# ArkCDC v2.0 Send/Recv Spec


- Send Config
- 전송 정보 설정
- 구간 암호화
- 전송 후 자동 삭제
- 전송 단위 설정
- 시점 지정 전송
- Send/Recv 체크포인트
- 통계
- Send/Recv 모듈 상태
- Timeout 설정
- 파일 유무 대기 시간
- 메타데이터 관리
- Module Interface Thread API
- 예외 처리


---

## Send Config

| 항목 | Default | 설명 |
|---|---|---|
| TARGET_IP | - | 대상 IP 주소 |
| TARGET_PORT | - | Target Agent 포트 |
| SOURCE_TRACINGFILE | - | 전송할 Tracing file의 alias |
| SOURCE_TRACINGFILE_DEST | $ARKCDC_HOME/trace | 전송할 Tracing file의 위치 |
| TARGET_TRACINGFILE | - | 작성할 Tracing file의 alias<br>Recv 모듈의 alias가 된다. |
| TARGET_TRACINGFILE_DEST | $ARKCDC_HOME/trace | 작성할 Tracing file의 위치 |
| TRANSMISSION_UNIT | 8M | 지정된 Size 단위로 Tracing file 전송 |
| AUTO_DELETE | no | 전송 완료 된 Tracing 파일 삭제 할지 지정 |
| SECURITY | plain | send-recv간 구간 암호화 여부 설정 |
| LOG_LEVEL | global.conf LOG_LEVEL(INFO) | 로그 레벨 설정 |
| LOG_BACKUP | global.conf LOG_BACKUP (none) | 로그 백업 정책 설정 |
| TRANSFER_TIMEOUT | 10S | timeout 설정. 10S 부터 60S까지 설정 가능 |
| TRACINGFILE_CHECK_INTERVAL | 100ms | Tracing File이 없는 상황에서 파일 체크 간격 설정. 1ms 부터 60s까지 설정 가능 |

| Config 추가 버전 |
|---|
| v. 1.4 or older |
| v. 1.5 |
| v. 2.0 |

## 전송 정보 설정

### 목적

Send 모듈이 읽을 Tracing 파일과 Target 쪽으로 전송하고 기록할 Tracing 파일에 대한 정보를 입력하기 위해 필요하다.

## 기능

전송을 위해 Target 쪽 서버의 IP 주소와 포트를 입력해야 한다.

```
TARGET_IP = "<IP_Address>"
TARGET_PORT = "<Port>"
```

- ***TARGET_IP***
    - 0 ~ 9 이외의 문자는 지원되지 않는다.
    - 최대 길이: 15
- ***TARGET_PORT***
    - 1024 ~ 65535 사이의 정수가 지원된다.

Source에서 읽을 Tracing 파일 정보와 Target 쪽에서 기록 할 Tracing 파일 정보를 입력해야 한다.

```
SOURCE_TRACINGFILE = "<alias>"
SOURCE_TRACINGFILE_DEST = "<dest>"
TARGET_TRACINGFILE = "<alias>"
TARGET_TRACINGFILE_DEST = "<dest>"
```

- ***SOURCE_TRACINGFILE & TARGET_TRACINGFILE***
    - A-Z, a-z, 0-9 _ 이외 모든 문자는 지원되지 않는다.
- ***SOURCE_TRACINGFILE_DEST & TARGET_TRACINGFILE_DEST***
    - 작성 된 경로가 존재하지 않을 경우 기본 경로인 $ARKCDC_HOME/trace가 사용 된다.
    - 기본 경로도 존재하지 않을 경우 모듈이 종료된다.

# 구간 암호화

## 목적

네트워크 전송 구간의 데이터를 보호하기 위해 필요하다.

## 기능

openssl을 이용하여 SSL/TLS 처리 된 구간 암호화 전송을 지원한다.

***SECURITY*** Config를 통해 설정이 가능하다.

```
SECURITY="[ plain | ssl ]";
```

- ***SECURITY***
    - 기본 값: plain
    - 대소문자 구분 없음

# 전송 후 자동 삭제

## 목적

사용이 완료 된 Tracing 파일이 저장 공간을 과도하게 차지하지 않기 위해 자동으로 삭제하는 기능이 필요하다.

## 기능

Tracing이 전송 된 후 Source에서 파일을 자동 삭제할 수 있다

***AUTO_DELETE*** Config를 통해 설정이 가능하다.

```
AUTO_DELETE="[ yes | no | y | n ]";
```

- ***AUTO_DELETE***
    - 기본 값: global.conf의 AUTO_DELETE 참조 (global.conf AUTO_DELETE(no))

# 전송 단위 설정

## 목적

## 기능

설정 된 단위 만큼 network 전송 단위를 조정할 수 있다.

***TRANSMISSION_UNIT*** Config를 통해 설정이 가능하며 기본 값은 8MB이다.

Send는 ***TRANSMISSION_UNIT***에 설정된 값 만큼 레코드를 모은 다음 한꺼번에 전송한다.

사용 방법은 다음과 같다:

```
TRANSMISSION_UNIT="n [ k | K | m | M ]";
```

- ***TRANSMISSION_UNIT***
    - 기본 값: 8MB
    - 최솟값: 1KB
    - 지원 포맷: 정수에 이은 문자 k, K, m, M
    - 위 표기에 따른 정수 변환 값이 1,048,576-104,857,600이하 (1MB ~ 100MB)

# 시점 지정 전송

## 목적

전송 시작 시점을 원하는 시점으로 제어하기 위해 필요하다.

## 기능

scn을 사용하여 원하는 시점으로 시작 지점을 설정할 수 있다

사용 방법은 다음과 같다:

```
--start-dsn
```

# Send/Recv 체크포인트

# 목적

Send가 재시작 했을 때 마지막으로 전송한 지점부터 이어서 전송할 수 있도록 전송 시점을 기록한다.

또, Send/Recv 프로세스의 상태 및 진행 상황을 파악 할 수 있다.

# 기능

Send/Recv Checkpoint는 크게 Send Checkpoint와 Recv Checkpoint로 나뉜다.

## Send Checkpoint

Send Checkpoint는 다음과 같이 구성된다:

- Startup Checkpoint: Send 모듈 최초 시작 시점.
- Current Checkpoint: Recv 모듈로부터 전송 완료 메시지를 받은 시점
- Write Checkpoint: Recv 모듈이 작성을 완료한 시점

## Recv Checkpoint

Recv Checkpoint는 다음과 같이 구성된다:

- Safe Checkpoint: 전송 받은 레코드 작성을 완료하고 Send 모듈에게 작성 완료를 알린 시점
- Current Checkpoint: 전송 받은 각 레코드의 작성 시점

# 작성 및 갱신 시점

## 작성

- 모듈 Start (Send Checkpoint - Startup)

## 갱신 시점

메모리

- Recv가 TRANSMISSION_UNIT만큼 전송 받은 레코드 작성을 완료한 시점 (Recv Checkpoint - Current)

체크포인트 파일

- Recv가 작성 완료를 알린 시점 (Send Checkpoint - Current, Write / Recv Checkpoint - Safe)
- 모듈 Stop (Send Checkpoint - Current / Recv Checkpoint - Current)
- 기본 Update 시간: 10초

## 예외 케이스

- 체크포인트 파일이 없을 경우:
    - 파일을 새로 생성하여, 현재 확인 할 수 있는 정보로 체크포인트를 작성한다.
- 체크포인트 파일 권한이 없을 경우:
    - 프로세스 시작 시:
        - 체크포인트 파일에 권한이 없다는 메시지를 로그에 남긴다.
        - Send 또는 Recv 프로세스가 시작되지 않아야 한다.
    - 프로세스 실행 중 발생 시:
        - 체크포인트 갱신 시점에 권한이 없을 경우, 체크포인트 파일에 권한이 없다는 메시지를 로그에 남긴다.
        - 프로세스를 중단 시킨다.

# Send/Recv Lag

Send 프로세스 Lag는 Source Redo에서 레코드가 발생한 시간과 Send가 해당 레코드를 전송한 시간 간에 간격이다. (레코드가 전송 된 시간 - 해당 레코드의 Timestamp)

Recv Lag는 Source Redo에서 레코드가 발생한 시간과 Recv가 레코드를 작성한 시간 간에 간격이다. (레코드가 작성 된 시간 - 해당 레코드의 Timestamp)

각 Lag는 Send/Recv 체크포인트 파일에 기록되며, 체크포인트 파일이 갱신 될 때 함께 갱신된다.

Send 프로세스 Lag는 다음 명령어로 조회할 수 있다.

```
admgr> getlag
admgr> getlag all
admgr> getlag send
admgr> getlag send all
```

Recv 프로세스 Lag는 다음 명령어로 조회할 수 있다.

```
admgr> getlag
admgr> getlag all
admgr> getlag recv
admgr> getlag recv all
```

# 통계

전송 데이터를 모니터링하기 위해 통계를 기록한다.

통계는 각 모듈별로 기록되며, 통계 파일명 포맷은 다음과 같다:

```
send_[alias]_[timestamp].stat
```

다음은 통계 파일의 예시이다:

```
Start of Statistics at [timestamp]

DDL extraction statistics:

*** Total statistics since [timestamp] ***
    Total operations                        n
    Total creates                           n
    Total alters                            n
    Total drops                             n
    Total others                            n

Send [schema].[table]

*** Total statistics since [timestamp] ***
    Total inserts                           n
    Total updates                           n
    Total deletes                           n
    Total truncates                         n
    Total operations                        n

*** Daily statistics since [timestamp] ***
    Total inserts                           n
    Total updates                           n
    Total deletes                           n
    Total truncates                         n
    Total operations                        n

*** Hourly statistics since [timestamp] ***
    Total inserts                           n
    Total updates                           n
    Total deletes                           n
    Total truncates                         n
    Total operations                        n
```

# Send/Recv 모듈 상태

다음은 Send/Recv 모듈 상태이다:

- Running
  Send/Recv가 동작 중인 상태
- Stopped
  Send/Recv가 정상 중지된 상태.
- Error
  에러가 발생하여 Send/Recv가 중지 된 상태

Send/Recv는 위 모듈 상태를 admgr, Checkpoint 파일, Manager 및 미들웨어에 제공해야 한다.

필요에 따라 모듈 상태를 추가해서 사용할 수 있다. (e.g. Manager의 경우 Agent 통신 문제로 인한 상태로 UNKNOWN이 있다)

# Timeout 설정

Send/Recv 프로세스 간에 응답 대기 시간을 설정 할 수 있다. (기본 값: 10초)

```
TRANSFER_TIMEOUT = "n [ s | S | m | M ]"
```

- ***TRANSFER_TIMEOUT***
  - 기본 값: 10s
  - 최솟값: 1s
  - 지원 포맷: 정수에 이은 문자 s, S, m M

# 파일 유무 대기 시간

Tracing 파일이 없거나 Size가 0일 경우 Tracing 파일을 찾는 주기를 ***TRACINGFILE_CHECK_INTERVAL*** Config를 통해 설정할 수 있다.

```
TRACINGFILE_CHECK_INTERVAL = "n [ ms | MS | s | S | m | M ]"
```

- ***TRACINGFILE_CHECK_INTERVAL***
  - 기본 값: 100ms
  - 최솟값: 1ms
  - 최댓값: 60s
  - 지원 포맷: 정수에 이은 문자 ms, Ms, s, S, m M

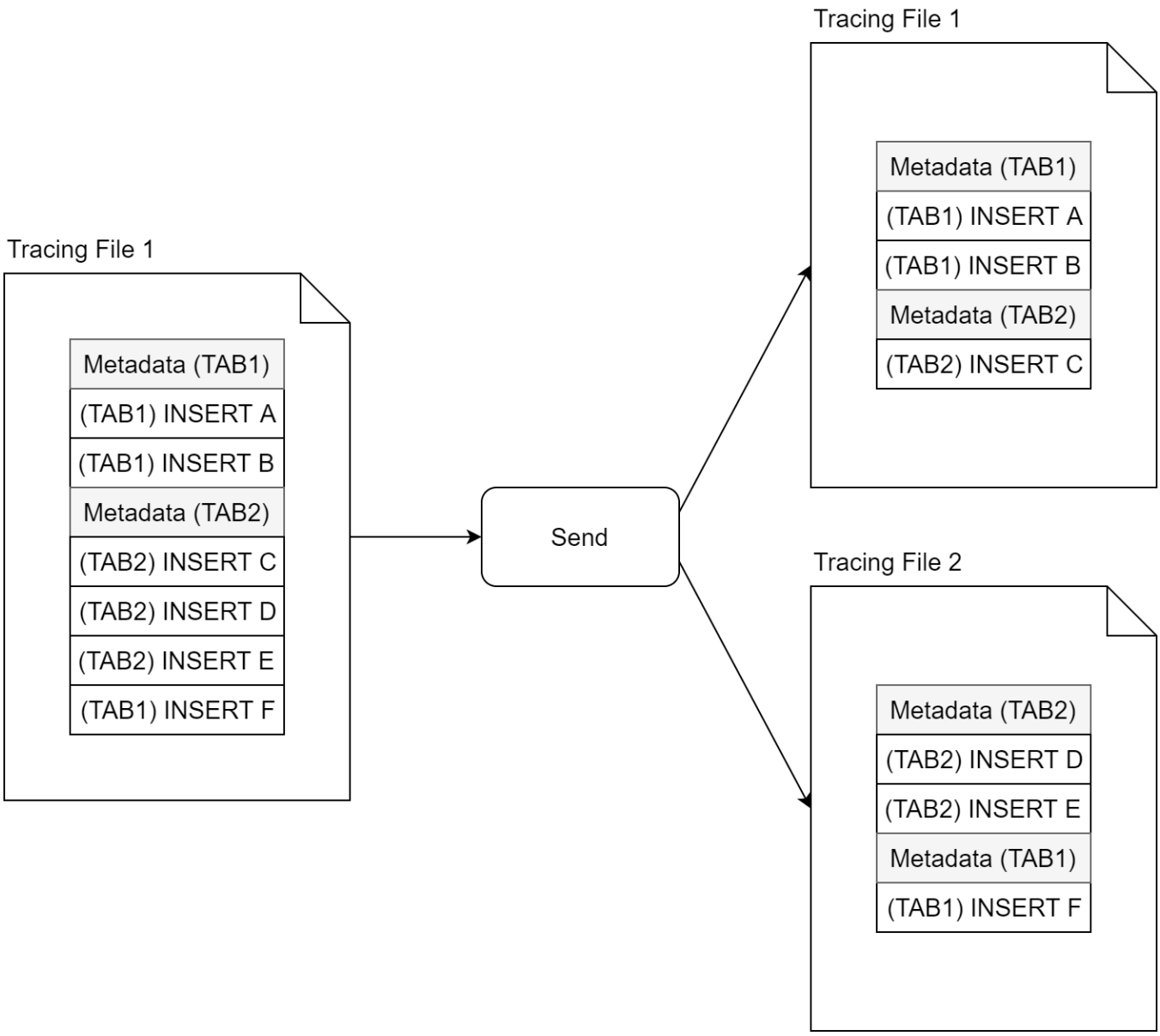
# 메타데이터 관리

Send Start/Stop 또는 대용량 트랜잭션으로 인해, 동일한 메타데이터의 레코드가 여러 Tracing 파일에 걸쳐 기록될 경우,

각 Tracing 파일마다 메타데이터가 기록되어야 한다.

예를 들어 다음 그림에서 Tracing File 1은 2개의 Tracing 파일로 나뉜다.

이때 두 번째 Tracing File에도 메타데이터가 기록되어야 한다.




# Module Interface Thread API

Module Interface Thread API는 모듈과 통신하는 스레드로, 별도 스레드가 존재한다.
Module Interface Thread API 로 얻을 수 있는 정보는 다음과 같다:

- 통계 정보
- 체크포인트
- LAG 정보
- 추출 중인 트랜잭션 정보 (트랜잭션 추출 중단 등)
- Alive 체크
- 시작, 종료 요청

# 예외 처리

## 공통

- 모듈 동작에 예외 케이스를 만나면 사용자가 인지할 수 있는 로그를 작성한다.
- Tracing file을 읽거나, 작성할 경로가 존재하지 않는 경우
    - 기본 경로 ($ARKCDC_HOME/trace)
        - 해당 디렉토리를 생성하고 다음 절차를 진행한다.
    - 그 외
        - 모듈을 종료한다.
- 전송 및 수신 중이던 파일이 삭제되는 경우, 모듈을 종료한다.
- 파일을 읽기/쓰기가 불가능한 경우 모듈을 종료한다.
- 네트워크 장애로 연결이 끊기는 경우 모듈 종료.

## Send

- Target agent에 recv 프로세스가 정상적으로 실행되지 않는 경우, 모듈을 종료한다.
- DSN 지정 시작(--start-dsn) 시 체크포인트에 기록된 전송 완료 시점보다 빠른 경우 중복 전송이 발생할 수 있으므로 모듈을 종료한다.
- Recv 모듈과 연결한 Socket이 disconnect 되면 모듈을 종료한다.

## Recv

- Agent와 연계해서 동작하는 것이 기본 동작이다.
- 단독 실행되는 경우
    - Tracing file size가 기본값인 50MB로 셋팅된다.
    - SSL 암호화 지원이 불가능하다.
    - send config의 ***TARGET_TRACINGFILE_DEST*** 참조가 불가능하다.
- 모듈 시작 시, 새로운 파일으로 작성한다.
- Send 모듈과 연결한 Socket이 disconnect 되면 모듈을 종료한다.
- Tracing file이 스위치 될 때, 동일한 이름의 파일이 존재하면 해당 파일을 삭제하고 새 파일에 작성한다.
- Tracing file이 저장되는 disk의 usage가 global config의 DISK_USAGE_CAPACITY 값 만큼 채워질 경우 수신을 정지한다.
    - disk usage가 DISK_USAGE_CAPACITY 값 아래로 내려가면 재수신한다.
- 모듈 시작시 port binding 에 실패하는 경우 모듈을 종료한다.