

TCP 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. 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	23
2.3. 아웃바운드 룰 설정	28
3. 예제	32
3.1. 개요	32
3.2. 어댑터 생성	34
3.2.1. 인바운드 어댑터 생성	34
3.2.2. 아웃바운드 어댑터 생성	35
3.3. 엔드포인트 생성	36
3.3.1. 인바운드 엔드포인트 생성	36
3.3.2. 아웃바운드 엔드포인트 생성	38
3.4. 스튜디오 리소스 생성	40
3.4.1. 거래그룹 생성	40
3.4.2. 거래 생성	41
3.4.3. 메시지 생성	42
3.4.4. 아웃바운드 룰 생성	42
3.4.5. 플로우 생성	43
3.5. 스튜디오 리소스 설정	44
3.5.1. 거래그룹 설정	45
3.5.2. 거래 설정	47
3.5.3. 아웃바운드 룰 설정	49
3.5.4. 플로우 설정	50
3.6. 배포	55
3.7. 거래 테스트	56
3.7.1. 클라이언트 프로그램	56
3.7.2. 서버 프로그램	57
3.7.3. 테스트 수행	59

1. 소개

본 장에서는 AnyLink TCP 어댑터의 기본적인 동작방식에 대해서 설명한다.

1.1. 개요

TCP 어댑터는 프로토콜이 TCP/IP 기반으로 정의되어 있을 경우 사용한다. TCP 어댑터는 다양한 메시지 길이 지정 방식을 통해 고정 길이의 메시지뿐만 아니라 메시지의 특정 부분의 값에 따라 수신하는 메시지의 형태가 달라지거나 딜리미터를 사용하는 메시지일 경우에도 처리가 가능하다.

TCP 어댑터는 인바운드와 아웃바운드 모두 지원하고 엔드포인트에 설정된 통신 방향에 따라서 처리하는 동작 방식이 달라진다.

1.2. 어댑터 동작 방식

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

각각의 모드에 따른 동작 방식은 다음과 같다.

- 서버 모드

서버 모드는 선택된 포트로 연결이 들어 오기를 대기한다.

- 클라이언트 모드

클라이언트 모드는 지정된 IP/호스트, 포트로 연결을 한다. 연결 관리 여부에 따라 연결을 미리 생성할지 아니면 요청할 때 연결을 생성할지 결정된다.

서버/클라이언트 모드에 따라서 거래의 방향성이 정해지는 것은 아니다. 서버 모드라하여 거래가 항상 인바운드로 들어오는 것이 아니고 아웃바운드로 수행될 수도 있다. 마찬가지로 클라이언트 모드라고 하여 항상 아웃바운드로만 수행되는 것이 아니라 인바운드로도 수행될 수 있다.

TCP 엔드포인트를 설정할 때 통신 방향을 설정할 수 있다. 통신 방향은 해당 엔드포인트를 인바운드, 아웃바운드 또는 양방향으로 사용할 방식만 선택하고 동작 방식에는 영향을 주지 않는다. TCP 아웃바운드 룰에서 엔드포인트를 선택할 때 아웃바운드와 양방향만 선택할 수 있다.

2. 환경설정

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

2.1. 어댑터 설정

TCP 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

RUNNING

3

SHUTDOWN

에러로그

JB

JEUS ▶

admin (admin)

업무시스템리모트 에이전트어댑터어댑터 클러스터메시지 큐패지

search

이름아이디

엔드포인트 열기

ALL

ANL001

DB

EB_ML

FILE

FTP

HTTP

IMS

LOG

MO

SAP

SMTP

TCP

TMAX

TUXEDO

UDP

WEBDAV

WEB_SERVICE

클러스터001

전체

어댑터 목록

엔드포인트 그룹 목록

엔드포인트 목록

어댑터 목록

업무시스템 이름전체어댑터 종류전체어댑터 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

◀◀

1234567▶▶

페이지당 목록 수: 10

어댑터 목록 화면

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

기본정보

어댑터 아이디*

TCP_ADT

어댑터 이름*

TCP_ADT

어댑터 종류*

TCP

설명

TCP 어댑터

업무시스템 이름*

ANL001

배포 사유

상세설정

접속 제어

아웃바운드 스크드 플 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 TCP를 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
설명	생성하는 어댑터에 대한 설명을 입력한다. 입력하지 않아도 무관하다.

항목	설명
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.

• [접속 제어] 탭

항목	설명
IP Control	엔드포인트에 연결을 시도하는 Client를 제어한다. 어댑터에 지정한 설정은 하위 엔드포인트 그룹, 엔드포인트에서 사용할 수 있다. <ul style="list-style-type: none"> • OFF : 기본값으로, 모든 Client의 연결을 허용한다. • Allow : 지정한 IP Rule에 해당하는 Client의 연결만을 허용한다. • Deny : 지정한 IP Rule에 해당하는 Client의 연결을 차단한다.

상세설정
접속 제어

IP Control ?

IP Control
☐ OFF
☒ Allow
☐ Deny

Allow

+ 추가

No	IP Rule	삭제
1		- 삭제

어댑터 등록 화면 - [접속 제어]

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*

TCP_ADT

어댑터 이름*

TCP_ADT

어댑터 종류*

TCP

설명

TCP 어댑터

업무시스템 이름*

ANL001

배포 사유

상세설정

엔드포인트 그룹 목록

엔드포인트 목록

접속 제어

No

엔드포인트 아이디

엔드포인트 이름

통신 방향

버전

상태

최종 변경일자

담당자

+ Add

- Del

▶ Start

■ Stop

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], [메시지/장애처리], [메시지 변경 송신], [SSL], [TCP 소켓 옵션], [접속 제어] 탭으로 구성된다. [연결정보] 탭의 '연결방식' 항목에 따라서 각 탭의 구성이 달라진다. 각 탭의 자세한 설명은 해당 절의 설명을 참고한다.

기본정보

업무시스템 이름

ANL001

어댑터 종류

TCP

어댑터 이름

TCP_ADT

엔드포인트 아이디*

TCP_EP

엔드포인트 이름*

TCP_EP

엔드포인트 그룹 이름

엔드포인트 상태

Running

부팅 초기값

Running

통신 방향

Inbound

설명

TCP 엔드포인트

배포 사유

연결정보

상세설정

메시지/장애처리

메시지 변경 옵션

SSL

TCP 소켓 옵션

접속 제어

연결방식

연결방식

Server

Client

기본 인코딩

서버연결유지

No

Agent 사용여부

사용

미사용

엔드포인트 등록 화면

- 기본 정보 (* : 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	엔드포인트의 그룹 이름이다.

항목	설명
엔드포인트 상태	<p>생성할 때 상태 초기값을 선택한다.</p> <ul style="list-style-type: none"> • Running : Running 상태이면 엔드포인트가 시작된 상태가 된다. • Stopped : Stopped 상태이면 엔드포인트가 중지된 상태가 된다.
부팅 초기값	<p>부팅할 때 상태 초기값을 선택한다.</p> <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	<p>엔드포인트의 통신 방향을 설정한다.</p> <ul style="list-style-type: none"> • Inbound : 외부로부터 메시지를 받기위한 엔드포인트이다. • Outbound : 외부로 메시지를 전달하기 위한 엔드포인트이다. • Both : 외부와 메시지를 주고받기 위한 엔드포인트이다.
설명	생성하는 엔드포인트에 대한 설명을 입력한다. 입력하지 않아도 무관하다.
배포 사유	배포 사유를 작성한다. 작성한 사유는 [배포관리] > [배포이력] 메뉴에서 조회하면 확인할 수 있다.

2.2.1. 서버 모드 엔드포인트

다음은 [엔드포인트 등록 화면](#)의 각 탭 화면에 대한 설명이다.

'연결방식' 항목을 'Server'로 선택한 경우 **[상세설정]**, **[메시지/장애처리]**, **[메시지 변경 송신]**, **[SSL]**, **[TCP 소켓 옵션]**, **[접속 제어]** 탭으로 구성된다.

- **[연결정보] 탭** (* : 필수 입력항목)

연결정보

상세설정

메시지/장애처리

메시지 변경 옵션

SSL

TCP 소켓 옵션

접속 제어

연결방식

연결방식

Server

Client

기본 인코딩

서버 연결유지

No

Agent 사용여부

사용

미사용

주소

IP/호스트 (*)

포트 (*)

0

타임아웃/연결수

Idle 타임아웃*

0

ms

동기/비동기 여부

동기

비동기

최대 연결 수*

1

새 연결 허용

TCP 연결 핸들러

연결 플로우 사용 유무

연결 플로우 아이디

연결 종료 플로우 사용 유무

연결 종료 플로우 아이디

불 타임아웃시 연결유지

TCP 타임아웃

타임아웃 close 소켓

타임아웃 플로우 클

타임아웃 플로우 아이디

타임아웃 헬스체크

요청 메시지

응답 타임아웃

s

재시도 횟수

헬스체크 메시지

별도 메시지 길이 지정

헤더 사이즈 포함 유무

오프셋 위치

오프셋 길이

헬스체크 식별오프셋

+

추가

오프셋 위치

오프셋 길이

식별값

-

삭제

헬스체크 응답

응답 메시지

거대그룹/거래

요청 메시지 변환

헬스체크 수신 모드

헬스체크 식별오프셋

+

추가

오프셋 위치

오프셋 길이

식별값

-

삭제

응답 Ack 사용

엔드포인트 등록 화면 - [연결정보] - 서버 모드

연결방식

항목	설명
연결 방식	엔드포인트가 동작하는 연결 방식을 설정한다. <ul style="list-style-type: none"> ◦ Server : 엔드포인트가 서버 모드로 동작한다. ◦ Client : 엔드포인트가 클라이언트 모드로 동작한다.
기본 인코딩	엔드포인트에서 사용할 기본 인코딩을 설정한다. 메시지에 설정된 인코딩이 우선 적용되고 메시지에 설정되지 않았을 경우에 적용된다. 메시지와 엔드포인트 모두 설정이 되지 않은 경우에는 시스템 기본 인코딩으로 적용된다. Java 표준 Charset과 EBCDIC 타입을 설정할 수 있다.
서버연결유지	<ul style="list-style-type: none"> ◦ No : 연결 중 오류가 발생하면 Client와의 연결을 끊는다. (기본값) ◦ Yes : 거래를 찾지 못하였을 경우에 연결을 유지하도록 한다.
Agent 사용 여부	Agent 사용 여부를 선택한다. <ul style="list-style-type: none"> ◦ 사용 : AnyLink 리모트 에이전트가 설치된 서버의 엔드포인트로 지정한다. Agent를 '사용'으로 설정하였을 경우 리모트 에이전트가 다운되어 있을 때는 수정이 불가하다. 리모트 에이전트 설정은 AnyLink WebAdmin 안내서의 "리모트 에이전트"를 참고한다. ◦ 미사용 : 로컬 서버의 엔드포인트로 지정한다.
Agent 아이디	엔드포인트의 ' Agent 사용여부 ' 항목이 '사용'인 경우에 연결할 리모트 에이전트 아이디를 설정한다.

◦ TCP EP 연결정보

[2023.02 stable 버전 기능추가]

TCP 오류시 연결유지 설정이 yes인 경우의 발생 조건에 ADT_00001 에러코드를 추가하였다.

- **TCP EP > 연결정보 > 오류시 연결유지: yes** 를 설정한 경우 ADT_00001 에러 발생시 세션을 유지한다.

[ADT_00001] biztx not found in endpointConnection

엔드포인트 등록 화면 - [연결정보] - 연결방식 - 오류시 연결 유지

◦ 주소

항목	설명
IP/호스트 (*)	노드의 IP/호스트를 설정한다.

항목	설명
포트 (*)	Listen할 포트 번호를 설정한다.

◦ 타임아웃/연결수

항목	설명
Idle 타임아웃 (ms)	<p>Idle 타임아웃을 설정한다. 커넥션이 아무런 동작을 하고 있지 않을 때 해당 설정된 값이 지나면 타임아웃이 발생한다.</p> <p>기본값은 0이고 0으로 설정할 경우에는 타임아웃이 발생하지 않는다.</p> <p>(단위: ms)</p>
동기/비동기 여부	<p>동기/비동기 여부를 설정한다.</p> <ul style="list-style-type: none"> ◦ 동기 : 메시지를 송신하면 응답이 들어오기 전까지 사용할 수 없게 하고, 메시지를 수신하면 응답을 줄때까지 사용할 수 없도록 한다. ◦ 비동기 : 메시지 송수신 여부와 상관없이 항상 사용 가능하도록 한다.
아웃바운드 엔드포인트 설정	<p>엔드포인트가 'Inbound'이면서 '비동기'인 경우에 아웃바운드 엔드포인트를 설정할 수 있다.</p> <p>인바운드 엔드포인트로 응답을 송신하려 할 경우 지정한 아웃바운드 엔드포인트로 응답을 보낸다. 지정할 아웃바운드 엔드포인트는 인바운드 엔드포인트와 동일한 어댑터 내에 있어야 한다.</p>
최대 연결수 *	<p>최대 연결 수를 설정한다.</p> <p>서버 모드의 경우 연결이 최대 연결 수를 넘어가면 신규 연결을 받지 않는다.</p>
새 연결 허용	엔드포인트에 최대 연결수만큼 연결되어 있을 때 신규 연결이 요청되면 연결 중 가장 오래된 연결을 끊고 신규 연결을 허용한다.

◦ Agent 연결정보

항목	설명
재시도 간격	엔드포인트의 ' Agent 사용여부 ' 항목이 '사용'인 경우에 재시도 간격을 설정할 수 있다. 재시도 간격에 따라 리모트 에이전트가 상대 서버에 연결한다.

◦ TCP 연결 핸들러

항목	설명
연결 플로우 사용 유무	커넥션이 연결되었을 때 플로우 호출 여부를 설정한다.

항목	설명
연결 플로우 아이디	<p>커넥션이 연결되었을 때 호출할 플로우 아이디를 설정한다.</p> <p>연결 플로우는 로그인 전문같이 커넥션이 연결되면 주고받아야 하는 전문이 있을 경우 사용한다.</p> <ul style="list-style-type: none"> ◦ 형식 <div>거래 SYSID:FlowID:Start event ID</div> ◦ 예시 <div>com.tmax.al7.flow.test:al7_flow_Id:flow_start_event_Id</div>
연결 종료 플로우 사용 유무	커넥션의 연결이 종료되었을 경우 플로우 호출 여부를 설정한다.
연결 종료 플로우 아이디	<p>커넥션의 연결이 종료되었을 경우 호출할 플로우 아이디를 설정한다.</p> <p>연결 종료 플로우는 로그아웃같이 커넥션을 종료할 때 주고받아야 하는 전문이 있을 경우 사용한다.</p> <ul style="list-style-type: none"> ◦ 형식 <div>거래 SYSID:FlowID:Start event ID</div> ◦ 예시 <div>com.tmax.al7.flow.test:al7_flow_Id:flow_start_event_Id</div>
롤 타임아웃시 연결유지	<p>비동기 거래에서 아웃바운드 롤 타임아웃이 발생하는 경우 연결을 끊지 않고 유지하고 싶을 때 설정한다.</p> <p>'통신 방향' 항목이 'Outbound'나 'Both'이며, 클라이언트 모드인 경우 '연결 관리' 항목 'Yes'이거나 '연결 유지' 항목이 'Yes'일 때 설정 가능하다.</p>

◦ TCP 타임아웃

항목	설명
타임아웃 close 소켓	<p>커넥션에서 타임아웃이 발생했을 경우 소켓에 대한 close 여부를 설정한다 (타임아웃 플로우 콜과 동시에 사용할 수 없다).</p> <p>타임아웃 close 소켓은 Idle 타임아웃에 의해서 타임아웃이 발생한 경우 소켓을 close할 경우 사용한다.</p>
타임아웃 플로우 콜	커넥션에서 타임아웃이 발생했을 경우 플로우 호출 여부를 설정한다.

항목	설명
타임아웃 플로우 아이디	<p>커넥션에서 타임아웃이 발생했을 경우 호출할 플로우 아이디를 설정한다 (타임아웃 close 소켓과 동시에 사용할 수 없다).</p> <p>타임아웃 플로우는 로그아웃같이 커넥션을 종료할 때 주고받아야 하는 전문이 있을 경우 사용한다.</p> <ul style="list-style-type: none"> ◦ 형식 <div>거래 SYSID:FlowID:Start event ID</div> ◦ 예시 <div>com.tmax.al7.flow.test:al7_flow_Id:flow_start_event_Id</div>
타임아웃 헬스 체크	<p>커넥션에서 타임아웃이 발생했을 경우 헬스체크 메시지를 상대방에게 보낼 때 설정한다.</p>
요청 메시지	<p>커넥션에서 타임아웃이 발생했을 경우 보낼 헬스체크 메시지를 입력한다. 문자 형식으로 입력하며, 제공되는 함수를 함께 사용할 수 있다.</p> <ul style="list-style-type: none"> ◦ <code>\${time:}</code> : 요청 메시지를 보내는 시각을 생성한다. <div>(예: <code>\${time:yyyyMMDD}</code>, <code>\${time:HHmmss}</code>)</div> ◦ <code>\${seq:자리수}</code> : 자리수만큼의 sequence 값을 생성한다. <p>엔드포인트가 재배포될 때 다시 1부터 시작하며, 자리수는 최대 9까지 설정할 수 있다. 10 이상의 자리수는 9로 설정된다.</p> <div>(예: <code>\${seq:3}</code>, <code>\${seq:6}</code>)</div> ◦ <code>\${rand:자리수}</code> : 자리수만큼의 random 값을 생성한다. <p>자리수는 최대 9까지 설정할 수 있다. 10 이상의 자리수는 9로 설정된다.</p> <div>(예: <code>\${rand:3}</code>, <code>\${rand:6}</code>)</div>
응답 타임아웃	<p>커넥션 타임아웃으로 전송한 헬스체크 메시지에 대한 응답 타임아웃을 설정하고 싶을 때 설정한다.</p> <p>응답 타임아웃 시간 내에 응답이 없을 경우 소켓의 연결을 끊는다.</p>

항목	설명
재시도 횟수	<p>커넥션에서 타임아웃이 발생했을 경우 헬스체크 메시지 전송의 재시도 횟수를 지정할 때 설정한다.</p> <p>이 설정되어 있을 때 응답 타임아웃을 재시도 간격으로 사용한다. 재시도 횟수동안 응답이 없을 경우 소켓의 연결을 끊는다.</p>

◦ 헬스체크 메시지

헬스체크 메시지는 메시지/장애처리의 설정만 적용된다. 메시지 변경 옵션은 적용되지 않는다.

항목	설명
별도 메시지 길이지정	<p>수신시 헬스체크 메시지에만 적용되는 통신헤더를 설정하고 싶을 경우 설정한다.</p> <p>'헤더 사이즈 포함 유무', '오프셋', '길이'만 설정 가능하며 '헤더 길이'는 메시지/장애처리의 설정값을 사용한다. 그 외 메시지/장애처리의 통신헤더 설정값은 무시한다.</p> <ul style="list-style-type: none"> ◦ 헤더 사이즈 포함 유무 : 헤더 길이가 헤더에 포함될 때 설정한다. ◦ 오프셋 위치 : 헤더의 시작 위치를 설정한다. ◦ 오프셋 길이 : 헤더의 시작 위치로부터의 길이를 설정한다.
헬스체크 식별오프셋	<p>헬스체크 메시지인지를 검증하는 값에 대해 설정한다.</p> <ul style="list-style-type: none"> ◦ 오프셋 위치 : 식별값이 있는 곳의 시작 위치를 설정한다. ◦ 오프셋 길이 : 식별값이 있는 곳의 시작 위치로부터의 길이를 설정한다. ◦ 식별값 : 식별값을 설정한다. '?' expression을 사용하여 메시지의 특정 부분만을 식별값으로 설정할 수 있다. <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>예)</p> <p>식별값 설정 : AA??CC</p> <p>수신 메시지 : AABBCB 일 경우 헬스체크 메시지로 판정</p> </div>

◦ 헬스체크 응답

항목	설명
응답 메시지	<p>헬스체크 메시지가 일치할 때 응답으로 보낼 메시지를 설정한다.</p> <p>문자 형식으로 입력하며, 제공되는 함수를 함께 사용할 수 있다. 메시지 설정에 대한 자세한 내용은 TCP 타임아웃 영역의 '요청 메시지' 항목을 참고한다.</p> <p>응답 메시지를 선택하는 경우 추가적으로 응답 엔드포인트를 설정할 수 있다. 미선택하는 경우 요청받은 엔드포인트로 응답을 보낸다.</p> <ul style="list-style-type: none"> ◦ 응답 엔드포인트 : 해당 엔드포인트 외에 다른 엔드포인트로 응답을 보내고자 할 때 설정한다. ◦ 아웃바운드 엔드포인트 설정 : 응답으로 보낼 엔드포인트를 설정한다. 엔드포인트는 해당 엔드포인트와 동일한 어댑터 내의 엔드포인트만을 설정할 수 있다.
거래그룹/거래	헬스체크 메시지가 일치할 때 호출할 거래그룹/거래를 선택한다.
요청 메시지 변환	<p>헬스체크 메시지가 일치할 때 수신한 헬스체크 메시지의 일부분을 변환하여 응답한다.</p> <ul style="list-style-type: none"> ◦ 오프셋 위치 : 변환할 시작 위치를 설정한다. ◦ 오프셋 길이 : 변환할 시작 위치로부터의 길이를 설정한다. ◦ 변환값 : 변환값을 설정한다. 변환값 설정에 대한 자세한 설명은 TCP 타임아웃 영역의 '요청 메시지' 항목을 참고한다.

◦ 헬스체크 수신모드

항목	설명
헬스체크 식별오프셋	<p>헬스체크 메시지인지를 검증하는 값에 대해 설정한다.</p> <ul style="list-style-type: none"> ◦ 오프셋 위치 : 식별값이 있는 곳의 시작 위치를 설정한다. ◦ 오프셋 길이 : 식별값이 있는 곳의 시작 위치로부터의 길이를 설정한다. ◦ 식별값 : 식별값을 설정한다. '?' expression을 사용하여 메시지의 특정 부분만을 식별값으로 설정할 수 있다. ◦ 예 <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>식별값 설정 : AA??CC 수신 메시지 : AABBC 일 경우 헬스체크 메시지로 판정</p> </div>

◦ 응답 Ack 사용

클러스터 환경에서 요청에 대한 응답이 다른 서버로 들어올 경우 요청을 보낸 서버로 응답을 전달하기 전 응답을 보낸 서버에 ACK을 송신할 때 사용한다. '**통신 방향**' 항목이 'Inbound' 또는 'Both'일 때 사용할 수 있다.

항목	설명
응답 메시지	수신한 메시지에 대한 응답 메시지를 정의한다. 응답 메시지를 송신하고 설정한 거래를 시작한다.

• [상세설정] 탭

엔드포인트 등록 화면 - [상세설정]

항목	설명
메시지 핸들러	[핸들러] 버튼을 클릭한 후 메시지 핸들러 선택 화면 (엔드포인트 등록 화면 - [상세설정] - 메시지 핸들러 선택 화면)에서 메시지를 선택한다.
거래그룹/거래/룰	메시지를 수신할 때 엔드포인트가 받아서 처리해야 할 메시지에 대한 거래를 리스트에서 선택한다.
메시지 분할 거래	TCP 메시지 분할 유틸리티가 동작한 플로우에서 Intermediate 이벤트로 호출될 거래를 설정한다. 호출된 거래에서는 TCP 메시지 분할 유틸리티의 분할 길이만큼씩 메시지를 읽는다. 호출되기 전 거래에서는 메시지 길이지정 방식에 따라 메시지를 읽는다.
배치모드	대화형 모드로 거래를 수행할 때 선택한다. 일반적으로 배치 스타일로 거래를 만들 경우 선택한다. 배치 모드를 선택하면 거래를 호출할 때나 응답 처리할 때에 스레드를 변경하지 않고 거래가 수행된다. '통신 방향' 항목이 'Inbound' 또는 'Both'일 때 설정할 수 있다.
멀티 셀렉터 모드	'배치모드' 일 때 사용 가능하다. 한 스레드가 거래를 실행 중일 때 새로운 요청이 들어오면 다른 스레드가 거래를 잡아 수행할 수 있도록 한다.
복수거래설정	수신한 메시지의 특정 값에 따라 거래그룹/거래/룰을 선택하고자 할 때 사용한다. '복수거래설정' 을 선택하면 '복수헤더사용'과 '헤더 추가'를 할 수 있다.
복수헤더사용	헤더 값에 따라 다른 거래그룹/거래/룰을 사용하고자 할 때 선택한다. 헤더 구분필드를 입력해야 한다. <ul style="list-style-type: none"> 오프셋 : 비교할 헤더 값의 시작 위치를 설정한다. 길이 : 비교할 헤더 값의 시작 위치로부터의 길이를 설정한다.

항목	설명
헤더 추가	'복수거래설정'을 선택한 경우 추가할 수 있다. 복수 헤더를 사용하는 경우 [추가] 버튼을 클릭하면 헤더의 정보를 추가하거나 [삭제] 버튼을 클릭해서 삭제할 수 있다(엔드포인트 등록 화면 - [상세설정] - 복수거래설정 - 헤더 추가 참고).

다음은 '메시지 핸들러' 항목에 입력할 메시지를 선택한 후 **[선택]** 버튼을 클릭한다.

엔드포인트 등록 화면 - [상세설정] - 메시지 핸들러 선택 화면

'복수거래설정'을 선택한 후 '헤더 추가' 항목에 **[추가]** 버튼을 클릭하면 다음의 화면에서 헤더 정보를 추가할 수 있다.

엔드포인트 등록 화면 - [상세설정] - 복수거래설정 - 헤더 추가

항목	설명
헤더 값	복수헤더를 사용하는 경우 헤더 값에 따라 다른 거래그룹/거래/룰을 설정한다.

항목	설명
식별필드 설정	<p>특정 거래를 선택하기 위해 메시지의 식별필드를 설정한다.</p> <ul style="list-style-type: none"> 오프셋길이 : 메시지의 특정 부분으로 식별한다. 오프셋 길이를 추가할 수 있고, 메시지에서 비교할 값의 위치와 길이를 설정한다. 파싱핸들러 : 파싱핸들러를 호출하여 식별한다.
아웃바운드 룰 키 사용	TCP 아웃바운드 룰의 서비스 키 설정에 따라 거래그룹/거래/룰을 호출한다.
거래룰식별리스트	<p>메시지 핸들러나 파싱 핸들러를 통해 추출된 값과 식별값을 비교하여 거래그룹/거래/룰을 지정한다. 아웃바운드 룰 키를 사용하지 않았을 때 설정 가능하다.</p> <ul style="list-style-type: none"> 식별값 : 지정한 거래그룹/거래/룰을 타기 위한 식별값을 설정한다. 거래그룹/거래/룰 : 식별값이 일치할 경우 해당 거래그룹/거래/룰을 호출한다.

'복수거래설정'의 '식별필드 설정' 항목을 '파싱핸들러' 선택한 경우 다음의 화면에서 파싱 핸들러를 선택한 후 [선택] 버튼을 클릭한다.

엔드포인트 등록 화면 - [상세설정] - 파싱 핸들러 선택 화면

• [메시지/장애처리] 탭

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

메시지 길이 지정 방식

메시지 길이 지정 방식
헤더에 길이정보 포함
Outbound 별도 설정

헤더 종류
통신헤더
헤더 길이
Bytes

Primitive 사용 유무

헤더 사이즈 포함 유무

오프셋
길이

메시지 정합성 검사

길이정보 키 사용 유무
+ 추가

StxEtx 사용유무

TCP 에이전트 사용

엔드포인트 등록 화면 - [메시지/장애처리]

항목	설명
메시지 길이 지정 방식	<p>메시지 길이 지정 방식을 설정한다.</p> <ul style="list-style-type: none"> ◦ 헤더에 길이정보 포함 : 메시지의 헤더에 길이정보가 있는 경우 ◦ 고정길이 : 전체 메시지 길이가 고정으로 지정되어 있는 경우 ◦ End 문자 지정 : 메시지의 끝에 특정 예약된 문자가 있는 경우 ◦ VO : VO Agent를 사용하는 경우 ◦ TCP Agent : TCP Agent를 사용하는 경우 ◦ EOD : End of Data로 한 번에 메시지가 들어오는 경우 (TCP 특성상 네트워크가 느릴 경우 오류 발생 소지가 높으므로 사용에 주의한다.) ◦ StxEtxChar : 메시지의 시작과 끝에 특정 예약된 문자가 있는 경우 ◦ EOR : 메가박스과 통신하는 메시지의 끝에 '0xffef'의 특정 예약된 문자가 있는 경우
Outbound 별도 설정	인바운드와 아웃바운드의 메시지 길이 지정 방식이 상이할 경우 선택한다.

다음은 '메시지 길이 지정 방식'에 따라 등록하는 항목에 대한 설명이다.

- 헤더에 길이 정보 포함

항목	설명
헤더 종류	<p>헤더 종류를 선택한다.</p> <ul style="list-style-type: none"> ◦ 통신헤더 : 통신 만을 위한 헤더로 메시지로 길이 만을 위한 정보로 헤더 메시지를 어댑터로 전달하지 않는다. ◦ 공통헤더 : 길이 정보가 포함된 공통헤더로 헤더 메시지를 어댑터로 전달한다.
헤더 길이	헤더 길이를 설정한다.

항목	설명
Trim 기능 설정 사용 여부	길이 정보가 일반 적인 문자가 아닌 공백이 포함되어 올 경우 선택한다. 공통헤더의 경우만 선택이 가능하다. (예: __10)
Primitive 사용 유무	길이 정보가 일반적인 문자(예: 0010) 이 아닌 Primitive 타입(예: int)으로 들어올 경우 선택한다.
Primitive 종류	Primitive 타입의 종류를 선택한다. ◦ int : int 타입으로 설정된 경우
헤더 사이즈 포함 유무	헤더에서 메시지 길이 정보를 추출하였을 때 헤더의 사이즈가 길이 정보에 포함되어 있는지 여부를 설정한다. ◦ 예) <div>헤더 사이즈 포함 Yes : 0014AAAAABBBBB 헤더 사이즈 포함 No : 0010AAAAABBBBB</div>
전문 길이 설정	아웃바운드, 공통헤더일 경우에만 사용가능하다. 전문 길이를 계산하여 공통 헤더 위치에 사이즈를 넣어준다. 헤더 사이즈 포함 유무에 따라 길이 정보가 다르게 설정된다. ◦ 예) 다음은 오프셋이 2, 길이가 4, 헤더 사이즈 포함 유무는 false, 전문 길이는 14일 경우 예제이다. <div>원본 메시지 : KB____AAAAABBBBB --> 변환된 메시지 : KB0014AAAAABBBBB</div>
오프셋	메시지의 길이 정보의 시작 위치를 설정한다.
길이	메시지 길이 정보의 시작 위치로부터의 길이를 설정한다.
메시지 정합성 검사	메시지의 정합성을 검사한다. 헤더 길이로부터 설정한 식별값의 위치만큼 읽은 후 메시지와 일치하는지 검증한다. 식별값과 같을 경우 헤더 처리 후의 메시지를 어댑터로 전달한다. 정합성이 맞지 않을 경우 연결을 끊는다. ◦ 식별값 : 헤더와 비교할 값을 설정한다. ◦ 오프셋 : 식별값의 시작 위치를 설정한다. ◦ 길이 : 식별값의 시작 위치로부터의 길이를 설정한다. ◦ 예) <div>헤더 길이 : 4, 식별값 : AA, 오프셋 : 4, 길이 : 2 수신한 메시지 : 0010AABBCCDD 1. 헤더 길이로부터 설정한 식별값의 위치(오프셋 4, 길이 2)까지인 '0010AA'를 읽음 2. 오프셋과 길이로 AA 추출 3. 식별값 AA와 일치하므로 어댑터로 AABBCCDD 전달</div>

항목	설명
길이 정보 키 사용 유무	<p>길이 정보 키 사용 유무를 설정한다.</p> <p>길이 정보 키란 헤더에서 길이 정보를 추출하였을 때 해당 정보가 길이 정보가 아닌 특정한 키 값일 경우에 사용된다. "?" 기호는 모든 글자를 의미한다. 매칭되는 것이 여러 개일 경우 제일 처음 매칭되는 값을 사용한다. Key에 '\${default-key}' 예약어를 사용하면 매칭되는 값이 없을 때 선택된다.</p> <ul style="list-style-type: none"> ◦ Key : 헤더값에 매칭되는 Key를 설정한다. ◦ 길이 : 메시지 길이를 설정한다. ◦ 예) <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>헤더 종류 : 통신헤더, 헤더 길이 : 10, 오프셋 : 0, 길이 : 4 Key : 0001, 길이 : 10 Key : 0002, 길이 : 30 수신한 메시지 : 0001ABCDEF1234567890 1. 헤더 길이만큼 읽은 후 오프셋, 길이에 맞는 0001을 추출 2. 0001이 Key의 첫번째와 같으므로 다음에 읽어야 할 길이는 10으로 설정 3. 남은 길이인 10만큼 메시지를 모두 읽음</p> </div> <p>길이에 표현식을 사용할 수 있다. 사칙연산을 사용할 수 있으며 정수로만 계산되고 괄호는 사용할 수 없다. 사칙연산의 우선순위는 무시된다.</p> <ul style="list-style-type: none"> ◦ \${offset(N)} : 메시지의 N번째 위치의 값 추출 ◦ \${offset(N-M)} : 메시지의 N번째 위치부터 M번째 위치까지의 값 추출 ◦ 사칙연산 예) <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>수신한 메시지 : 0010100ABCD001 길이 : \${offset(0-2)}*\${offset(7-9)} 결과 : 100*001 = 100</p> </div>
StxEtx 사용유무	<p>헤더에 시작 문자(Stx), 끝 문자(Etx)가 있을 경우 사용한다.</p> <ul style="list-style-type: none"> ◦ Stx 문자 : 헤더의 시작 문자를 설정한다. ◦ Etx 문자 : 헤더의 끝 문자를 설정한다. ◦ StxEtx 문자 포함유무 : Stx, Etx 문자를 헤더에 포함할지 여부를 설정한다. 설정 시 헤더길이는 StxEtx가 포함된 길이여야 하며, 미설정시 StxEtx 길이만큼 헤더 길이를 늘려 읽는다. ◦ HEX 타입 유무 : Stx, Etx 문자를 HEX 형식으로 지정하는 경우 사용한다. 예를 들면 '0A'(new line)으로 설정한다.
TCP 에이전트 사용	<p>리모트 에이전트와 내부 Hub간의 통신을 위해 설정한다.</p>

◦ 고정 길이

항목	설명
고정 길이	메시지의 길이를 설정한다.

◦ End 문자 지정

항목	설명
End문자	End 문자를 설정한다.
End Char 포함 유무	End 문자를 메시지에 포함할지 여부를 설정한다.
HEX 모드	End 문자를 HEX 형식으로 지정하는 경우 사용한다. 예를 들면 '0A'(new line)으로 설정한다.

◦ StxEtxChar

항목	설명
Stx 문자	시작 문자를 설정한다.
Etx 문자	끝 문자를 메시지에 포함할지 여부를 설정한다.
StxEtx 문자 포함유무	Stx, Etx 문자를 메시지에 포함할지 여부를 설정한다.
HEX 타입 유무	Stx, Etx 문자를 HEX 형식으로 지정하는 경우 사용한다. 예를 들면 '0A'(new line)으로 설정한다.

• [메시지 변경 송신] 탭

연결정보
상세설정
메시지/장애편리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

메시지 송신시

메시지 추가 위치

메시지 추가 문자열

메시지 삭제 위치

메시지 삭제 길이

EBCDIC 코드 변환
☒

코드 변환 종류

ASCII to EBCDIC ▾

오프셋

길이

메시지 수신시

메시지 추가 위치

메시지 추가 문자열

메시지 삭제 위치

메시지 삭제 길이

지환 이전 문자

지환 이후 문자

EBCDIC 코드 변환
☒

코드 변환 종류

ASCII to EBCDIC ▾

오프셋

길이

엔드포인트 등록 화면 - [메시지 변경 옵션]

◦ 메시지 송신시

메시지를 변환하여 송신한다.

항목	설명
메시지 추가 위치	메시지에 추가할 문자열의 위치를 설정한다.
메시지 추가 문자열	메시지에 추가할 문자열을 설정한다.
메시지 삭제 위치	메시지 내의 삭제할 시작 위치를 설정한다.
메시지 삭제 길이	메시지의 삭제 시작 위치로부터의 길이를 설정한다.
EBCDIC 코드 변환	메시지의 특정 부분을 EBCDIC에서 ASCII로 변환하거나 ASCII에서 EBCDIC으로 변환할 때 설정한다.
코드 변환 종류	<ul style="list-style-type: none"> ◦ ASCII to EBCDIC : ASCII에서 EBCDIC으로 변환할 경우 ◦ EBCDIC to ASCII : EBCDIC에서 ASCII로 변환할 경우
오프셋	EBCDIC 코드 변환이 적용될 메시지의 시작 위치를 설정한다.
길이	EBCDIC 코드 변환이 적용될 메시지의 시작 위치로부터의 길이를 설정한다.

◦ 메시지 수신시

수신한 메시지를 변환하여 어댑터에 넘긴다. 설명되지 않은 항목은 **메시지 송신시** 영역의 설명을 참고한다.

항목	설명
치환 이전 문자	메시지에서 변환할 문자를 설정한다. 개행문자(\n)나 탭(\r), 특수문자(\u) 또한 설정할 수 있다.
치환 이후 문자	치환 이전 문자를 변환할 문자를 설정한다.

• [SSL] 탭

[연결정보] 탭의 '연결방식' 항목을 'Server'로 선택한 경우 키 스토어를 설정할 수 있으며, 'Client'로 설정한 경우 Trust Store와 키 스토어를 설정할 수 있다.

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

SSL

SSL 사용 설정
☒ 사용
☐ 미사용

Trust Store 설정

스토어 위치
스토어 타입
선택

클라이언트 인증
☒ 예
☐ 아니오
스토어 패스워드

키 스토어 설정

스토어 위치
스토어 타입
선택

Private Key 패스워드
스토어 패스워드

엔드포인트 등록 화면 - [SSL]

◦ 키 스토어 설정

항목	설명
스토어 위치	Key store의 저장 위치를 설정한다.

항목	설명
스토어 타입	Key store의 저장 형식을 설정한다. (기본값: jks)
Private Key 패스워드	개인 키를 접근하기 위한 암호를 설정한다.
스토어 패스워드	Key store를 접근하기 위한 암호를 설정한다.

• [TCP 소켓 옵션] 탭

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

TCP 소켓 옵션

TCP 소켓 옵션 사용 유무
☒ 사용
☐ 미사용

TCP 소켓 옵션 설정

TCP_NODELAY ☒

SO_LINGER (s)

Send Buffer (byte)

Receive Buffer (byte)

SO_REUSEADDR ☒

엔드포인트 등록 화면 - [TCP 소켓 옵션]

◦ TCP 소켓 옵션 설정

항목	설명
TCP_NODELAY	Nagle 알고리즘을 무력화하는 소켓 옵션인 TCP_NODELAY을 설정한다. (기본값: 체크)
SO_LINGER (s)	초기값인 '-1'이 아닌 경우 연결 종료 후 FIN_ACK 패킷을 받을 때까지 대기한다. 소켓 옵션인 SO_LINGER의 타임아웃 시간을 설정한다. (기본값: -1)
Send Buffer (byte)	송신 TCP 버퍼 사이즈를 설정한다. 소켓 옵션인 SO_SNDBUF을 설정한다. (기본값: 1313280)
Receive Buffer (byte)	수신 TCP 버퍼 사이즈를 설정한다. 소켓 옵션인 SO_RCVBUF을 설정한다. (기본값: 531000)
SO_REUSEADDR	TIME_WAIT 상태에 있는 소켓을 재사용할 수 있도록 한다. 소켓 옵션인 SO_REUSEADDR을 설정한다. (기본값: 체크)

• [접속 제어] 탭

'IP Control' 항목을 'Allow'와 'Deny'로 설정한 경우 여러 개의 IP를 추가할 수 있다.

연결정보

상세설정

메시지/장애처리

메시지 변경 옵션

SSL

TCP 소켓 옵션

접속 제어

IP Control ?

IP Control

☐ Parent
☐ OFF
☒ Allow
☐ Deny

Allow

+ 추가

No	IP Rule	삭제
1		- 삭제

엔드포인트 등록 화면 - [접속 제어]

◦ IP Control

항목	설명
IP Control	<p>엔드포인트에 연결을 시도하는 특정 IP의 클라이언트를 제어한다.</p> <ul style="list-style-type: none"> ◦ Parent : 상위 엔드포인트 그룹이나 어댑터의 IP Control을 따른다. (기본값) ◦ OFF : 모든 클라이언트의 연결을 허용한다. ◦ Allow : 지정한 IP Rule에 해당하는 클라이언트의 연결만을 허용한다. ◦ Deny : 지정한 IP Rule에 해당하는 클라이언트의 연결을 차단한다.

2.2.2. 클라이언트 모드 엔드포인트

[연결정보] 탭의 '연결방식' 항목을 'Client'로 선택한 경우 [엔드포인트 등록 화면](#)의 각 탭 화면에 대한 설명이다. 설명되지 않은 탭의 설명은 [서버 모드 엔드포인트](#)를 참고한다.

'연결방식' 항목을 'Client'로 선택한 경우 [상세설정], [메시지/장애처리], [메시지 변경 송신], [SSL], [TCP 소켓 옵션] 탭으로 구성된다.

- [연결정보] 탭 (* : 필수 입력항목)

연결정보

상세설정

메시지/장메시리

메시지 변경 옵션

SSL

TCP 소켓 옵션

연결방식

연결방식

Server

Client

기본 인코딩

연결 관리 여부

Yes

Agent 사용여부

사용

미사용

주소

IP/호스트 (*)

포트 (*)

0

로컬 IP/호스트

로컬 포트

0

타임아웃/연결수

연결 타임아웃*

3000

ms

Idle 타임아웃*

0

ms

동기/비동기 여부

동기

비동기

최소 연결 수*

1

최대 연결 수*

1

재시도

재시도 간격

3000

ms

서브 재시도 간격 사용 유무

TCP 연결 핸들러

연결 플로우 사용 유무

연결 플로우 아이디

연결 종료 플로우 사용 유무

연결 종료 플로우 아이디

콜 타임아웃시 연결유지

TCP 타임아웃

타임아웃 close 소켓

타임아웃 플로우 콜

타임아웃 헬스체크

타임아웃 플로우 아이디

요청 메시지

응답 타임아웃

s

재시도 횟수

헬스체크 메시지

Backup 주소

IP/호스트

포트

엔드포인트 등록 화면 - [연결정보] - 클라이언트 모드

연결방식

항목	설명
연결 방식	<p>엔드포인트가 동작하는 연결 방식을 설정한다.</p> <ul style="list-style-type: none"> Server : 엔드포인트가 서버 모드로 동작한다. Client : 엔드포인트가 클라이언트 모드로 동작한다.
기본 인코딩	<p>엔드포인트에서 사용할 기본 인코딩을 설정한다.</p> <p>메시지에 설정된 인코딩이 우선 적용되고 메시지에 설정되지 않았을 경우에 적용된다. 메시지와 엔드포인트 모두 설정이 되지 않은 경우에는 시스템 기본 인코딩으로 적용된다.</p> <p>Java 표준 Charset과 EBCDIC 타입을 설정할 수 있다.</p>
연결 관리 여부	<p>엔드포인트가 클라이언트 모드일 때 연결 관리 여부를 설정한다.</p> <ul style="list-style-type: none"> Yes : 연결을 관리한다. 최소 연결수만큼의 연결을 유지하고 연결이 끊어지면 재시도 간격을 주기로 재연결을 맺는다. No : 연결을 관리하지 않는다.

항목	설명
연결 유지 여부	엔드포인트가 클라이언트에 ' 연결 관리 여부 '가 'No'인 경우에 연결 유지 여부를 설정한다. <ul style="list-style-type: none"> ◦ Yes : 해당 연결로 들어온 거래가 종료 되어도 연결을 끊지 않는다. ◦ No : 해당 연결로 들어온 거래가 종료되면 해당 연결을 끊는다.
Agent 사용 여부	Agent 사용 여부를 선택한다. <ul style="list-style-type: none"> ◦ 사용 : AnyLink 리모트 에이전트가 설치된 클라이언트의 엔드포인트로 지정한다. ◦ 미사용 : 로컬 클라이언트의 엔드포인트로 지정한다.
Agent 아이디	엔드포인트의 ' Agent 사용여부 ' 항목이 '사용'인 경우에 연결할 리모트 에이전트 아이디를 설정한다.

◦ 주소

항목	설명
IP/호스트 (*)	연결할 IP/호스트를 설정한다.
포트 (*)	연결할 포트 번호를 설정한다.
로컬 IP/호스트	연결에 사용할 로컬 IP/호스트를 설정한다.
로컬 포트	연결에 사용할 로컬 포트를 설정한다.

◦ 타임아웃/연결수

항목	설명
연결 타임아웃 (ms)	연결 타임아웃(Connect timeout)을 설정한다. 연결 요청을 해서 연결될 때까지의 타임아웃이다. (단위: ms)
Idle 타임아웃 (ms)	Idle 타임아웃을 설정한다. 커넥션이 아무런 동작을 하고 있지 않을때 해당 설정된 값이 지나면 타임아웃이 발생한다. 기본값은 0이고 0으로 설정할 경우에는 타임아웃이 발생하지 않는다. (단위: ms)
동기/비동기 여부	동기/비동기 여부를 설정한다. <ul style="list-style-type: none"> ◦ 동기 : 메시지를 송신하면 응답이 들어오기 전까지 사용할 수 없게 하고, 메시지를 수신하면 응답을 줄때까지 사용할 수 없도록 한다. ◦ 비동기 : 메시지 송수신 여부와 상관없이 항상 사용 가능하도록 한다.

항목	설명
최소 연결수 *	<p>최소 연결수를 설정한다.</p> <p>클라이언트에 '연결 관리 여부' 항목이 'Yes'인 경우에만 설정할 수 있다. 최소 연결수가 설정되면 엔드포인트는 항상 최소 연결을 유지하고 최소 연결수가 설정된 값보다 작으면 재시도 간격을 주기로 하여 재연결을 맺는다.</p>
최대 연결수 *	<p>최대 연결 수를 설정한다.</p> <p>클라이언트 모드의 경우 연결이 최대 연결수를 넘어가면 거래가 연결을 받을 때까지 대기한다. 아웃바운드 룰의 요청처리 타임아웃 시간만큼 대기하고 해당 시간 내에 연결을 가지지 못하면 타임아웃 처리된다.</p>

◦ 재시도

항목	설명
재시도 간격 (ms)	<p>재시도 간격을 설정한다. (단위: ms)</p> <p>클라이언트에 '연결 관리 여부' 항목이 'Yes'인 경우에만 설정할 수 있다. 재시도 간격을 주기로 재연결을 맺는다.</p>
서브 재시도 간격 사용 유무	<p>재시도 간격으로 재시도 횟수만큼 시도한 뒤 다른 재시도 간격을 사용하고자 할 때 설정한다.</p> <p>클라이언트에 '연결 관리 여부' 항목이 'Yes'인 경우에만 설정할 수 있다. 서브 재시도 간격을 주기로 재연결을 맺는다.</p>
재시도 횟수	서브 재시도 전 재시도 횟수를 설정한다. ' 서브 재시도 간격 사용 유무 ' 항목을 선택해야 설정할 수 있다.
서브 재시도 간격 (ms)	서브 재시도 간격을 설정한다. (단위: ms)

◦ TCP 연결 핸들러

클라이언트에 '**연결 관리 여부**'가 'Yes'거나 '**연결 유지 여부**'가 'Yes'인 경우에만 설정할 수 있다. 상세 내용은 [서버 모드 엔드포인트](#)를 참고한다.

◦ TCP 타임아웃

클라이언트에 '**연결 관리 여부**'가 'Yes'거나 '**연결 유지 여부**'가 'Yes'인 경우에만 설정할 수 있다. 상세 내용은 [서버 모드 엔드포인트](#)를 참고한다.

◦ 헬스체크 메시지

상세 내용은 [서버 모드 엔드포인트](#)를 참고한다.

◦ Backup 주소

항목	설명
IP/호스트	연결에 실패할 때 사용할 백업 IP/호스트를 설정한다. Backup 주소는 하나만 설정 가능하다. 클라이언트에 '연결 관리 여부' 항목이 'Yes'인 경우에만 설정할 수 있다.
포트	연결에 실패할 때 사용할 백업 포트를 설정한다. 클라이언트에 '연결 관리 여부' 항목이 'Yes'인 경우에만 설정할 수 있다.

• [상세설정] 탭

화면에 대한 내용은 [서버 모드 엔드포인트](#)를 참고한다.

• [메시지/장애처리] 탭

화면에 대한 내용은 [서버 모드 엔드포인트](#)를 참고한다.

• [메시지 변경 송신] 탭

화면에 대한 내용은 [서버 모드 엔드포인트](#)를 참고한다.

• [SSL] 탭

[연결정보] 탭의 '연결방식' 항목을 'Server'로 선택한 경우 키 스토어를 설정할 수 있으며, 'Client'로 설정한 경우 Trust Store와 키 스토어를 설정할 수 있다.

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

SSL

SSL 사용 설정

☒ 사용
☐ 미사용

Trust Store 설정

스토어 위치

스토어 타입

선택 ▾

클라이언트 인증

☒ 예
☐ 아니오

스토어 패스워드

키 스토어 설정

스토어 위치

스토어 타입

선택 ▾

Private Key 패스워드

스토어 패스워드

엔드포인트 등록 화면 - [SSL]

◦ Trust Store 설정

항목	설명
스토어 위치	Trust Store의 저장 위치를 설정한다.
스토어 타입	Trust Store의 저장 형식을 설정한다. (기본값: jks)
클라이언트 인증	클라이언트 인증 여부를 선택한다. '예'로 설정하면 Key Store를 설정할 수 있다.
스토어 패스워드	해당 Trust Store를 접근하기 위한 암호를 설정한다.

2.3. 아웃바운드 룰 설정

본 절에서는 아웃바운드 룰 설정방법에 대해서 설명한다. TCP 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] > [TCP 아웃바운드 룰]**를 선택하면 다음과 같이 **TCP 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 설정 화면이 나타난다.

아웃바운드 룰 정의

- 프로토콜: TCP
- 아웃바운드 룰 ID *: tcpOutboundRule
- 아웃바운드 룰 이름 *: TCP아웃바운드룰
- 요청처리 타임아웃(ms): 10,000

Endpoint * (Group)

서비스 키 설정

서비스 키

요청 메시지

이름	메시지 ID	타입 ID	배열 선택

☐ 코릴레이션

메시지 ID	필드 ID

응답 헤더 메시지

☐ 응답 헤더 메시지 사용
헤더 메시지 필드 정보를 읽어 정상과 정상 오류응답을 구분한다.

이름	메시지 ID	타입 ID

☐ 응답 종류 확인 필드 메시지 ID 필드 ID

Field Value	Type

정상 응답 메시지

이름	메시지 ID	타입 ID	배열 선택

☐ 코릴레이션

메시지 ID	필드 ID

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.

항목	설명
Endpoint(Group) *	아웃바운드 룰이 등록될 엔드포인트 또는 엔드포인트 그룹을 선택한다. 해당 프로젝트가 업무시스템에 할당되고 서버에 접속한 상태여야만 선택이 가능하다.
서비스 키 설정	복수거래의 아웃바운드 룰 키를 사용하는 경우 서비스 키에 따라 아웃바운드 룰을 매칭할 때 설정한다.
설명	아웃바운드 룰에 대한 설명을 입력한다.
응답 수신 방식	메시지에 대한 응답 수신 방식을 선택한다. <ul style="list-style-type: none"> • SYNC : 외부로 요청을 보냈던 엔드포인트로 응답이 들어온다. • ASYNC : 응답이 요청을 보냈던 엔드포인트가 아닌 다른 엔드포인트로 들어 올 수 있다. 거래의 호출 서비스의 비동기 응답 거래를 설정해서 사용할 수 있다. 어댑터에서 코릴레이션하기 때문에 같은 어댑터를 사용해야 하고 서로 다른 어댑터로는 사용이 불가능하다. • ONEWAY : 응답이 없는 ONEWAY이다.
ByPass	메시지 변환(marshal/unmarshal) 없이 요청으로 들어온 전문을 그대로 전송하고 싶을 경우 선택한다. ByPass가 체크 되면 요청 메시지, 응답 헤더 메시지, 정상 응답 메시지를 설정할 수 없고 요청으로 들어온 전문을 그대로 사용한다.
DynamicInetAddress 사용	수신한 메시지의 특정 필드로 IP, Port를 설정한다. <ul style="list-style-type: none"> • Host : 수신한 메시지의 특정 필드로 Host를 설정한다. 검색을 누르면 메시지 ID와 필드 ID를 선택할 수 있다. • Port : 수신한 메시지의 특정 필드로 Port를 설정한다. 검색을 누르면 메시지 ID와 필드 ID를 선택할 수 있다.

• 아웃바운드 룰 메시지 정의

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

항목	설명
요청 메시지	아웃바운드 룰에서 사용될 요청 메시지를 선택한다. <ul style="list-style-type: none"> • 코릴레이션 : 응답 수신 방식이 ASYNC일 때 사용될 요청 메시지의 코릴레이션 필드를 선택한다.
정상 응답 메시지	아웃바운드 룰에서 사용될 응답 메시지를 선택한다. <ul style="list-style-type: none"> • 코릴레이션 : 응답 메시지로 아웃바운드 룰을 찾을 때 응답 메시지의 코릴레이션 필드를 선택한다.

항목	설명
응답 헤더 메시지	<p>응답 헤더 메시지를 선택한다.</p> <p>응답 헤더 메시지는 응답 메시지가 정상과 업무 오류로 나누어져 있을 경우 사용된다(일반적인 에러가 아닌 정상 메시지 형태로 업무 오류 메시지를 수신할 경우 사용).</p> <ul style="list-style-type: none"> 응답 종류 확인 : 응답 헤더 메시지를 사용할 경우에 정상 응답 메시지와 업무 오류 응답 메시지를 구분하기 위한 메시지 필드와 해당 값에 따른 정상 또는 업무 오류를 입력한다.

• TCP 아웃바운드 룰 타임아웃

[2023.02 stable 버전 기능추가]

TCP 동기방식(Request 전송모드)이고 아웃바운드 룰 '요청처리 타임아웃' ADT_00028 에러가 발생한 경우 요청 재시도 기능이 추가되었다.

- TCP 아웃바운드 룰의 응답 수신 방식이 SYNC인 경우 **재시도 횟수** 항목이 활성화된다. (기본값: 0)

```
<ns0:tcpOutboundRuleInfo>
<ns0:callType>Sync</ns0:callType>
<ns0:retryNum>3</ns0:retryNum>
</ns0:tcpOutboundRuleInfo>
```

- '요청처리 타임아웃' ADT_00028 에러가 발생한 경우 '재시도 횟수' 만큼 재시도를 요청하며 기존 연결과 동일한 세션으로 요청한다.
- 재시도 시 WARNING 레벨 로그로 로깅한다.

[WARNING] 1040 | anylink.adapter.framework.SharedThreadAdapter | | | | | Tcp Outbound Retry - Remaining: 1 | #]

TCP 아웃바운드 룰

아웃바운드 룰 정의

- 프로토콜 TCP
- 아웃바운드 룰 ID * TCP
- 아웃바운드 룰 이름 * TCP
- 요청처리 타임아웃(ms) : 1,000
- Endpoint * (Group) EP_OUT_0002 Q 검색
- 서비스 키 설정

서비스 키

- 설명
- 응답 수신 방식 SYNC
- ByPass ☐
- DynamicIpAddress 사용 ☐
- 재시도 횟수 4

TCP 아웃바운드 룰 타임아웃

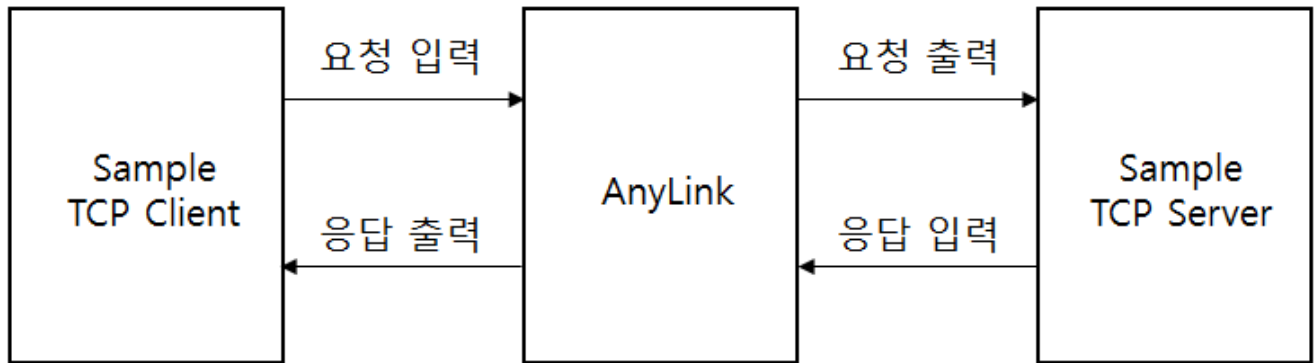
3. 예제

본 장에서는 AnyLink TCP 어댑터를 사용한 예제를 설명한다.

3.1. 개요

사용된 전문은 모두 Fixed Length에 통신헤더(4byte) 방식을 기본으로 한다.

다음은 TCP 기본 거래 구성이다.



기본 거래 구성도

다음은 거래 순서를 수행하는 과정에 대한 설명이다.

1. Sample TCP Client를 통해 요청 입력 전문을 전송한다.
2. 인바운드 어댑터로 전문 수신 후 서비스 플로우를 수행한다.
3. 서비스 플로우에서 요청 입력 전문을 요청 출력 전문으로 매핑한다.
4. 아웃바운드 어댑터를 통해 요청 출력 전문을 Sample TCP Server로 전송한다.
5. Sample TCP Server에서 응답 입력 전문을 전송한다.
6. 서비스 플로우에서 응답 입력 전문을 응답 출력 전문으로 매핑한다.
7. 인바운드 어댑터로 응답 출력 전문을 Sample TCP 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

필드명	설명	타입	크기
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 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. 인바운드 어댑터 생성

다음은 TCP 인바운드 어댑터를 생성하는 예제이다.

기본정보

어댑터 아이디*	<input type="text" value="TCP_IN_ADT"/>	어댑터 이름*	<input type="text" value="TCP_IN_ADT"/>	어댑터 종류*	<input type="text" value="TCP"/>
설명	<input type="text" value="TCP 인바운드 어댑터"/>			업무시스템 이름*	<input type="text" value="ANL001"/>
배포 사유	<input type="text"/>				

아웃바운드 스레드 풀 아이디

인바운드 어댑터 등록 화면

• 기본정보

항목	설정값
어댑터 아이디	TCP_IN_ADT
어댑터 이름	TCP_IN_ADT
어댑터 종류	TCP
설명	TCP 인바운드 어댑터
업무시스템	ANL001

• [상세설정] 탭

항목	설정값
아웃바운드 스레드 풀 아이디	사용 안함

3.2.2. 아웃바운드 어댑터 생성

다음은 TCP 아웃바운드 어댑터를 생성하는 예제이다.

기본정보

어댑터 아이디*	<input type="text" value="TCP_OUT_ADT"/>	어댑터 이름*	<input type="text" value="TCP_OUT_ADT"/>	어댑터 종류*	<input type="text" value="TCP"/>
설명	<input type="text" value="TCP 아웃바운드 어댑터"/>			업무시스템 이름*	<input type="text" value="ANL001"/>
배포 사유	<input type="text"/>				

상세설정	접속 제어
아웃바운드 스레드 풀 아이디 <input type="text" value="미사용"/>	

아웃바운드 어댑터 등록 화면

• 기본정보

항목	설정값
어댑터 아이디	TCP_OUT_ADT
어댑터 이름	TCP_OUT_ADT
어댑터 종류	TCP
설명	TCP 아웃바운드 어댑터
업무시스템	ANL001

• [상세설정] 탭

항목	설정값
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 TCP 인바운드 엔드포인트를 등록하는 예제이다. 본 예제에서는 [SSL] 탭은 설정하지 않는다.

• 기본정보

기본정보					
업무시스템 이름	ANL001	어댑터 종류	TCP	어댑터 이름	TCP_IN_ADT
엔드포인트 아이디*	TCP_IN_EP	엔드포인트 이름*	TCP_IN_EP	엔드포인트 그룹 이름	
엔드포인트 상태	Running	부팅 초기값	Running	통신 방향	Inbound
설명	TCP 인바운드 엔드포인트				
배포 사유					

인바운드 엔드포인트 등록 화면 - 기본정보

항목	설정값
엔드포인트 아이디	TCP_IN_EP
엔드포인트 이름	TCP_IN_EP
엔드포인트 상태	Running
부팅 초기값	Running
통신 방향	Inbound
설명	TCP 인바운드 엔드포인트

• [연결정보] 탭

연결정보

상세설정

메시지/장애처리

메시지 변경 옵션

SSL

TCP 소켓 옵션

접속 제어

연결방식

연결방식

Server

Client

기본 인코딩

서버연결유지

No

Agent 사용여부

사용

미사용

주소

IP/호스트 (*)

포트 (*)

30000

타임아웃/연결수

Idle 타임아웃*

60000

ms

동기/비동기 여부

동기

비동기

최대 연결 수*

5

새 연결 허용

TCP 연결 핸들러

연결 종료 사용 유무

연결 종료 아이디

연결 종료 종료 사용 유무

연결 종료 종료 아이디

TCP 타임아웃

타임아웃 close 소켓

타임아웃 종료 콜

타임아웃 종료 아이디

타임아웃 헬스체크

요청 메시지

응답 타임아웃

s ?

재시도 횟수

?

헬스체크 메시지

응답 Ack 사용

인바운드 엔드포인트 등록 화면 - [연결정보]

항목	설정값
연결 방식	Server
서버연결유지	No
포트	30000
Idle 타임아웃	60000
동기/비동기 여부	동기
최대 연결 수	5

• [상세설정] 탭

3. 예제 | 37

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

메시지 핸들러
핸들러

거래그룹/거래/룰
manual.tcp.tcpCase1.tx01
선택

메시지 분할 거래
☐

배치모드
☐

복수거래설정
☐

인바운드 엔드포인트 등록 화면 - [상세설정]

항목	설정값
거래그룹/거래	manual.tcp.tcpCase1.tx01 (거래그룹/거래 항목은 직접 입력해도 되고 차후 거래그룹/거래 생성 후 선택하여도 무방하다.)

• [메시지/장애처리] 탭

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션

메시지 길이지정 방식

헤더에 길이정보 포함
Outbound 별도 설정

메시지 길이지정 방식
헤더에 길이정보 포함

Outbound 별도 설정

헤더 종류
통신헤더

헤더 길이
4
Bytes

Primitive 사용 유무
☐

헤더 사이즈 포함 유무
☐

오프셋
0

길이
4

메시지 정합성 검사
☐

길이정보 키 사용 유무
☐
?
+ 추가

StxExt 사용유무
☐

TCP 에이전트 사용
☐

인바운드 엔드포인트 등록 화면 - [메시지/장애처리]

항목	설정값
메시지 길이지정 방식	헤더에 길이정보 포함
헤더 종류	통신 헤더
헤더 길이	4
오프셋	0
길이	4

3.3.2. 아웃바운드 엔드포인트 생성

다음은 TCP 아웃바운드 엔드포인트를 등록하는 예제이다. 아웃바운드 엔드포인트는 [상세설정] 탭을 설정하지 않는다. 본 예제에서는 [SSL] 탭은 설정하지 않는다.

• 기본정보

기본정보

업무시스템 이름	ANL001	어댑터 종류	TCP	어댑터 이름	TCP_OUT_ADT
엔드포인트 아이디*	TCP_OUT_EP	엔드포인트 이름*	TCP_OUT_EP	엔드포인트 그룹 이름	
엔드포인트 상태	Running	부팅 초기값	Running	통신 방향	Outbound
설명	TCP 아웃바운드 엔드포인트				
배포 사유					

아웃바운드 엔드포인트 등록 화면 - 기본정보

항목	설정값
엔드포인트 아이디	TCP_OUT_EP
엔드포인트 이름	TCP_OUT_EP
엔드포인트 상태	Running
부팅 초기값	Running
통신 방향	Outbound
설명	TCP 아웃바운드 엔드포인트

• [연결정보] 탭

연결정보	상세설정	메시지/장애처리	메시지 변경 옵션	SSL	TCP 소켓 옵션
------	------	----------	-----------	-----	-----------

연결방식

연결방식 ☐ Server ☒ Client 기본 인코딩

연결 관리 여부 연결 유지 여부

Agent 사용여부 ☐ 사용 ☒ 미사용

주소

IP/호스트 (*) 포트 (*)

로컬 IP/호스트 로컬 포트

타임아웃/연결수

연결 타임아웃* ms Idle 타임아웃* ms

동기/비동기 여부 ☒ 동기 ☐ 비동기

최대 연결 수*

아웃바운드 엔드포인트 등록 화면 - [연결정보]

항목	설정값
연결 방식	Client
연결 관리 여부	No
연결 유지 여부	No
IP/호스트	192.168.15.75 (Sample TCP 서버가 구동되어 있는 장비의 장비의 IP를 지정한다.)

항목	설정값
포트	38000
연결 타임아웃	3000
Idle 타임아웃	60000
동기/비동기 여부	동기
최대 연결 수	5

• [메시지/장애처리] 탭

연결정보
상세설정
메시지/장애처리
메시지 변경 옵션
SSL
TCP 소켓 옵션
접속 제어

메시지 길이 지정 방식

메시지 길이 지정 방식
헤더에 길이정보 포함
Outbound 볼드 설정

헤더 종류
통신헤더
헤더 길이
4
Bytes

Primitive 사용 유무

헤더 사이즈 포함 유무

오프셋
0
길이
4

메시지 정합성 검사

길이정보 키 사용 유무
+ 추가

Stx/Ex 사용유무

TCP 에이전트 사용

아웃바운드 엔드포인트 등록 화면 - [메시지/장애처리]

항목	설정값
메시지 길이 지정 방식	헤더에 길이정보 포함
헤더 종류	통신 헤더
헤더 길이	4
오프셋	0
길이	4

3.4. 스튜디오 리소스 생성

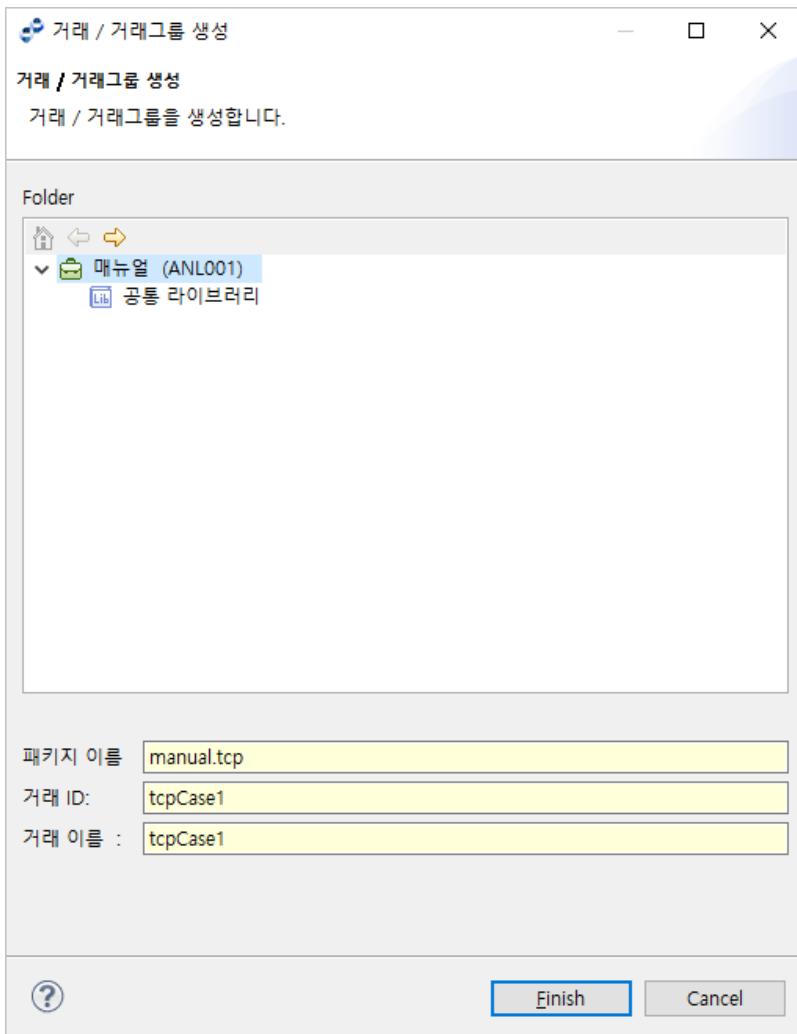
본 절에서는 스튜디오에서 리소스를 생성하는 방법에 대해서 설명한다.



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.

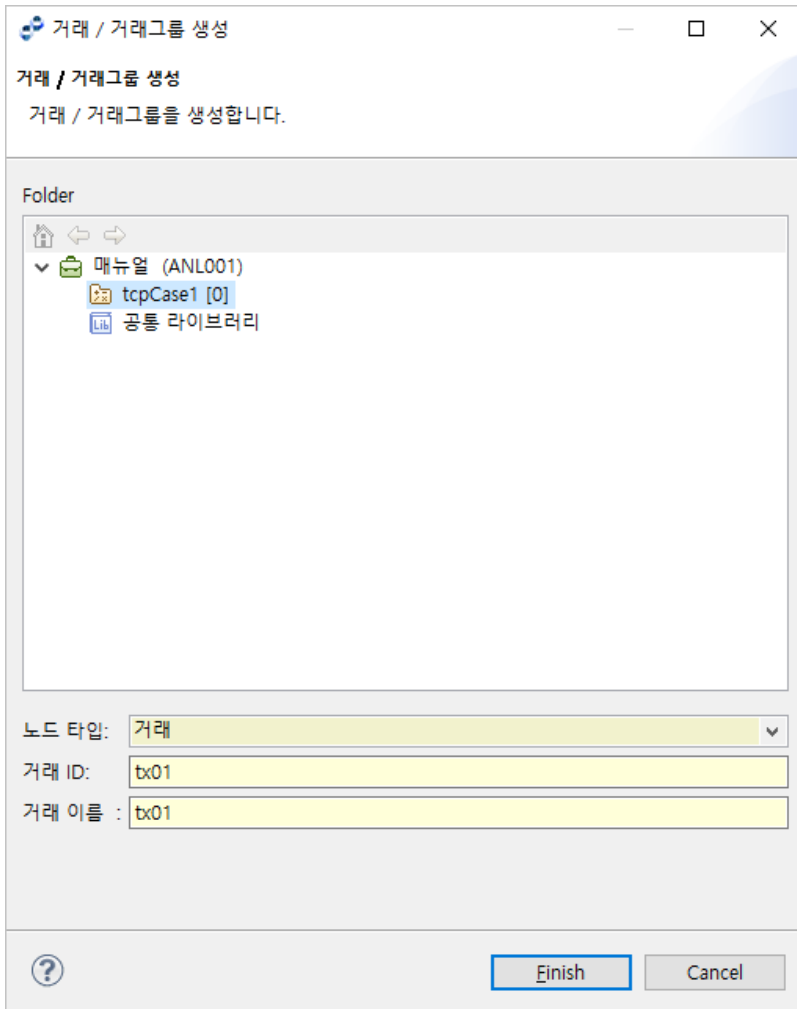


거래그룹 생성

항목	설정값
패키지 이름	manual.tcp
거래 이름	tcpCase1
거래 ID	tcpCase1

3.4.2. 거래 생성

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



거래 생성

항목	설정값
노드 타입	거래
거래 이름	tx01
거래 ID	tx01

3.4.3. 메시지 생성

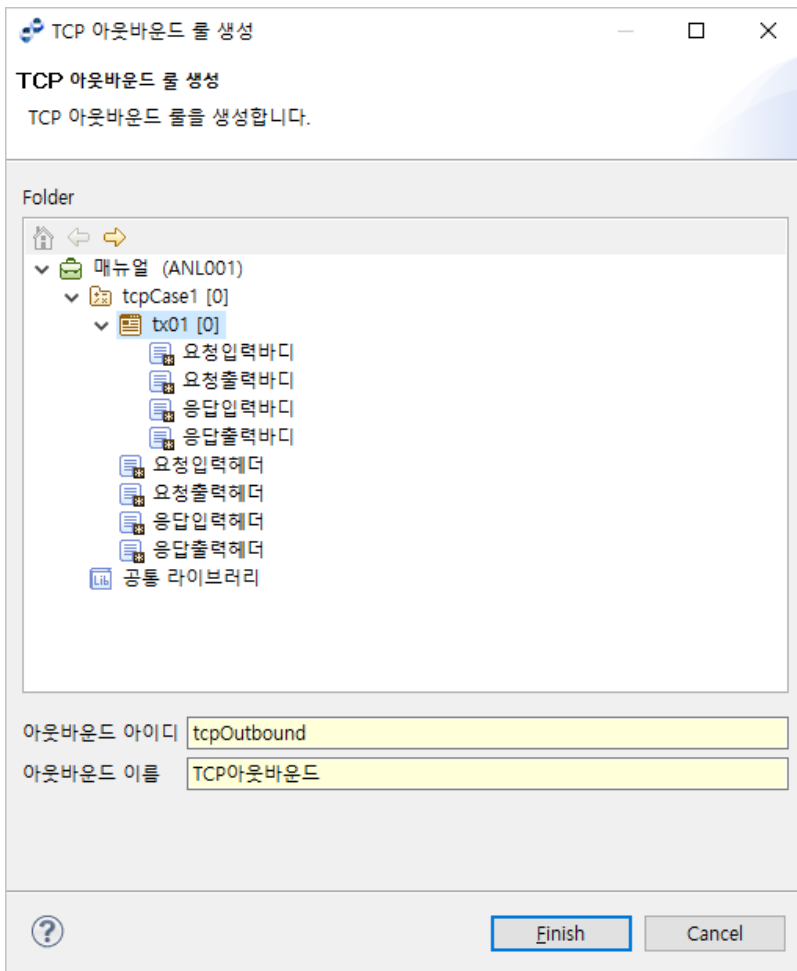
요청 입력 헤더/바디, 요청 출력 헤더/바디, 응답 입력 헤더/바디, 응답 출력 헤더/바디 메시지를 생성한다.



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

3.4.4. 아웃바운드 룰 생성

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

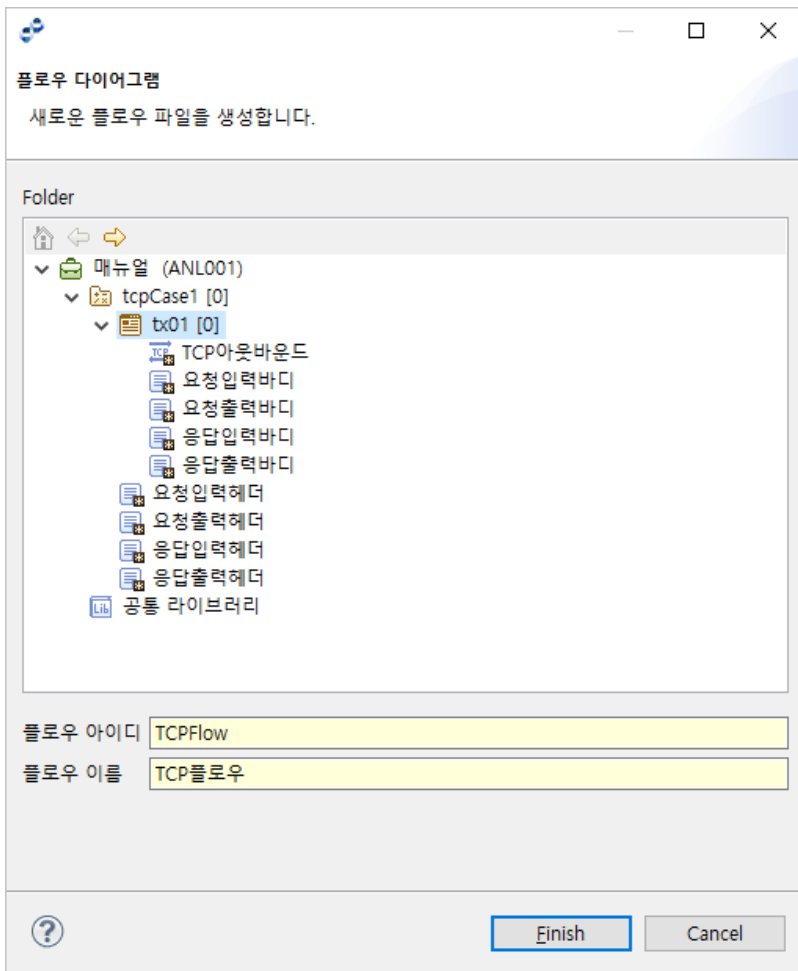


아웃바운드 룰 생성

항목	설정값
아웃바운드 이름	TCP아웃바운드
아웃바운드 아이디	tcpOutbound

3.4.5. 플로우 생성

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



플로우 룰 생성

항목	설정값
플로우 아이디	TCPFlow
플로우 이름	TCP플로우

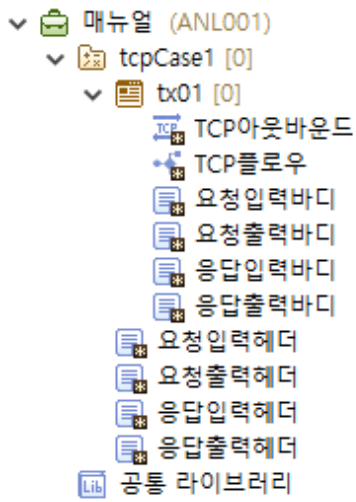
3.5. 스튜디오 리소스 설정

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

다음은 거래그룹을 설정하는 예제 이다. **[파싱 옵션]** 탭은 설정하지 않는다.

• [거래그룹 정보] 탭

거래그룹 정의

기본정보

- 거래그룹 ID: tcpCase1
- 패키지 이름: manual.tcp
- 거래그룹 이름: tcpCase1
- 거래 타입: NONE
- 버전: 0
- 설명:
- Bypass 설정: NO
- XA 설정: NO

요청 메시지

이름	메시지 ID	타입 ID	배열 선택
요청 입력헤더	ReqInHeader	Default Type	설정 안함

네임스페이스 URI: NONE

정상 응답 메시지

이름	메시지 ID	타입 ID	배열 선택
응답 출력헤더	ResOutHeader	Default Type	설정 안함

네임스페이스 URI: NONE

업무오류 응답 메시지

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

네임스페이스 URI: NONE

오류 응답처리

- 오류 응답 방식: NONE

거래그룹 설정 - [거래그룹 정보]

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

- [거래그룹 옵션] 탭

거래그룹 옵션

거래그룹 옵션 정의

- 거래 플로우 타임아웃(ms)
- 연결 고정
- 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

거래그룹 설정 - [거래그룹 옵션]

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

- [파싱정보] 탭

[illegible]

거래그룹 설정 - [파싱 정보]

항목	설정값
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

거래 정의

기본정보

- 거래 ID: tx01
- 패키지 이름: manual.tcpCase1
- 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
로컬 파트
그룹: NONE
그룹번호

정상 응답 메시지

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

네임스페이스 URI
로컬 파트
그룹: NONE
그룹번호

업무오류 응답 메시지

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

네임스페이스 URI
로컬 파트
그룹: NONE
그룹번호

오류 응답처리

- 오류 응답 방식: PARENT

거래 설정 - [거래정보]

항목	설정값
요청 메시지	요청입력바디
정상 응답 메시지	응답출력바디
Bypass 설정	NO
XA 설정	NO
오류 응답 방식	PARENT
서비스 타입	FLOW
서비스 이름	TCPFlow_Message_STRT_Event_69400C(랜덤으로 생성됨)

• [거래 옵션] 탭

3. 예제 | 47

거래 옵션

거래 옵션 정의

- 거래 플로우 타임아웃(ms)
- 타임아웃 설정 안함 ☒

- 연결 고정
- 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
- 중복거래 보관 주기
- 거래 우선순위

거래 설정 - [거래 옵션]

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

• [파싱 정보] 탭

파싱 정보

파싱 정보 정의

거래 식별코드

코드값	종류
[A001]	REQUEST

거래 설정 - [파싱 정보]

◦ 거래 식별코드

항목	설정값
코드값	A001
종류	REQUEST

3.5.3. 아웃바운드 룰 설정

다음은 TCP 아웃바운드 룰 설정 예제이다.

TCP 아웃바운드 룰

아웃바운드 룰 정의

- 프로토콜: TCP
- 아웃바운드 룰 ID: tcpOutbound
- 아웃바운드 룰 이름: TCP아웃바운드
- 요청처리 타임아웃(ms): 10,000
- Endpoint (Group): TCP_OUT_EP Q 검색
- 서비스 키 설정

서비스 키

➕ 추가 ➖ 삭제

• 설명

- 응답 수신 방식: SYNC ▼
- ByPass: ☐
- DynamicIpAddress 사용: ☐

요청 메시지

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

⏮ ⏭ ↺ ↻ ⏩ ⏪ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ ⏰ ⏱ ⏲ ⏳ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ ⏰ ⏱ ⏲ ⏳

☐ 코릴레이션

메시지 ID	필드 ID

➕ 추가 ➖ 삭제

응답 헤더 메시지

☐ 응답 헤더 메시지 사용

헤더 메시지 필드 정보를 읽어 정상과 정상 오류응답을 구분한다.

이름	메시지 ID	타입 ID

⏮ ⏭ ↺ ↻ ⏩ ⏪ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ ⏰ ⏱ ⏲ ⏳ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ ⏰ ⏱ ⏲ ⏳

☐ 응답 종류 확인 필드 메시지 ID 필드 ID Q 검색

Field Value	Type

➕ 추가 ➖ 삭제

정상 응답 메시지

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

⏮ ⏭ ↺ ↻ ⏩ ⏪ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ ⏰ ⏱ ⏲ ⏳ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ ⏰ ⏱ ⏲ ⏳

☐ 코릴레이션

메시지 ID	필드 ID

➕ 추가 ➖ 삭제

아웃바운드 룰 설정

항목	설정값
요청처리 타임아웃(ms)	10000
응답 수신 방식	Sync
Endpoint	TCP_OUT_EP
요청 메시지	요청출력헤더(ReqOutHeader) 요청출력바디(ReqOutBody)
정상 응답 메시지	응답입력헤더(ResInHeader) 응답입력바디(ResInBody)

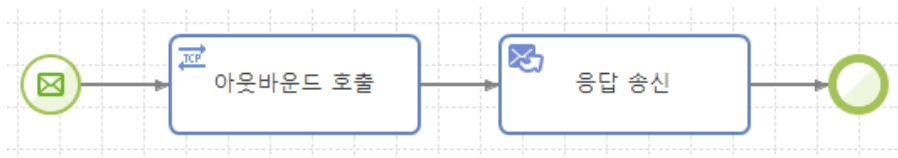
3.5.4. 플로우 설정

다음의 과정으로 플로우를 설정한다.

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

플로우 그리기

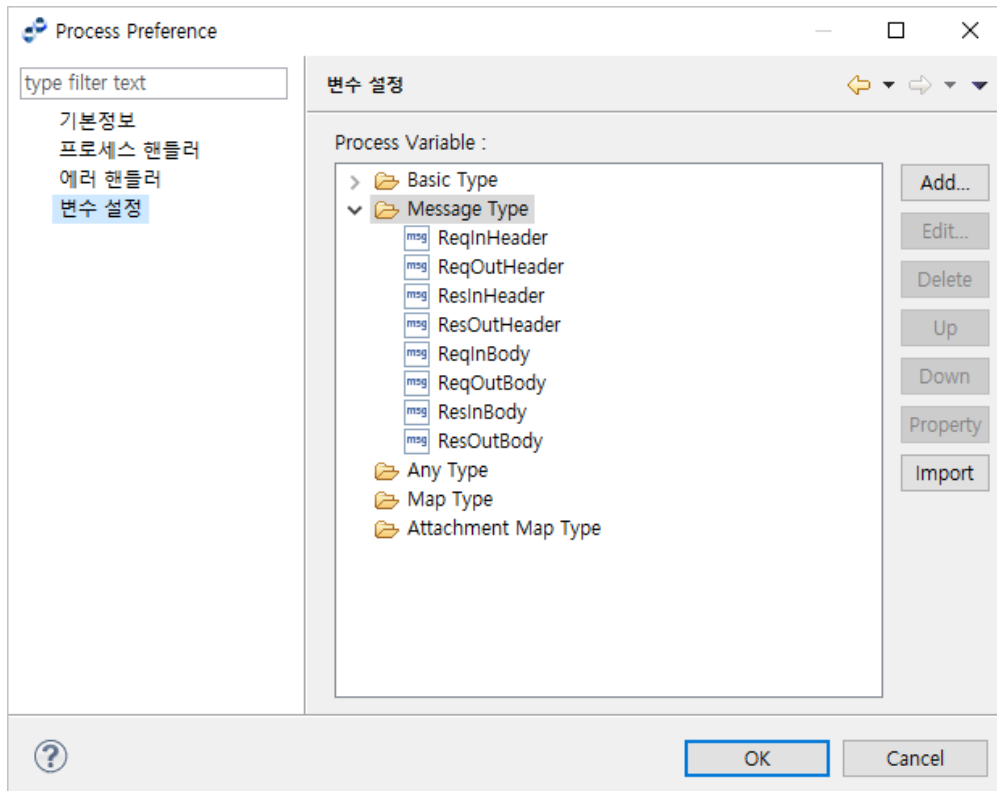
다음은 예제에 대한 서비스 플로우이다.



서비스 플로우 에디터 - 기본 그림

변수 설정

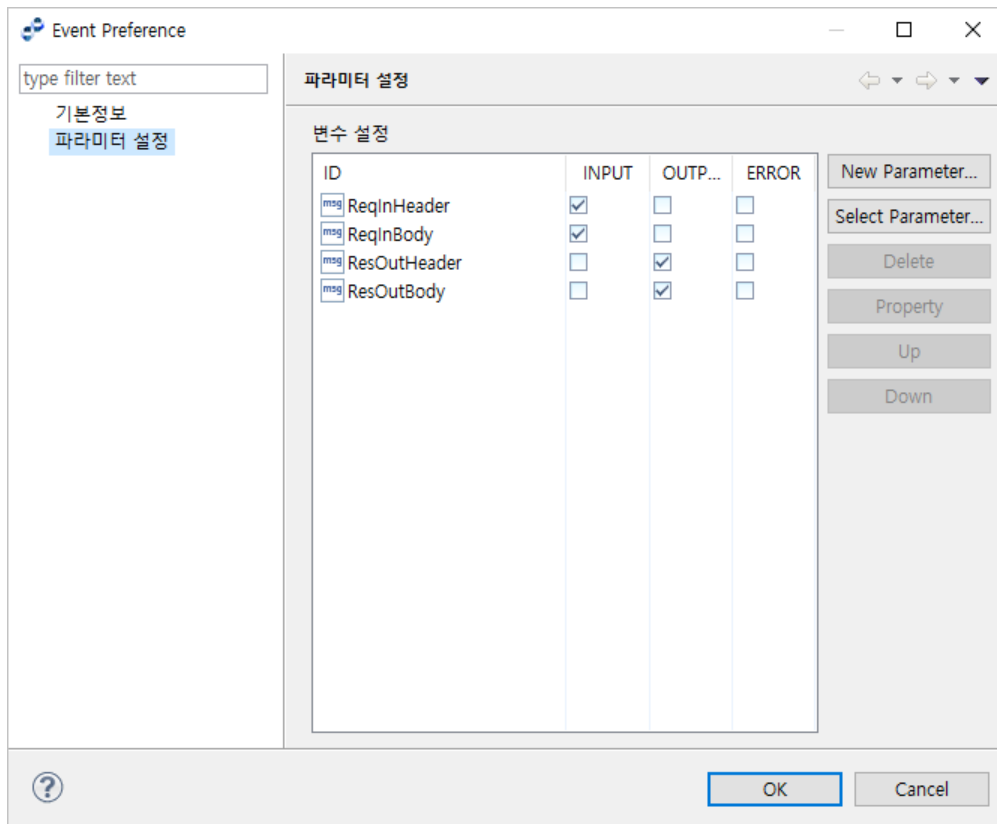
변수 설정을 하기 위해 플로우 에디터 바탕의 컨텍스트 메뉴에서 **[Property]**를 선택한다.



Property Preference - 변수 설정

메시지 이벤트 설정

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

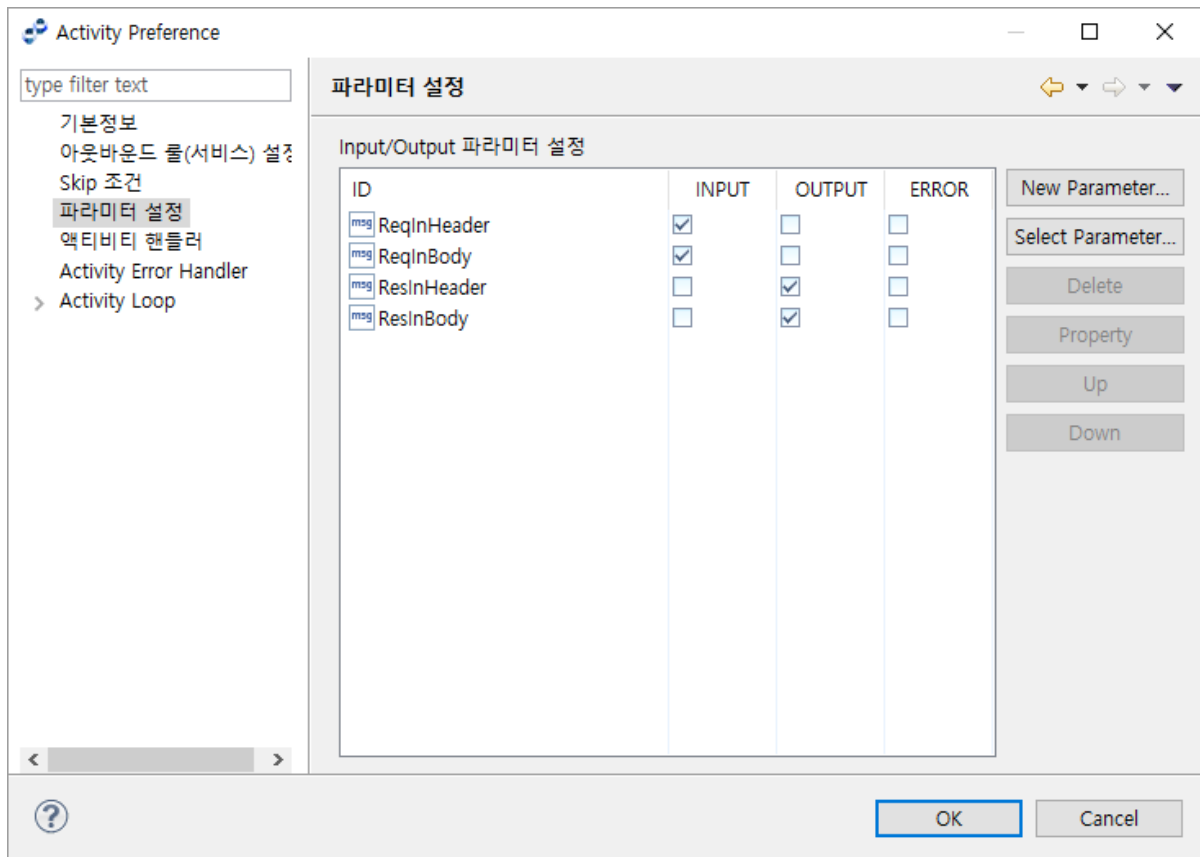


Event Preference - 파라미터 설정

아웃바운드 호출 설정

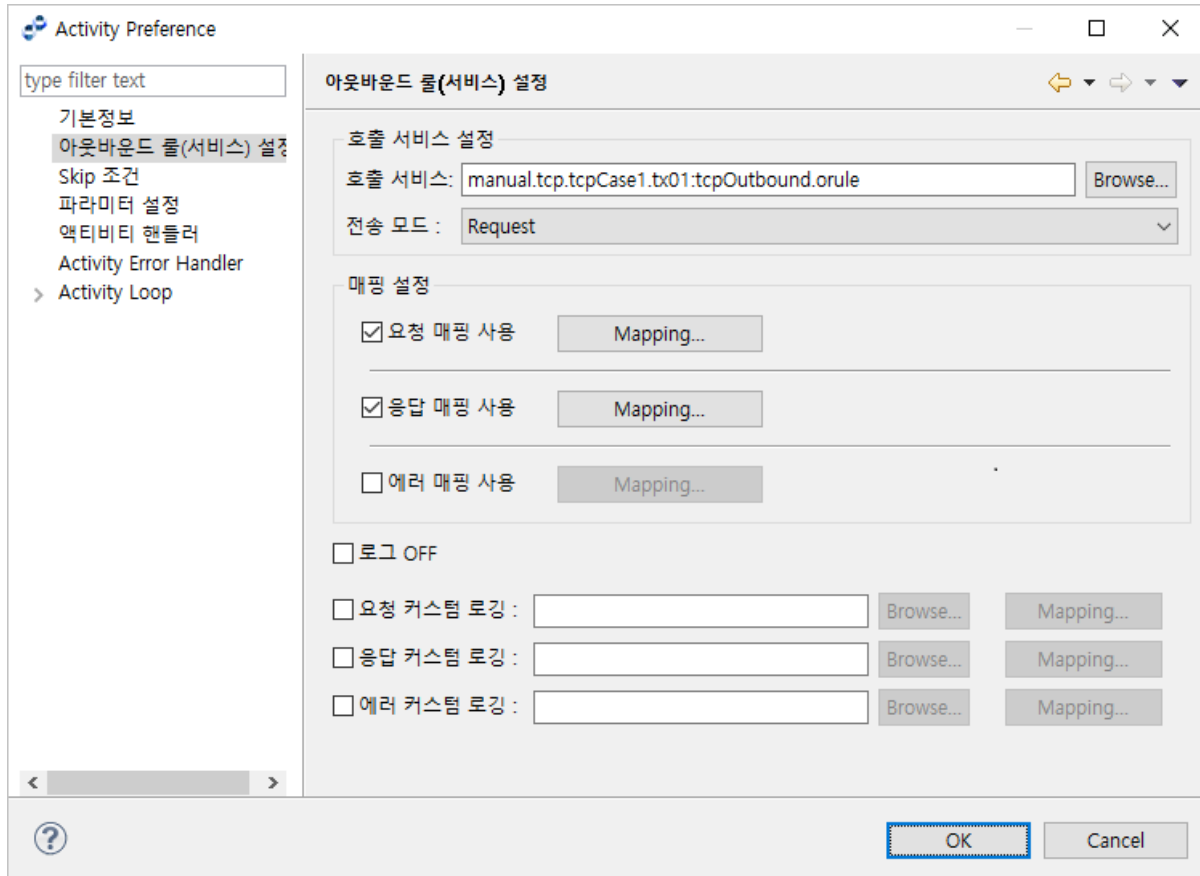
다음의 과정으로 아웃바운드 호출을 설정한다.

1. 플로우 에디터의 [아웃바운드 호출] > [아웃바운드 룰] 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.

New Mapping

Mapping
Create a new Mapping

Mapping

Source

- ReqInHeader
 - 종별코드
 - 거래코드
 - 날짜
 - 아이디
- ReqInBody
 - 지점번호
 - 담당자번호
 - 이름
 - 계정
 - 데이터

Target

- ReqOutHeader
 - 거래코드
 - 아이디
- ReqOutBody
 - 이름
 - 계정
 - 데이터

Properties

Property	Value

Location Path Expression

Properties

Property	Value

아웃바운드 호출 - 아웃바운드 룰(서비스) 설정 - 요청 매핑

New Mapping

Mapping
Create a new Mapping

Mapping

Source

- ResInHeader
 - 응답코드
 - 아이디
- ResInBody
 - 응답일자
 - 관리정보

Target

- ResInHeader
 - 응답코드
 - 아이디
- ResInBody
 - 응답일자
 - 관리정보

Properties

Property	Value

Location Path Expression

Properties

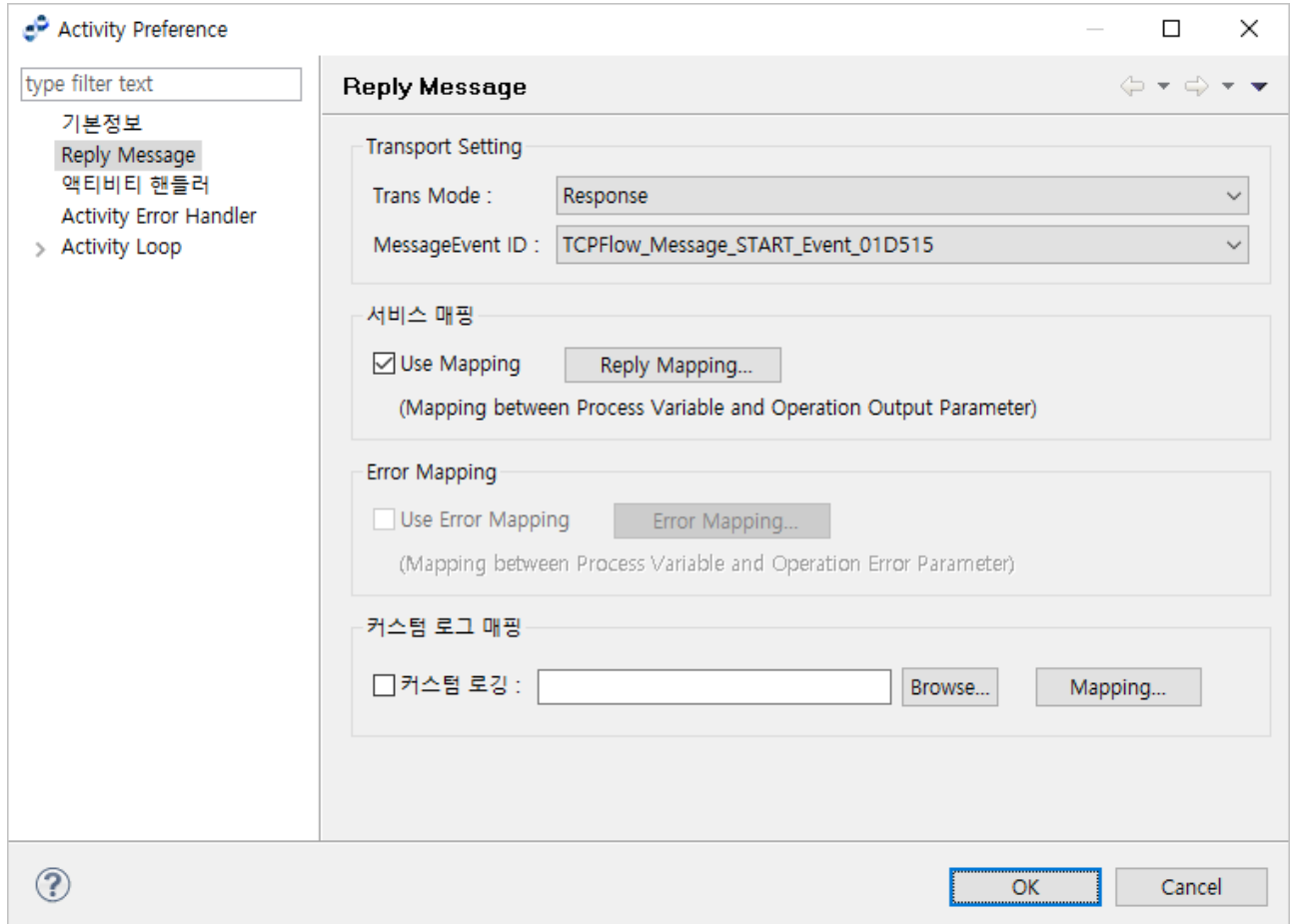
Property	Value

아웃바운드 호출 - 아웃바운드 룰(서비스) 설정 - 응답 매핑

응답 호출 설정

다음의 과정으로 응답 호출 설정을 설정한다.

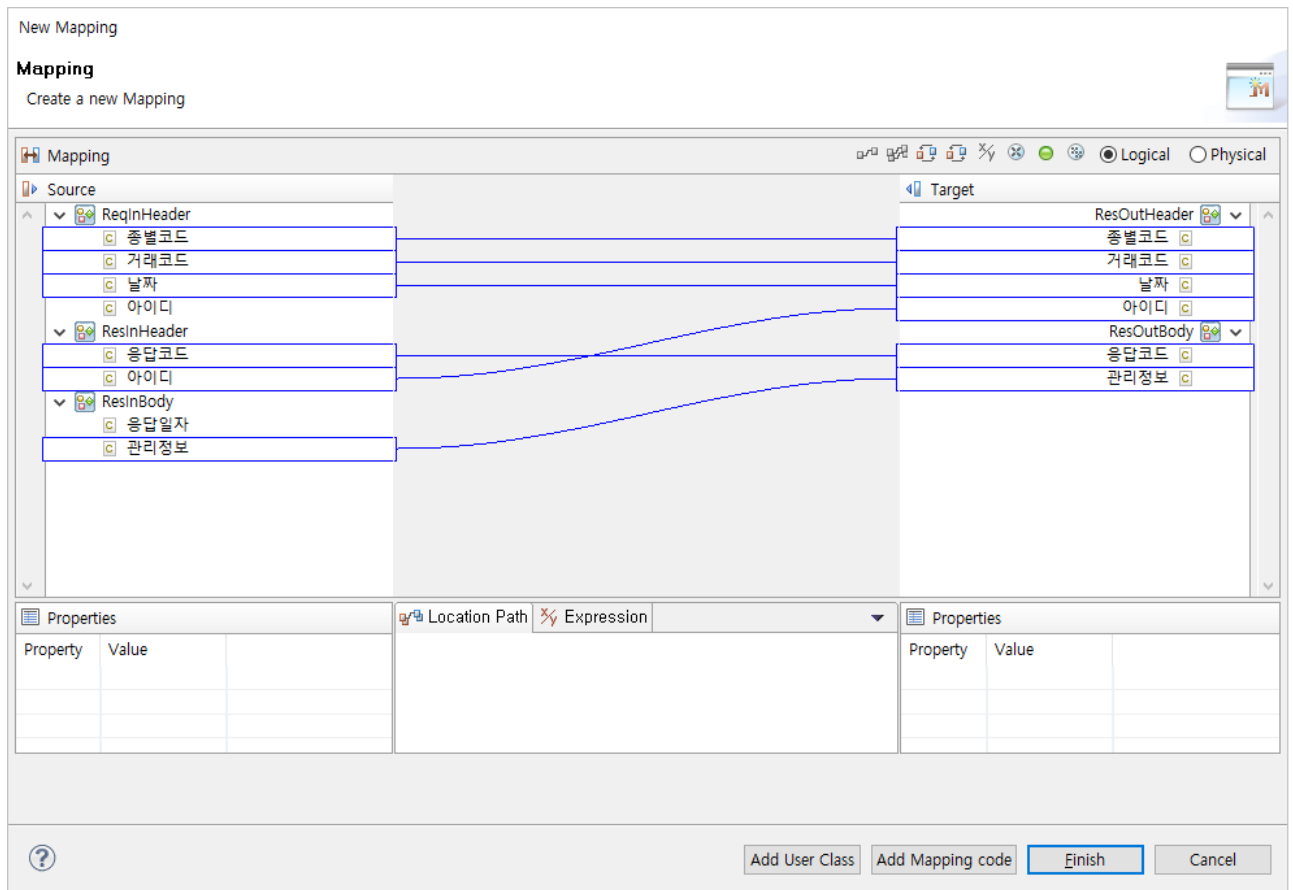
1. 플로우 에디터의 [응답 송신] > [Reply 메시지] 컨텍스트 메뉴에서 [Property]를 선택한다. **Activity Preference 화면**에서 [Reply Message]를 선택한다.



응답 송신 - Reply Message

2. **Activity Preference 화면**에서 'Use Mapping'을 체크하고 [Reply Mapping] 버튼을 클릭해서 각 매핑정보를 설정한다.

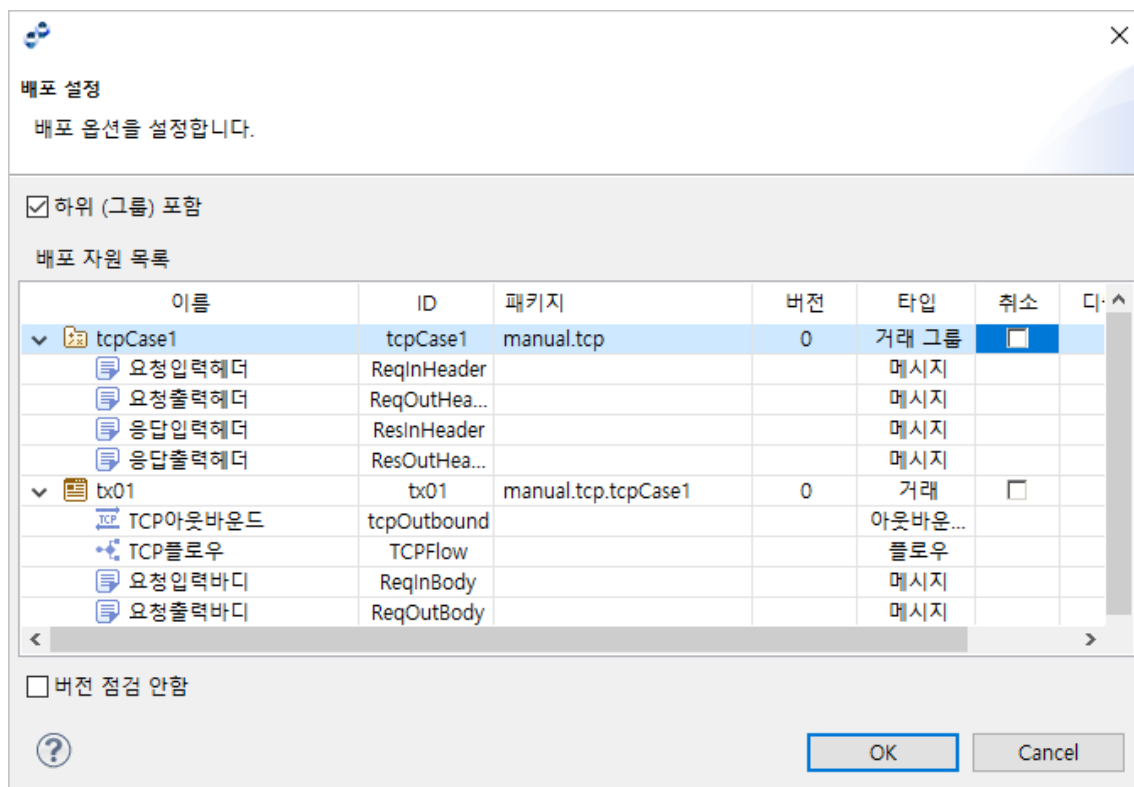
Mapping 화면의 Source 영역의 컨텍스트 메뉴에서 [Add source]를 선택하여 ReqInHeader, ResInHeader, ResInBody를 선택한 후 매핑한다.



응답 송신 - Reply Message - 서비스 매핑 - Reply Mapping

3.6. 배포

거래그룹/거래 배포정보를 설정한다.



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, 서버 프로그램과 테스트 수행 결과를 설명한다.

3.7.1. 클라이언트 프로그램

<TCPClient.java>

```
package manual.tcp;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.Socket;

public class TCPClient {
    BufferedReader br;
    BufferedWriter bw;
    InputStream is;
    OutputStream os;
    int headerLength = 4;

    String ip;
    int port;
    String message;

    public TCPClient(String ip, int port, String message, int headerLength) {
        this.ip = ip;
        this.port = port;
        this.message = message;
        this.headerLength = headerLength;
    }

    public void run() {
        try {
            System.out.println("Start Client ip:" + ip + " port:" + port);
            long startTime = System.currentTimeMillis();

            Socket socket = new Socket(ip, port);
            socket.setSoTimeout(10000);
            is = socket.getInputStream();
            os = socket.getOutputStream();

            br = new BufferedReader(new InputStreamReader(is));
            bw = new BufferedWriter(new OutputStreamWriter(os));

            System.out.println("TCPSimpleClient:Connected to " + socket.getRemoteSocketAddress());
```

```

        bw.write(message);
        bw.flush();
        System.out.println("TCPSimpleClient:Send message : " + message);

        char[] headerbuf = new char[headerLength];
        br.read(headerbuf, 0, headerLength);
        System.out.println("recv length: " + new String(headerbuf));
        int bodybufLength = Integer.parseInt(new String(headerbuf));

        char[] bodybuf = new char[bodybufLength];
        br.read(bodybuf);
        System.out.println("TCPSimpleClient:recv data: " + new String(bodybuf));

        bw.flush();
        br.close();
        bw.close();

        is.close();
        os.close();

        socket.close();

        long endTime = System.currentTimeMillis();
        long lTime = endTime - startTime;
        System.out.println("TCPSimpleClient:TIME : " + lTime + "(ms)");

    } catch (Throwable t) {
        t.printStackTrace();
    }
}

public static void main(String[] args) {
    String sendMessage = "0106AAAAA00120160701KIMCHEOLSUB01P1034LEEYEONGSUKNIGHT
THISISSAMPLEDATA";
    TCPCClient client = new TCPCClient("192.168.14.106", 30000, sendMessage, 4);
    client.run();
}
}

```

3.7.2. 서버 프로그램

<TCPServer.java>

```

package manual.tcp;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.ServerSocket;
import java.net.Socket;

```

```

public class TCPServer extends Thread {

    BufferedReader br;
    BufferedWriter bw;
    InputStream is;
    OutputStream os;
    int port;
    int headerLength = 4;

    public TCPServer(int port, int headerLength) {
        this.port = port;
        this.headerLength = headerLength;
    }

    @Override
    public void run() {
        ServerSocket serverSocket = null;
        try {
            System.out.println("Start TCP Server");
            serverSocket = new ServerSocket(port);
            while (true) {
                Socket socket = serverSocket.accept();
                System.out.println("Accept from client:" + port);

                is = socket.getInputStream();
                os = socket.getOutputStream();
                br = new BufferedReader(new InputStreamReader(is));
                bw = new BufferedWriter(new OutputStreamWriter(os));

                System.out.println("read...");

                char[] headerbuf = new char[headerLength];
                br.read(headerbuf, 0, headerLength);
                System.out.println("recv length: " + new String(headerbuf));

                int bodybufLength = Integer.parseInt(new String(headerbuf));
                char[] bodybuf = new char[bodybufLength];

                br.read(bodybuf, 0, bodybufLength);

                System.out.println("recv data: " + new String(bodybuf));

                System.out.println("write...");
                String retMessage = "00261000KIMCHEOLSU20160720SP01";
                System.out.println("send message: " + retMessage);
                bw.write(retMessage);
                System.out.println("write done");

                bw.flush();
                br.close();
                bw.close();
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (serverSocket != null) {
                    serverSocket.close();
                }
            }
        }
    }
}

```

```

        }
    } catch (Throwable t) {
    }
}

public static void main(String[] args) {
    TCPServer server = new TCPServer(38000, 4);
    server.start();

    try {
        Thread.sleep(600000);
    } catch (InterruptedException e) {
    }
}
}

```

3.7.3. 테스트 수행

- 클라이언트 결과 화면

```

Start Client ip:192.168.14.106 port:30000
TCPSimpleClient:Connected to /192.168.14.106:30000
TCPSimpleClient:Send message : 0106AAAAA00120160701KIMCHEOLSUB01P1034LEEYEONGSUKNIGHT
THISISSAMPLEDATA
recv length: 0034
TCPSimpleClient:recv data: AAAAA00120160701KIMCHEOLSU1000SP01
TCPSimpleClient:TIME : 268(ms)

```

- 서버 결과 화면

```

Start TCP Server
Accept from client:38000
read...
recv length: 0086
recv data: A001KIMCHEOLSULEEYEONGSUKNIGHT      THISISSAMPLEDATA
write...
send message: 00261000KIMCHEOLSU20160720SP01
write done

```