XLGAMES

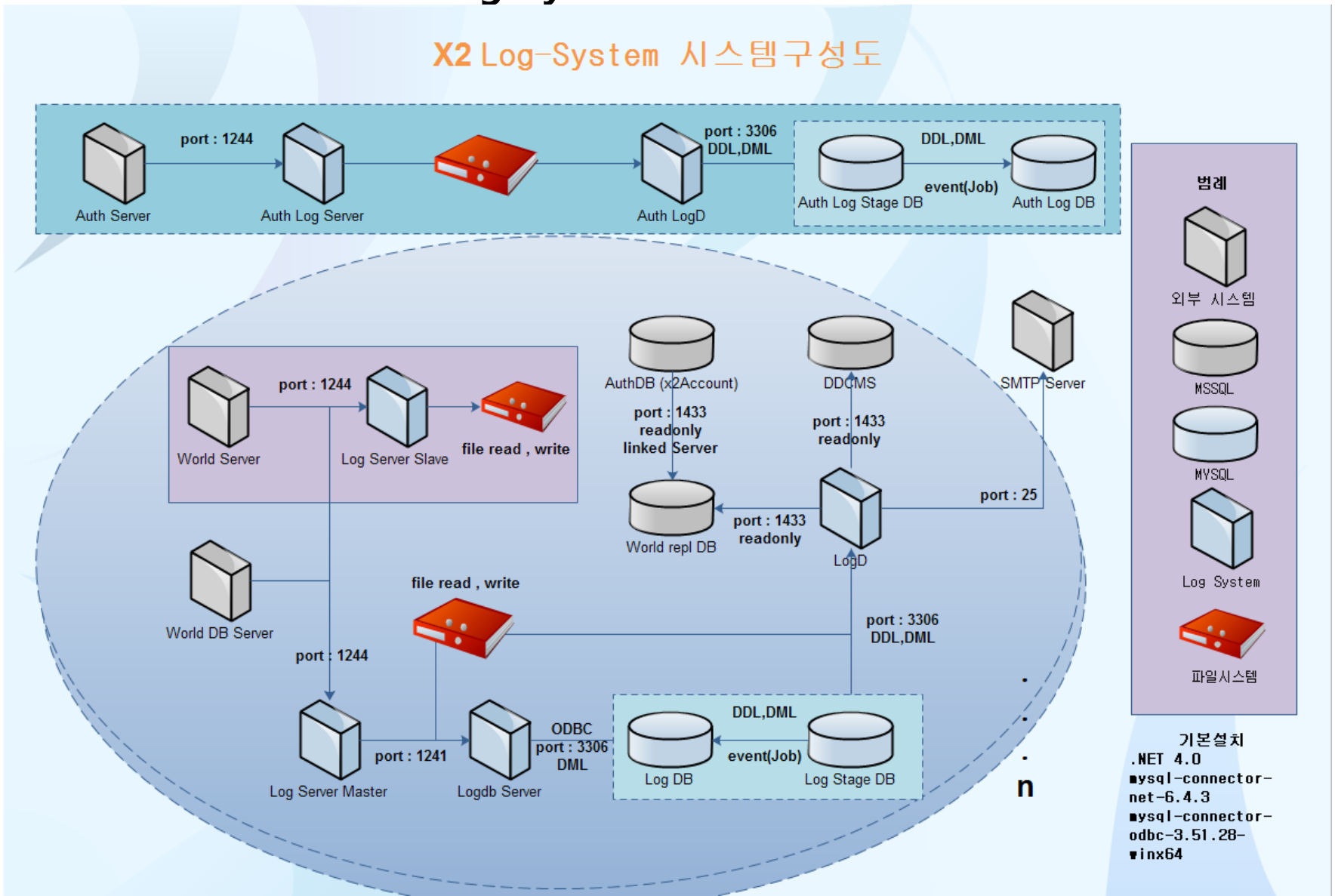
X2 Log System

해당 파일은 아키에이지 로스시스템 구성도, 개발 내용 및
로그시스템 운영시 대응했던 내용 위주로 만들었습니다.

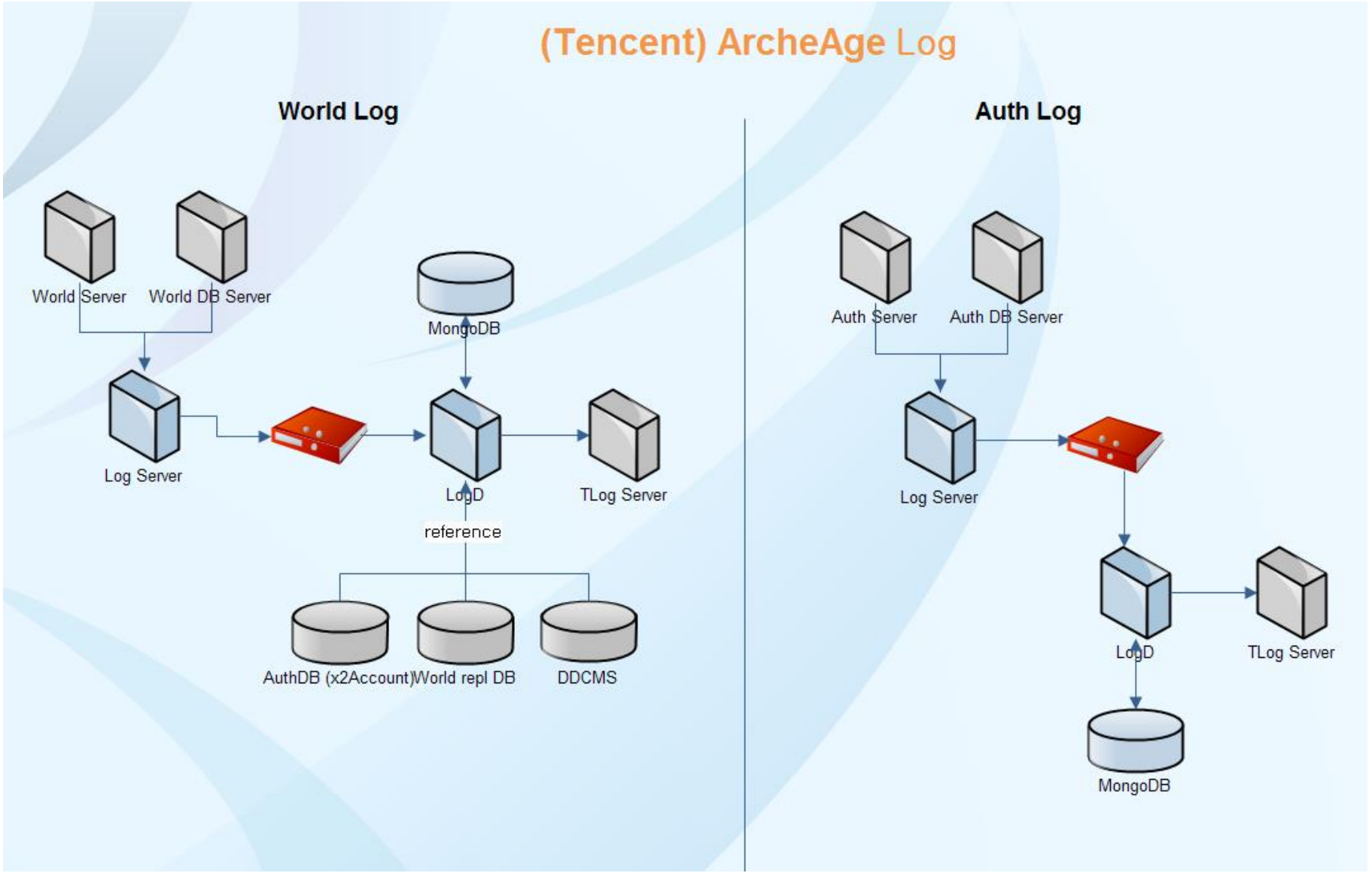
제목	X2 로그 시스템
Version	ver 1.0
작성자	오준석
작성일	2020-03-05

목차

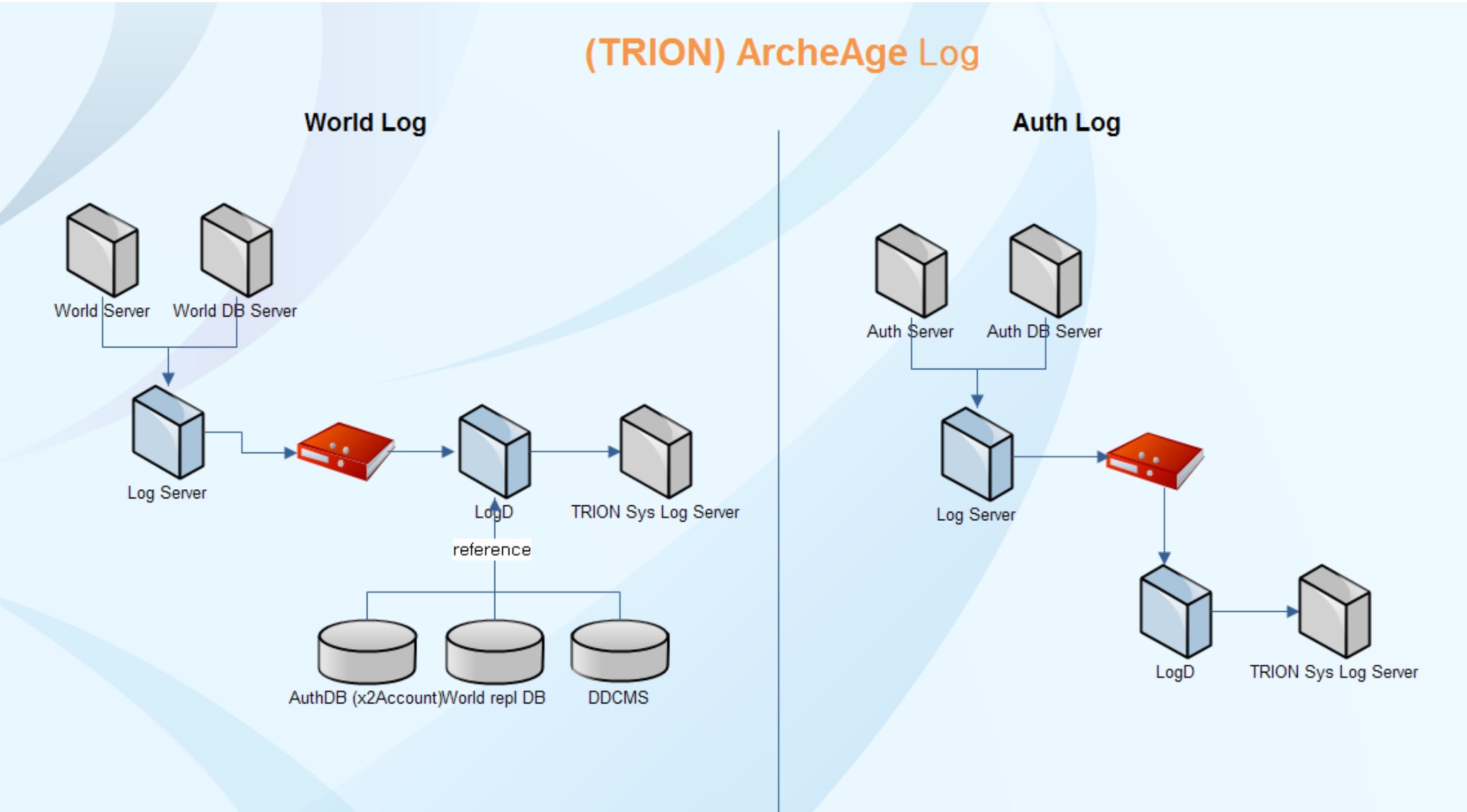
1. X2 Log-System 시스템 구성도



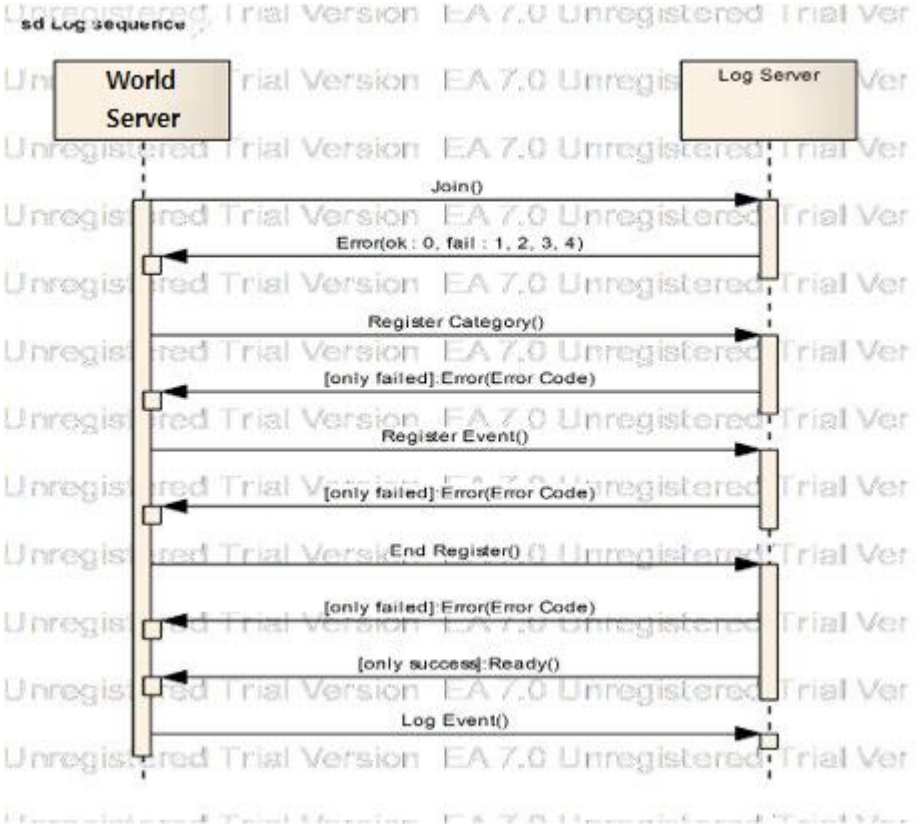
1. X2 Log-System 시스템 구성도



1. X2 Log-System 시스템 구성도



2. 로그서버 구성 및 흐름도



기본 패킷 프로토콜 형식 : Byte

Packet Header		Body				
		Message				
Packet Length	Packet Type	uint16	int32	int64	string	
2 byte (uint16)	2 byte (uint16)	2 byte	4 byte	8 byte	2 (string size)	가변 문자열

⇨ Packet Header의 Length 에 입력하는 값은 length(2바이트) 자신의 길이는 포함하지 않는다.

⇨ Body에 들어갈 각 변수를 전송할 때에는 해당 변수의 타입의 크기만큼 직렬화한다. (String or char[] 형태는 제외)

⇨ String 형태의 값을 Body에서 전송할 때에는 실제 문자열 앞에 String Size를 2 Byte (uint16) 크기로 직렬화 되어야 한다.

⇨ String 형태의 값을 직렬화 할 때에 입력하는 String Size에는 널 문자의 길이는 포함되지 않도록 한다.

Error : Join 또는 Register Category, Event 중 오류 발생시 응답으로 전송 (Log Server → Server)

Packet Header		Body
Packet Length	Packet Type	Error Code
2 byte (uint16)	2 byte (uint16)	1 byte (uint8)

⇨ Error Code : 오류가 발생한 원인 코드

Error Code 정리

코드	값	설명
No Error	0	성공
Invalid Version	1	프로토콜 버전이 맞지 않음
Invalid Category	2	카테고리 Index 값이 올바르지 않음
Invalid Event	3	이벤트 Index 값이 올바르지 않음
File Error	4	파일에 로그를 저장하는데 실패함

Ready : End Register까지 성공했을 때 응답으로 전송 (Log Server → Server)

Packet Header		Body
Packet Length	Packet Type	
2 byte (uint16)	2 byte (uint16)	

⇨ Packet Type 만으로 처리, Message Body 가 없음

Join → 연결 후 Join으로 프로토콜 버전을 체크한다.
Register Category → 로그 카테고리를 등록한다. 로그를 저장하기 위해서는 반드시 1개 이상의 카테고리는 반드시 등록해야 한다. (카테고리 이름은 String 타입)
Register Event → 카테고리 하위에 로그 이벤트를 등록한다. 카테고리 별로 1개 이상의 이벤트는 반드시 등록해야 한다. (이벤트 이름은 String 형태)
Register Event → 다수의 이벤트를 하나의 카테고리에 등록할 수 있다.
… →
Register Category → 다수의 카테고리를 등록할 수 있다.
… →
End Register → 모든 카테고리와 이벤트를 등록했음을 알린다.

Log Event → 카테고리, 이벤트 등록 완료 후 실제 로그를 전송한다.

2. 로그서버 구성 및 흐름도

Log Event : 실제 로그 데이터를 전송하는 패킷 (Server → Log Server)

Packet Header		Body	
Packet Length	Packet Type	Stored Byte Size	Formatted Log String
2 byte (uint16)	2 byte (uint16)	2 byte (uint16)	가변 길이 (char [] Max : 4096)

- Stored Byte Size : 뒤에 직렬화 할 Formatted Log String 의 길이를 입력
- Formatted Log String : 형식화된 로그 문자열로 별도로 정해진 형식대로 저장된 형식. 직렬화는 길이를 포함하지 않도록 한다 (길이는 앞에 Stored Byte Size에서 저장되기 때문)

Formatted Log String 형식

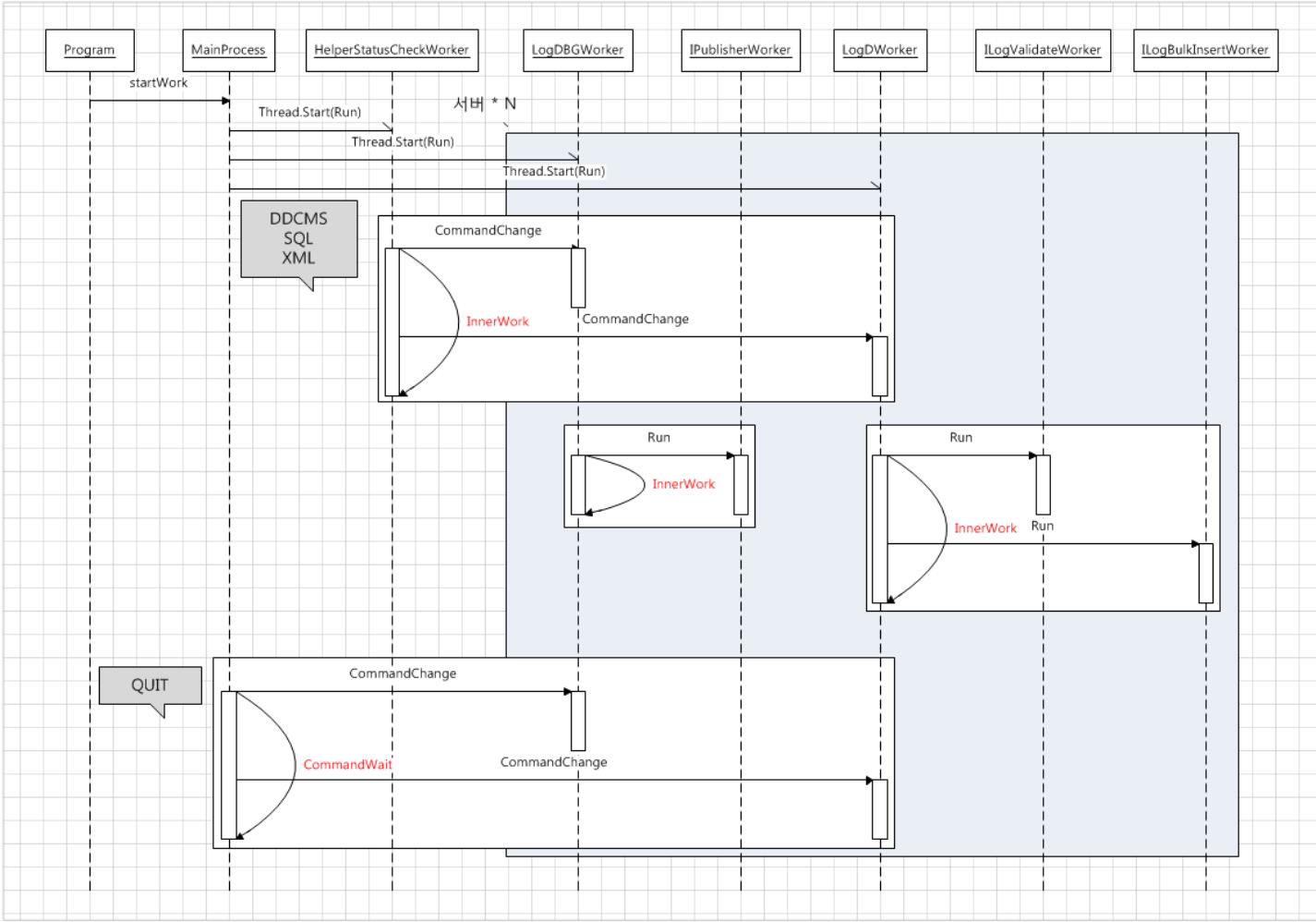
Header					Body				
					Add Value		Add Value		
Log Size	Category	Event	Action ID	Log Time	Value Type	Value	Value Type	String Size (String Type만 해당)	Value
2 (uint16)	4 (int32)	4 (int32)	8 (int64)	8 (int64)	4 (uint32)	가변 길이	4 (uint32)	2 (uint16)	가변 길이

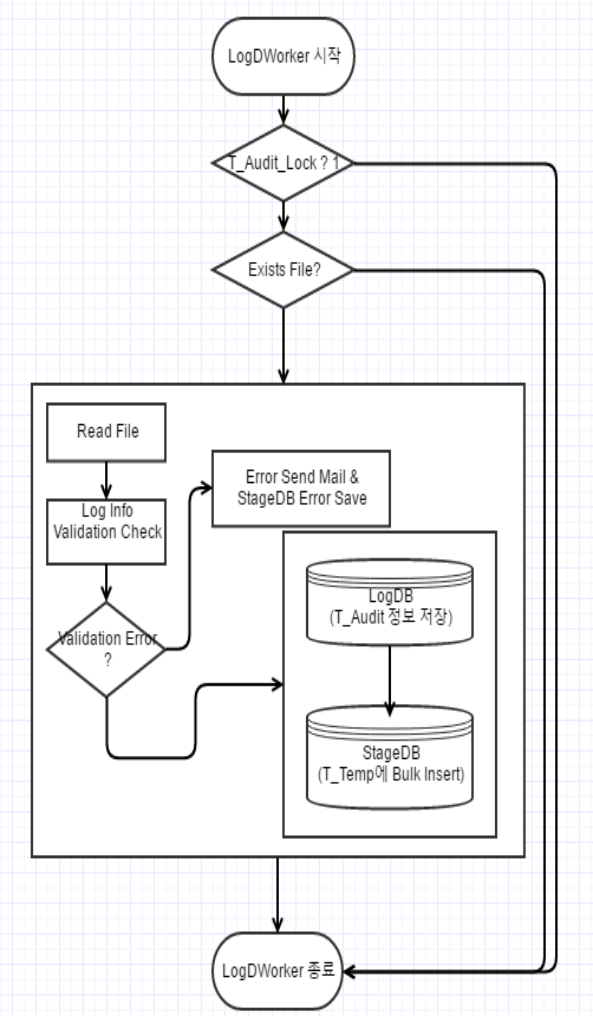
- Log Size : Formatted Log String의 Header를 포함하는 전체 길이 (Log Event 패킷의 Stored Byte Size 항목과 동일한 값이어야 함).
- Category : 로그의 카테고리 Index 입력.
- Event : 로그의 이벤트 Index 입력.
- Action ID : 로그를 특정 Action 단위로 그룹화 할 필요가 있을 때 해당 그룹의 ID를 입력.
- Log Time : 로그가 발생한 시간을 _time64() 값으로 입력.
- Add Value : 로그로 보내는 각 데이터를 하나의 Add Value로 보고, 여러 값을 전송하게 될 경우 Add Value 형식으로 연결된 형태로 전송하면 된다.
- Add Value 에서 Value Type이 String 형태인 경우(or char[]) Value Type 뒤에 String의 Size를 추가로 입력해야 한다. (Size는 널 문자 포함 길이로 입력).
- String Type의 Value를 직렬화 할 때에는 Size에 널 문자를 포함한 길이를 입력한 것과 같이 널 문자도 포함하여 직렬화 한다.

Value Type (Enum) : 4 byte

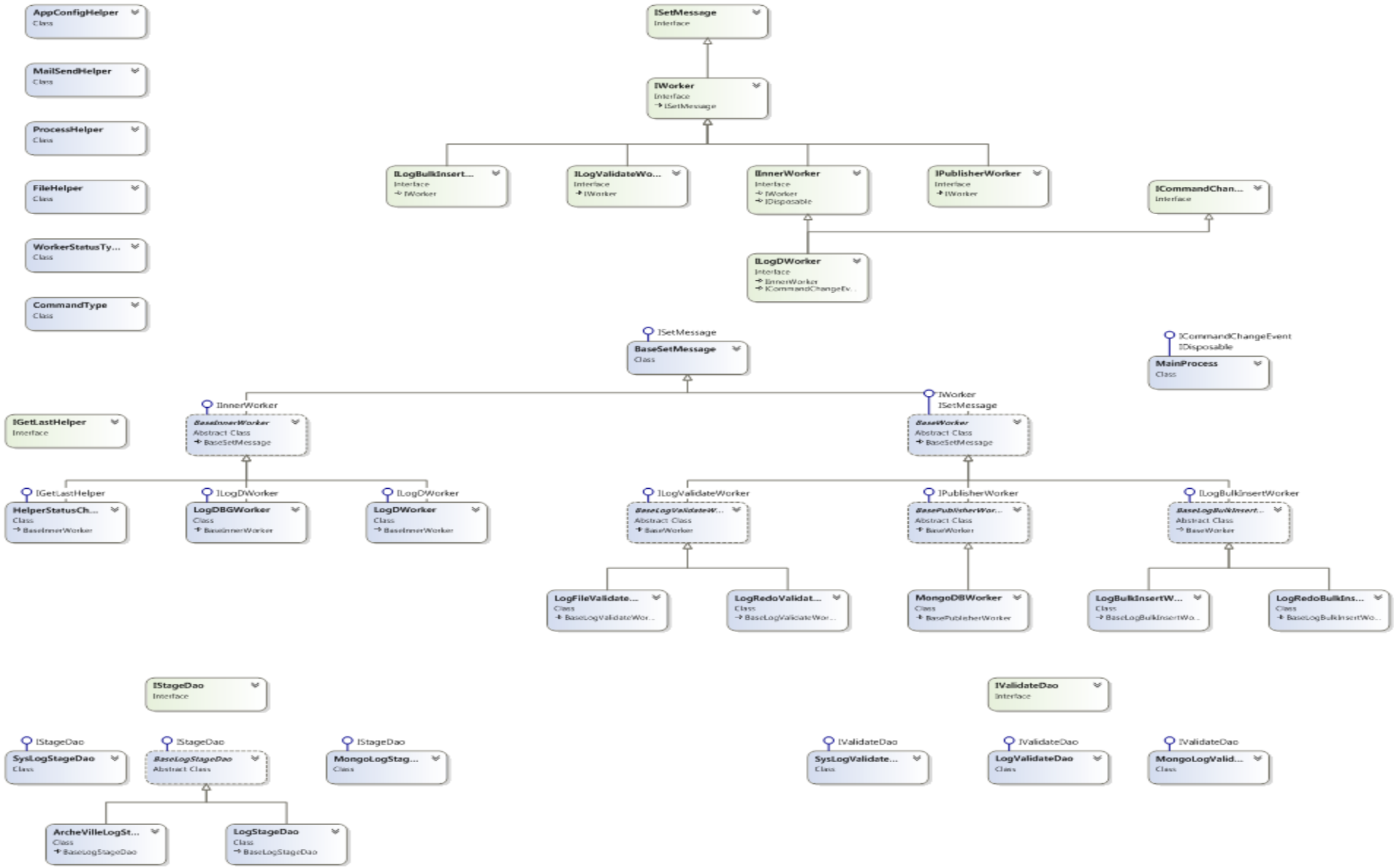
Type	Value	Type	Value	Type	Value	Type	Value
Boolean	1	Int16	5	Uint16	9	Dword	13
Float	2	Int32	6	Uint32	10	String	14
Double	3	Int64	7	Uint64	11	Delimeter	15
Int8	4	Uint8	8	Long	12	EndOfLog	16

3. 로그D 내부 구성 및 흐름도



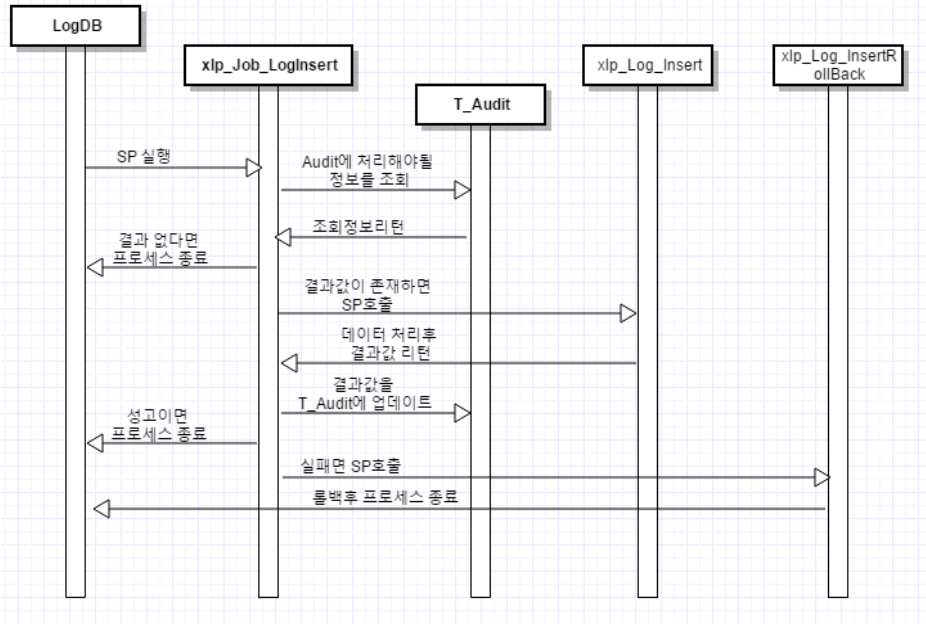


3. 로그D 내부 구성 및 흐름도



4. LogDB 시퀀스 다이어그램 및 구성

LogDB_Insert_Process



StageDB 구성 : LogDB로 데이터를 이관하기 전에 데이터를 임시 보관함.

구분	테이블명	설명
시스템 테이블	T_Stage_Error_Log	LogD에서 원시로그 Validation 체크시 오류가 발생한 데이터를 저장
임시테이블	T_ServerId_Datetime_MasterOrSlave	LogD에서 임시 테이블을 생성하여 데이터를 저장함 EX) 테이블패턴 : T_1001_20160607121522_M

1. Tables

1. **x2AuthLogDB / x2LogStageAuth : 인증정보 저장**
2. **x2LOGDB_001 / x2LogStage : 월드 게임정보 저장**

테이블 구분	테이블명	설명
시스템	Audit	LogD가 x2LogDB_Stage DB에 테이블 생성후 데이터를 저장하면 해당 정보 생성함
	T_Audit_Lock	LogD의 처리상태 (0, 1)
	tPatchVersion	LogDB의 패치정보를 저장함. (수동저장)
물리파티션	T_Log_yyyyMMdd	모든 로그정보를 저장함 (chat, system, skill 제외)
	T_Log_Chat_yyyyMMdd	Chat 로그만 저장함
	T_Log_System_yyyyMMdd	System 로그만 저장함
	T_Log_SkillUse_yyyyMMdd	Skill 로그만 저장함
논리파티션	T_Log2Actability	능력관련 정보
	T_Log2Auction	경매관련 정보
	T_Log2CharacterEtc	캐릭터 기타 정보
	T_Log2Connect	캐릭터 연결정보
	T_Log2Doodad	두대드 정보
	T_Log2Exp	경험치 정보
	T_Log2Expedition	원정대 정보
	T_Log2Item	아이템 정보
	T_Log2LevelUp	레벨업 정보
	T_Log2Mail	메일 수신/발신 정보
	T_Log2Money	머니 변화 정보
	T_Log2MonsterKilled	몬스터 정보
	T_Log2Quest	퀘스트 정보
	T_Log2ShopSales	샵 구매정보
기타	T_Log_ServerStatus	서버 상태를 저장함
	T_Log_ZoneCount	월드서버의 존에 동접 정보저장
	T_WebDaily	웹페이지에서 사용하나 국가마다 다름

4. LogDB 시퀀스 다이어그램 및 구성

2. Stored Procedules

구분	프로시저명	설명
시스템	xlp_Job_DropPartition	파티션 삭제
	xlp_Job_DropTable	테이블 삭제
	xlp_Job_LogInsert	T_Audit에서 처리할 테이블 정보를 가져옴
	xlp_Log_Insert	xlp_Job_LogInsert의 데이터를 실제 처리
	xlp_Log_InsertRollBack	xlp_Log_Insert 처리 오류시 롤백 처리
	xlp_Job_LogStats	캐릭터별 레벨업 시간 및 경험치 업데이트
	xlp_Job_MakePartition	파티션 생성
	xlp_Job_MakeTable	테이블 생성
	xlp_Job_TRUNCATE_T_SkillUseTable	T_Log_SkillUse_yyyyMMdd 테이블 드롭
	xlp_Log_Alter_Table_FactionID	원정대 정보 수정
	p_Batch_Web	일단위 캐릭터의 기타 정보를 저장
	xlp_Log_ZoneCount	LogServer에서 존별 접속량을 저장
	xlp_Log_Ping	LogServer에서 LogDB가 동작중인지 체크
조회	xlp_log_Main	조회 분기용 SP
	xlp_Log_Page	xlp_log_Main에서 호출함. ➢ T_Log_yyyyMMdd의 정보 조회
	xlp_Log_PageConnection	xlp_log_Main에서 호출함. ➢ T_Log2Connect 로그테이블 정보를 조회
	xlp_Log_PageExpedition	xlp_log_Main에서 호출함. ➢ T_Log2Connect 로그테이블 정보를 조회
	xlp_Log_PageItem	T_Log2Item 로그테이블 정보 조회 (미사용)
	p_Log2Actability	T_Log2Actability테이블 정보 조회
	p_Log2Auction	T_ Log2Auction테이블 정보 조회
	p_Log2CharacterDead	T_Log2CharacterEtc테이블 정보 조회
	p_Log2Connect	T_ Log2Connect테이블 정보 조회
	p_Log2Doodad	T_ Log2Doodad테이블 정보 조회
	p_Log2Exp	T_ Log2Exp테이블 정보 조회
	p_Log2HonorPoint	T_Log2CharacterEtc테이블 정보 조회
	p_Log2Item301	T_ Log2Item테이블 정보 조회
	p_Log2Item302	T_ Log2Item테이블 정보 조회
	p_Log2Item303	T_ Log2Item테이블 정보 조회
	p_Log2Item305	T_ Log2Item테이블 정보 조회
	p_Log2Item308	T_ Log2Item테이블 정보 조회
	p_Log2Item310	T_ Log2Item테이블 정보 조회
	p_Log2ItemTotal	T_ Log2Item테이블 정보 조회
	p_Log2Labor	T_ Log2Labor테이블 정보 조회
	p_Log2LevelUp	T_ Log2LevelUp테이블 정보 조회
	p_Log2Mail	T_ Log2Mail테이블 정보 조회
	p_Log2Money	T_ Log2Money테이블 정보 조회
	p_Log2MonsterKilled	T_ Log2MonsterKilled테이블 정보 조회
	p_Log2PK	T_ Log2PK테이블 정보 조회
	p_Log2Quest401	T_ Log2Quest테이블 정보 조회
	p_Log2Quest402_404	T_ Log2Quest테이블 정보 조회
	p_Log2Quest403_405	T_ Log2Quest테이블 정보 조회
	p_Log2ShowSales	T_Log2ShowSales테이블 정보 조회
	p_Log2ShowSales_Get	T_Log2ShowSales테이블 정보 조회
	xlp_Log_ZoneMove	존 이동 정보 조회
	xlp_Get_ZoneCount	로그정보에서 존별 캐릭터 접속량 조회
	Set_MakePartitionTable	논리 파티션 테이블 초기 생성 스크립트 (미사용)
	TruncateTable	테이블 초기화 (미사용)
	CallSpRoof	초기 셋팅시 테이블 및 파티셔닝 설정 (미사용)
	Insert_dic	삭제 대상 테이블 정보 Insert (미사용)

3. Functions

함수명	설명
BigIntToInt	Bigint Type 데이터를 Int Type로 캐스트
F_CharacterLastMoney	캐릭터 현재 머니량을 리턴
F_ColumnCheckReturnValue	테이블의 컬럼 존재여부를 확인하여 원하는 값을 리턴
F_DateCheckReturnTable	조회 대상 테이블 정보를 리턴
F_getItemEnchantGrade	아이템 강화후의 등급을 리턴
F_findAbilityName	능력명 리턴
F_LaborPower	노동력 리턴
F_LevelUpExp	레벨업 경험치 리턴
F_LevelUpTime	레벨업 시간 리턴
F_SplitValueCount	Split된 정보의 개수
IntToBigInt	Int Type 데이터를 Bigint Type로 캐스트
SPLIT_STR	구분자로 문자열을 잘라 해당 인덱스 정보를 리턴
xlf_Ability_Sort	캐릭터 능력치 정렬

4. Events

이벤트명	설명
Events_LogProc	1분 단위로 StageDB에 저장된 데이터를 처리
Events_LogStats	5분단위 레벨업 정보 업데이트
Events_MakePartition	일단위 파티션 생성
Events_MakeTable	일단위 테이블 생성
Events_Truncate_T_SkillUse	일단위 T_SkillUse 테이블 삭제
Events_WebBatch	일단위 캐릭터의 상세정보를 저장

향후 업무 및 문제점 논의

구분	시스템구분	내용
공통	LogD	➢
	Database	➢ LogDB 처리 오류 발생시 오류 트랙킹 시스템이 존재하지 않음
x2 (Archeage)	x2LogD	➢ LogD 구동중 LogFormat이 변경되었을 경우, ddcms.xml 파일에 존재하지 않는 속성이 있으면 오류를 발생하지 않는다. ➢ MongoDB의 간헐적 연결실패 (Tencent)
	x2LogDB	➢

5. 로그 개발 및 운영시 해결했던 일감 정리

#238433 "520 미삭제 하우징 로그" 필드 갯수가 맞지 않는 현상
- •LogD 프로그램에서 로그 형식 정의를 한 x2format.xml 파일과 실제 로그서버에서 만든 원시로그 데이터와의 로그 필드 갯수가 같지않아 원시로그를 토대로 LogDViewr(Node.js) 로 만든 툴에서 각 필드의 속성을 파악하고 부족한 필드를 x2foramt.xml에 필드 추가.

#238574 [AAKR][로그] Summary 메일링 기능 수정
- •LogD 프로그램은 매일 새벽 4시에 LogDB의 감사테이블 정보, LogsatgeDB 에러 테이블 정보, LogD 동작중 파일 이동 정보가 있을때만 dest 메일서버에 송신하게 변경.

#241543 [ABOX] (개선) 게임 로그 - 접속 관련 로그 - 계정 접속, 종료 로그
- •기존 로그 eventID: 1 계정접속로그, 2 계정 접속종료 로그에 아키에이지 프로그램 cpu Type 값 필드 추가.

#242536 [AAKR][Log] xlp_Log_InsertRollBack SP 개선
- •xlp_Log_InsertRollBack sp로 지우는게 실패가나면 LogDB는 Job을돌면서 재처리를 하게되는데 정상으로 지워지기 전에 StageDB의 테이블명을 재처리 가능하게 이름을 변경해서 계속 실패나기 전까지 재처리를 한다.
  따라서 삭제 SP가 정상 작동 했을때만 STAGELogDB의 테이블명을 변경하게 바꿔 무한루프에 빠지는 현상 개선.

#243071 [AATW] 라이브 로그 이상 확인 요청
- •Log DB(mysql)의 일부 파일이 누락되어 T_Log2HeirLevelUp 테이블에 접근되지 않아서 였던 것으로 확인되었습니다. mysql repair table 해당 명령어로 누락된 파일정보를 재구성 하였다.
  https://dev.mysql.com/doc/refman/8.0/en/repair-table.html

#243523 [AATW][CS17] 신규서버 구축 가이드 요청 - LogD 작업
- •1) LogDB, LogStageDB 설치 스크립트 01_x2LogDB, 02_x2LogStage, 03_Today Table and Partition Create
  2) LogSystem 설정 가이드 LogD, config.xml, Log4Net, infra_log_server.cfg
  해당 정보 워드에 정리해서 가이드 배포.

#244166 [AATW] 5.5.1v QA2 LogD 업데이트 에러 발생 - QA3 logserver 로그파일 open 관련 에러 발생
- •LogD 프로그램중 X2DDCMS(MSSQL) 아키에이지 게임 DB에 버전 정보가 다르다는 에러 로그를 확인후 버전 확인후 정상처리.

#249430 [ABOX][5.0.0.6]5.5.1v 업데이트 이후, 전체 로그 검색에서 전장 신청 로그

#[AATW][Live] 대만 11,12 각성 장비 로그, 노르예트 무한대전 관련 로그 조회가 안되는 현상
- •대만 PM분이 해당 버전의 로그 바이너리 및 포멧을 배포하지 않아 로그 처리가 되지않음.
  추가(변경) 로그만 처리하는 바이너리 배포후 재처리 가이드 배포.

#248921 [DBA][LOG] 35 누이서버 임시DB 데이터 재처리 요청
T_35_20190725092819_M 테이블의 데이터를 임시db에 Insert시 Error Code: 1153. Got a packet bigger than 'max_allowed_packet' bytes 0.032 sec 오류가 있습니다.
요청쿼리:
set global max_allowed_packet=8000000;
update T_Audit set is_BatchDone=0 , errorCode=null where idx_audit=474726; 후 정상처리.

#257986 [LOG] 동남아 배포 마스터
LogDB, LogD,LogServer,LogDBServer 관련 바이너리,포멧 및 스크립트를 동남아 버전에 맞게 번역적용후 전달.

5. 로그 개발 및 운영시 해결했던 일감 정리

#246047 [AAKR][로그] 로그D 개선
1.LogD 프로젝트의 Info,error 로그에 현재 버전 로그, x2foramt.xml, sql_ddcms.xml 파일에대한 버전 추가.

2.Sqlddcms.xml 파일에 아키에이지 직업명 관련 쿼리를 하드코딩한 부분을 실제
직업명 db table에 접근하여 가져오게 변경.

3.LogD 에서 로그의 저장 유형 확장성을 위해 LogD 맵핑후 JSON형식으로 파일생성(Newtonsoft.Json 라이브러리에
IsValidJson() json파일 형식 유효성 검사 함수 사용.

```
public bool MakeJsonFilewithLib(LogFileInfoModel logFileInfo, IDictionary<int, IDictionary<EventInfo, EntryList>> xmletnrylist, IDictionary<DBTableType, IList<ILogModel>> result, Valida
{
    StopWatch stopWatch = null;
    try
    {

        // ...중략...
        jsonstring.Append("]");
        // json validate check!
        if (IsValidJson(jsonstring.ToString()) == false)
        {
            logger.Info("json validate check Error File: " + jsonFileName);
            return false;
        }
        // [ ] 문자열 제거
        jsonstring.Remove(0, 1);
        jsonstring.Remove(jsonstring.ToString().Length - 1, 1);
        //Json 파일 만들기
        System.IO.File.WriteAllText(logFileInfo.destinationFilePath + jsonFileName, jsonstring.ToString());

        return true;
    }
    finally
    {
        if (logger.IsDebugEnabled && stopWatch != null)
        {
            stopWatch.Stop();
            logger.Debug("MakeJson [" + stopWatch + "]");
        }
    }
}
```

5. 로그 개발 및 운영시 해결했던 일감 정리

#246047 [AAKR][로그] 로그D 개선
4.LogDB로그 처리시 LogD x2foramt.xml 파일의 각 로그 필드들의 변수 타입과 1차,2차테이블의 타입이 다른게 있는지 여부 파악.

로그 정의 포멧에 변수 Type과 LogDB 에 칼럼타입이 같은 타입인지 확인. (로그 유효성 사전 검사)

```
<event id="301"      name="event301"     desc="아이템 획득 로그" >
   <entry ref="eventBasic"       />
   <entry ref="characterBase"    />
   <entry ref="characterEtc" />
   <entry name="itemUID"             type="string"       min="1"               desc="아이템 액션 유니크 키 - 2011-12-28 추가" />
   <entry name="itemType"               type="int"          min="0"               desc="캐릭터가 획득한 아이템의 종류(명칭)" />
   <entry name="itemTypeName,itemTypeCategoryId,itemTypeCategoryName"           type="title"        tokens="3"            desc="캐릭터가 획득한 아이템의 명칭-카테고리ID-카테고리이름" >
     <values>
       <entry ddcmstype="item" ddcmsvalue="{itemType}"    ddcmstext="name,categoryId,categoryName"   params="{ddcmsResult[0]},{ddcmsResult[1]:0},{ddcmsResult[2]}" >{0}|{1}|{2}</entry>
     </values>
   </entry>
   <entry name="itemID"              type="long"         min="0"               desc="획득한 아이템의 DB ID" />
   <entry name="itemgrade"              type="int"                                desc="아이템 등급" />
   <entry name="itemgradeTitle"      type="title"    >
     <values>
       <entry ddcmstype="itemgrade"  ddcmsvalue="{itemgrade}"    ddcmstext="name"    params="{ddcmsresult}" >{0}</entry>
     </values>
   </entry>
   <entry name="addAmountItem"          type="int"          min="-100"  max="100"   desc="획득한 아이템 수량" />
   <entry name="slotTotalItem"          type="int"          min="0"     max="100"   desc="아이템 획득 후 캐릭터의 인벤 총 수량">구현됨?</entry>
   <entry name="totalItem"              type="long"         min="0"                 desc="장비, 가방, 은행을 포함한 총 개수" >구현됨?</entry>
   <entry name="itemAcquireReason"      type="string"       min="1"                 desc="아이템 습득 이유 (사냥, 채집, 제작 등)" />
   <entry name="itemSlotType"           type="int"          min="0"     max="100"   desc="획득 아이템 슬롯 위치 " />
   <entry name="itemSlotTypeTitle"      type="title"                                desc="획득 아이템 슬롯 위치 " >
     <values>
       <entry ddcmstype="slot" ddcmsvalue="{itemSlotType}"    ddcmstext="name"   params="{ddcmsResult}" >{0}</entry>
     </values>
   </entry>
   <entry name="itemSlotGroup"          type="int"          min="0"     max="100"   desc="획득 아이템 슬롯 위치 " />
   <entry name="itemSlotIndex"          type="int"          min="0"     max="100"   desc="획득 아이템 슬롯 위치 " />
   <entry ref="lootSource"   />
</event>
```

```
CREATE TABLE IF NOT EXISTS T_Log2Actability(
    nid int(10) unsigned  NOT NULL auto_increment,
    dateLog datetime  NULL ,
    AccountID bigint(20)  NULL ,
    AccountName varchar(150)  NULL ,
    CharacterID int(11)  NULL ,
    CharacterName varchar(64)  NULL ,
    Race tinyint(4)  NULL ,
    Ability1 tinyint(4)  NULL ,
    Ability2 tinyint(4)  NULL ,
    Ability3 tinyint(4)  NULL ,
    Level tinyint(3) unsigned zerofill  NULL ,
    ZoneID int(11)  NULL ,
    Loc varchar(50)  NULL ,
    UsePoint int(11)  NULL ,
    ActabilityType int(11)  NULL ,
    ActabilityName varchar(50)  NULL ,
    Reason varchar(50)  NULL ,
    Audit int(10) unsigned  NULL ,
    KEY AccountID (AccountID),
    KEY AccountName (AccountName),
    KEY ActabilityType (ActabilityType),
    KEY audit (Audit),
    KEY CharacterName (CharacterName),
    KEY dateLog (dateLog),
    KEY nid (nid),
    KEY Reason (Reason)
) ENGINE = MYISAM DEFAULT CHARSET=utf8;
```

5. 로그 개발 및 운영시 해결했던 일감 정리

#246047 [AAKR][로그] 로그D 개선
5.캐릭터 정보, 계정 정보 데이터를 위한 WorldDB 접속을 파일단위로 수정.

```
    public override IDbConnection GetConnection()
    {
        return new MySqlConnection(GetConnectionString());
    }

    private DataBindingSource ExecuteDataBinder(SqlParameter sqlParameter, bool beginTransaction, int commandTimeout, IDbConnection connection, IDbTransaction transaction)
    {
        IDbCommand cmd = null;
        bool dbConnectionIsNull = true;
        if (connection != null) dbConnectionIsNull = false;
        IDataAdapter iDataAdapter = null;

        DataBindingSource ds = new DataBindingSource();
        ds.RecordSet = new DataSet();
        ds.Output = new DataTable("Output");
        ds.Return = 0;

        try
        {
            if (dbConnectionIsNull) connection = dataSource.GetConnection();
        }
        catch{};;
        // 중략...
    }

internal class CharacterDao
{
    IDataSource dataSource;
    DBManager db;
    IXmlPropertyHelper sqlHelper;
    SqlConnection mssqlconnection;
    IDbTransaction tran;
    // 바뀌기전 코드
    public CharacterModel GetCharacter(long characterId)
    {
        SqlParameter sql = new SqlParameter(ValidConfigHelper.SQLname_SqlCharacterTemplate, sqlHelper);
        sql.AddParameter("@id", "long", characterId);
        return DataModelUtil.GetDataModel<CharacterModel>(db.ExecuteDataSet(sql,3).Tables[0]);
    }
    // connection 정보를 멤버변수로 추가하여 재사용
    public CharacterModel GetCharacter(long characterId)
    {
        LogD.Common.Data.Parameter.SqlParameter sql = new LogD.Common.Data.Parameter.SqlParameter(ValidConfigHelper.SQLname_SqlCharacterTemplate, sqlHelper);
        sql.AddParameter("@id", "long", characterId);
        return DataModelUtil.GetDataModel<CharacterModel>(db.ExecuteDataSet(sql, 3, this.mssqlconnection, tran).Tables[0]);
    }
}
```

5. 로그 개발 및 운영시 해결했던 일감 정리

#246047 [AAKR][로그] 로그D 개선

6.EventID: 1103 채팅로그에 특수문자 Emoji 4Byte짜리 딸기 이모티콘이 와서 로그 처리중 에러발견.
정규식 방법으로 생각했으나 정규식 버전 업그레이드가 잦음.

13:58:09,606 [LogD,11,M,1,bulk] ERROR DBManager - MySql.Data.MySqlClient.MySqlException (0x80004005):
Incorrect string value: '\xF0\x9F\x8D\x93\xEC\xA7...' for column 'strLog' at row 1

LogDB의 Character, Collation 확인하니 utf8, utf8_general_ci 형식 이었다.

SHOW VARIABLES WHERE Variable_name LIKE 'character_set_%' OR Variable_name LIKE 'collation%';

구글링중 mysql은 칼럼단위로도 캐릭터셋을 변경 가능하여

LogD 프로젝트에서 LogDB connectionstring에 charset=utf8mb4; 추가

stageTable에 strLog 컬럼 charset변경 (sql_games.xml)

sqlStageMakeTable 에 strLog를 strLog varchar(10000) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL, 로변경!

SPLIT_STR 함수 변경 ,xlp_Job_MakeTable 변경 하여 문제 해결.

```
foreach (Entry entry in entryList)
{
    try
    {
        #region 캐릭터 정보 입력이 아니라면
        if (CharacterFieldType.NONE.Equals(entry.GetCharacterFieldType()))
        {
            캐릭터 정보 입력이 아니고, 타이틀 형식이라면
            반복 데이터 형
            #region //캐릭터 정보 입력도 아니고, 타이틀 형식도 아님 // 즉 원시 로그 자체
            else
            {
                // 20200116 topojs8 eventID 채팅로그 chatMessage안에 4byte Emoji 이모티콘 예외처리
                if (tokens[1]=="1103" && entry.name == "chatMessage" )
                {
                    Regex regex = new Regex("([₩u2000-₩u3300]|[₩ud000-₩udfff]|₩ud83d[₩ud000-₩udfff]|₩ud83e[₩ud000-₩udfff])");
                    if (regex.IsMatch(tokens[tokenIndex]))
                    {
                        try
                        {
                            tokens[tokenIndex]  = Regex.Replace(tokens[tokenIndex], "(₩u00a9|₩u00ae|[₩u2000-₩u3300]|₩ud83c[₩ud000-₩udfff
                        }
                        catch
                        {

                        }

                    }
                }
                #region 원시데이터 입력
                setLogValue(validatorCheckModel.GetResult(), entry, tokens[tokenIndex]);
```

Result Grid | Filter Rows: | Export:

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
collation_connection	utf8_general_ci
collation_database	utf8_general_ci
collation_server	utf8_general_ci

5. 로그 개발 및 운영시 해결했던 일감 정리

#255256 [X2][LOG] (신규) ABox '소속 서버 항목' 추가 작업
1.Live 환경에 1차테이블,2차테이블에 정보를 추가해야하는데 정기점검시간에 가능한지 확인
2.
•테스트 Table : T_Log2Doodads (약 71,000,000 rows, 일자별 range partition)
•소요 시간: 26 min 58 sec
Live환경에서 로그db 테이블 필드 추가시
7100만rows 기준 27준 소요 되었습니다.
현재 35번 월드 T_Log2Item테이블의 13억rows 기준
약 9시간 이상 소요될것으로 판단되어 어려울것 같습니다.

전체로그(1차 테이블)에만 '캐릭터 소속 월드' 정보가 나오도록 작업 진행
-> 로그중 캐릭터 기본정보가 있는 필드만 자료구조로 ref되어있는데 해당 정보에 n4ServerNum 필드 추가 후 stagedb에 n4ServerNum 필드 추가후 insert.
Abox insert sp, LogDB처리 sp에도 칼럼 추가. 프로그램팀과 사용하는 로그 정리및 캐릭터 기본정보를 제외한 필요한 서버필드 로그 추가작업.

#로그서버 꺼질시 자동 시작되게 windows 타임스케줄러에 등록. Batch파일 <<watchdog.bat>>

```
@ECHO OFF
 set count=0
 set sleeptime=5
 :loop
     set nowdatetime=%date% %time%
     for /f "tokens=1,2,3,4,5,6 delims=-:." %%a in ("%nowdatetime%") do ( set bootlogdatetime=%%a³a %%b¿u %%cAI %%d¼A %%eºÐ %%fAE )

     wmic process list brief | find /i "infra_log_server.exe"

     set result=%ERRORLEVEL%

     if %count% equ 0 (
            if "%result%" == "1" (
                echo %bootlogdatetime% 에 서비스가 멈춤
                echo %bootlogdatetime% 에 서비스가 멈춤 >> Log_ARM.txt
                set /a count+=1
                echo 서비스를 재실행 합니다.


  start /B cmd /C CALL "C:\Work\repos-infradev\trunk\Servers\code_infra\bin64\infra_log_server.exe"
  set count=0
  timeout /t %sleeptime% > NUL

  goto :loop
           ) else (
           set count=0
           )
     echo 서비스가 정상 작동중입니다.
     )
     timeout /t %sleeptime% > NUL

     goto :loop

 :end

 echo pause

 pause
```

5. 로그 개발 및 운영시 해결했던 일감 정리

#256704 [X2][LOG] 캐릭터 소속서버와다른서버에서 플레이시 월드DB정보를 가져오지 못하는 현상
1)
소속서버와 다른 정원서버에서 캐릭터가 플레이시 아래 EventID 로그들의 현재
정상적으로 조회가 불가능합니다. (해당 서버에 worldb만 참조하기때문에)
1.인증db를 참조하여 로그처리함.
2.인증db에 7개의 로그d서버를 한줄당 처리하다보니 cpu가 90퍼센이 상 발생
3. 인증db쿼리를 확인해보니

```
select
        [account_id]    as accountId
    ,[world_id]          as wid
    ,[char_id]       as id
    ,[name]              as name
    ,[race]              as race
    ,[gender]            as gender
        ,acc.account        as accountName
    From
    [x2account].[x2account].[dbo].[account_characters]
   ac  with(nolock)
    inner join
    [x2account].[dbo].[accounts]
    acc with(nolock)
    on ac.[account_id] = acc.[id]
    Where   ac.[char_id] = @id
```

에서 where 조건에 cid가 인덱스가 잡히지 않아 fullscan 하였습니다. aid를 and조건에 추가하였고
중복된 인증db정보를 select 을 방지하기 위해 authCache를 사용하여 6시간 주기로 해당 캐시를 자
료구조화해 cid가 캐시자료구조 안에 없을때만 쿼리를 조회하게 했습니다.
캐시를 6시간 주기로 한 이유는 캐릭터정보와,계정명은 바뀔일이 거의 없기때문에 운영팀과 상의후에
진행하였습니다.

```
// auth db 기준의 캐릭터 정보
private LRUCache<long, CharacterModel> AuthlruCache;

    // 파일단위로 로그처리하는 함수
    public override void InnerWork()
    {

        lock (this)
        {
            if (CommandType.QUIT.Equals(nextCommand))
            {
                return;
            }

            currCommand = nextCommand;
            nextCommand = CommandType.NONE;
        }
        //캐릭터 삭제 캐싱 제거
        if (serverProcessInfo.useWorldDB && nextCharacterCacheRemoveDateTime < DateTime.Now)
        {
            setNextCharacterCacheRemoveDateTime();
            getCharacterHelper(this.sqlHelper).RemoveCharacterList();
        }
        //6시간 주기로 AuthLRUCache를 비우고 새로운 정보로 로드
        if (serverProcessInfo.useWorldDB && nextAuthCharacterCacheRemoveDateTime < DateTime.Now)
        {
            setAuthCharacterCacheRemoveDateTime(authCharacterCacheRemoveCheckHour);
            getCharacterHelper(this.sqlHelper).RemoveAuthlruCache();
        }

        for (index = 0; index < checkList.Count; index++)
        {
             line = checkList[index];
             GetValidator().Check(checkModel, ref line, index);
        }
        //...중략
    }


//인증캐시 사용부분
 if (!AuthlruCache.ContainsKey(characterId))
 {
     temp = characterDao.GetCharacterLinkedAuth(characterId, accountId);
     if (temp != null)
     {

         addCharacter(AuthlruCache, characterId, temp);
     }
     else
     {
         return null;
     }
 }
```

5. 로그 개발 및 운영시 해결했던 일감 정리

#?로그 Viewer Web(Node.js)에 로그정의서 툴 추가 노드js 웹 로그뷰어 -> 동작방식 이해하고 포멧
#?중국,북미 로그D 코드 분석 ...mysql mongodb sqllight mssql 썻던것과 이유 관련 LogD 소스분석