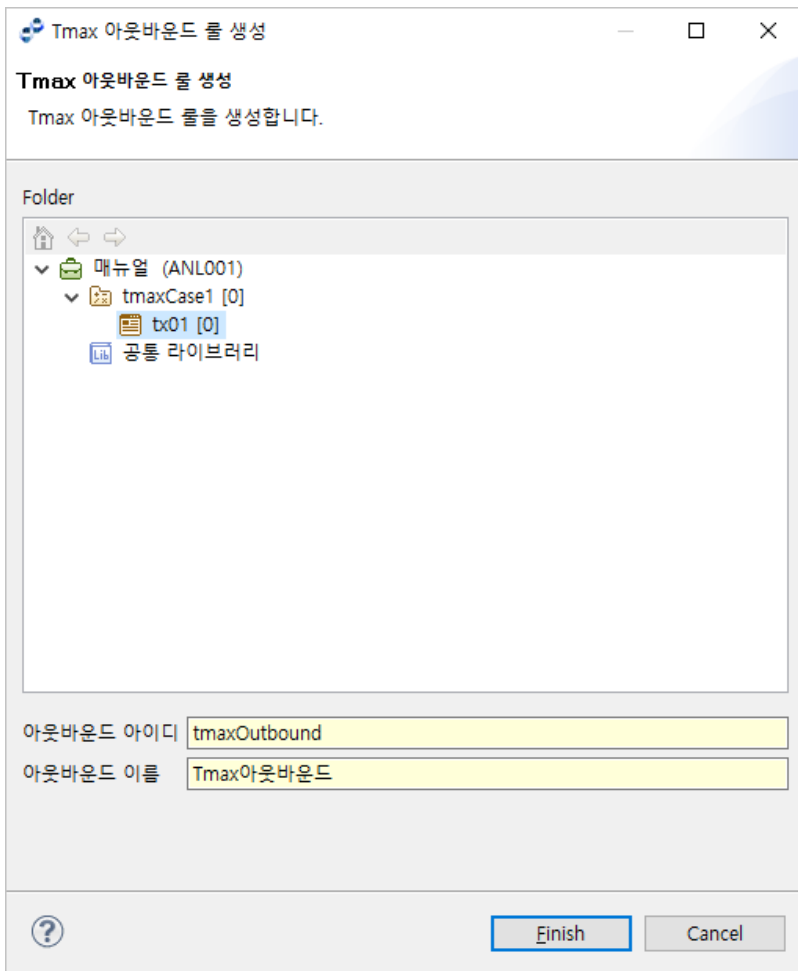
요청 입력 헤더/바디, 요청 출력 헤더/바디, 응답 입력 헤더/바디, 응답 출력 헤더/바디 메시지를 생성한다.



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

### 3.4.4. 아웃바운드 룰 생성

거래그룹 네비게이터에서 **거래**를 선택한 다음 컨텍스트 메뉴에서 **[새로만들기] > [아웃바운드 룰] > [Tmax 아웃바운드 룰]**을 선택한다.

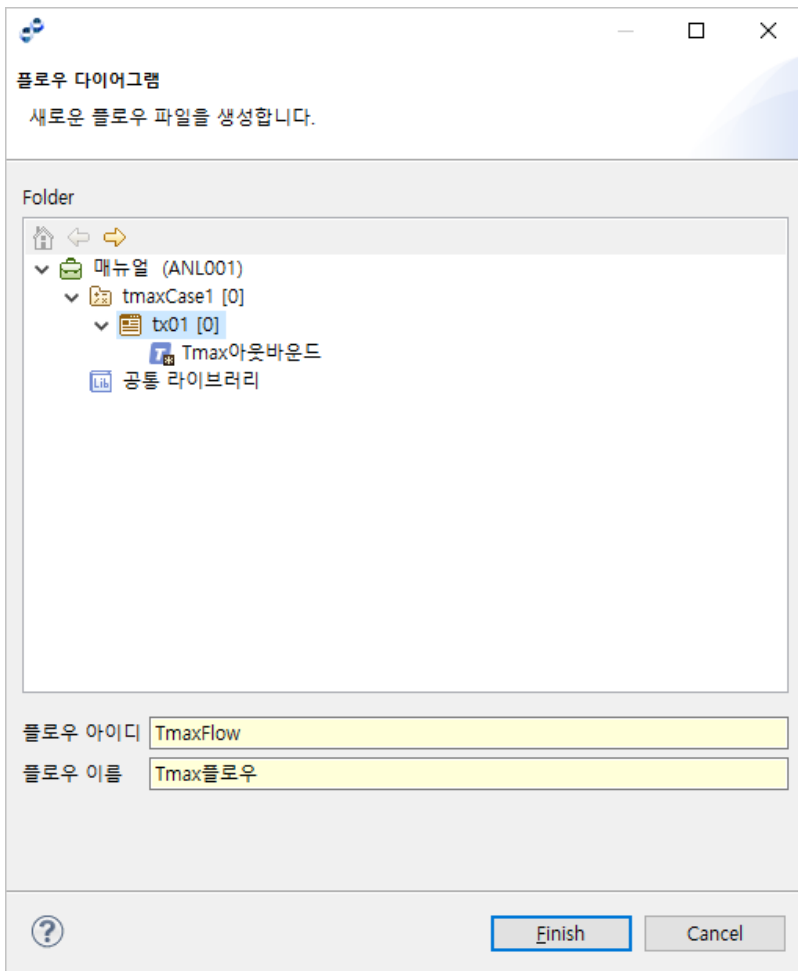


아웃바운드 룰 생성

항목	설정
아웃바운드 아이디	tmaxOutbound
아웃바운드 이름	Tmax아웃바운드

### 3.4.5. 플로우 생성

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



플로우 룰 생성

항목	설정
플로우 아이디	TmaxFlow
플로우 이름	Tmax플로우

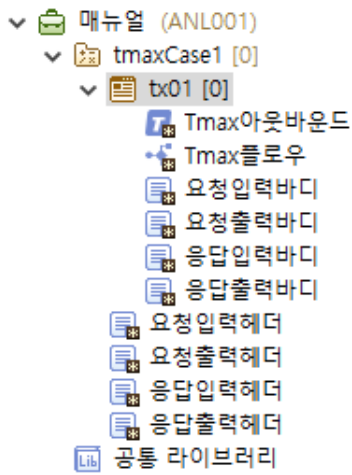
### 3.5. 스튜디오 리소스 설정

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

### 3.5.1. 거래그룹 설정

본 절에서는 거래그룹을 설정하는 예제를 설명한다. **[파싱 옵션]** 탭은 설정하지 않는다.

#### • [거래그룹 정보] 탭

기본정보

거래그룹 ID

tmaxCase1

패키지 이름

manual.tmax

거래그룹 이름

tmaxCase1

거래 타입

NONE

버전

0

설명

Bypass 설정

NO

XA 설정

NO

요청 메시지

이름	메시지 ID	타입 ID	배열 선택
요청입력헤더	ReqInHeader	Default Type	설정 안함

네임스페이스 URI

로컬 파트

그룹

그룹번호

정상 응답 메시지

이름	메시지 ID	타입 ID	배열 선택
응답출력헤더	ResOutHeader	Default Type	설정 안함

네임스페이스 URI

로컬 파트

그룹

그룹번호

업무오류 응답 메시지

이름	메시지 ID	타입 ID	배열 선택
----	--------	-------	-------

네임스페이스 URI

로컬 파트

그룹

그룹번호

오류 응답처리

오류 응답 방식

NONE

거래그룹 설정 - [거래그룹 정보]

항목	설정
요청 메시지	요청입력헤더
정상 응답 메시지	응답출력헤더
Bypass 설정	NO
XA 설정	NO
오류 응답 방식	NONE

#### • [거래그룹 옵션] 탭

#### 거래그룹 옵션 정의

• 거래 플로우 타임아웃(ms) 60,000

• 외부 GUID 사용 NO  
• GUID 타입 STRING

• 응답 유무 REQUEST\_RESPONSE

• 전달 보장 설정  
• 전달 보장  
• 큐 ID  
• 개별 설정  
• 재시도 간격(s)  
• 최대 재시도 횟수  
• 만료 시간(s)

• 중복거래 확인 여부 NO

- 중복거래 확인 필드

메시지 ID	필드 ID

• 중복거래 보관 주기

• 거래 우선순위 MEDIUM

#### 거래그룹 설정 - [거래그룹 옵션]

항목	설정
거래 플로우 타임아웃(ms)	60000
응답 유무	REQUEST_RESPONSE
거래 우선순위	MEDIUM

#### • [파싱정보] 탭

##### 파싱 정보 정의

거래 식별코드

코드값	종류

하위 거래 식별 방법 MESSAGE

거래 식별코드 매핑

메시지 ID ReqInHeader.umsg
필드 ID tx\_code

값	거래 식별코드

#### 거래그룹 설정 - [파싱 정보]

항목	설정
하위 거래 식별 방법	MESSAGE
메시지 ID	ReqInHeader.umsg
필드 ID	tx_code

### 3.5.2. 거래 설정

다음은 거래를 설정하는 예제이다. [파싱 옵션] 탭은 설정하지 않는다.

#### • [거래정보] 탭

기본정보

- 거래 ID: tx01
- 파라미터 이름: manual.tmax.tmaxCase1
- 거래 이름: tx01
- 버전: 0
- 거래 타입: PARENT
- Bypass 설정: NO
- XA 설정: NO
- 설명

호출 서비스

- 비동기 응답 거래: ☐
- 서비스 타입: FLOW
- 서비스 이름:
- 매핑 설정: ☐

요청 메시지

이름	메시지 ID	타입 ID	배열 선택
요청입력바디	ReqInBody	Default Type	설정 안함

- 네임스페이스 URI
- 포맷 파츠
- 그룹: NONE
- 그룹번호

정상 응답 메시지

이름	메시지 ID	타입 ID	배열 선택
응답출력바디	ResOutBody	Default Type	설정 안함

- 네임스페이스 URI
- 포맷 파츠
- 그룹: NONE
- 그룹번호

업무오류 응답 메시지

이름	메시지 ID	타입 ID	배열 선택
----	--------	-------	-------

- 네임스페이스 URI
- 포맷 파츠
- 그룹: NONE
- 그룹번호

오류 응답처리

- 오류 응답 방식: PARENT

## 거래 설정 - [거래정보]

항목	설정
요청 메시지	요청입력바디
정상 응답 메시지	응답출력바디
Bypass 설정	NO
XA 설정	NO
오류 응답 방식	PARENT
서비스 타입	FLOW
서비스 이름	TmaxFlow_Message_STRT_Event_697118(랜덤으로 생성됨)

## • [거래 옵션] 탭



거래 옵션 정의

- 거래 플로우 타임아웃(ms)
- 타임아웃 설정 안함 ☒

- 외부 GUID 사용
- GUID 타입

- 응답 유무

- 전달 보장 설정 ☐
- 전달 보장
- 큐 ID
- 개별 설정 ☐
  - 재시도 간격(s)
  - 최대 재시도 횟수
  - 만료 시간(s)

- 중복거래 확인 여부 
  - 중복거래 확인 필드

메시지 ID	필드 ID

+ 추가 - 삭제
- 중복거래 보관 주기
- 거래 우선순위

거래 설정 - [거래 옵션]

항목	설정
거래 플로우 타임아웃(ms)	미설정
타임아웃 설정 안함	체크
응답 유무	PARENT
거래 우선순위	MEDIUM

• [파싱 정보] 탭

파싱 정보 정의

거래 식별코드

코드값	종류
[A001]	REQUEST

+ 추가 - 삭제

거래 설정 - [파싱 정보]

◦ 거래 식별코드

항목	설정
코드값	A001
종류	REQUEST

### 3.5.3. 아웃바운드 룰 설정

다음은 Tmax 아웃바운드 룰 설정 예제이다.

TMAX 아웃바운드 룰

아웃바운드 룰 정의

- 프로토콜: TMAX
- 아웃바운드 룰 ID \*: tmaxOutbound
- 아웃바운드 룰 이름 \*: Tmax아웃바운드
- 요청처리 타임아웃(ms): 10,000
- Endpoint \*: TMAX\_OUT\_EP

Q 검색

설정

- Tmax 응답 유무: true
- Tmax XA 트랜잭션: false
- Tmax 서비스 타임아웃(ms): 5000
- Tmax 서비스 ID: TEST\_SERVICE
- ByPass: ☐

요청 메시지

이름	메시지 ID	타입 ID	배열 선택
요청출력헤더	ReqOutHeader	Default Type	설정 안함
요청출력바디	ReqOutBody	Default Type	설정 안함

가져오기

+

추가

-

삭제

응답 메시지

이름	메시지 ID	타입 ID	배열 선택
응답입력헤더	ResInHeader	Default Type	설정 안함
응답입력바디	ResInBody	Default Type	설정 안함

가져오기

+

추가

-

삭제

아웃바운드 룰 설정

항목	설정
요청처리 타임아웃(ms)	10000
Endpoint	TMAX_OUT_EP
Tmax 응답 유무	true
Tmax XA 트랜잭션	false
Tmax 서비스 타임아웃	5000
Tmax 서비스 ID	TEST_SERVICE
요청 메시지	요청출력헤더(ReqOutHeader) 요청출력바디(ReqOutBody)
정상 응답 메시지	응답입력헤더(ResInHeader) 응답입력바디(ResInBody)

### 3.5.4. 플로우 설정

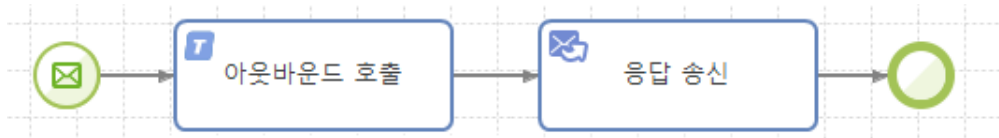
다음의 과정으로 플로우를 설정한다.

1. 플로우 그리기
2. 변수 설정
3. 메시지 이벤트 설정
4. 아웃바운드 호출 설정

## 5. 응답 호출 설정

### 플로우 그리기

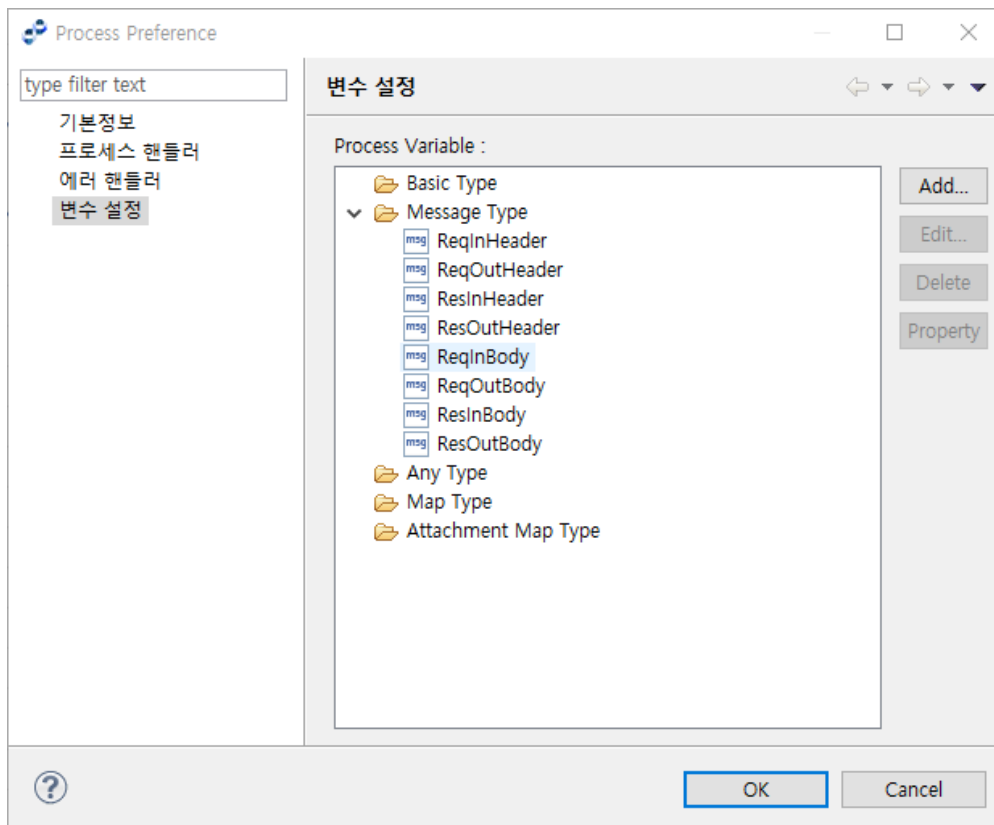
다음은 예제에 대한 서비스 플로우이다.



서비스 플로우 에디터 - 기본 그림

### 변수 설정

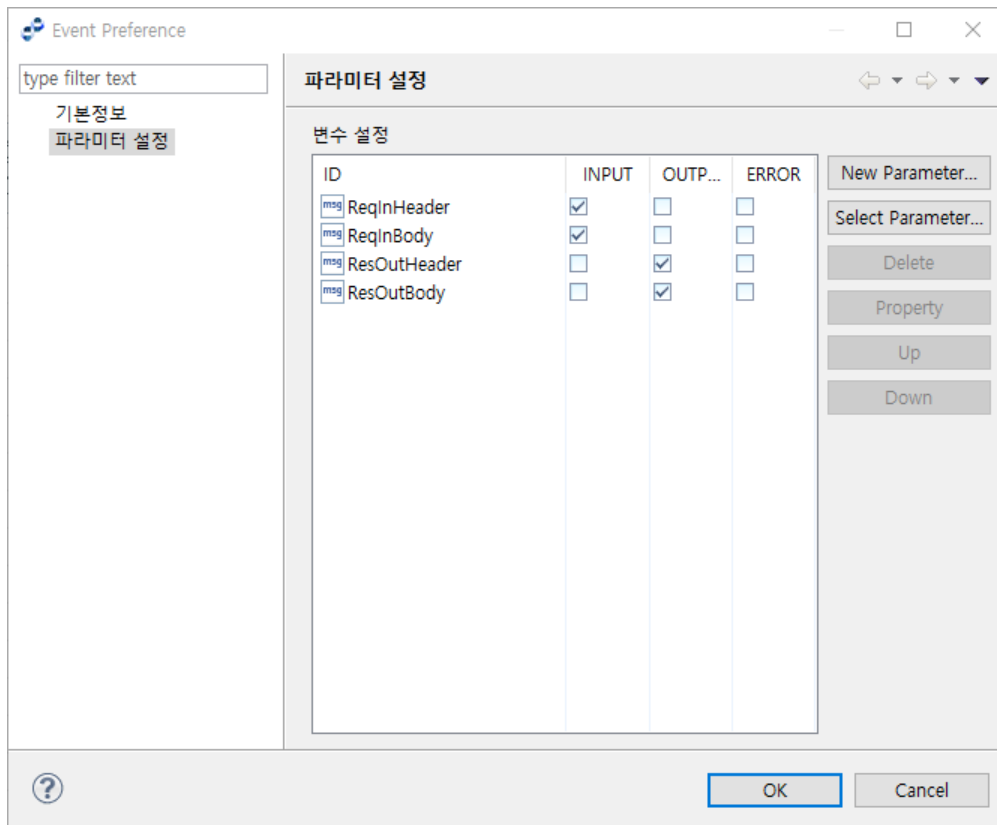
변수 설정을 하기 위해 플로우 에디터 바탕의 컨텍스트 메뉴에서 **[Property]**를 선택한다.



Property Preference - 변수 설정

### 메시지 이벤트 설정

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

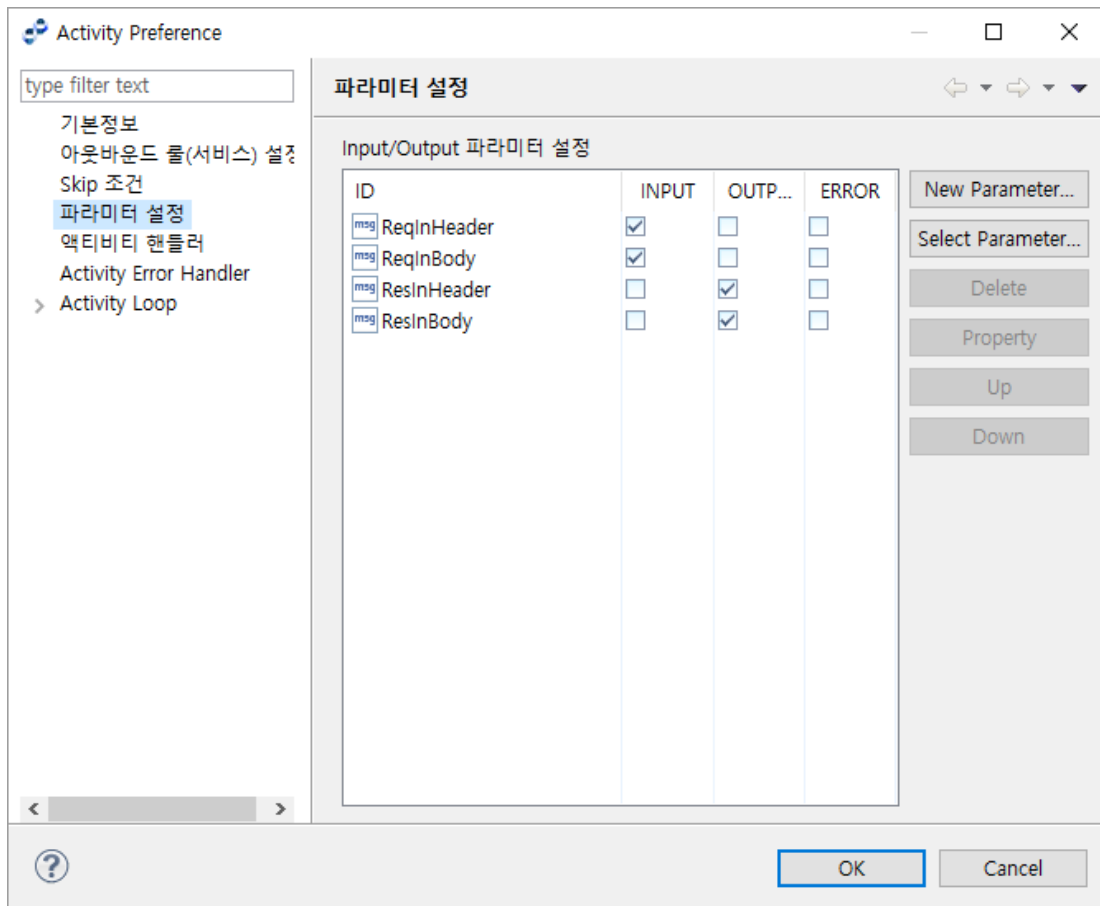


Event Preference - 파라미터 설정

## 아웃바운드 호출 설정

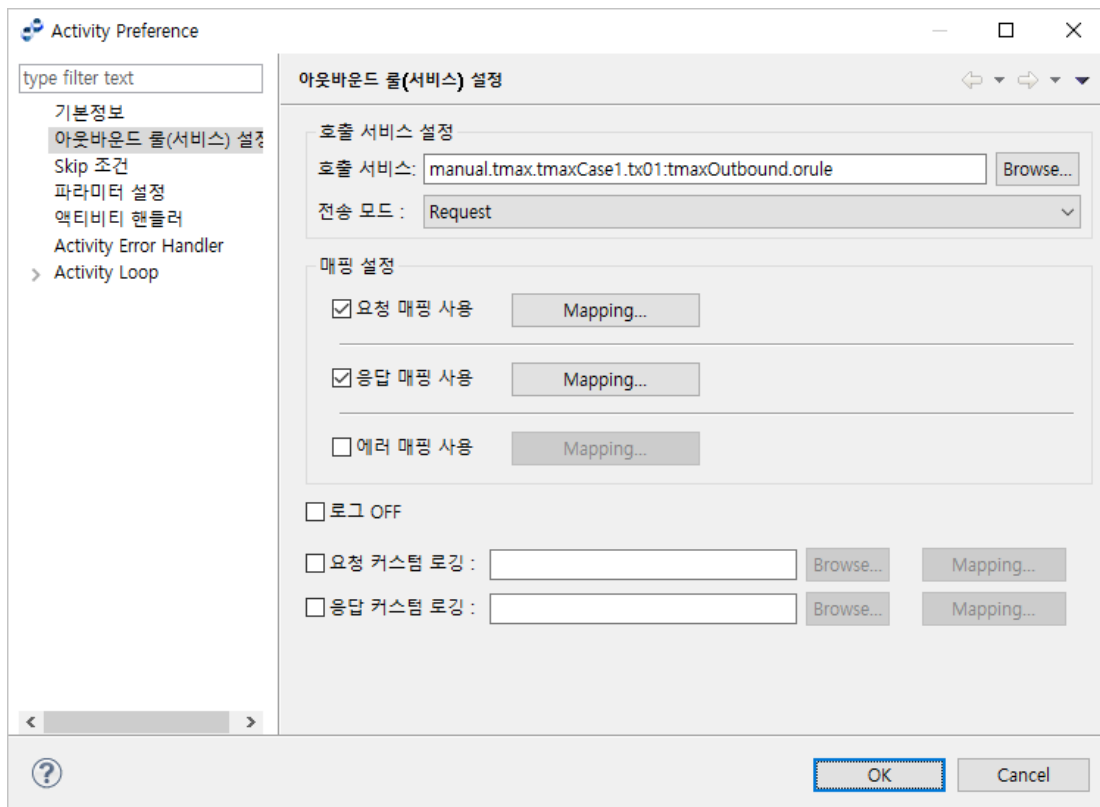
다음의 과정으로 아웃바운드 호출을 설정한다.

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



아웃바운드 호출 - 파라미터 설정

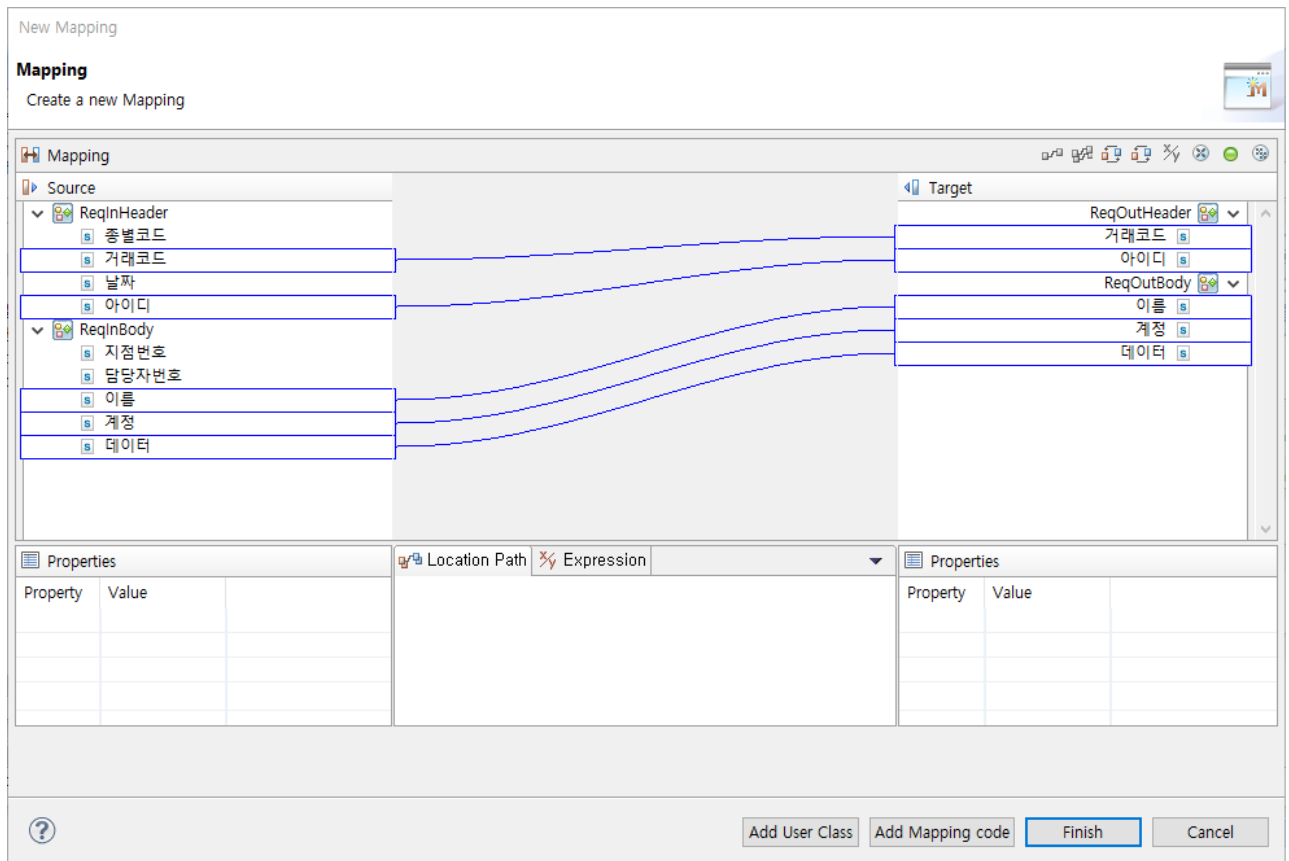
2. **Activity Preference** 화면에서 [아웃바운드 룰(서비스) 설정]을 선택한다.



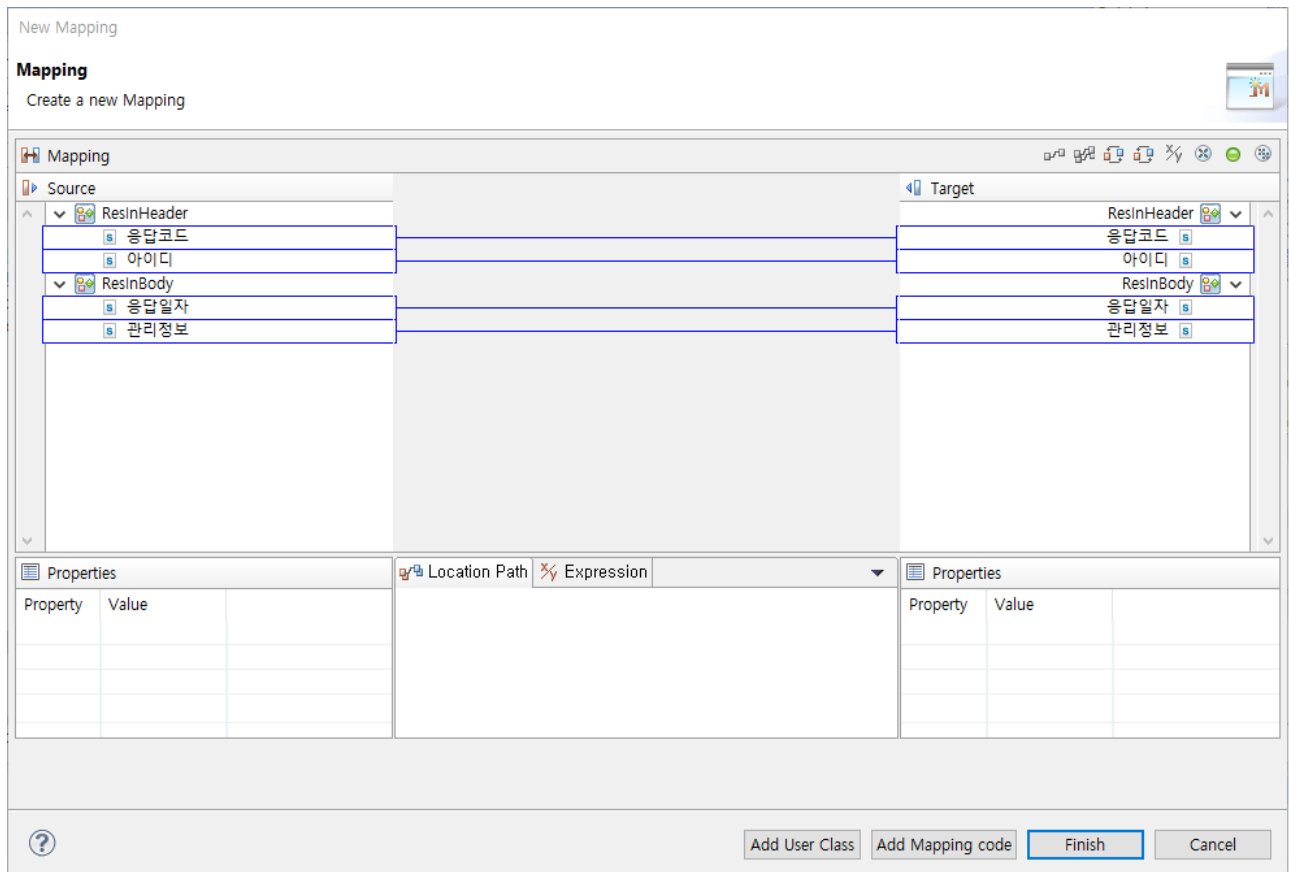
아웃바운드 호출 - 아웃바운드 룰(서비스) 설정

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

각 매핑정보를 설정한다.



아웃바운드 호출 - 아웃바운드 룰(서비스) 설정 - 요청 매핑

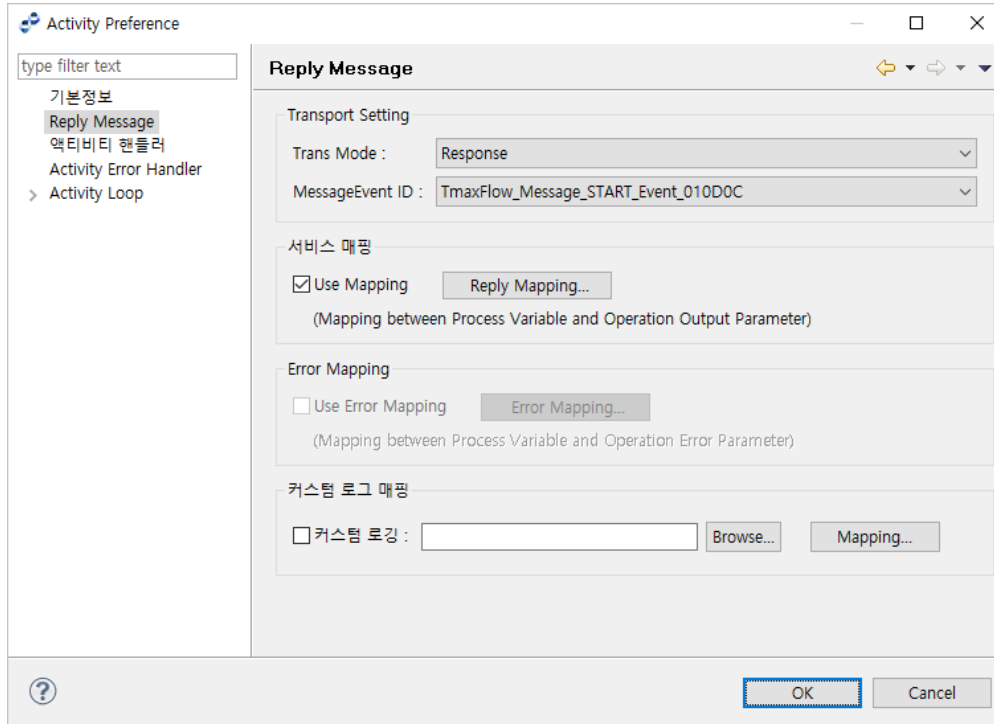


아웃바운드 호출 - 아웃바운드 룰(서비스) 설정 - 응답 매핑

## 응답 호출 설정

다음의 과정으로 응답 호출을 설정한다.

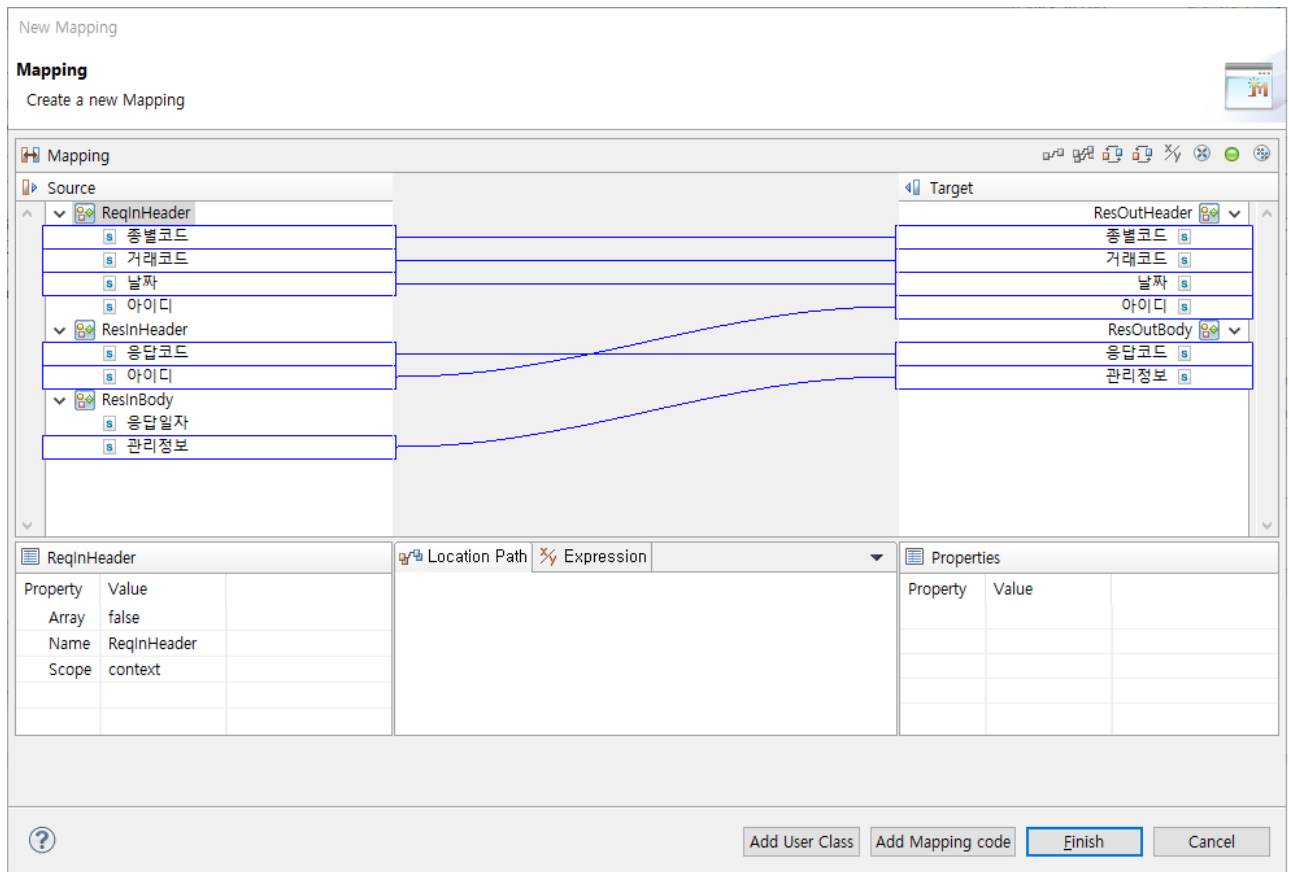
1. 플로우 에디터의 [응답 송신] > [Reply 메시지]의 컨텍스트 메뉴에서 [Property]를 선택한다. **Activity Preference 화면**에서 [Reply Message]를 선택한다.



응답 송신 - Reply Message

2. 'Use Mapping'을 체크하고 [Reply Mapping] 버튼을 클릭해서 각 매핑정보를 설정한다.

**Mapping 화면**의 Source 영역의 컨텍스트 메뉴에서 [Add source]를 선택하여 ReqInHeader, ResInHeader, ResInBody를 선택한후 매핑한다.



응답 송신 - Reply Message - 서비스 매핑 - Reply Mapping

## 3.6. 배포

거래그룹/거래 배포정보를 설정한다.



**배포 설정**

배포 옵션을 설정합니다.

☒ 하위 (그룹) 포함

배포 자원 목록

이름	ID	패키지	버전	타입	취소	다
▼ tmaxCase1	manual.tm...	manual.tmax.tmaxCase1	0	거래그를	<input type="checkbox"/>	
응답출력해더				메시지		
응답입력해더				메시지		
요청출력해더				메시지		
요청입력해더				메시지		
▼ tx01	manual.tm...	manual.tmax.tmaxCase1.tx01	0	거래	<input type="checkbox"/>	
Tmax아웃바운드				아웃바운...		
Tmax플로우				Flow		
응답출력바디				메시지		
응답입력바디				메시지		

☐ 버전 점검 안함

거래그룹 / 거래 배포



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

### 3.7. 거래 테스트

본 절에서는 거래 테스트를 위한 클라이언트 프로그램, Tmax 환경 설정 및 서버 프로그램과 테스트 수행하는 방법을 설명한다.

### 3.7.1. 클라이언트 프로그램

Tmax 클라이언트 프로그램을 사용하기 위해서는 Tmax 라이브러리(webt.jar , jclient.jar)가 필요하다.

<TmaxClient.java>

```
package manual.tmax;

import tmax.webt.WebtAttribute;
import tmax.webt.WebtBuffer;
import tmax.webt.WebtConnection;
import tmax.webt.io.WebtStringBuffer;

public class TmaxClient {
    String host;
    int port;
    String serviceName;
```

```

public TmaxClient(String host, int port, String serviceName) {
    this.host = host;
    this.port = port;
    this.serviceName = serviceName;
}

public String call(String inputMessage) {
    WebtBuffer sndBuffer = makeWebtBuffer(inputMessage);
    WebtConnection connection = new WebtConnection(host, port);
    WebtBuffer recvBuffer = connection.tpcall(sndBuffer, serviceName, WebtAttribute.TPNOFLAGS);
    return new String(recvBuffer.getBytes());
}

public WebtBuffer makeWebtBuffer(String message) {
    WebtBuffer webtBuffer = new WebtStringBuffer();
    webtBuffer.setString(message);
    return webtBuffer;
}

public static void main(String[] args) {
    try {
        TmaxClient client = new TmaxClient("192.168.1.87", 7282, "SENDCLNT");

        String sendMessage = "AAAAA00120160701KIMKILSU  B01P1034MOONYOU  KNIGHT
THISISSAMPLEDATA                                ";

        System.out.println("SEND:" + sendMessage);
        String recvMessage = client.call(sendMessage);
        System.out.println("RCV:" + recvMessage);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

### 3.7.2. Tmax 환경 설정 및 서버 프로그램

Tmax 환경 설정 및 프로그램 빌드에 관한 자세한 내용은 Tmax 안내서를 참고한다.

- Tmax JAVA GW 설정

```

*SERVICE
SENDCLNT      SVRNAME = jgw_t1toa1

*GATEWAY
jgw_t1toa1    GWTYPE=JEUS_ASYNC,
              NODENAME="tmaxi7",
              PORTNO=8801,
              RGWADDR="192.168.14.106",
              RGWPORTNO=31000,
              CLOPT = "-r"

```

- Tmax 서비스 설정

```
*SERVICE
TEST_SERVICE          SVRNAME = server_test
```

- Tmax 서비스 프로그램

<server\_test.c>

```
#include <usrinc/atmi.h>
#include <stdio.h>
#include <string.h>

TEST_SERVICE(TPSVCINFO *msg)
{
    char *rcvbuf;
    char *recvData;

    recvData = (char *)tpalloc("STRING", NULL, msg->len + 1);
    memcpy(recvData, msg->data, msg->len);

    printf(" Data recv = %s\n", recvData);
    printf(" Data Length = %d\n", msg->len);

    rcvbuf=(char *)tpalloc("STRING", NULL, 1024);
    if(rcvbuf==NULL) {
        printf("tpalloc failed (rcvbuf) : %s\n", tpstrerror(tperrno));
        tpreturn(TPFAIL, -1, NULL, 0, 0);
    }

    strcpy(rcvbuf, "1000KIMKILSU 20160720SP01");
    printf(" Data send = %s\n", rcvbuf);

    tpfree(recvData);

    tpreturn( TPSUCCESS, 0, rcvbuf, 0 , 0 );
}
```

### 3.7.3. 테스트 수행

- 클라이언트 결과 화면

```
SEND:AAAAA00120160701KIMKILSU B01P1034MOONYOU KNIGHT THISISSAMPLEDATA
[2016.08.03 19:42:46:819] (main)webt.properties file is not found
[2016.08.03 19:42:46:857] (main)[WEBT-1109][000000] start to connect server 192.168.1.87:7282 for
20000 msec
[2016.08.03 19:42:46:877] (main)[WEBT-1102][000000] connection established
socket(192.168.1.87:7282)
[2016.08.03 19:42:46:880] (main)[WEBT-1111][000000] socket list add
tmax.webt.io.WebtSocket@5901d85a
[2016.08.03 19:42:46:882] (main)[WEBT-1037][000000] buffer transmitted
[_tpstart(0x0):seq(0):msgtype(1):cd(1):errcode(0):rcode(0):asize(76):reserved(0x0):
magic(0x270f)][xid:00 00 00 00 : 00 00 00 00 : 00 00 00 00 : F
[2016.08.03 19:42:46:882] (main)[WEBT-1034][000000] receive start for 60000 msec
```

```
[2016.08.03 19:42:46:882] (main)[WEBT-1035][000000] buffer received [---(0x0):seq(0):
msgtype(1001):cd(8):errcode(0):rcode(16):asize(56):reserved(0x0):magic(0x270f)]
[xid:00 00 00 00 : 00 00 00 00 : 00 00 00 1e : F
[2016.08.03 19:42:46:882] (main)[WEBT-1041][000000] matching received.. seqno(0:0), type(1:1001)
[2016.08.03 19:42:46:883] (main)[WEBT-1042][000000] matching result.. seqno match = true, type
match = true
[2016.08.03 19:42:46:883] (main)[WEBT-1305][000000] set maximum dialoge session size 8
[2016.08.03 19:42:46:883] (main)[WEBT-1003][000000] tmax session established.. tmax version
[5.0.2.1]
[2016.08.03 19:42:46:883] (main)[WEBT-1005][000000] node count [1]
[2016.08.03 19:42:46:883] (main)[WEBT-1006][000000] rq count [0]
[2016.08.03 19:42:46:883] (main)[WEBT-1007][000000] topend product count [0]
[2016.08.03 19:42:46:884] (main)[WEBT-1056][000000] node registered [0:tmaxi7]
[2016.08.03 19:42:46:884] (main)[WEBT-1011][000000] compress threshold -1 byte
[2016.08.03 19:42:46:885] (main)[WEBT-1037][000000] buffer transmitted [SENDCLNT(0x0):
seq(1):msgtype(3):cd(0):errcode(0):rcode(0):asize(107):reserved(0x0):magic(0x270f)]
[xid:00 00 00 00 : 00 00 00 00 : 00 00 00 00 : F
[2016.08.03 19:42:46:885] (main)[WEBT-1034][000000] receive start for 30000 msec
[2016.08.03 19:42:46:935] (main)[WEBT-1035][000000] buffer received [---(0x0):seq(1):
msgtype(1003):cd(0):errcode(0):rcode(0):asize(1024):reserved(0x0):magic(0x270f)]
[xid:00 00 00 00 : 00 00 00 00 : 00 00 00 00 : F
[2016.08.03 19:42:46:935] (main)[WEBT-1041][000000] matching received..
seqno(1:1), type(3:1003)
[2016.08.03 19:42:46:936] (main)[WEBT-1042][000000] matching result..
seqno match = true, type match = true
RECV:AAAAA00120160701KIMKILSU 1000SP01
```

- 서버 결과 화면(Tmax ulog)

```
Data recv = A001KIMKILSU MOONYOU KNIGHT THISISSAMPLEDATA
Data Length = 86
Data send = 1000KIMKILSU 20160720SP01
```