

2022년 3월 22일 화요일

DASH 운용소프트웨어 매뉴얼

미디어플랫폼팀 정동욱

목차

1. 빌드

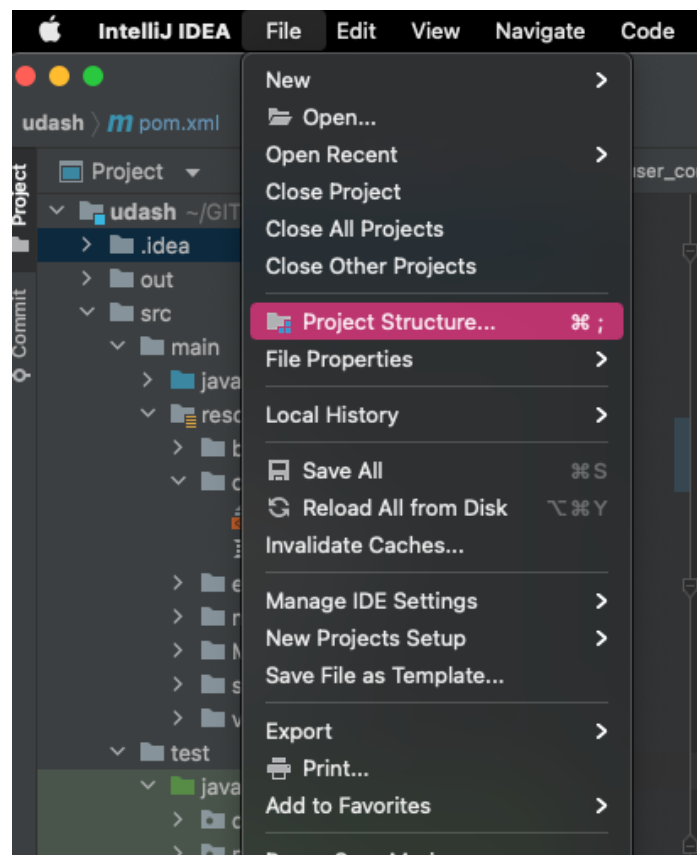
2. 설치

3. 설정

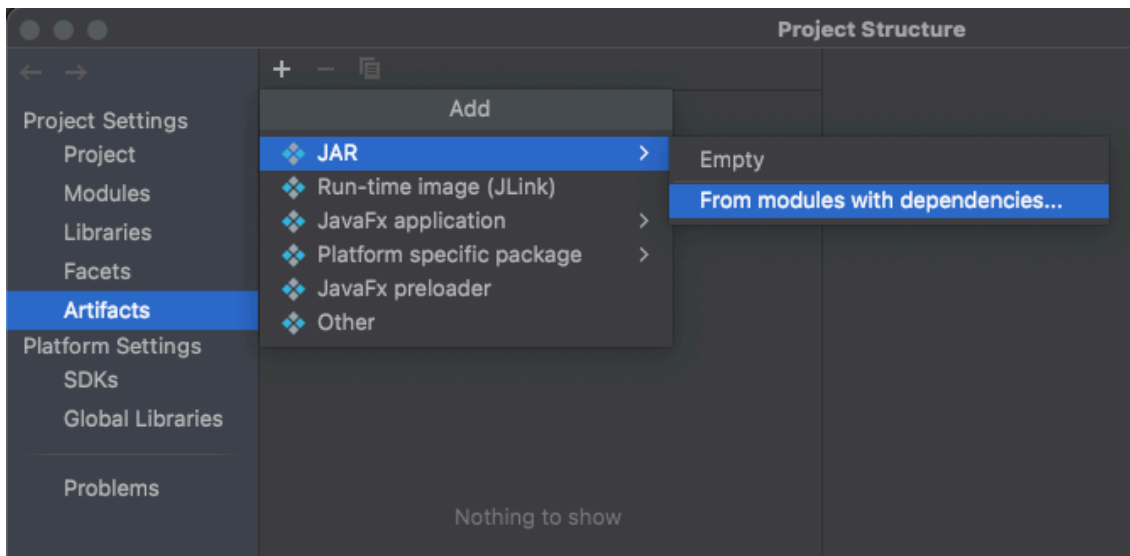
4. 실행

1. 빌드

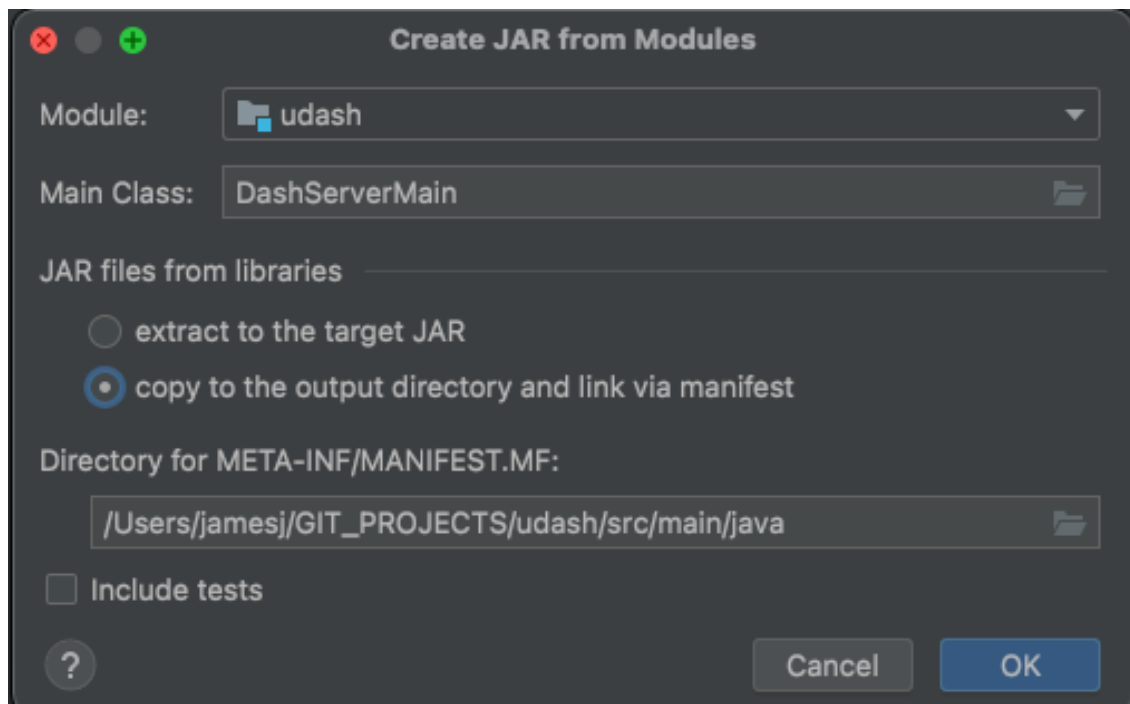
- JAVA 11 버전 사용
- 먼저 상단의 [File] 메뉴 선택 후, [Project Structure] 메뉴 진입



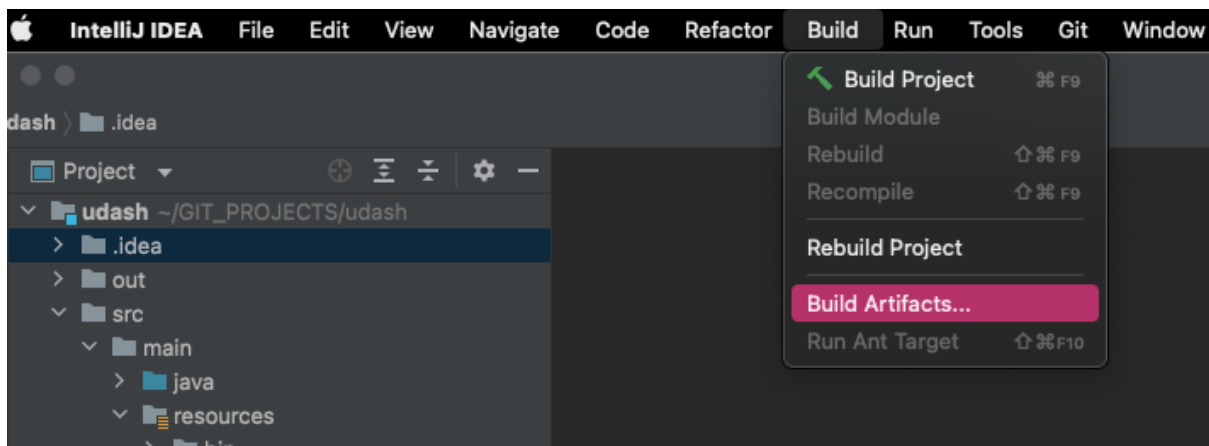
- [Artifacts] 메뉴 선택 후, [+] 버튼 클릭하여
[JAR] > [From modules with dependencies] 메뉴 클릭



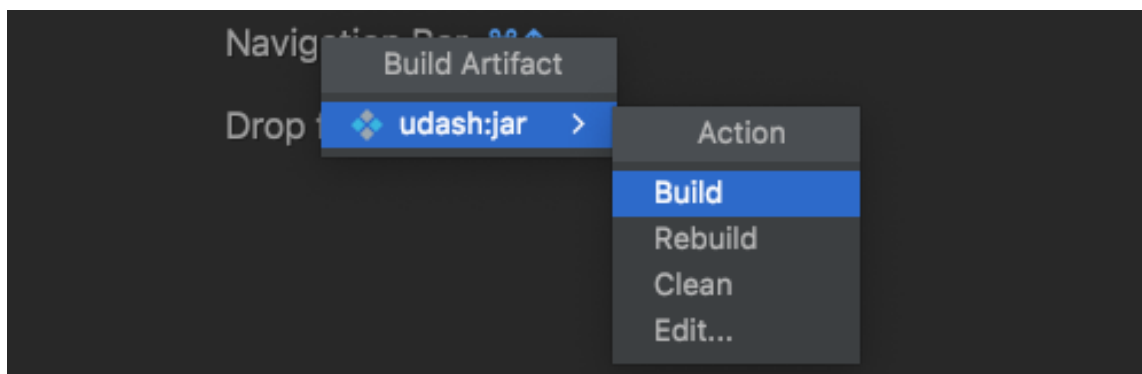
- [Main Class] 에서 [DashServerMain] 선택 후,
[copy to the output directory and link via manifest] 옵션 선택 후,
[OK] 버튼 클릭



- [copy to the output directory and link via manifest] 옵션을 선택한 이유는,
현 프로젝트에 사용되는 외부 라이브러리가 많기 때문에, (빌드 시간 오래 걸림)
다른 라이브러리를 미리 복사해놓고,
UDASH 만 따로 빌드하기 위함이다.
- 상단의 [Build] 메뉴 선택 후,
[Build Artifacts...] 메뉴 클릭



- [Build] 메뉴 클릭



2. 설치

2-1) Maven 으로 RPM 빌드 후(mvn rpm:rpm),
설치하려는 리눅스 서버에 RPM 설치 진행

```
[udash@MEDIA-DEV33 udash]$ l
합계 16
drwxr-xr-x 9 udash ustream 107 3월 22 08:05 .
drwx----- 8 udash ustream 236 3월 21 21:59 ..
drwxr-xr-x 2 root root 20 3월 17 11:51 bin
drwxr-xr-x 2 udash ustream 46 3월 21 21:59 config
drwxr-xr-x 2 udash ustream 12288 3월 17 12:15 lib
drwxr-xr-x 2 udash ustream 245 3월 21 21:52 logs
drwxr-xr-x 4 udash ustream 30 3월 17 14:42 media
drwxr-xr-x 2 udash ustream 28 3월 18 22:14 media_info
drwxr-xr-x 2 udash ustream 43 3월 14 11:47 validation_xsd
```

2-2) RPM 으로 설치가 어려운 경우,
아래 포함된 폴더 및 파일은 필수적으로 존재해야 한다.

- **bin** : 프로그램 실행 폴더
 - **run.sh** : 프로그램 실행 스크립트 파일
- **config** : 프로그램 설정 폴더
 - **user_conf.ini** : 프로그램 필수 설정 파일
 - **logback.xml** : 프로그램 로그 설정 파일
- **media_info** : 허용 가능한 Streaming key 관리 폴더
 - **white_list.txt** : 허용 가능한 Streaming key 관리 파일
 - static, dynamic key 모두 관리 가능
 - 프로그램 내에서 [1초] 마다 주기적으로
[~/udash/media_info] 폴더에 있는 [video_list.txt] 파일이 변경되었는지 확인
 - 변경된 내용이 있으면 프로그램 내에 즉각적으로 반영함

- **validation_xsd** : MPD(이하 [*.mpd]) 파일을
검증하기 위한 파일들을 저장한 폴더
 - **DASH-MPD.xsd**
 - **xlink.xsd**
- **logs** : 실행 로그와 GC 로그 파일들을 저장하는 폴더
- **media** : Static 또는 dynamic media stream 을 저장하는 폴더

3. 설정

[\${프로젝트 경로}/config/user_conf.ini] 파일

```
[COMMON]
ID=100
SERVICE_NAME=UDASH
LONG_SESSION_LIMIT_TIME=1800000
ENABLE_CLIENT=false
THREAD_COUNT=10
SEND_BUF_SIZE=33554432
RECV_BUF_SIZE=16777216

[SERVER]
# rtmp, dash
STREAMING=dash
ENABLE_PRELOAD_WITH_DASH=true
HTTP_LISTEN_IP=192.168.5.222
HTTP_LISTEN_PORT=5858
HTTP_LISTEN_PORT_BEGIN_OFFSET=2
HTTP_LISTEN_PORT_END_OFFSET=10000
PREPROCESS_LISTEN_IP=127.0.0.1
PREPROCESS_LISTEN_PORT=5860

[CLIENT]
ENABLE_GUI=false
CAMERA_PATH=/live/livestream
HTTP_TARGET_IP=192.168.7.33
HTTP_TARGET_PORT=5858
PREPROCESS_INIT_IDLE_TIME=5000
PREPROCESS_TARGET_IP=192.168.7.33
PREPROCESS_TARGET_PORT=5870

[MEDIA]
MEDIA_BASE_PATH=/home/uangel/udash/media
MEDIA_LIST_PATH=/home/uangel/udash/media_info/video_list.txt
CLEAR_DASH_DATA_IF_SESSION_CLOSED=true
CHUNK_FILE_DELETION_INTERVAL_SEC=30
AUDIO_ONLY=true

[MPD]
ENABLE_VALIDATION=false
REPRESENTATION_ID_FORMAT=$RepresentationID$
CHUNK_NUMBER_FORMAT=$Number%05d$
SEGMENT_DURATION=2
WINDOW_SIZE=10
VALIDATION_XSD_PATH=/home/uangel/udash/validation_xsd/DASH-MPD.xsd

[RTMP]
RTMP_PUBLISH_IP=192.168.7.33
RTMP_PUBLISH_PORT=1945
```

1. Section

1. COMMON

: 서비스 관련 설정

1. ID : 프로그램 사용자 ID
2. SERVICE_NAME : 프로그램 이름 > HTTP Response 에서 사용됨
3. LONG_SESSION_LIMIT_TIME : 오래된 미디어 스트리밍 세션 삭제 타임아웃 시간
4. ENABLE_CLIENT : 카메라를 사용한 라이브 스트리밍 여부
5. THREAD_COUNT : Netty NioEventLoopGroup 의 쓰레드 풀 크기
6. SEND_BUF_SIZE : 메세지 송신 최대 버퍼 크기
7. RECV_BUF_SIZE : 메세지 수신 최대 버퍼 크기

2. SERVER

1. STREAMING : 어떤 프로토콜을 사용하여 미디어 스트림을 다운로드 받을지 결정하는 옵션
> 현재 [rtmp], [dash] 로 설정 가능
> 정의된 값이 아닌 값으로 설정 시 프로그램 구동되지 않음
2. ENABLE_PRELOAD_WITH_DASH : 프로그램 초기 구동 시, static media stream 을 dash 를 사용하여 미리 다운로드 받을지 결정하는 옵션
3. HTTP_LISTEN_IP : HTTP 소켓의 listen IP
4. HTTP_LISTEN_PORT : HTTP 소켓의 listen Port
5. HTTP_LISTEN_PORT_BEGIN_OFFSET : 서버에서는 다른 노드로 미디어 스트림을 요청할 때 HTTP communication 이 발생하므로 해당 노드로 보내기 위한 HTTP 소켓 이 필요하다. 즉, GET 메시지를 보내기 위한 Target ip, port 와 응답을 수신하기 위한 Listen ip, port 가 필요한데, 이 때 Listen port 의 시작 지점을 정의한다. 시작 offset 과 [HTTP_LISTEN_PORT] 와의 합이 최대 포트 번호(65536)를 초과하면 안된다. 0 보다 큰 값을 가져야 한다. (시작 offset 이 0 이면, [HTTP_LISTEN_PORT] 부터 번호 가 시작되므로 중복된 주소로 바인딩되는 이슈가 발생한다.)
6. HTTP_LISTEN_PORT_END_OFFSET : 앞서 설명한 바에 따라, Listen port 의 종료 지점을 정의한다. 시작 offset 보다 크면 안된다. 종료 offset 과 [HTTP_LISTEN_PORT] 와의 합이 최대 포트 번호(65536)를 초과하면 안된다. 0 보다 큰 값을 가져야 한다. (종료 offset 이 0 이면, 시작 offset 의 정의가 무의미해질뿐만 아니라, [HTTP_LISTEN_PORT] 부터 번호가 시작되므로 중복된 주소로 바인딩되는 이슈 가 발생한다.)
7. PREPROCESS_LISTEN_IP : Preprocess 메세지 수신할 IP
8. PREPROCESS_LISTEN_PORT : Preprocess 메세지 수신할 PORT

3. CLIENT

1. ENABLE_GUI : RTMP 서버로 송출할 Local Stream (camera) 을 사용자가 GUI 를 통해 볼 수 있도록 하는 옵션 (true: on, false: off)
2. CAMERA_PATH : RTMP 서버에 publish 할 URI
3. HTTP_TARGET_IP : HTTP 소켓의 target IP
4. HTTP_TARGET_PORT : HTTP 소켓의 target Port
5. PREPROCESS_INIT_IDLE_TIME : Preprocess 메세지 송신 전 유휴 시간 (ms)
> 미디어 스트림이 RTMP 서버로 Publishing 된 다음 전처리가 진행되어야함
6. PREPROCESS_TARGET_IP : Preprocess 메세지 송신할 IP
7. PREPROCESS_TARGET_PORT : Preprocess 메세지 송신할 PORT

4. MEDIA

: 미디어 파일 및 관련 파일 경로 설정

1. MEDIA_BASE_PATH : MPEG_DASH MPD 및 세그먼트 파일들이 저장될 경로 (static, dynamic 모두 포함)
2. MEDIA_LIST_PATH : 허용 URI 리스트 파일 경로
3. CLEAR_DASH_DATA_IF_SESSION_CLOSED : 클라이언트가 RTMP Unpublished 상태일 때, 로컬에 저장된 DASH 데이터를 모두 지울 것인지에 대한 여부
4. CHUNK_FILE_DELETION_INTERVAL_SEC : 클라이언트로서 Remote media stream 을 가져올 때, 오래된 미디어 청크 파일을 삭제할 시간 간격 (단위: 초)
5. AUDIO_ONLY : UDASH 가 서버로서 동작할 때, 오디오 데이터만 미디어 세그먼트로 저장할지에 대한 여부

5. MPD

: MPD 관련 설정

1. ENABLE_VALIDATION : MPD 검증 로직 수행 여부 결정 (true or false)
2. REPRESENTATION_ID_FORMAT : MPD 의 하위 필드인 [Representation] 필드에 속해 있는 [SegmentTemplate] 필드에서 [Initialization segment] 와 [Media segment] 필드값에 사용될 [Representation] 필드의 ID 형식을 정의, 초기 세그먼트 파일의 이름을 정의(Initialization segment)
3. CHUNK_NUMBER_FORMAT : [Media segment] 필드에서 사용되고, [REPRESENTATION_ID_FORMAT] 설정과 함께 미디어 세그먼트 파일의 이름을 정의 (Media segment)
4. VALIDATION_XSD_PATH : MPD 파일을 검증하기 위한 룰이 작성된 [xsd] 확장자 파일 경로
5. SEGMENT_DURATION : 미디어 세그먼트 하나당 부여할 시간 (초)
6. WINDOW_SIZE : 지정한 스트림 키에 의해 로컬에 저장된 미디어 세그먼트 개수를 정의. WINDOW_SIZE * 2 개수 만큼 미디어 세그먼트 개수가 일정하게 유지된다. 오래된 미디어 세그먼트는 차례대로 삭제된다.

6. RTMP

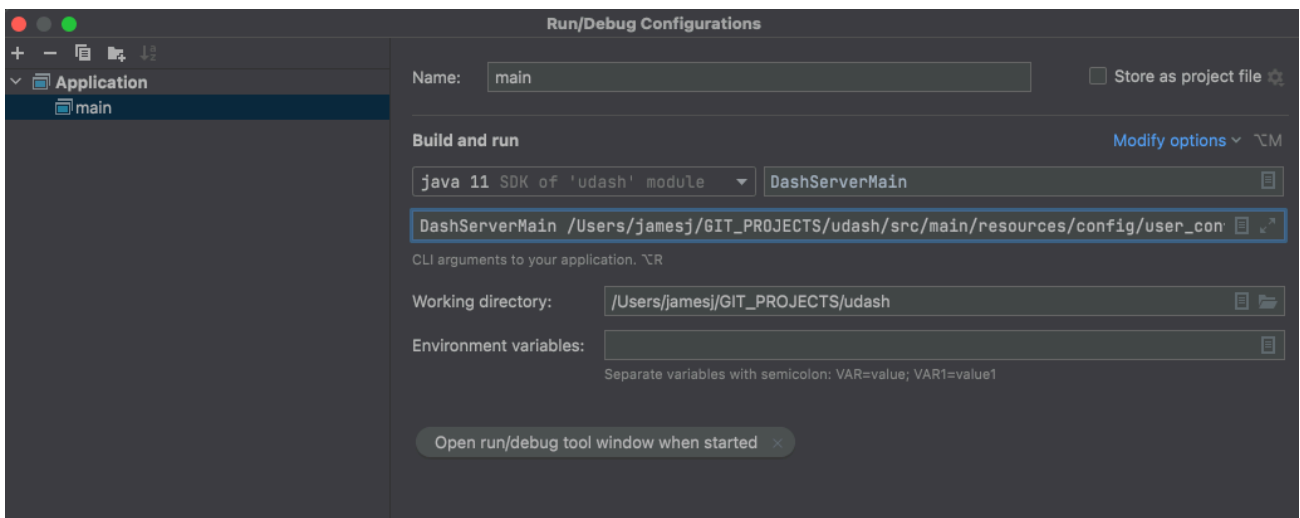
: RTMP 설정

1. RTMP_PUBLISH_IP : Publish 할 RTMP 서버 IP
2. RTMP_PUBLISH_PORT : Publish 할 RTMP 서버 Port

4. 실행

4-1) IntelliJ 에서 실행

- [Run/Debug Configurations] 메뉴 진입 후,
[DashServerMain] 을 main 함수로 지정 후,
프로그램 매개변수는 아래와 같이 작성



- 프로그램 매개변수
 - #0 : DashServerMain
 - #1 : \${user_conf.ini 절대 경로}
- 프로그램 실행 전,
[user_conf.ini], [logback.xml] 설정 파일 반드시 확인

4-2) Linux 에서 실행

1. [~/udash/bin] 경로 진입
2. ./run.sh 스크립트 파일로 프로그램 제어
 1. [start] 명령으로 프로그램 시작
 2. [stop] 명령으로 프로그램 중지
 3. [status] 명령으로 프로그램 실행 여부 조회
3. 프로그램 실행 후,
[logs] 폴더 안에 로그 파일로 프로그램 진행 상황 모니터링