대분류/20
정보통신

중분류/01
정보기술

소분류/02
정보기술개발

세분류/02
응용SW엔지니어링

능력단위/06

NCS학습모듈

통합 구현

LM2001020206_16v4

교육부

NCS 학습모듈은 교육훈련기관에서 출처를 명시하고 교육적 목적으로 활용할 수 있습니다. 다만 NCS 학습모듈에는 국가(교육부)가 저작재산권 일체를 보유하지 않은 저작물들(출처가 표기되어 있는 도표, 사진, 삽화, 도면 등)이 포함되어 있으므로 이러한 저작물들의 변형, 복제, 공연, 배포, 공중 송신 등과 이러한 저작물들을 활용한 2차 저작물의 생성을 위해서는 반드시 원작자의 동의를 받아야 합니다.

NCS 학습모듈의 이해

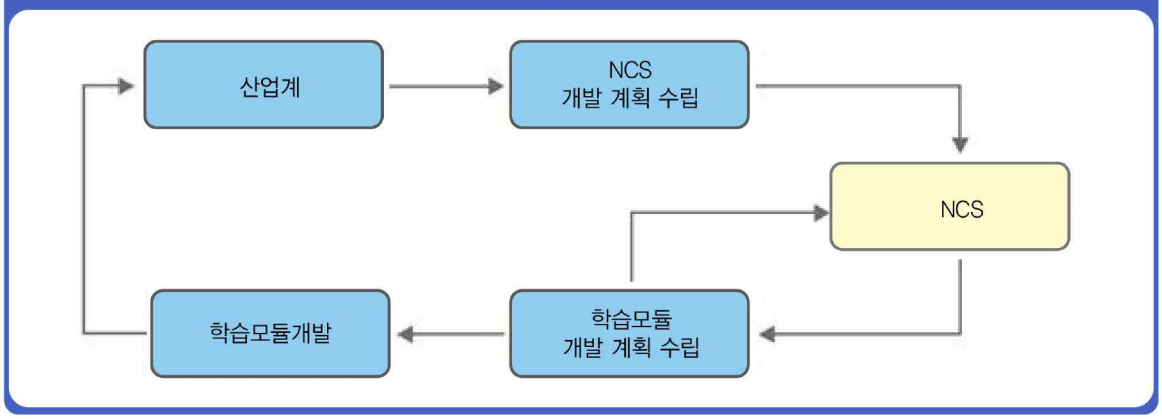
※ 본 학습모듈은 「NCS 국가직무능력표준」 사이트(http://www.ncs.go.kr) 에서 확인 및 다운로드 할 수 있습니다.

(1) NCS 학습모듈이란?

- 국가직무능력표준(NCS: National Competency Standards)이란 산업현장에서 직무를 수행하기 위해 요구되는 지식·기술·소양 등의 내용을 국가가 산업부문별·수준별로 체계화한 것으로 산업현장의 직무를 성공적으로 수행하기 위해 필요한 능력(지식, 기술, 태도)을 국가적 차원에서 표준화한 것을 의미합니다.

- 국가직무능력표준(이하 NCS)이 현장의 '직무 요구서'라고 한다면, **NCS 학습모듈은 NCS의 능력단위를 교육훈련에서 학습할 수 있도록 구성한 '교수·학습 자료'입니다.** NCS 학습모듈은 구체적 직무를 학습할 수 있도록 이론 및 실습과 관련된 내용을 상세하게 제시하고 있습니다.

NCS 및 학습모듈 개발 절차

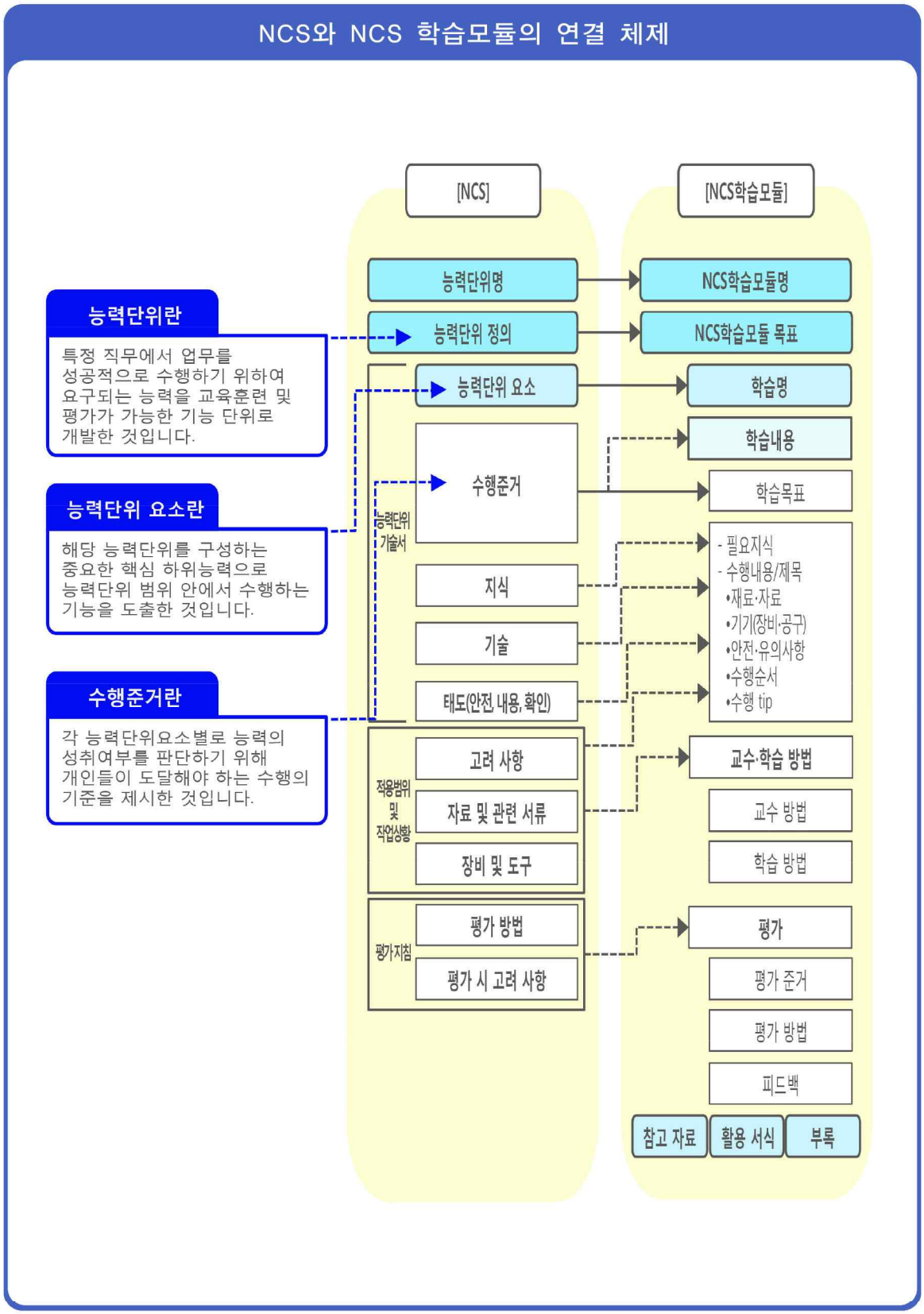


- **NCS 학습모듈은 다음과 같은 특징을 가지고 있습니다.**

  첫째, NCS 학습모듈은 산업계에서 요구하는 직무능력을 교육훈련 현장에 활용할 수 있도록 성취목표와 학습의 방향을 명확히 제시하는 가이드라인의 역할을 합니다.

  둘째, NCS 학습모듈은 특성화고, 마이스터고, 전문대학, 4년제 대학교의 교육기관 및 훈련기관, 직장교육기관 등에서 표준교재로 활용할 수 있으며 교육과정 개편 시에도 유용하게 참고할 수 있습니다.

● NCS와 NCS 학습모듈 간의 연결 체제를 살펴보면 아래 그림과 같습니다.



(2) NCS 학습모듈의 체계

- NCS 학습모듈은 1.학습모듈의 위치, 2.학습모듈의 개요, 3.학습모듈의 내용 체계, 4.참고 자료, 5.활용 서식/부록 으로 구성되어 있습니다.

1. NCS 학습모듈의 위치

- NCS 학습모듈의 위치는 NCS 분류 체계에서 해당 학습모듈이 어디에 위치하는지를 한 눈에 볼 수 있도록 그림으로 제시한 것입니다.

예시 : 이·미용 서비스 분야 중 네일미용 세분류

NCS-학습모듈의 위치

대분류	이용·숙박·여행·오락·스포츠
중분류	이·미용
소분류	이·미용 서비스

세분류
헤어미용
피부미용
메이크업
네일미용
이용

능력단위	학습모듈명
네일 샵 위생 서비스	네일숍 위생서비스
네일 화장물 제거	네일 화장물 제거
네일 기본 관리	네일 기본관리
네일 랩	네일 랩
네일 팁	네일 팁
젤 네일	젤 네일
아크릴릭 네일	아크릴 네일
평면 네일아트	평면 네일아트
융합 네일아트	융합 네일아트
네일 샵 운영관리	네일숍 운영관리

학습모듈은

NCS 능력단위 1개당 1개의 학습모듈 개발을 원칙으로 합니다. 그러나 필요에 따라 고용 단위 및 교과단위를 고려하여 능력단위 몇 개를 묶어서 1개의 학습모듈로 개발할 수 있으며, NCS 능력단위 1개를 여러 개의 학습모듈로 나누어 개발할 수도 있습니다.

2. NCS 학습모듈의 개요

구 성

- NCS 학습모듈 개요는 학습모듈이 포함하고 있는 내용을 개략적으로 설명한 것으로서 학습모듈의 목표, 선수 학습, 학습모듈의 내용 체계, 핵심 용어로 구성되어 있습니다.

학습모듈의 목표	해당 NCS 능력단위의 정의를 토대로 학습목표를 작성한 것입니다.
선수 학습	해당 학습모듈에 대한 효과적인 교수·학습을 위하여 사전에 이수해야 하는 학습모듈, 학습 내용, 관련 교과목 등을 기술한 것입니다.
학습모듈의 내용 체계	해당 NCS 능력단위요소가 학습모듈에서 구조화된 방식을 제시한 것입니다.
핵심 용어	해당 학습모듈의 학습 내용, 수행 내용, 설비·기자재 등 가운데 핵심적인 용어를 제시한 것입니다.

활 용 안 내

예시 : 네일미용 세분류의 '네일 기본관리' 학습모듈

네일 기본관리 학습모듈의 개요

학습모듈의 목표

고객의 네일 보호와 미적 요구 충족을 위하여 효과적인 네일 관리로 프리에지 형태 만들기, 큐티클 정리하기, 컬러링하기, 보습제 도포하기, 마무리를 할 수 있다.

선수학습

네일숍 위생서비스(LM1201010401_14v2)

학습모듈의 내용체계

학습	학습내용	NCS 능력단위요소 코드번호	NCS 능력단위요소 요소명칭	수준
1. 프리에지 형태 만들기	1-1. 네일 파일에 대한 이해와 활용 1-2. 프리에지 형태 파일링	1201010403_12v2.1	프리엣지 모양 만들기	3
2. 큐티클 정리하기	2-1. 네일 기본관리 매뉴얼 이해 2-2. 큐티클 관리	1201010403_14v2.2	큐티클 정리하기	3
3. 컬러링하기	3-1. 컬러링 매뉴얼 이해 3-2. 컬러링 방법 선정과 작업 3-3. 젤 컬러링 작업	1201010403_14v2.3	컬러링	3
4. 보습제 도포하기	4-1. 보습제 선정과 도포 4-2. 각질제거	1201010403_14v2.4	보습제 바르기	2
5. 네일 기본관리 마무리하기	5-1. 유분기 제거 5-2. 네일 기본관리 마무리와 정리	1201010403_14v2.5	마무리하기	3

핵심 용어

프리에지, 니퍼, 푸셔, 폴리시, 네일 파일, 스퀘어형, 스퀘어 오프형, 라운드형, 오발형, 포인트형

학습모듈의 목표는

학습자가 해당 학습모듈을 통해 성취해야 할 목표를 제시한 것으로, 교수자는 학습자가 학습모듈의 전체적인 내용흐름을 파악할 수 있도록 지도하는 것이 필요합니다.

선수학습은

교수자나 학습자가 해당 모듈을 교수 또는 학습하기 이전에 이수해야 할 학습내용, 교과목, 핵심 단어 등을 표기한 것입니다. 따라서 교수자는 학습자가 개별 학습, 자기 주도 학습, 방과 후 활동 등 다양한 방법을 통해 이수할 수 있도록 지도하는 것이 필요합니다.

핵심 용어는

학습모듈을 통해 학습되고 평가되어야 할 주요 용어입니다. 또한 당해 모듈 또는 타 모듈에서도 핵심 용어를 사용하여 학습내용을 구성할 수 있으며, 「NCS 국가직무능력표준」 사이트(www.ncs.go.kr)에서 색인(찾아보기) 중 하나로 이용할 수 있습니다.

3. NCS 학습모듈의 내용 체계

구 성

● NCS 학습모듈의 내용은 크게 학습, 학습 내용, 교수·학습 방법, 평가로 구성되어 있습니다.

학습	해당 NCS 능력단위요소 명칭을 사용하여 제시한 것입니다. 학습은 크게 학습 내용, 교수·학습 방법, 평가로 구성되며 해당 NCS 능력단위의 능력단위 요소별 지식, 기술, 태도 등을 토대로 학습 내용을 제시한 것입니다.
학습 내용	학습 내용은 학습 목표, 필요 지식, 수행 내용으로 구성하였으며, 수행 내용은 재료·자료, 기기(장비·공구), 안전·유의 사항, 수행 순서, 수행 tip으로 구성한 것입니다. 학습모듈의 학습 내용은 업무의 표준화된 프로세스에 기반을 두고 실제 산업현장에서 이루어지는 업무활동을 다양한 방식으로 반영한 것입니다.
교수·학습 방법	학습 목표를 성취하기 위한 교수자와 학습자 간, 학습자와 학습자 간의 상호작용이 활발하게 일어날 수 있도록 교수자의 활동 및 교수 전략, 학습자의 활동을 제시한 것입니다.
평가	평가는 해당 학습모듈의 학습 정도를 확인할 수 있는 평가 준거, 평가 방법, 평가 결과의 피드백 방법을 제시한 것입니다.

활 용 안 내

예시 : 네일미용 세분류의 '네일 기본관리' 학습모듈의 내용

학습 1	프리에지 형태 만들기(LM1201010403_14v2.1)
학습 2	큐티클 정리하기(LM1201010403_14v2.2)
학습 3	**컬러링하기(LM1201010403_14v2.3)**
학습 4	보습제 도포하기(LM1201010403_14v2.4)
학습 5	네일 기본관리 마무리하기(LM1201010403_14v2.5)

학습은

해당 NCS 능력단위요소 명칭을 사용하여 제시하였습니다. 학습은 일반교과의 '대단원'에 해당되며, 모듈을 구성하는 가장 큰 단위가 됩니다. 또한 완성된 직무를 수행하기 위한 가장 기존적인 단위로 사용할 수 있습니다.

3-1. 컬러링 매뉴얼 이해

학습내용은

요소 별 수행준거를 기준으로 제시하였습니다. 일반교과의 '중단원'에 해당합니다.

학습목표	• 고객의 요구에 따라 네일 폴리시 색상의 침착을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다. • 작업 매뉴얼에 따라 네일 폴리시를 얼룩 없이 균일하게 도포할 수 있다. • 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다.

학습목표는

모듈 내의 학습내용을 이수했을 때 학습자가 보여줄 수 있는 행동수준을 의미합니다. 따라서 일반 수업시간의 과목목표로 활용할 수 있습니다.

필요 지식 /

1 컬러링 매뉴얼

컬러링 작업 전, 아세톤 또는 네일 폴리시 리무버를 사용하여 손톱표면과 큐티클 주변, 손톱 밑 부분까지 깨끗하게 유분기를 제거해야 한다. 컬러링의 순서는 Base coating 1회 → Polishing 2회 → 칼라수정 → Top coating 1회 → 최종수정의 순서로 한다. 베이스코트는 착색을 방지하고 발림성 향상을 위해 가장 먼저 도포하며 컬러링의 마지막에 컬러의 유지와 광택을 위해 톱코트를 도포한다. 네일 보강제(Nail Strengthner)를 바를 시에는 베이스코트를 도포하기 전에 사용한다.

필요지식은

해당 NCS의 지식을 토대로 해당 학습에 대한 이해와 성과를 높이기 위해 알아야 할 주요 지식을 제시하였습니다. 필요지식은 수행에 꼭 필요한 핵심 내용을 위주로 제시하여 교수자의 역할이 매우 중요하며, 이후 수행순서 내용과 연계하여 교수·학습으로 진행할 수 있습니다.

수행 내용 / 컬러링 매뉴얼 실습하기

재료 · 자료

- 컬러링 관련 네일 미용 자료들
- 정리바구니, 베이스코트, 네일 폴리시, 톱코트, 오렌지우드스틱, 탈지면, 폴리시리무버, 디스펜서 등

기기(장비 · 공구)

- 컴퓨터, 빔 프로젝터, 스크린 등

안전 · 유의사항

- 컬러링 재료들의 냄새를 직접적으로 맡지 않도록 유의한다.
- 컬러링 제품들이 대부분 유리병에 들어 있기 때문에 깨지지 않도록 각별히 조심한다.
- 컬러링 제품들은 상온에 마르기 때문에 개봉 후 뚜껑을 잘 닫도록 한다.

수행 순서

[1] 네일 폴리시를 바르게 잡는다.

1. 손바닥에 네일 폴리시를 놓고 약지 소지를 이용하여 네일 폴리시를 잡는다.
2. 폴리시를 쥔 손의 엄지와 검지로 고객의 작업손가락을 잡는다.
3. 폴리시를 쥔 손의 중지 손가락을 곧게 펴서 받침대가 되도록 한다.
4. 반대편 손으로 네일 폴리시의 뚜껑을 열고 소지 손가락을 펴서 네일 폴리시를 쥔 중지 손가락 위에 받쳐놓는다.
5. 다양한 형태의 폴리시를 잡아본다.

수행 tip

- 흰색이 많이 섞인 네일 폴리시의 경우는 붓의 각도를 높이 세워 빠르게 브러시 작업을 해야 붓 자국이 나지 않는다.
- 컬러링은 기본 2회 정도이나 컬러에 따른 도포량과 컬러감에 따라 1~3회 사이로 증감할 수 있다.

수행 내용은

모듈에 제시한 것 중 기술(Skill)을 습득하기 위한 실습 과제로 활용할 수 있습니다.

재료·자료는

수행 내용을 수행하는데 필요한 재료 및 준비물로 실습 시 필요 준비물로 활용할 수 있습니다.

기기(장비·공구)는

수행 내용을 수행하는데 필요한 기본적인 장비 및 도구를 제시하였습니다. 제시된 기기 외에도 수행에 필요한 다양한 도구나 장비를 활용할 수 있습니다.

안전·유의사항은

수행 내용을 수행하는데 안전상 주의해야 할 점 및 유의사항을 제시하였습니다. 수행 시 유념해야 하며, NCS의 고려사항도 추가적으로 활용할 수 있습니다.

수행 순서는

실습과제의 진행 순서로 활용할 수 있습니다.

수행 tip은

수행 내용에서 수행의 수월성을 높일 수 있는 아이디어를 제시하였습니다. 따라서 수행tip은 지도상의 안전 및 유의사항 외에 전반적으로 적용되는 주안점 및 수행과제 목적에 대한 보충설명, 추가사항 등으로 활용할 수 있습니다.

학습3 교수·학습 방법

교수 방법

- 컬러링 제품의 성분과 칼라별 점도의 차이, 베이스코트와 톱코트의 역할, 폴리시 잡는 방법, 큐어링 시간 등의 내용을 화면 자료와 함께 설명한다.
- 서식지를 활용하여 네일 컬러링 방법을 그림으로 그려 보게 한 뒤, 다양한 컬러링의 매뉴얼을 그려서 숙지하도록 한다.
- 젤 컬러링 시 주의사항을 계속 숙지시키도록 하며, 큐어링 시간에 대해 작성하도록 한다.

학습 방법

- 컬러링을 위한 재료의 필요성과 사용방법을 숙지하고 컬러링 매뉴얼 과정에 맞추어 작업 내용을 이해한다.
- 컬러링의 다양성에 대한 용어를 숙지하고 진행과정에 맞추어 내용을 작업한다.
- 젤 컬러링 시 적합한 큐어링 시간을 선택해서 큐어링 해본다.

교수·학습 방법은

학습목표를 성취하는데 필요한 교수 방법과 학습 방법을 제시하였습니다.

교수 방법은

해당 학습활동에 필요한 학습내용, 학습내용과 관련된 학습자료명, 자료 형태, 수행내용의 진행 방식 등에 대하여 제시하였습니다. 또한 학습자의 수업참여도를 제고하기 위한 방법 및 수업진행상 유의사항 등도 제시하였습니다. 선수학습이 필요한 학습을 학습자가 숙지하였는지 교수자가 확인하는 과정으로 활용할 수도 있습니다.

학습 방법은

해당 학습활동에 필요한 학습자의 자기주도적 학습 방법을 제시하였습니다. 또한 학습자가 숙달해야 할 실기능력과 학습과정에서 주의해야 할 사항 등으로 제시하였습니다. 학습자가 학습을 이수하기 전에 반드시 숙지해야 할 기본 지식을 학습하였는지 스스로 확인하는 과정으로 활용할 수 있습니다.

학습3 평 가

평가는

해당 NCS 능력단위 평가방법과 평가 시 고려 사항을 준용하여 작성하였습니다. 교수자 및 학습자가 평가항목 별 성취수준을 확인하는데 활용할 수 있습니다.

평가 준거

- 평가자는 학습자가 학습 목표 및 평가 항목에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습내용	평가항목	성취수준 상	중	하
컬러링 매뉴얼 이해	- 고객의 요구에 따라 네일 폴리시 색상의 침착을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다.			
	- 작업 매뉴얼에 따라 네일 폴리시를 얼룩 없이 균일하게 도포할 수 있다.			
	- 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다.			

평가 준거는

학습자가 해당 학습을 어느 정도 성취하였는지를 평가하기 위한 기준을 제시하고 있습니다. 학습목표와 연계하여 단위수업 시간에 평가항목 별 성취수준을 평가하는데 활용할 수 있습니다.

평가 방법

- 작업장 평가

학습내용	평가항목	성취수준 상	중	하
컬러링 매뉴얼 이해	- 고객의 요구에 따라 네일 폴리시 색상의 침착을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다.			
	- 작업 매뉴얼에 따라 네일 폴리시를 얼룩 없이 균일하게 도포할 수 있다.			
	- 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다.			

평가 방법은

NCS 능력단위의 평가방법을 준용하였으며, 평가 준거에 따른 평가방법을 2개 이상 제시하였습니다. 평가방법으로는 포트폴리오, 문제해결 시나리오, 서술형 시험, 논술형 시험, 사례연구, 평가자 체크리스트, 작업장 평가 등이 있으며, NCS의 능력단위 요소 별 수행 수준을 평가하는데 가장 적절한 방법을 선정하여 활용할 수 있습니다.

피 드 백

1. 작업장 평가
   - 작업 결과물을 확인하여 수정사항을 제시하고 수정 부분을 인지하도록 한다.

피드백은

평가 후에 학습자들에게 평가 결과를 피드백하여 부족한 부분을 알려주고, 학습 결과가 미진한 경우, 해당 부분을 다시 학습하여 학습목표를 달성하는 데 활용할 수 있습니다.

4. 참고 자료

참고자료

- 김미원(2011). 『Nail Study』. 서울: 사)한국네일지식서비스협회.
- 민방경(2015). 『미용사(네일)필기』. 서울: 예문사.
- 박은주(2014). 『네일미용』. 서울: 정담미디어.

참고자료는

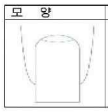
해당 학습모듈의 필요지식에 대한 출처와 인용한 참고자료 및 사이트를 제시하였습니다.

5. 활용 서식/부록

활용서식

프리에지 형태 실습지

1. 프리에지 형태의 이해

모 양	이 름	특 징
	( ) Square nail	·강한 느낌의 사각형태 ·네일의 양끝 모서리 부분이 90° 사각의 형태이다. ·( ) ·발톱의 형태 활용 ·내인성 발톱의 보정시에 적용

활용서식은

평가 서식, 실습시트 등 교수학습 시 활용 가능한 다양한 서식들로 구성하였습니다. 과제 진행에서 평가에 이르기까지 필요한 서식을 해당 학습모듈의 특성에 맞춰 개발하거나 기존의 양식을 활용하여 제시하였습니다

부록

네일 기본관리 도구와 재료 목록

목록	비고	준비
위생가운	흰색	작업자 착용
위생 마스트	흰색	작업자 착용
보호안경	투명한 렌즈 (안경으로 대체 가능)	작업자 착용
재료정리함	재질, 색상 무관	작업대

부록은

활용서식 이외에 교수학습과정에서 참고할 수 있는 자료가 있는 경우 제시하였습니다.

[NCS-학습모듈의 위치]

대분류	정보 통신
중분류	정보 기술
소분류	정보 기술 개발

세분류
SW아키텍처
응용SW 엔지니어링
임베디드SW 엔지니어링
DB엔지니어링
NW엔지니어링
보안엔지니어링
UI/UX엔지니어링
시스템SW 엔지니어링
빅데이터 플랫폼구축
핀테크 엔지니어링
데이터아키텍트

능력단위	학습모듈명
요구사항 확인	요구사항 확인
데이터 입출력 구현	데이터 입출력 구현
통합 구현	**통합 구현**
정보시스템 이행	정보시스템 이행
제품소프트웨어 패키징	제품소프트웨어 패키징
서버프로그램 구현	서버프로그램 구현
인터페이스 구현	인터페이스 구현
애플리케이션 배포	애플리케이션 배포
프로그래밍 언어 활용	프로그래밍 언어 활용
응용 SW 기초 기술 활용	응용 SW 기초 기술 활용
애플리케이션 리팩토링	애플리케이션 리팩토링
인터페이스 설계	인터페이스 설계
애플리케이션 요구사항 분석	애플리케이션 요구사항 분석

기능 모델링	기능 모델링
애플리케이션 설계	애플리케이션 설계
정적모델 설계	정적모델 설계
동적모델 설계	동적모델 설계
화면 설계	화면 설계
화면 구현	화면 구현
애플리케이션 테스트 관리	애플리케이션 테스트 관리
애플리케이션 테스트 수행	애플리케이션 테스트 수행
소프트웨어공학 활용	소프트웨어공학 활용
소프트웨어개발 방법론 활용	소프트웨어개발 방법론 활용

차 례

통합 구현의 개요 ---------- 1

학습 1. 연계 데이터 구성하기

1-1 연계 요구 사항 분석 ---------- 3

1-2 연계 데이터 식별 및 표준화 ---------- 14

- 교수·학습 방법 ---------- 23
- 평가 ---------- 24

학습 2. 연계 메커니즘 구성하기

2-1 연계 메커니즘 정의 ---------- 26

2-2 연계 장애 및 오류처리 구현 ---------- 41

2-3 연계 데이터 보안 적용 ---------- 56

- 교수·학습 방법 ---------- 64
- 평가 ---------- 66

학습 3. 내외부 연계 모듈 구현하기

3-1 연계 모듈 구현 환경 구성 및 개발 ---------- 69

3-2 연계 테스트 및 검증 ---------- 90

- 교수·학습 방법 ---------- 101
- 평가 ---------- 103

활용 서식 ---------- 107

통합 구현 학습모듈의 개요

학습모듈의 목표

모듈 간의 분산이 이루어진 경우를 포함하여 단위 모듈 간의 데이터 관계를 분석하고, 이를 기반으로 한 메커니즘 모듈 간의 효율적인 연계를 구현하여 검증할 수 있다.

선수학습

현행 시스템 분석(2001020201_14v2.1), 요구 사항 확인(2001020201_14v2.2), 배치 프로그램 구현(2001020203_14v2.3), 개발자 단위 테스트(2001020203_14v2.4), 개발자 테스트 케이스 설계(2001020207_14v2.1), 개발자 통합 테스트(2001020207_14v2.2)

학습모듈의 내용 체계

학습	학습 내용	NCS 능력단위 요소	
		코드 번호	요소 명칭
1. 연계 데이터 구성하기	1-1. 연계 요구 사항 분석 1-2. 연계 데이터 식별 및 표준화	2001020206_16v4.1	연계 데이터 구성하기
2. 연계 메커니즘 구성하기	2-1. 연계 메커니즘 정의 2-2. 연계 장애 및 오류 처리 구현 2-3. 연계 데이터 보안 적용	2001020206_16v4.2	연계 매카니즘 구성하기
3. 내외부 연계 모듈 구현하기	3-1. 연계 모듈 구현 환경 구성 및 개발 3-2. 연계 테스트 및 검증	2001020206_16v4.3	내외부 연계 모듈 구현하기

핵심 용어

연계 데이터, 연계 메커니즘, 송수신 현황 모니터링, 데이터 보안, 연계 모듈, EAI(Enterprise Application Integration)/ESB(Enterprise Service Bus), 엔터티 관계도(ERD: Entity Relationship Diagram), 연계(인터페이스) 정의서, 연계(인터페이스) 설계서

학습 1	연계 데이터 구성하기
학습 2	연계 메커니즘 구성하기
학습 3	내외부 연계 모듈 구현하기

1-1. 연계 요구 사항 분석

학습 목표	• 개발하고자 하는 응용 소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구 사항을 분석할 수 있다.

필요 지식 /

1 통합 구현 개념

통합 구현은 사용자의 요구 사항을 해결하고, 새로운 서비스 창출을 위해 단위 기능을 하는 모듈 간의 연계와 통합이다.

통합 구현은 시스템 아키텍처 구성, 송수신 방식, 송수신 모듈 구현 방법 등에 따라 다양하므로 구축하고자 하는 환경과 사용자 요구 사항에 따라 적합한 통합 구현 방법을 설계한다.

[그림 1-1] 통합 구현(연계 시스템 아키텍처) 개념도

다양한 통합 구현 방식은 일반적으로 송신 시스템과 모듈, 중계 시스템, 수신 시스템과 모듈로 구성되며, 각 구성 요소별 수행 역할 및 기능은 다음과 같다.

1. 송신 시스템과 모듈

송신 시스템은 전송하고자 하는 데이터를 생성하여 필요에 따라 변환 후 송신하는 송신 모듈과 데이터 생성 및 송신 상태를 모니터링하는 기능으로 구성된다.

송신 모듈은 내부 연계 시에도 필수 요소에 해당하며 일반적으로 연계 솔루션이 적용될 경우에는 데이터 생성 처리만 구현한다.

2. 중계 시스템

중계 시스템은 주로 외부 시스템 간의 연계 시에 적용되는 아키텍처로, 내외부 구간의 분리로 보안성이 강화되고, 인터넷 망(Internet)과 인트라넷 망(Intranet)을 연결할 수도 있다.

중계 모듈은 송신된 데이터의 오류 처리 및 수신 시스템의 데이터 형식으로 변환 또는 매핑 등을 수행한다.

3. 수신 시스템과 모듈

수신 시스템은 수신 받은 데이터를 정제(Cleansing)하고, 응용 애플리케이션이나 데이터베이스의 테이블에 적합하도록 변환하여 반영하는 수신 모듈과 연계 데이터의 수신 현황 및 오류 처리, 데이터 반영을 모니터링 하는 기능으로 구성된다.

4. 연계 데이터

송수신되는 데이터로 의미를 갖는 속성, 길이, 타입 등이 포함된다. 송수신되는 연계 데이터 형식은 크게 데이터베이스(DB: Database)의 테이블과 칼럼, 파일로 분류할 수 있으며 파일은 세분화하여 text, xml, csv 등 다양한 형식으로 구분할 수 있다.

5. 네트워크

송신 시스템과 수신 시스템, 송신 시스템과 중계 시스템, 중계 시스템과 수신 시스템을 연결해주는 통신망으로 유선 또는 무선, 인터넷 서비스 공급자(ISP: Internet Service Provider) 사업자의 공중 망 또는 사설 망(전용선 포함)과 같은 유무선의 물리적인 망과 송수신 규약을 위한 프로토콜(Protocol)을 의미한다.

2 연계 요구 사항 분석

통합 구현을 위해 연계 시스템 아키텍처를 설계하고, 연계 데이터를 정의하기 위해서는 사용자 요구 사항 분석은 중요한 작업이다.

사용자 요구 사항은 연계 데이터와 연계 환경 구성을 위해 성능, 보안, 데이터 발생 패턴 등 다양한 관점에서 분석한다.

요구 사항 분석 시 입력물과 도구 및 기법, 분석 결과물은 다음과 같다.

입력물		도구 및 기법		출력물
시스템 구성도 (H/W, N/W, S/W) 응용 애플리케이션 구성 ERD·테이블(코드) 정의서	⇨	사용자 인터뷰, 체크리스트 설문지 및 설문 조사 델파이 기법 연계 솔루션 비교 분석	⇨	**요구 사항 분석서** **인터페이스 정의서** (데이터 정의서) 회의록

[그림 1-2] 요구 사항 분석 입력물, 도구 및 기법, 결과물

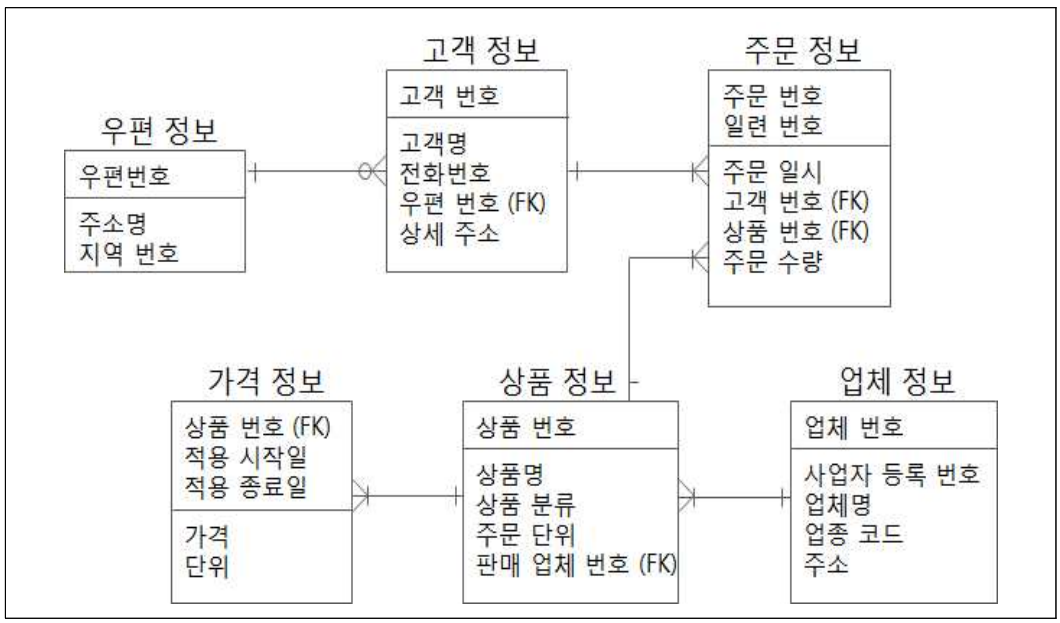
1. 요구 사항 분석 시 입력물

송수신 시스템과 운영되는 데이터의 이해를 통해 사용자의 요구 사항의 정확한 분석이 가능하다. 식별된 사용자 요구 사항은 송수신 시스템 운영 환경과 데이터를 적용하여 연계 방식, 연계 주기, 연계 데이터 등을 분석한다.

(1) 시스템 구성도: 송수신 시스템의 네트워크, 하드웨어, 시스템 소프트웨어 구성

(2) 응용 애플리케이션 구성: 송신 측에서 연계 데이터가 발생하는 응용 애플리케이션의 메뉴 구조도, 화면 및 화면 설계서, 사용자 인터페이스 정의서
연계할 데이터의 발생 시점 및 주기, 발생 패턴 등을 확인

(3) 엔터티 관계도(ERD: Entity Relationship Diagram)·테이블(공통 코드) 정의서 : 데이터 모델링 기술서, 테이블 간의 연관도, 테이블별 칼럼(속성)이 정의된 테이블 정의서, 공통 코드 및 공통 코드값에 대한 설명서
사용자 요구 사항에서 데이터의 송수신 가능 여부, 데이터 형식 및 범위 등을 확인



[그림 1-3] 엔터티 관계도(ERD: Entity Relationship Diagram) 예시

<표 1-1> 테이블 정의서

테이블 ID		TB_EAS010				테이블명		채용 공고
데이터베이스명		채용 대행 DB				테이블 스페이스명		EA_DATA_TS010
트리거 구성		없음						
테이블 설명		채용 대행 서비스 대상 채용 공고의 상세 정보를 관리한다.						
초기 건수	증가량 (일)	보관 주기	최대 건수	용량	비고			
1,000	20	영구	100,000	300MB				
칼럼명	칼럼 ID	타입 및 길이	Not Null	PK	FK	IDX	기본 값	제약 조건
채용 공고 번호	EMPL_NTNO	VARCHAR(30)	Y	Y		IDX_EAS010_1		
채용 공고명	EMPL_KORN	VARCHAR(150)	Y			IDX_EAS010_1		
:	:	:	:	:	:	:	:	:

<표 1-2> 공통 코드 정의서

코드 분류(그룹)	코드ID	코드명	코드 설명	사용 여부
CD001	00	진행 상태	민원인이 제출한 신청서의 처리 단계 진행 상태	Y
CD001	01	제출		Y
CD001	02	접수		Y
CD001	03	승인		Y
:	:	:	:	:

2. 요구 사항 분석 시 도구 및 기법

효과적으로 연계 요구 사항을 분석하기 위해 설문 조사, 인터뷰, 체크리스트(Checklist)와 같은 도구 및 기법을 활용한다.

(1) 사용자 인터뷰, 핵심 사용자 그룹 면담(FGI: Focus Group Interview): 사용자 면담 또는 시스템 관리자 및 서비스 활용자와 같은 핵심 그룹 면담

연계 데이터 정의, 연계 데이터의 활용 목적, 필요성 등을 식별하기 위함으로 사용자 인터뷰 전 연계 대상 시스템의 응용 애플리케이션 기능, 서비스의 확인이 필요함

(2) (연계 분석) 체크리스트(Checklist): 연계 데이터와 연계 시스템 아키텍처 정의를 위해 시스템 운영 환경, 성능, 보안, 데이터 발생 등 다각도의 관점에서 고려 사항 점검 및 확인

<표 1-3> 연계 분석 체크리스트

<table>
<tr><th colspan="2">분야</th><th>확인 항목</th></tr>
<tr><td rowspan="2">시스템</td><td>네트워크</td><td>□ 1. 송신 시스템의 네트워크(Network) 구간?
□ 인터넷 망 □ 인트라넷 망
□ 2. 수신 시스템의 네트워크(Network) 구간?
□ 인터넷 망 □ 인트라넷 망
□ 3. 송수신 시스템 간 연계 시 인터넷 망과 연결 가능 여부?
□ Yes □ No
:</td></tr>
<tr><td>보안</td><td>□ 1. 연계 데이터의 보안 수준은?
ㅇ 채널(Channel, 통신 구간) ㅇ 데이터
□ 2. 데이터 암호화 시 적용 가능 알고리즘은?
:</td></tr>
<tr><td>응용</td><td>데이터</td><td>□ 1. 연계 대상 데이터의 시스템 관리 여부?
□ 송신 시스템 □ 수신 시스템
□ 2. 연계 대상 데이터 중 코드나 도메인(일정 범위의 값) 데이터는 ?
□ 송신 시스템 ( )
□ 수신 시스템 ( )
□ 3. 연계 데이터의 발생 및 반영 주기는?
ㅇ 실시간 ㅇ 분 단위 (분)
ㅇ 시간 단위 (시간) ㅇ 일 단위 (일)
□ 4. 송신 시스템에서 연계가 필요한 데이터?
<table>
<tr><th>테이블명</th><th>속성명</th><th>PK 여부</th><th>코드 여부</th><th>연계 필요 여부</th><th>연계 가능 여부</th></tr>
<tr><td rowspan="6">채용 공고</td><td>채용 공고 ID</td><td>Y</td><td></td><td>Y</td><td>Y</td></tr>
<tr><td>채용 공고명</td><td></td><td></td><td>Y</td><td>Y</td></tr>
<tr><td>채용 분야</td><td></td><td>Y</td><td>Y</td><td>Y</td></tr>
<tr><td>학력 코드</td><td></td><td>Y</td><td></td><td></td></tr>
<tr><td>자격 코드</td><td></td><td>Y</td><td></td><td></td></tr>
<tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr>
<tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr>
</table>
:</td></tr>
</table>

(3) 설문지 및 설문 조사: 서비스 활용 목적에 따라 연계가 필요한 데이터를 식별하고, 연계 주기 등을 분석하기 위해 설문 조사 항목을 통해 자료를 수집

객관식 문항으로 예상 답변을 일정 범위 이내로 한정할 수도 있음

(4) 델파이 기법: 통합 구현 및 연계 전문가, 시스템 아키텍처, 업무 전문가 등 각 분야 전문가로 부터 연계 데이터 및 사용자 요구 사항 식별

(5) 연계 솔루션(Solution) 비교 분석: EAI, ESB, Open API 등 다양한 연계 방식과 연계 솔루션별 연계 시의 성능, 보안, 데이터 처리, 모니터링 등의 장단점을 비교함

3. 요구 사항 분석 결과물

사용자 및 시스템 관리자 인터뷰, 체크리스트 등을 활용하여 연계 데이터와 연계 아키텍처

를 설계하기 위한 요구 사항을 식별하고 분석한다.

식별된 요구 사항은 요구 사항 정의서 또는 요구 사항 분석서에 요구 사항 유형, 요구 사항 설명, 해결 방안, 요구 사항에 대한 제약 조건, 중요도, 출처의 내용으로 작성하되, 연계 주기, 연계 방식, 연계 데이터를 식별하고 확인할 수 있도록 기술해야 한다.

연계 데이터와 관련한 요구 사항을 상세하고 설명하고 분석하기 위해서는 엔터티 관계도(ERD: Entity Relationship Diagram) 및 테이블 정의서와 같은 입력물을 활용한다.

<표 1-4> 연계 요구 사항 분석서

요구 사항 ID	요구 사항명	요구 사항 유형	요구 사항 설명	~
RM-IFM-001	조직 정보 LDAP과 연계	기능	- 정부 부처 조직정보, 사용자 발령 정보 등의 현행화를 위해 정부 디렉토리(LDAP: Light Directory Access Protocol)과 연계하여 반영 - 연계 정보에 개인 정보 포함으로 사용자 사전 동의 필요 - 조직 정보 및 사용자 정보가 변경 시마다 변경된 정보를 실시간으로 연동	~
RM-IFM-002	자산 정보 dBrain과 연계	기능	- 자산 관리 시스템과 디지털 예산 회계 dBrain 시스템의 dBrain 번호, 자산명, 소유 기관, 취득일 정보를 연계 - 연계 시 보안을 위해 VPN(Virtual Private Network)을 도입하여 송수신되는 정보 암호화 - 변경 여부와 관계없이 해당 기관의 전체 자산 정보를 1일 1회 연동	~
RM-IFM-003	연계 시스템 구축	비기능 (인프라, 표준화	- 외부 시스템과의 연계를 위해 중계 서버를 구성하여 외부에서 내부시스템에 직접 접근할 수 없도록 설계 - 연계 방식은 Open API나 EAI 등 비용과 보안, 유지 보수 등을 고려하여 표준화하여 구축	~
RM-IFM-004	연계 데이터 보안	비기능 (보안)	- 연계 데이터는 중요도에 따라 통신 구간 암호화, 데이터 암호화를 적용해야 함 - VPN과 같은 보안 장비, 솔루션 도입, 암호화 알고리즘 적용 시 국가정보원의 기준을 통과한 제품 및 알고리즘이어야 함	~
RM-IFM-005	연계 시 성능	비기능 (성능)	- 실시간 연계의 경우 운영하는 서비스 성능에 영향을 주지 않아야 함 - 보안 기능으로 인해 성능 저하가 발생하지 않도록 설계 반영	~
:	:	:	:	:

수행 내용 / 연계 요구 사항 분석하기

재료 · 자료

- 사용자 요구 사항 인터뷰 내용(면담서), 회의록
- 하드웨어 및 시스템 소프트웨어, 네트워크 등 시스템 구성도(송신 시스템, 수신 시스템)
- 테이블 정의서(송신 시스템, 수신 시스템)
- 코드 정의서(송신 시스템, 수신 시스템)
- 연계 분석을 위한 체크리스트(Checklist)
- 요구사항 분석서

기기(장비 · 공구)

- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

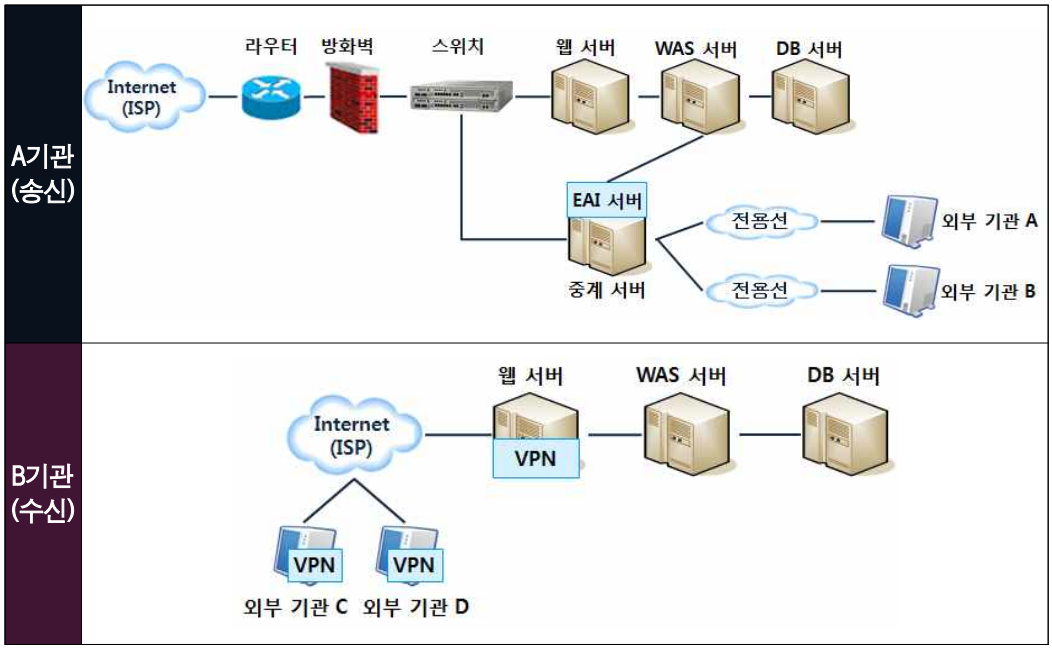
1 연계 요구 사항 분석을 위한 정보 수집 및 현황 확인 등 사전 작업을 준비한다.

연계 요구 사항 분석은 사용자 인터뷰 및 면담을 통해 식별되고, 확인된다.

사용자 인터뷰 전에 송신 시스템 및 수신 시스템 환경과 관련된 시스템 구성도, 각 송수신 시스템에서 관리하는 데이터와 관련된 테이블 정의서, 코드 정의서 등을 확인 시 사용자 요구 사항에 대한 이해를 향상시킬 수 있다.

사용자 인터뷰를 통해 연계 요구 사항을 분석 시 필요한 사항을 정확하게 도출하기 위해 연계 분석을 위한 체크리스트(Checklist) 등을 사전에 작성하여 활용한다.

1. 송신 시스템과 수신 시스템의 하드웨어 구성, 시스템 소프트웨어 구성, 네트워크 현황을 확인한다.



[그림 1-4] 시스템 구성도

2. 송신 시스템과 수신 시스템에서 연계하고자 하는 데이터와 관련된 테이블 정의서, 코드 정의서를 확인한다.

<표 1-5> 채용 공고(TB_EAS010) 테이블 정의서(송신 시스템)

한글 칼럼명	영문 칼럼명	타입	길이	PK 구분	Not Null 구분	연관 테이블
채용 공고 ID	EMPL_NOTC_ID	String	10	PK	Not Null	
채용 공고명	EMPL_NOTC_CONT	String	500		Not Null	
채용 분야 코드	EMPL_CATE_CODE	String	2			공통 코드
채용 예정 인원	EMPL_EXPT_PERS	Int	3		Not Null	
자격 코드	LICN_CODE	String	4			공통 코드
어학 코드	LANG_CODE	String	3			공통 코드
지역명	ZONE_NM	String	30			
공고 내용	NOTC_CONT	String	LOB		Not Null	
담당자 ID	MNGE_ID	String	7			사용자 정보
공고 등록일	NOTC_REGT_DT	String	8		Not Null	
접수 시작일	RECV_STRT_DT	String	8		Not Null	
접수 종료일	RECV_END_DT	String	8		Not Null	

<표 1-6> 사용자(TB_SYS010) 테이블 정의서(송신 시스템)

한글 칼럼명	영문 칼럼명	타입	길이	PK 구분	Not Null 구분	연관 테이블
사용자ID	USER_ID	String	7	PK	Not Null	
사용자명	USER_NM	String	50		Not Null	
부서 코드	DEPT_CODE	String	5		Not Null	부서 정보
담당 업무	MNGE_BUSI	String	200			

<표 1-7> 코드 정의서(송신 시스템)

코드 분류(그룹)	코드 분류(그룹)명	공통 코드	코드명	사용 여부
LICN_CODE	자격 코드	1001	Database	Y
		1002	OS(Operating System)	Y
		1003	Network	Y
		1004	보안	Y
		2001	청소년 상담사	Y
		2002	청소년 지도사	Y
		2003	사회 복지사	Y
		3001	변리사	Y
		3002	세무사	Y
		3003	손해 평가사	Y
SCHL_CODE	어학 코드	100	영어	Y
		105	독일어	Y
		110	일본어	Y
		115	프랑스어	Y
		120	터키어	Y
		125	러시아어	Y
		130	중국어	Y
EMPL_CATE_CODE	채용 분야 코드	A1	기획	Y
		A2	서무	Y
		B1	전산	Y
		B2	프로그래머	
		C1	회계	Y
		D1	건설	Y
		E1	제조	Y

<표 1-8> 공고 정보(NOTICE_INFO) 테이블 정의서(수신 시스템)

한글 칼럼명	영문 칼럼명	타입	길이	PK 구분	Not Null 구분	연관 테이블
채용 공고일	NOTICE_DATE	String	8	PK	Not Null	
채용 공고 ID	NOTICE_IDEN	String	5	PK	Not Null	
채용 공고명	NOTICE_STATE	String	1,000		Not Null	
채용 구분	EMPLOY_TYPE	String	1		Not Null	
채용 분야	CATE_STATE	String	100			
채용 인원	PERSON	Int	5			
자격 분야	QUALI_FIELD	String	3			공통 코드
언어 코드	LANGUAGE_CODE	String	5			공통 코드
지역 코드	REGION_CODE	String	3			공통 코드
채용 내용	CONTENTS	String	4,000		Not Null	
담당자명	MANAGER	String	50			
채용 접수일	APPLY_DATE	Date	7			
마감일	MAGAM_DATE	Date	7			
경력 사항	CAREER_CNT	String	4,000			
지원 시 참고 사항	REFERENCE	String	4,000			

<표 1-9> 코드 정의서(수신 시스템)

코드 분류(그룹)	공통 코드		코드 분류(그룹)	공통 코드	코드명
자격 분야 (COMMON_CODE_001)	1	컴퓨터	언어 코드 (COMMON_CODE_002)	1	TOEIC(영어)
	2	사회 복지		2	TOEFL(영어)
	3	회계		3	G-TELP(영어)
	8	청소년 및 사회 복지		4	JLPT(일본)
	10	일반 행정		5	JPT(일본)
	25	경영		6	HSK(중국)
	301	세무		7	TCF(프랑스)
	379	경매 및 감정		8	DELF/DALF (프랑스)
	500	부동산		9	TORFL(러시아)
	:	:		10	TOMER(터키)

3. 확인된 각 시스템 구성과 데이터 현황 정보를 활용하여 연계 요구 사항을 도출하기 위해 필요한 질의 사항을 시스템(네트워크, 보안 등) 관점과 응용(데이터, 공통 코드 등) 관점에서 연계 분석 체크리스트('<표 1-3> 연계 분석 체크리스트' 참고)를 작성한다.

2 사용자 인터뷰 및 면담을 수행하고 회의록을 작성한다.

<표 1-10> 사용자 인터뷰 내용

사용자 인터뷰 및 면담 내용 ('00.00.00.)
- 정보를 연계하고자 하는 각 기관 시스템 운영자, 응용 애플리케이션 분석 및 설계가, 서비스 운영자의 인터뷰 및 면담을 진행함 - (서비스 운영자) 채용 정보의 실시간 연계를 원칙으로 하되, 실시간 연계가 어려울 경우 실시간에 근접한 주기로 연계가 필요함 - (서비스 운영자) 연계된 채용 정보는 최대한 다양하게 검색될 수 있어야 함 - (응용 애플리케이션 분석 및 설계가) 연계된 정보가 다양하게 검색하려면 최대한 정보를 코드화하여 연계하여야 하며, 각 기관에서 서비스하고 있는 코드 범위의 확인이 필요함 - (응용 애플리케이션 분석 및 설계가) 각 기관에서 사용하는 코드의 매핑 가능 여부를 확인하고 매핑 수준을 정의해야 함 - (서비스 운영자) 연계되는 채용 정보에서 사용하는 코드 정보의 매핑이 불가할 경우에는 매핑 기준을 새로 정의해야 함 - (시스템 운영자) 정보 연계 시에는 각 기관의 보안 정책을 준수하여야 하며, 연계 정보의 중요성에 따라 채널(Channel) 보안이나 데이터 보안까지 보안 수준을 다양하게 적용할 수 있어야 함 - (시스템 운영자) 연계 정보의 성공 및 실패 여부, 실패 시 원인을 확인할 수 있도록 모니터링 기능이 구현되어야 함

1. 사전에 준비한 시스템 구성도, 테이블 정의서, 코드 정의서, 연계 분석을 위한 체크리스트(Checklist) 등을 배포 및 공유하고, 사용자 인터뷰를 진행한다.

2. 사용자 인터뷰 완료 후 인터뷰 시 진행 내용, 확인 사항 등을 회의록에 정리한다.

3. 정리된 회의록 내용은 사용자 인터뷰 시의 참석자에게 확인을 받는다.

3 연계 요구 사항 분석서를 작성한다.

회의록 내용 중 연계 시스템 구성 및 데이터 식별에 해당하는 내용에 요구사항 ID를 부여하여 형식과 항목에 맞게 연계 요구 사항 분석서('<표1-4> 연계 요구 사항 분석서' 참고)를 작성한다.

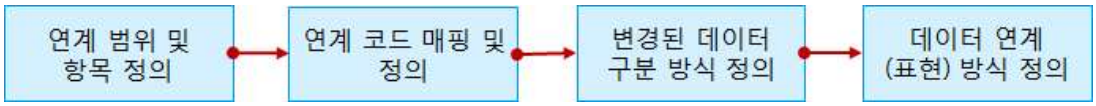
수행 tip

- 요구 사항 분석서의 요구 사항 ID는 일반적으로 인터페이스 1개 단위로 분류하여 작성한다.

1-2. 연계 데이터 식별 및 표준화

학습 목표	• 개발하고자 하는 응용 소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다. • 개발하고자 하는 응용 소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.

필요 지식 /



[그림 1-5] 연계 데이터 식별 및 표준화 절차

연계 데이터 구성은 연계 데이터를 식별하고, 식별된 연계 데이터를 표준화하는 과정이다. 이 수행 결과로 연계(인터페이스) 정의서를 작성한다.

코드나 텍스트, 아이디(ID: Identification) 등으로 구성된 정보 중 연계 가능 범위와 항목을 식별하고, 식별된 항목 중 코드화된 정보의 포함 여부를 확인한다. 코드화된 정보는 송신 시스템과 수신 시스템에서 사용 가능 하도록 코드 매핑 정보를 제공하거나 송수신 시스템 시스템 간 상이한 코드 정보를 표준화한다.

정의와 표준화가 완료된 연계 정보는 각 시스템에 반영하기 위한 데이터 발생 구분 정보를 추가하고, 데이터베이스(DB: Database) 또는 파일 등으로 데이터 연계 방식을 정의한다.

1 연계 범위 및 항목 정의

시스템 간에 연계하려는 정보를 상세화하며 범위와 항목을 정의한다. 상세화하는 방법은 연계 필요 정보를 정보 그룹에서 그룹을 구성하는 단위 항목으로 확인한다.

예) <u>채용 정보</u> 연계 필요 ⇒ 송신 시스템과 수신 시스템에서 채용 정보와 관련된 <u>테이블 또는 파일</u> 확인(송신 시스템의 채용 정보, 수신 시스템의 공고 정보) ⇒ 테이블 또는 파일의 각 <u>구성 항목</u> 확인 [송신 시스템의 '<표1-5> 테이블 정의서', 수신 시스템의 '<표1-8> 테이블 정의서' 참고]

송신 시스템과 수신 시스템에서 연계하고자 하는 각 항목의 데이터 타입 및 길이, 코드화 여부 등을 확인한다.

송신 시스템과 수신 시스템의 연계 항목이 상이할 경우, 일반적으로 연계 정보가 활용되는

수신 시스템 기준으로 적용 및 변환한다.

<표 1-11> 연계 범위 및 항목 정의 시 확인 사항

연계 정보		확인 사항	예시 (표 1-5, 1-8 참고)
송신	수신		
String	Number	한 가지의 데이터 타입으로 정의 및 적용(일반적으로 수신 시스템 기준으로 적용)	- 송신 시스템 접수 시작일 String vs 수신 시스템 채용 접수일 Date ⇒ String 또는 Date 둘 중 하나로 동일하게 적용함
Number	String		
String	Date		
Date	String		
Not Null	Null	송신 시스템에 입력된 값 그대로 연계	- 송신 시스템 채용 예정 인원 vs 수신 시스템 채용 인원
정의되지 않은 항목 또는 Null	Not Null	**연계 불가,** Unique하게 임의로 부여한 값으로 처리 시 수신 시스템에서 문제 발생 여부 확인하여 적용	- 수신 시스템의 채용 구분(1: 경력, 2: 신입, 3: 무관) ⇒ '3' 무관으로 처리 가능 여부 확인하여 적용
송신 시스템 데이터 크기 > 수신 시스템 데이터 크기		**연계 불가**하나, 송신 시스템 데이터 크기를 수신 시스템 데이터 크기로 줄였을 경우 잘려지는 데이터의 필요성 여부 확인	- 송신 시스템 공고 내용 LOB(Large Object) vs 수신 시스템 채용 내용 String 4,000 ⇒ 수신 시스템의 String 4,000으로 적용 시 문제점 확인
송신 시스템 데이터 크기 < 수신 시스템 데이터 크기		수신 시스템의 데이터 크기로 변환하여 연계	- 송신 시스템 채용 공고명 String 500 vs 수신 시스템 채용 공고명 String 1,000 ⇒ 수신 시스템의 String 1,000으로 적용
송신 시스템 코드 상세화 > 수신 시스템 코드 상세화		수신 시스템 기준으로 매핑 또는 표준화	- 송신 시스템 자격 코드 vs 수신 시스템 자격 분야 ⇒ 송신 시스템 자격 코드를 수신 시스템 자격 분야 코드로 매핑 또는 표준화
송신 시스템 코드 상세화 < 수신 시스템 코드 상세화		**연계 불가**, 연계 가능 항목에서 제외함	- 송신 시스템 어학 코드 vs 수신 시스템의 언어 코드
텍스트	코드	**연계 불가**, 연계 가능 항목에서 제외함	- 송신 시스템 지역명 vs 수신 시스템 지역 코드
코드	텍스트	송신 시스템의 코드를 해당 코드명으로 전환 처리	- 송신 시스템의 담당자 ID, 수신 시스템의 담당자명 ⇒ 송신 시스템에서 사용자 테이블을 참고하여 사용자 ID에 해당하는 사용자명으로 매핑 또는 전환

2 연계 코드 변환 및 매핑(Mapping)

연계 대상 범위 및 항목을 식별한 이후에는 연계 정보 중 코드로 관리되는 항목을 변환해야 한다. 코드로 관리되는 정보는 정확한 정보로의 전환 및 검색 조건으로 활용하는 이점이 있다.

송수신되는 연계 정보에 포함된 코드를 변환하는 방법은 송신 시스템 코드를 수신 시스템 코드로 매핑 해주는 방법과 송수신 시스템에서 사용되는 코드를 통합하여 표준화한 후 전환하는 방법이 있다.

1. 송신 시스템 코드를 수신 시스템 코드로 매핑(Mapping)하는 방법

<표 1-12> 송수신 시스템 간의 코드 변환을 위한 코드 매핑 테이블 – <표 1-7>, <표 1-9> 코드 정의서 참고

송신 시스템 코드				수신 시스템 코드			
코드 그룹	코드 그룹명	코드	코드명	코드 그룹	코드 그룹명	코드	코드명
LICN_ CODE	자격 코드	1001	Database	COMMON _CODE_ 001	자격 분야	1	컴퓨터
		1002	OS(Operating System)			1	컴퓨터
		1003	Network			1	컴퓨터
		1004	보안			1	컴퓨터
		2001	청소년 상담사			8	청소년 및 사회 복지
		2002	청소년 지도사			8	청소년 및 사회 복지
		2003	사회 복지사			8	청소년 및 사회 복지
		3001	변리사			25	경영
		3002	세무사			301	세무
		3003	손해 평가사			3	회계

2. 송수신 시스템의 코드를 통합하여 표준화한 후 매핑(Mapping)하는 방법

<표 1-13> 송수신 시스템의 자격 코드를 통합하여 표준화한 테이블 – <표 1-7>, <표 1-9> 코드 정의서 참고

표준화 코드				송신 시스템 코드		수신 시스템 코드	
코드 그룹	코드 그룹명	코드	코드명	코드	코드명	코드	코드명
LICNS_ CODE	자격 코드	100	컴퓨터	-	-	1	컴퓨터
		110	컴퓨터-DB	1001	Database	-	-
		120	컴퓨터-OS	1002	OS	-	-
		130	컴퓨터-NW	1003	Network	-	-
		140	컴퓨터-보안	1004	보안	-	-
		200	청소년 및 사회 복지	-	-	8	청소년 및 사회 복지
		210	청소년 상담사	2001	청소년 상담사	-	-
		220	청소년 지도사	2002	청소년 지도사	-	-
		230	사회 복지사	2003	사회 복지사	-	-
		300	회계 및 세무	-	-	3	회계
				-	-	301	세무
		310	세무사	3002	세무사	-	-
		320	손해 평가사	3003	손해 평가사	-	-
		400	일반 행정 및 경영	-	-	10	일반 행정
				-	-	25	경영

3 연계 데이터 식별자와 변경 구분 추가

1. 식별자(PK: Primary Key)

송신된 정보가 수신 시스템의 어떤 데이터에 해당되는지 추출하기 위해서는 수신 시스템의 연계 정보에 송신 시스템의 식별키 항목을 추가하여 관리한다.

송신 시스템 채용 공고 테이블의 PK				수신 시스템 공고 정보 테이블의 PK			
한글 칼럼명	영문 칼럼명	타입	길이	한글 칼럼명	영문 칼럼명	타입	길이
채용 공고 ID	EMPL_NOTC_ID	String	10	채용 공고일	NOTICE_DATE	String	8
				채용 공고 ID	NOTICE_IDEN	String	5

⇩

수신 시스템의 공고 정보 테이블				
한글 컬럼명	영문 칼럼명	타입	길이	PK 구분
채용 공고일	NOTICE_DATE	String	8	PK
채용 공고 ID	NOTICE_IDEN	String	5	PK
:	:	:	:	:
송신 채용 공고 ID	EMPI_NOTC_ID	String	10	

수정, 삭제를 위해 송신 테이블의 PK 추가 ⇨

[그림 1-6] 송수신 시스템 테이블의 주 식별자(PK: Primary Key) – <표 1-7>, <표 1-9> 코드 정의서 참고

2. 변경 구분

송신 데이터를 수신 시스템에 반영하기 위해서 송신 정보를 수신 시스템의 테이블에 추가, 수정, 삭제할 데이터인지 식별해 주는 구분 정보를 추가한다.

<표 1-14> 송신 데이터의 변경 구분

한글 칼럼명	영문 칼럼명	타입	길이	설명
변경 일시	CHG_DTTM	Date	7	송신 시스템의 채용 정보가 추가, 수정, 삭제된 일자와 시간
변경 구분	CHG_TYPE	String	1	송신 시스템의 채용정보가 I(Insert): 추가, U(Update): 수정, D(Delete): 삭제 되었는지를 표시

3. 관리를 위한 정보

연계되는 정보의 송수신 여부, 송수신 일시, 오류 코드 등을 확인하고 모니터링하기 위해 인터페이스 테이블 또는 파일에 관리 정보를 추가한다.

<표 1-15> 연계 정보 모니터링을 위한 관리 정보

한글 칼럼명	영문 칼럼명	타입	길이	설명
송신 일시	SEND_DTTM	Date	7	- 송신 시스템에서 연계 데이터를 전송한 일시
수신 일시	RECV_DTTM	Date	7	- 수신 시스템에서 연계 데이터를 수신한 일시
처리 일시	PROC_DTTM	Date	7	- 수신 후 수신 시스템에 연계 데이터를 처리한 일시
처리 상태	PROC_STAT	String	1	- 송신 완료, 수신 완료, 처리 결과 성공, 오류 발생 등 처리 상태를 표시(R: 수신, S: 성공, E: 오류 발생)
오류 코드	ERR_CODE	String	5	- 처리 결과 오류가 발생한 경우, 오류 원인에 해당하는 코드 - 오류 코드 및 오류 내용은 사전에 정의해야 함
오류 내용	ERR_CONT	String	4,000	- 오류 발생 내용에 해당하는 시스템 Log

④ 연계 데이터 표현 방법

연계 대상 범위 및 항목, 코드 매핑 방식 등을 정의한 후 연계 데이터를 테이블이나 파일 등의 형식으로 구성한다.

구성된 연계 데이터는 응용 애플리케이션에서 연계 데이터를 생성하는 시점, 연계 주기, 적용되는 연계 솔루션의 지원 기능 등에 다르게 표현될 수 있다.

연계 데이터 표현의 기본적인 분류는 데이터베이스의 테이블과 파일의 형식이며, 파일의 경우에는 파일 형식에 따라 태그(Tag), 항목 분리자(딜리미터, Delimiter) 사용 등에 의해 상세화된다.

<표 1-16> 연계 데이터 표현 방식 분류

<table>
<tr><th>유형</th><th>분리자</th><th>단위</th><th>표현 예시</th></tr>
<tr><td>테이블</td><td>칼럼</td><td>Row</td><td>< 채용 정보 인터페이스 테이블 (EMPL_INFO_IF) >
<table><tr><th>채용 공고 ID</th><th>채용 공고명</th><th>~</th><th>처리 상태</th><th>송신 일시</th><th>수신 일시</th></tr><tr><td>:</td><td>:</td><td>~</td><td>:</td><td>:</td><td>:</td></tr></table></td></tr>
<tr><td rowspan="2">파일</td><td>태그
(Tag)</td><td>태그
(Tag)
그룹</td><td>- xml 파일 형식, 데이터 참조 시 parsing 필요
- xml 사례
<xml>
<empl_notc> <!-- 채용 정보 1 시작 -->
<empl_notc_id>20151009000001</empl_notc_id> <!-- 공고 ID -->
<empl_notc_nm> <!-- 공고명 -->
전산직 OS 운영 경험자 및 DBA 경력직 채용 공고
</emp_notc_nm>
<receive_start_dt>20151010</receive_start_dt><!-- 접수 시작일 -->
<receive_end_dt>20151030</receive_end_dt> <!-- 접수 종료일 -->
:
</empl_notc> <!-- 채용 정보 1 끝 -->
<empl_notc> <!-- 채용 정보 2 시작 -->
:
</empl_notc> <!-- 채용 정보 2 끝 -->
:
</xml></td></tr>
<tr><td>분리자
(Delimiter)</td><td>개행
문자
구분</td><td>- 콤마(Comma, ,), 콜론(Colon, :), 세미콜론(Semi-colon, ;) 등 정의한 분리자를 사용한 CSV, Text 파일 형식
- 콤마(Comma, ,) 분리자를 사용한 CSV(Comma Seperated Values) 파일 사례
채용 공고 ID, 채용 공고명, 접수 시작일, 접수 종료일,…, 처리 상태
2015100900001,전산직 OS 운영 경험자 및 DBA 경력직 채용 공고,20151010,20151030,…,송신</td></tr>
</table>

5 연계(인터페이스) 정의서 및 명세서

연계 항목, 연계 데이터 타입, 길이 등을 구성하고 형식을 정의하는 과정의 결과물로 연계(인터페이스) 정의서를 작성한다.

연계(인터페이스) 정의서에는 송신 시스템과 수신 시스템 간의 인터페이스 현황을 작성한다.

연계(인터페이스) 명세서는 연계(인터페이스) 정의서에 작성한 인터페이스 ID 별로 송수신하는 데이터 타입, 길이 등 인터페이스 항목을 상세하게 작성한다.

<표 1-17> 연계(인터페이스) 정의서 작성 양식

I/F 번호	송신		수신		인터페이스 방식	인터페이스 주기
	시스템명	인터페이스 ID/명	시스템명	인터페이스 ID/명		

<표 1-18> 연계(인터페이스) 명세서 작성 양식

I/F 번호						I/F명					
송신	I/F ID					수신	I/F ID				
	I/F명						I/F명				
주기 및 방식						DB 및 파일 형식		○ DB ○ FILE ( )			
송신						수신					
한글명	영문명	Type	길이	PK	Code 여부	한글명	영문명	Type	길이	PK	Code 여부
처리 내용											

수행 내용 / 연계 데이터 식별 및 표준화하기

재료 · 자료

- 연계 요구 사항 분석서
- 테이블 정의서(송신 시스템, 수신 시스템)
- 코드 정의서(송신 시스템, 수신 시스템)
- 연계(인터페이스) 정의서
- 연계(인터페이스) 명세서

기기(장비 · 공구)

- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

1 연계 요구 사항 분석서 및 테이블 정의서, 코드 정의서 등 필요 자료를 확인한다.

연계 요구 사항 분석서에서 연계가 필요한 정보 그룹을 식별하고, 연계 대상 정보 그룹에 해당하는 테이블 정의서 또는 파일 레이아웃(구조, Layout)을 확인하여 연계 범위와 항목을 정의한다.

1. 연계 요구 사항 분석서(학습 1-1. 연계 요구 사항 분석하기 수행 내용 결과물)에서 연계가 필요한 정보 그룹을 식별한다.
2. 송신 시스템과 수신 시스템에서 식별된 정보 그룹에 해당하는 테이블 정의서 또는 파일 레이아웃(구조, Layout), 코드 정의서 등을 준비한다.

- '<표 1-5> 테이블 정의서(송신 시스템)', '<표 1-6> 코드 정의서(송신 시스템)', '<표 1-8> 테이블 정의서(수신 시스템)', '<표 1-9> 코드 정의서(수신 시스템)' 참고

2 '<표 1-17> 연계(인터페이스) 정의서 작성 양식' 참고하여 연계(인터페이스) 정의서를 작성한다.

1. 송수신 시스템의 테이블 정의서(또는 파일 Layout)에서 연계하고자 하는 테이블 또는 파일 단위별 일련번호를 부여하고, 연계(인터페이스) 정의서를 작성한다.

- 인터페이스 ID: ① 송신 시스템 또는 수신 시스템의 연계 대상 테이블(파일) 영문명, 운영하는 시스템의 테이블(파일)명이거나, 연계를 위해 별도로 생성한 인터페이스(I/F: Interface) 테이블(파일)명
  ② 연계를 위해 인터페이스 테이블(파일)에 연계 데이터를 생성하고, 송수신하는 응용 프로그램

2. 연계(인터페이스) 단위별 주기, 인터페이스 방식 등을 확인하여 추가한다.

3 '<표 1-18> 연계(인터페이스) 명세서 작성 양식' 참고하여 연계(인터페이스) 명세서를 작성한다.

연계 요구 사항 분석 시 작성한 연계 요구 사항 분석서와 송신 시스템 및 수신 시스템의 테이블 정의서, 코드 정의서를 활용하여 연계 데이터를 식별하고 표준화한다.

정의된 연계 항목의 데이터 타입, 길이, 공통 코드 여부 등의 속성을 확인하면서 연계(인터페이스) 정의서 및 명세서를 작성한다.

연계(인터페이스) 정의서의 인터페이스별(인터페이스 번호별) 연계(인터페이스) 명세서를 작성한다.

1. 송신 시스템과 수신 시스템 간의 연계 항목을 매핑(Mapping) 및 정의한다.
   연계 항목 정의 시 데이터 타입, 길이, 필수 입력 여부, 코드화 여부 및 범위 등을 고려하여 연계 가능 여부를 결정('<표 1-11> 연계 범위 및 항목 정의 시 확인 사항 참고)하고 연계 가능한 항목을 식별하여 작성한다.
   【 확인 사항 : 데이터 타입, 데이터 길이, 필수 입력 여부(Not Null), 코드 여부 및 범위 】

2. 연계 항목 중 코드는 송신 시스템과 수신 시스템에서 사용되는 코드 정의서('<표 1-6> 코드 정의서(송신 시스템)', '<표 1-9> 코드 정의서(수신 시스템)' 참고) 등을 확인하여 연계 여부를 결정하고, 연계 가능 코드에 대해서 코드 매핑 내역서('<표 1-12> 송수신 시스템 간의 코드 변환을 위한 코드 매핑 테이블' 및 '<표 1-13> 송수신 시스템의 자격 코드를 통합하여 표준화한 테이블' 참고)를 작성한다.

3. 송신 시스템과 수신 시스템의 데이터를 식별할 수 있는 주 식별자 및 데이터 변경 구분을 확인한다.

   송신 시스템과 수신 시스템의 각 테이블(파일 Layout) 정의서를 참고하여, 송신 인터페이스 및 수신 인터페이스의 주 식별자(PK: Primary Key)를 구분한다.

   인터페이스별(인터페이스 번호별) 송신 데이터를 구분할 수 있는 식별자와 데이터 변경 구분 값을 정의한다.
   【 송신 데이터 구분 식별자: 변경 일시 또는 변경 일련번호 등 】
   【 변경 구분: 추가(I), 수정(U), 삭제(D) 】

4. 송수신 데이터의 모니터링 및 관리를 위해 인터페이스별(인터페이스 번호별) 관리 정보('<표 1-15> 연계 정보를 모니터링을 위한 관리 정보' 참고)를 정의한다.
【 관리 정보: 송신 일시, 수신 일시, 처리 일시, 처리 상태, 오류 코드, 오류 내용 등 】

수행 tip

- 연계(인터페이스) 정의서 및 명세서 작성 시 연계 데이터가 생성, 수정, 삭제되는 시점과 주기 등 데이터 특성을 응용 애플리케이션의 기능과 관련하여 확인한다.
- 응용 애플리케이션의 기능이란 메뉴 또는 화면에서 사용자와 인터랙션(Interaction)하는 단위 또는 데이터 처리 단위를 의미한다.

학습 1 교수 · 학습 방법

교수 방법

- 통합 구현은 실무에서도 적용하기 어려운 부분이므로 학생들이 이해하기 용이하도록 연계 시스템 아키텍처, 절차, 구성 요소, 구성 요소별 기능 및 역할 등 전반적인 사항과 개념을 설명하고, 상세 내용을 사례를 들어 설명해야 한다.
- 요구 사항 분석을 위해 필요한 회의록, 시스템 구성도, 연계 분석을 위한 체크리스트, 테이블(파일 Layout) 정의서, 코드 정의서 등 관련 자료의 실제 사례를 제시하며 각 자료의 필요성과 활용 방법, 작성방법 등을 설명한다.
- 연계 요구 사항 분석 과정 및 산출물 작성 방법을 사례를 제시하고 설명한다.
- 연계 데이터를 식별하고 정의하는 연계 범위 및 항목 정의, 연계 코드 변환 및 매핑 등 상세 과정을 작성하는 산출물과 연관지어 설명한다.
- 연계 데이터 식별 및 표준화 과정을 사례를 들어 설명하고 학생들이 실습해야 할 수행 내용을 자세히 설명해야 한다.

학습 방법

- 수행 내용에 따라 실습하는 과정에서 연계 요구 사항 분석, 연계 데이터 식별 및 표준화 작업, 연계 데이터 식별 및 표준화 과정별 작업의 필요성, 작업 결과 작성되는 산출물에 대해 학습한다.
- 연계 산출물 간 연관성, 선후 관계, 작성 목적과 활용에 대한 이해를 기반으로 시스템 및 운영 환경 등 주어진 조건에 적합한 산출물 선정과 각 산출물에서 작성해야 할 필수 항목들을 조정하여 산출물을 작성하는 실습을 한다.
- 연계 요구 사항 분석서의 작성 시점, 작성 목적, 작성 항목, 활용 방법 등을 이해하고 산출물을 작성하는 실습을 한다.
- 연계(인터페이스) 정의서의 작성 시점, 작성 목적, 작성 항목, 활용 방법 등을 이해하고 산출물을 작성하는 실습을 한다.
- 연계(인터페이스) 명세서의 작성 시점, 작성 목적, 작성 항목, 활용 방법 등을 이해하고 산출물을 작성하는 실습을 한다.

학습 1 평 가

평가 준거

- 평가자는 학습자가 수행 준거 및 평가 시 고려 사항에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.

- 평가자는 다음 사항을 평가한다.

학습 내용	평가 항목	성취수준		
		상	중	하
연계 요구 사항 분석	- 개발하고자 하는 응용 소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구 사항을 분석할 수 있다.			
연계 데이터 식별 및 표준화	- 개발하고자 하는 응용 소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다.			
	- 개발하고자 하는 응용 소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.			

평가 방법

- 평가자 체크리스트

학습 내용	평가 항목	성취수준		
		상	중	하
연계 요구 사항 분석	- 통합 구현의 개념과 연계 아키텍처에 대한 이해			
	- 연계 요구 사항 분석과정 이해 및 응용			
	- 연계 요구 사항 분석을 위한 산출물 작성 및 활용 방법 이해			
연계 데이터 식별 및 표준화	- 연계 데이터 정의 과정과 방법의 이해 및 응용			
	- 연계 코드 매핑(Mapping), 변환에 대한 이해 및 응용			
	- 연계(인터페이스) 정의서, 연계(인터페이스) 명세서의 작성과 활용 방법 이해 및 응용			

● 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
연계 요구 사항 분석	- 통합 구현의 개념과 연계 아키텍처 이해 및 응용			
	- 연계 요구 사항 분석 과정 이해 및 응용			
	- 연계 요구 사항 분석을 위한 산출물 작성 및 활용 방법 이해			
연계 데이터 식별 및 표준화	- 연계 데이터 정의 과정과 방법의 이해 및 응용			
	- 연계 코드 매핑(Mapping), 변환에 대한 이해 및 응용			
	- 연계(인터페이스) 정의서와 명세서의 차이 이해			
	- 연계(인터페이스) 정의서, 연계(인터페이스) 명세서의 작성과 활용 방법 이해 및 응용			
	- 연계(인터페이스) 정의서 및 연계(인터페이스) 명세서 작성의 완전성			

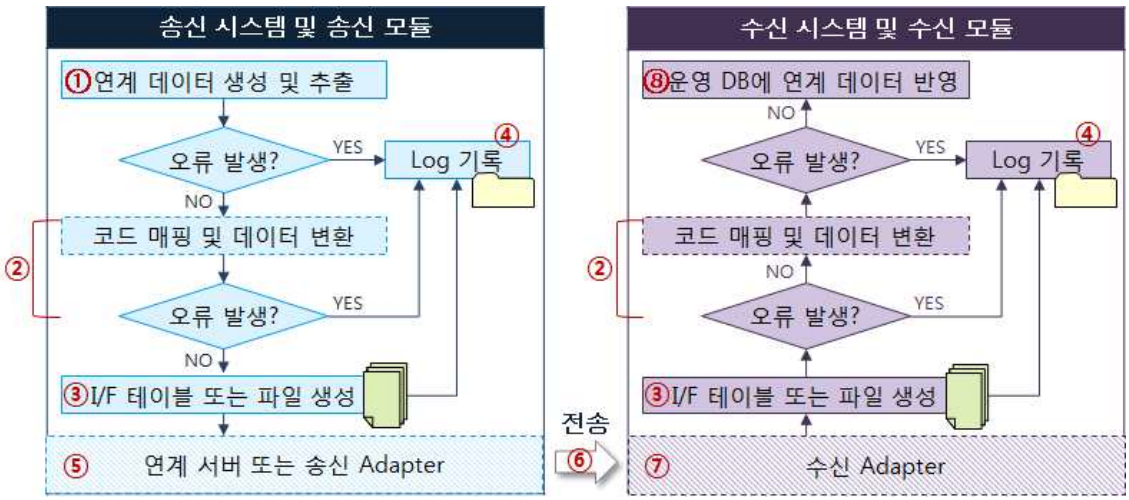
피 드 백

1. 체크리스트(Checklist)를 통한 관찰
   - 연계 요구 사항 분석서 작성 실습 과정에서 통합 구현의 개념과 연계 아키텍처 이해 및 응용, 연계 요구 사항 분석 과정에 대한 이해 및 응용, 연계 요구 사항 분석서 작성 및 활용 방법에 대한 이해를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 연계(인터페이스) 정의서 및 명세서 작성 실습 과정에서 연계 데이터 정의 과정과 방법의 이해 및 응용, 연계 코드 매핑(Mapping)과 변환에 대한 이해 및 응용, 연계(인터페이스) 정의서와 명세서의 차이에 대한 이해, 연계(인터페이스) 정의서와 명세서 작성 및 활용 방법에 대한 이해 및 응용 정도를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
2. 포트폴리오(Portfolio)
   - 제출한 연계 요구 사항 분석서의 충분성, 완전성, 적합성, 일관성, 구체성을 평가한 후에 필수 작성 항목 포함 여부 점검, 분석 내용의 구체성 정도, 연계 아키텍처 및 데이터 구성을 위한 내용의 적합성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 제출한 연계(인터페이스) 정의서 및 명세서의 충분성, 완전성, 적합성, 일관성, 구체성을 평가한 후에 송수신 시스템 간 연계 데이터의 적합성, 목적 달성을 위한 연계 항목의 충분성, 연계 데이터 표현 방식의 구체성 및 일관성 등 주요 사항에 대하여 표시하여 돌려준다.

2-1. 연계 메커니즘 정의

학습 목표	• 개발하고자 하는 응용 소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적인 데이터 송수신 방법을 정의할 수 있다. • 개발하고자 하는 응용 소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구 사항을 고려하여 연계 주기를 정의할 수 있다.

필요 지식 /



[그림 2-1] 연계 메커니즘

【 연계 메커니즘 구성 요소별 주요 기능 및 역할 】

연계 메커니즘은 데이터를 생성하여 전송하는 송신 체계와 수신하여 운영 DB에 반영하는 수신 체계로 구성된다.

송신 시스템은 운영 데이터베이스(DB: Database), 애플리케이션(AP: Application)으로 부터 연계 데이터를 인터페이스 테이블(I/F Table: Interface Table) 또는 파일(File - xml, csv, text)로 생성하여 송신한다.

수신 시스템은 수신한 인터페이스 테이블 또는 파일(File)의 데이터를 변환하여 운영 데이터베이스(DB: Database)에 반영한다.

송수신 시스템 사이에 데이터 송수신과 송수신 현황 모니터링 역할을 하는 중계(연계) 서버를 배치할 수도 있다.

중계(연계) 서버 배치는 성능과 보안의 품질 특성 중 보안의 더 중요하거나 송신 시스템이 위치한 네트워크와 수신 시스템이 위치한 네트워크가 상이한 경우 주로 설계하는 아키텍처 방식이다.

송신 시스템과 중계(연계) 서버, 수신 시스템의 각 역할이 중복되지 않도록 조정하여 아키텍처를 설계한 후 인터페이스 테스트와 통합 테스트를 통해서 기능을 검증한다.

【 연계 방식 】

연계 방식은 중간 매개체 없이 송신 시스템과 수신 시스템이 직접 연계되는 방식과 연계 솔루션과 같이 중간 매개체를 활용하여 연계하는 간접 연계 방식으로 분류할 수 있다.

일반적인 간접 연계 방식의 송신 시스템은 운영 데이터베이스(DB: Database)에서 연계 데이터를 생성 및 변환과 송신 로그(Log)를 모니터링하는 구현 대상 모듈과 솔루션(Solution)에서 제공하는 송수신 엔진(Engine), 어댑터(Adapter)로 구성된다.

간접 연계 방식의 수신 시스템은 데이터를 수신받아 인터페이스 테이블 또는 파일로 생성하는 송수신 엔진(Engine), 어댑터(Adapter)와 수신된 데이터를 변환하여 운영 DB에 반영하고, 데이터 수신 및 운영 DB 반영을 모니터링 하는 구현 모듈로 구성된다.

<표 2-1> 연계 방식별 장단점

방식	장점	단점
직접 연계	- 연계 및 통합 구현이 단순하며 용이함 - 개발 소요 비용 및 기간이 짧음 - 중간 매개체가 없으므로 데이터 연계 처리 성능이 대체로 좋음	- 시스템 간의 결합도가 높아서, 시스템 변경에 민감함(시스템 변경으로 인해 장애 발생 가능) - 보안을 위한 암·복호화 처리, 비즈니스 로직(Business Logic, 업무 처리 내용) 적용 등이 불가 - 연계 및 통합 가능한 시스템 환경이 제한적
간접 연계	- 서로 상이한 네트워크, 프로토콜(Protocol) 등 다양한 환경을 연계 및 통합 가능 - 시스템 간 인터페이스 변경 시에도 장애나 오류 없이 서비스 가능 - 보안이나 비즈니스(Business, 업무) 처리를 위한 로직(Logic)을 자유롭게 반영 가능	- 연계 아키텍처 및 메커니즘이 복잡하고, 중간 매개체로 인해 성능 저하 요소 존재 - 개발 및 적용을 위한 테스트 기간이 상대적 장기간 소요

<표 2-2> 연계 방식 분류

방식		연계 방식별 상세 설명	예시	송수신 서버(엔진)
직접 연계	DB Link	- 데이터베이스에서 제공하는 객체(오브젝트, Object) 이용 - 수신 시스템 디비(DB)에 송신 시스템에서 접근 가능한 디비링크(DBLINK: Database Link) 객체를 생성한 후 송신 시스템에서 디비링크명(DBLINK명)으로 직접 참조하여 연계	테이블 명 @DBLink명	X
	DB Connection Pool (WAS)	- 수신 시스템 와스(WAS)에서 송신 시스템 DB로 연결되는 커넥션 풀(Connection Pool) 생성 - 프로그램 소스에서 WAS에 설정된 Connection Pool명을 참고하여 구현	송신 시스템 데이터 소스(DS: Data Source) (WAS에서 설정한 Connection Pool명)	X
	JDBC	- 수신 시스템의 배치(Batch) 또는 온라인(On-Line) 프로그램에서 제이 디 비 씨(JDBC) 드라이버를 이용하여 송신 시스템의 DB와 연결 생성	jdbc:DBMS 제품명: DBMS 설치 서버 IP: Port:DB instance명	X
	화면 링크 (Link)	- 웹 애플리케이션 화면에서 하이퍼 링크(Hyper Link) 이용	<a href= “url” > LINK 화면명</a>	X
	API 또는 Open API	- 송신 시스템의 DB와 연결하여 데이터 제공하는 인터페이스(Interface) 프로그램	제공하는 컴포넌트명. 데이터 처리 메서드명 (파라미터 1~N);	△
간접 연계	연계 솔루션 (EAI)	- 실제 송수신 처리와 진행 현황을 모니터링 및 통제하는 이 에이 아이(EAI) 서버, 송수신 시스템에 설치되는 Adapter(Client)를 이용	메타빌드, 비즈마스터(Biz Master) 등	○
	Web Service/ ESB	- 웹 서비스가 설명된 더블유 에스 디 엘(WSDL)과 솝(SOAP) 프로토콜을 이용한 시스템 간 연계 - 미들웨어인 이 에스 비(ESB)에서 서비스(컴포넌트) 간 연동을 위한 변환 처리로 다중 플랫폼(Platform) 지원	WSDL, 유 디 디 아이(UDDI), 솝(SOAP), 서비스(Service), ESB(Enterprise Service Bus)	○
	Socket	- 소켓(Socket)을 생성하여 포트를 할당하고, 클라이언트(Client)의 요청을 연결하여 통신 - 네트워크 프로그램의 기반 기술	TcpServer.listen(); TcpServer.accept(); TcpClient.connect(); TcpClient.receive();	○

‘<그림 2-1> 연계 메커니즘’ 에서 송신 시스템의 ① ~ ④, 수신 시스템의 ②, ④, ⑧은 응용 시스템별로 별도로 구현한다. 연계 메커니즘의 ⑤, ⑥, ⑦은 연계 서버(엔진) 및 솔루션에서 지원하는 기능으로 별도로 구현할 필요가 없이 연계 서버(엔진)을 설치하고, 환경을 설정하여 구축한다.

<표 2-3> '<그림 2-1> 연계 메커니즘'의 요소별 관리 및 구현 주체

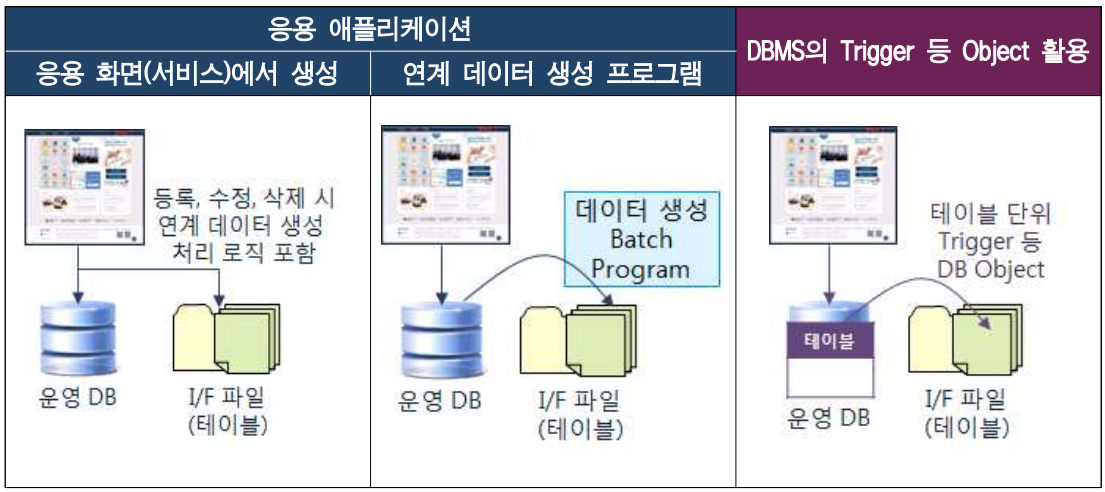
연계 메커니즘		관리 및 구현 주체	산출물
송신	① 연계 데이터 생성 및 추출	응용 시스템 별도 구현	연계 프로그램, Log 테이블(파일)
	② 코드 매핑 및 데이터 변환	응용 시스템 별도 구현	연계 프로그램, Log 테이블(파일) 코드 매핑 테이블
	③ I/F 테이블 또는 파일 생성	응용 시스템 별도 구현	연계 프로그램, Log 테이블(파일) I/F 테이블(파일)
	④ 로그(Log) 기록	응용 시스템 별도 구현, 연계 솔루션	연계 프로그램, Log 테이블(파일) 연계 서버(엔진) 및 Adapter
	⑤ 연계 서버 또는 송신 Adapter	연계 솔루션	연계 서버(엔진) 및 Adapter
⑥ 전송		연계 솔루션	연계 서버(엔진) 및 Adapter
수신	⑦ 연계 서버 또는 수신 Adapter	연계 솔루션	연계 서버(엔진) 및 Adapter
	③ I/F 테이블 또는 파일 생성	연계 솔루션	연계 서버(엔진) 및 Adapter
	② 코드 매핑 및 데이터 변환	응용 시스템 별도 구현	연계 프로그램, Log 테이블(파일) 코드 매핑 테이블
	④ 로그(Log) 기록	응용 시스템 별도 구현, 연계 솔루션	연계 프로그램, Log 테이블(파일) 연계 서버(엔진) 및 Adapter
	⑧ 운영 DB에 연계 데이터 반영	응용 시스템 별도 구현	연계 프로그램, Log 테이블(파일)

간접 연계 방식의 일반적인 연계 메커니즘에서 각 구성 요소와 구성 요소별 역할 및 기능은 아래 설명과 같다.

1 연계 데이터 생성 및 추출

운영 DB에서 연계 데이터를 추출하고 생성하는 부분으로 적용하는 연계 솔루션과는 관계없이 응용 시스템별로 별도로 구현한다. 응용 시스템에서 구현하는 방식으로는 응용 애플리케이션(프로그램)에서 생성하는 방법과 DB의 오브젝트를 이용하는 방법이 있다.

데이터를 추출하여 생성하는 과정에서 오류가 발생할 경우, 로그(Log) 테이블 또는 파일에 발생한 오류 내역을 발생 시점, 오류 코드, 오류 내용 등 상세하게 기록한다.



[그림 2-2] 연계 데이터 생성 및 추출 방법

② 코드 매핑(Mapping) 및 데이터 변환

송신 시스템에서 사용하는 코드를 수신 시스템의 코드로 매핑(Mapping) 및 변환하고, 데이터 타입(Type, 형식) 등이 상이할 경우 데이터 변환 작업을 수행한다.('<표 1-12> 송수신 시스템간의 코드 변환을 위한 코드 매핑 테이블' 및 '<표 1-13> 송수신 시스템의 자격 코드를 통합하여 표준화한 테이블' 참고)

코드 매핑(Mapping)과 데이터 변환은 송신 시스템에서 인터페이스(I/F: Interface) 테이블 또는 파일 생성 시에 수행하거나 수신 시스템에서 연계 데이터를 수신 시스템의 운영 데이터베이스(DB: Database)에 반영하기 전에 수행할 수 있다.

코드 매핑(Mapping)과 데이터 변환을 송신 시스템에서 수행할지 수신 시스템에서 수행할지의 여부는 시스템 특성과 환경에 따라 적합한 변환 시점을 결정하여 적용한다. 연계 아키텍처에서 중계 서버가 위치할 경우, 중계 서버에서 코드 매핑(Mapping)과 데이터 변환을 수행하도록 정의할 수도 있다.

코드 매핑(Mapping)이나 데이터 변환 과정에서 발생한 오류는 변환 일시, 오류 코드(Code) 및 오류 내용 등을 로그(Log) 테이블(Table)에 기록한다.

③ 인터페이스(I/F, Interface) 테이블(Table) 또는 파일 생성

연계 데이터는 데이터베이스의 테이블(Table) 또는 파일 형식으로 생성한다. 테이블과 파일 형식 여부는 적용하는 연계 솔루션(Solution)에서 지원하는 기능에 따라 선택할 수 있다.

인터페이스 테이블 또는 파일의 구조(칼럼명 또는 속성명, 데이터 타입, 길이 등), 레이아웃(Layout)을 사전에 협의하여 정의한다. 정의한 단위 테이블 및 파일을 인터페이스라고 한다.

인터페이스 테이블 또는 파일의 구조 및 레이아웃(Layout)은 '학습 1. 연계 데이터 구성'을 참고한다.

【 인터페이스(I/F: Interface) 테이블 또는 파일 레이아웃(Layout) 설계 】

일반적으로 송신 시스템의 인터페이스 테이블(파일)의 구조와 수신 시스템의 인터페이스 테이블(파일)의 구조는 동일하게 구성하나, 경우에 따라서 송수신 시스템의 인터페이스 테이블(파일)의 구조를 상이하게 설계한다.

송신 시스템의 인터페이스 테이블(파일)에서는 송신 관련 정보(연계 데이터 생성 일시, 송신 일시, 송신 상태 등)를 관리하기 위한 항목과 송신 시스템에서 필요한 항목을 추가하여 설계한다.

수신 시스템의 인터페이스 테이블(파일)에서는 수신 관련 정보(수신 일시, 처리 일시, 수신 상태 등)를 관리하기 위한 항목과 수신 시스템에서 필요한 항목을 추가하여 설계한다.

인터페이스(I/F) 테이블 또는 파일에 연계 데이터를 생성 단계 및 생성 과정에서 발생한 모든 오류 사항은 로그 테이블(파일)에 기록하여 송수신 현황과 오류 현황 모니터링 시 활용한다.

1. 송신 시스템

송신 시스템의 운영 DB에서 연계 데이터를 식별 및 추출하여 인터페이스 테이블(파일)로 생성할 때, 주민 번호와 같은 개인 정보 등의 이유로 보안이 필요한 데이터는 암호화하여 저장한다.(암·복호화 관련 연계 데이터 보안 구현 부분은 '학습 2-3. 연계 데이터 보안 적용하기' 참고)

<표 2-4> 송신 시스템의 인터페이스(I/F: Interface) 테이블 또는 파일 레이아웃(Layout)

연계 데이터(Data)	발생 소스(Source) 및 시점	용도 및 목적
생성 일시	추출 시 신규 생성하여 추가	연계 데이터 생성 및 송신 관리
변경 구분		
운영 DB에서 추출 및 생성한 데이터 1	송신 시스템의 운영 DB	수신 시스템에서 활용
:		
운영 DB에서 추출 및 생성한 데이터 N		
송신 일시	송신 시 업데이트(Update, 수정)	연계 데이터 생성 및 송신 관리 (처리 상태: 생성, 오류, 송신)
처리 상태		
오류 코드	추출, 매핑 등 단계별 결과 업데이트(Update, 수정)	
오류 내용		

2. 수신 시스템

송신 시스템에서 암호화되어 생성된 연계 데이터는 수신 시스템에서 운영 DB에 반영 시 암호화 시 적용된 동일 알고리즘으로 복호화 처리한다. 연계 데이터의 암·복호화 처리를 위해서는 송수신 시스템 간 암호화 알고리즘 및 키를 사전 협의하고 적용 방식 등을 공유·배포해야 한다.

<표 2-5> 수신 시스템의 인터페이스(I/F: Interface) 테이블 또는 파일 레이아웃(Layout)

연계 데이터	송신 시스템 I/F 테이블(파일)의 연관 항목	발생 소스(Source) 및 시점	용도 및 목적
수신 일시	-	수신 일시로 신규 추가	연계 데이터 수신 및 반영 관리
변경 구분	변경 구분	송신 시스템	운영 DB에 반영 시 처리 작업 구분
연계 데이터 1	운영 DB에서 추출 및 생성한 데이터 1		운영 DB에 반영
:	:		
연계 데이터 N	운영 DB에서 추출 및 생성한 데이터 N		
송신 일시	송신 일시		운영 DB에 반영 시 처리 순서 구분
처리 일시	-	운영 DB에 반영 시 업데이트(Update, 수정)	연계 데이터 수신 및 반영 관리 (처리 상태: 수신, 오류, 반영 완료)
처리 상태	-		
오류 코드	-	매핑 및 반영 등 단계별 결과 업데이트(Update, 수정)	
오류 내용	-		

④ 로그(Log) 기록

송신 시스템에서의 일련의 과정(연계 데이터 추출과 생성, 코드 매핑 및 변환, 인터페이스 테이블(파일)에 데이터 추가 등)과 수신 시스템에서 일련의 과정(데이터 수신 및 인터페이스 테이블(파일) 데이터 생성, 코드 매핑 및 변환, 운영 DB에 반영 등) 상 모든 활동에 대한 결과를 로그(Log) 테이블(파일)에 기록한다.

<표 2-6> 로그(Log) 기록 유형

Log 단위	로그(Log) 성격	로그 파일명 예시
1행 (건, Row 데이터)	연계 데이터별 송수신 현황 관리	- 인터페이스(I/F) 테이블 또는 파일에 속성으로 포함 - 로그 상세 이력 또는 데이터별 송수신 이력 현황
	연계 데이터 처리 단계별 오류 사항	- 송수신 로그(Log) 현황 또는 오류 현황 로그(Log) ※ 오류 기록과 오류 테이블 및 파일 구조(Layout) 등 상세 내용은 '학습 2-2 연계 장애 및 오류 처리 구현하기' 참고
테이블 또는 파일	송수신 처리 단계별 수행 결과 정보	- 송수신 현황 로그(Log)

로그(Log)를 파일로 기록할 경우에는 파일이 생성되는 위치인 디렉토리(Directory), 파일 생성 시점(일 또는 시간 등), 파일명 생성 규칙, 생성하는 파일 형식 등을 정의한다.

<표 2-7> 파일로 로그(Log) 기록 시 결정 사항

결정 사항	상세 내용	사례
파일 생성 위치	로그 파일이 생성되는 위치	예) 송수신 구분/생성 연도(YYYY)/생성 월(MM) /send/2015/09
파일 생성 시점	로그 파일이 생성되는 분, 시간, 일 등의 단위로 로그 발생량을 고려하여 정의	예) 일 단위로 로그 파일 생성 또는 시간 단위로 로그 파일 생성
파일명 생성 규칙	로그 파일의 명으로, 로그 파일이 언제 생성되었으며, 어떤 대상이 기록되는지 식별이 용이하도록 부여	예) 파일 생성 연월일(YYYYMMDD) + "-" + 연계(인터페이스) 단위명 20151002-TB_EAS010
파일 형식	로그 파일의 기록 형식	예) txt, log, xml

로그(Log) 파일에 처리 작업에 대한 결과를 기록하는 이유는 연계 데이터가 기준대로 추출 및 식별되어 정상적으로 송수신되고, 운영 DB에 반영되었는지 확인하며 송수신 과정에서 오류가 발생했을 경우, 오류 발생 현황과 원인을 분석하여 오류가 발생한 데이터를 재작업하기 위함이다.

송수신 시스템에서 모니터링하는 목적에 따라 작업 결과를 기록하는 로그(Log) 테이블(파일)의 관리 항목은 상이하게 정의할 수 있다.

1. 송신 시스템

<표 2-8> 송신 시스템 인터페이스 테이블(파일) 단위 로그(Log) 테이블(파일) 레이아웃(Layout)

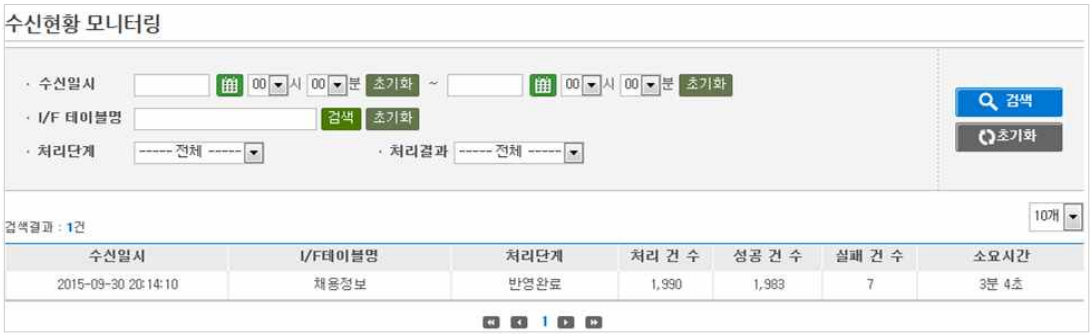
한글명	영문명	데이터 타입	데이터 길이	PK 여부	의미
송신 일시	SEND_DTTM	DATE	7	Y	I/F 테이블(파일) 단위로 데이터를 수신 시스템으로 전송한 일시
인터페이스 단위명 (테이블 또는 파일명)	IF_TABLE_NM	String	30	Y	I/F 테이블 또는 파일명
처리 단계	PROC_STEP	String	2		CF(Create Finish): 생성 완료 CE(Create Error): 생성 시 오류 SF(Send Finish): 전송 완료 SE(Send Error): 전송 시 오류
처리 건수	PROC_CNT	Number	5		데이터 생성 및 전송 대상 데이터 건수
성공 건수	SUCC_CNT	Number	5		생성(매핑) 및 전송 성공 데이터 건수
실패 건수 (오류 건수)	FAIL_CNT (ERR_CNT)	Number	5		생성(매핑) 및 전송 실패(오류) 데이터 건수
처리 시간	PROC_TM	DATE	7		I/F 테이블(파일) 단위별 생성(매핑) 및 전송 시 소요 시간(시분초)

[그림 2-3] 송신 시스템 로그(Log) 테이블(파일)에 활용한 송신 현황 모니터링 화면

2. 수신 시스템

<표 2-9> 수신 시스템 인터페이스 테이블(파일) 단위 로그(Log) 테이블(파일) 레이아웃(Layout)

한글명	영문명	데이터 타입	데이터 길이	PK 여부	의미
수신 일시	SEND_DTTM	DATE	7	Y	송신 시스템으로부터 I/F 테이블(파일) 단위로 데이터를 전송받아 I/F 테이블(파일)의 데이터를 생성한 일시
인터페이스 단위명 (테이블 또는 파일명)	IF_TABLE_NM	String	30	Y	I/F 테이블 또는 파일명
처리 단계	PROC_STEP	String	2		RF(Receive Finish): 수신 완료 RE(Receive Error): 수신 시 오류 AF(Apply Finish): 운영 DB 반영 완료 AE(Apply Error): 반영 시 오류
처리 건수	PROC_CNT	Number	5		수신 및 운영 DB 반영(코드 매핑, 데이터 전환) 대상 데이터 건수
성공 건수	SUCC_CNT	Number	5		수신 및 운영 DB 반영(코드 매핑, 데이터 전환) 성공 데이터 건수
실패 건수 (오류 건수)	FAIL_CNT (ERR_CNT)	Number	5		수신 및 운영 DB 반영(코드 매핑, 데이터 전환) 실패(오류) 데이터 건수
처리 시간	PROC_TM	DATE	7		I/F 테이블(파일) 단위별 수신 및 운영 DB 반영 시 소요 시간(시분초)



[그림 2-4] 수신 시스템 로그(Log) 테이블(파일)에 활용한 수신 현황 모니터링 화면

5 연계 서버 또는 송신 어댑터(Adapter)

연계 서버, 송수신 어댑터(Adapter)는 도입한 연계 솔루션에서 제공하는 연계 메커니즘의 구성 요소로, 도입한 연계 솔루션에 따라 지원하는 기능은 상이하다.

연계 서버는 송신 시스템 또는 수신 시스템 둘 중 한 곳에만 설치하며 인터페이스 테이블(파일)의 데이터를 전송 형식으로 변환, 송수신 대상 관리, 실제 송수신, 송수신 여부 관리 등 송수신과 관련된 모든 처리 수행한다.

일반적으로 연계 서버는 수신 시스템 구간에 위치하며, 연계 엔진, 중계 서버의 용어로도 사

용된다.

송신 어댑터(Adapter) 또는 송신 클라이언트(Client, 송신 모듈)는 데이터의 송신 처리를 담당하는 요소로 연계 서버가 설치된 경우에는 연계 서버에 포함되기도 한다. 그러나, 연계 서버 없이 연계 데이터가 생성되는 송신 시스템의 운영 서버에 설치되어 인터페이스 테이블(파일)의 데이터를 전송 형식으로 변환, 실제 송신 등을 수행한다.

6 전송

연계 서버(엔진)에 의해 송신 시스템에서 생성된 연계 데이터는 네트워크 환경에 따라 설정된 전송 형식으로 변환된 후 수신 시스템으로 전송된다.

【 연계 전송 주기 】

전송 주기는 실시간 또는 분, 시간, 일 단위의 주기로 설정할 수 있으나, 연계 데이터 생성 방식이나 시점 등에 영향을 받는다. 즉, 전송(연계) 주기는 ① 송신 시스템의 연계 데이터 추출 및 생성 방식, ② 연계 서버(엔진)나 송수신 어댑터(Adapter)의 연계 주기, ③ 수신 시스템의 연계 데이터를 운영 DB에 반영하는 방식에 의해 결정된다.

예를 들어, 연계 서버나 송수신 어댑터(Adapter)에서 연계 주기를 실시간으로 설정하더라도, 송신 시스템의 운영 DB에서 시간 주기의 배치(Batch) 프로그램으로 연계 데이터를 추출하여 생성한다면 연계 주기는 실시간보다 주기가 긴 시간 단위로 연계 주기가 적용된다.

<표 2-10> 연계 전송 주기 결정 요소

영역	수행 작업	연계 주기 영향 요소	예시 및 설명
송신 시스템	연계 데이터 추출 및 생성	추출 및 생성 방식 (데이터 추출 및 생성 프로그램 구현 방식)	- 실시간: 디비 엠 에스(DBMS) 오브젝트(Object) 중 트리거(Trigger) 및 에이피(AP)에서 생성 - 분/시간/일 주기: 배치 프로그램(Batch Program)에서 추출 및 생성
연계 서버	송수신	송수신 주기 설정값	- 연계 솔루션(엔진) 설정 방식에 따라 상이(일반적 연계 서버의 콘솔에서 지정, xml 파일에 저장됨)
수신 시스템	연계 데이터 운영 DB 반영	운영 DB 반영 프로그램 구현 및 실행 방식	- 실시간: 디비 엠 에스(DBMS) 오브젝트(Object) 중 트리거(Trigger)를 활용 및 연계 데이터 수신 후 운영 디비(DB) 반영 Program 실행 - 분/시간/일 주기: 배치 프로그램(Batch Program)에서 운영 DB 반영

7 연계 서버 또는 수신 어댑터(Adapter)

연계 서버 또는 수신 어댑터(Adapter)는 송신 시스템에서 연계 데이터를 전송 형식으로 전환하여 송신한 정보(패킷 또는 스트림 단위)를 다시 인터페이스(I/F) 테이블(파일)의 레이아웃(Layout)으로 전환하여 인터페이스(I/F) 테이블(파일)을 생성한다.

연계 서버(엔진)에서 실제 수신, 수신 대상 관리, 전송 형식을 인터페이스(I/F) 테이블(파일) 레이아웃(Layout)으로의 전환 등을 담당하므로 수신 어댑터(Adapter) 또는 수신 클라이언트(Client, 수신 모듈)를 설치하지 않아도 된다.

8 운영 DB에 연계 데이터 반영

수신된 인터페이스(I/F) 테이블(파일) 구조의 연계 데이터를 운영 DB에 반영하는 과정으로, 인터페이스(I/F) 테이블(파일)의 데이터를 운영 DB의 테이블에 반영하는 변환(매핑) 프로그램을 별도로 구현한다.

1. 변환(매핑) 프로그램 구현 내용

수신 인터페이스(I/F) 테이블(파일)의 데이터를 운영 DB의 테이블 구조와 값으로 변환하는 프로그램에는 수신된 연계 데이터가 송신 시스템의 운영 DB에서 사용하는 공통 코드, 데이터 형식으로 전환하고 확인하는 작업이 포함된다.

2. 변환(매핑) 프로그램 구현 방식

수신 인터페이스(I/F) 테이블(파일)의 데이터를 운영 DB에 반영하기 위한 구현 방식으로는 송신 시스템의 운영 DB에서 데이터를 추출 및 생성하는 구현 방식과 유사하다.

수신 인터페이스(I/F) 테이블(파일)에 트리거(Trigger)와 같은 데이터베이스 관리 시스템(DBMS: Database Management System) 오브젝트(Object)를 활용하여 실시간으로 반영하도록 구현하거나 수신된 인터페이스(I/F) 테이블의 데이터를 로드(Load)하여 운영 DB로의 전환 및 변환하는 응용 프로그램을 구현한다.

3. 연계 데이터 반영 주기

수신 시스템 운영 DB에 수신된 연계 데이터를 반영하는 방식을 응용 프로그램으로 별도로 구현했을 경우에는 운영 DB 반영 응용 프로그램의 실행 방식에 따라 실시간 또는 주기적으로 연계 데이터를 반영할 수 있다.

연계 데이터가 수신되는 즉시 운영 DB 반영 프로그램을 실행하면 운영DB에 수신 데이터를 실시간으로 반영할 수 있다.

연계 데이터를 운영 DB에 주기적으로 반영하고자 할 경우, 운영 DB 반영 프로그램의 실행 주기(분, 시간, 일 등)를 설정하면 연계 데이터가 수신된 후 설정된 주기가 경과해야 운영 DB에 반영된다.

수행 내용 / 연계 메커니즘 정의하기

재료 · 자료

- 하드웨어 및 시스템 소프트웨어, 네트워크 등 시스템 구성도(송신 시스템, 수신 시스템)
- 연계 요구 사항 분석서
- 테이블 정의서(송신 시스템, 수신 시스템)
- 코드 정의서(송신 시스템, 수신 시스템)
- 연계(인터페이스) 정의서
- 연계(인터페이스) 명세서
- 송수신 절차도 또는 연계 메커니즘 정의서, 연계 아키텍처 설계서
- 인터페이스(I/F: Interface) 테이블(파일 Layout) 정의서
- 로그 테이블(파일 Layout) 정의서
- 코드 매핑 테이블(파일 Layout) 정의서

기기(장비 · 공구)

- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

1 송수신 절차 또는 연계 메커니즘 정의서를 작성한다.

연계 메커니즘은 송수신 시스템의 운영 환경과 운영되는 데이터 특성(발생 시점, 활용 목적 등)에 대한 이해를 기반으로 정의할 수 있다.

1. 송신 시스템과 수신 시스템의 하드웨어 구성, 시스템 소프트웨어 구성, 네트워크 현황 또는 연계 아키텍처 설계서를 확인한다.

   연계 아키텍처 설계서가 없을 경우, 필요에 따라 송수신 시스템의 시스템 구성도를 참고한다. 송수신 시스템의 구성도는 ‘<그림 1-4> 시스템 구성도’와 아래의 ‘송수신 시스템 운영 환경’을 참고한다.

【 송수신 시스템 운영 환경 】

① 송신 시스템과 수신 시스템은 각각 인터넷 망과 내부망인 인트라넷 망에 연결되어 있음

② 송신 시스템과 수신 시스템은 별도 전용선을 설치하여 연계할 예정으로, 중간에 중계 서버를 도입하여 연계 서버를 설치할 계획임

③ 중계 서버에는 연계 서버 엔진의 송수신 처리 기록(Logging)과 모니터링을 위한 별도의 데이터베이스 관리시스템(DBMS: Database Management System)을 설치하여 운영할 예정임

④ 송신 시스템과 수신 시스템은 각각 웹(WEB), 웹 애플리케이션 서버(WAS: Web Application Server), DB 서버가 분리되어 있으며, 웹(WEB), 와스(WAS), 디비 엠 에스(DBMS)는 모두 상용 시스템 소프트웨어(SW: Software)가 설치되어 서비스되고 있음

2. 확인된 연계 아키텍처 설계서(송수신 시스템의 시스템 구성도)를 참고하여 송수신 절차 또는 연계 메커니즘을 '<그림 2-1> 연계 메커니즘' 형식으로 작성한다.

3. 위의 2항에서 작성한 연계 메커니즘과 '<표 2-3> 연계 메커니즘의 요소별 관리 및 구현 주체'를 참고하여 요소별 고려 요소를 아래의 '<표 2-11> 연계 메커니즘 정의서(설계서)'에 정리한다.

<표 2-11> 연계 메커니즘 정의서(설계서) 양식

영역(구간)	연계 메커니즘	구성 요소 및 구현 방식	고려 요소	메커니즘 결정 사항 (예: 실시간 처리 등)

2 연계 메커니즘 구간(영역)별 구성 요소를 정의 및 설계한다.

연계 메커니즘을 정의하면서, 구성 요소별 연계 전송 주기, 연계 프로그램 구현 방식, 연계(인터페이스) 테이블 구조(파일 Layout) 및 생성 위치 등을 설계한다.

송수신 주기를 결정하고 연계(인터페이스) 테이블(파일) 항목을 설계하기 위해서는 연계 요구 사항 분석서의 사용자 요구 사항과 송수신 시스템의 테이블 정의서, 코드 정의서 등을 활용한다.

송수신 주기와 연계(인터페이스) 방식(테이블 vs 파일)에 따라 연계 데이터를 생성하고, 수신된 연계 데이터를 운영 DB에 반영하는 응용 프로그램의 구현 방식이 결정되거나 제한될 수 있다.

1. 송신 시스템의 응용 애플리케이션과 운영 DB, 데이터 발생 시점 및 주기 등을 확인한 후, 운영 DB에서 데이터 추출 및 생성 방법과 구현 방식을 정의, 설계한다.

   데이터 추출 및 생성 방법을 정의 시 송수신 주기에 대한 요구 사항(실시간 처리 vs 분·시간·일 단위 주기적 연계 등)을 확인한다.

   1번 3항에서 작성했던 '<표 2-11> 연계 메커니즘 정의서(설계서)' 에 설계한 데이터 추출 및 생성 방법과 구현 방식을 추가한다.

2. 송신 시스템과 수신 시스템의 코드 정의서를 확인하여, 코드 변환 필요성 및 가능 여부, 변환 시점 등을 코드 매핑 내역서로 정의하고 코드 매핑 테이블을 설계한다.

   【 결정 사항: 코드 변환 필요 여부, 코드 변환 가능 대상, 코드 변환 시점 】

   코드 변환 필요 여부와 코드 변환 가능 대상 확인, 코드 매핑 정의서, 코드 매핑 테이블에 대한 상세 내용은 '학습 1-2. 연계 데이터 식별 및 표준화의 2 연계 코드 변환 및 매핑' 을 참고한다.

   코드 변환은 ① 송신 시스템에서 연계 데이터 추출 및 생성 후 인터페이스 테이블(파일)에 추가하는 시점 또는 ② 중계 서버가 존재할 경우, 중계 서버에서 송수신하는 시점, ③ 수신 시스템의 인터페이스 테이블(파일)에서 데이터를 로드(Load)하여 운영 DB에 반영하는 시점 중 한 곳에 적용한다.

   결정한 코드 변환 시점은 1번 3항에서 작성했던 '<표 2-11> 연계 메커니즘 정의서(설계서)' 에 추가 및 변경한다.
   결정한 코드 변환 시점에 의해 변경된 사항은 1번 2항에서 작성한 연계 메커니즘에 반영하여 수정한다.

   '학습 1-2. 연계 데이터 식별 및 표준화' 수행 내용 3번 2항에서 작성한 코드 매핑 내역서의 레이아웃(구조, Layout)으로 코드 매핑 테이블(파일)을 '<표 1-1> 테이블 정의서' 형식에 맞게 작성 및 설계한다.

3. '학습 1-2. 연계 데이터 식별 및 표준화' 에서 작성한 연계(인터페이스) 정의서 및 명세서에 송수신 현황 모니터링을 위한 관리 정보를 추가한 레이아웃(구조, Layout)으로 인터페이스 테이블(파일)을 설계한다.
   인터페이스 테이블은 '<표 1-1> 테이블 정의서' 형식으로 작성 및 설계한다.

4. 연계 메커니즘 단계별 진행 상태를 모니터링하고, 처리 결과를 확인할 수 있는 로그 테이블을 '<표 1-1> 테이블 정의서' 형식으로 정의 및 설계한다.

【 참고 】

① <표 2-8> 송신 시스템 인터페이스 테이블(파일) 단위 로그(Log) 테이블(파일) 레이아웃(Layout)

② <표 2-9> 수신 시스템 인터페이스 테이블(파일) 단위 로그(Log) 테이블(파일) 레이아웃(Layout)

5. 수신 시스템의 운영 DB 환경, 운영 DB 및 응용 애플리케이션에서 연계 데이터 활용 목적과 데이터 성격 등을 확인한 후, 운영 DB에 연계 데이터를 반영하는 주기 및 반영 프로그램 구현 방식을 정의, 설계한다.

운영 DB에 연계 데이터 반영 방식을 정의 시 송수신 주기에 대한 요구 사항(실시간 처리 vs 분·시간·일 단위 주기적 연계 등)을 확인한다.

1번 3항에서 작성했던 '<표 2-11> 연계 메커니즘 정의서(설계서)' 에 결정한 수신 시스템의 운영 DB에 연계 데이터를 반영하는 프로그램 구현 방식, 주기 등을 추가 작성한다.

3 연계 메커니즘 구간(영역)별 구성 요소의 정의 및 설계 결과는 1번 2항에서 작성했던 '<그림 2-1> 연계 메커니즘' 형식의 연계 메커니즘과 1번 3항에서 작성했던 '<표 2-11> 연계 메커니즘 정의서(설계서)' 에 반영하며, 연계 요구 사항과 불일치 사항, 구현 불가능한 사항이 없는지 최종 점검한다.

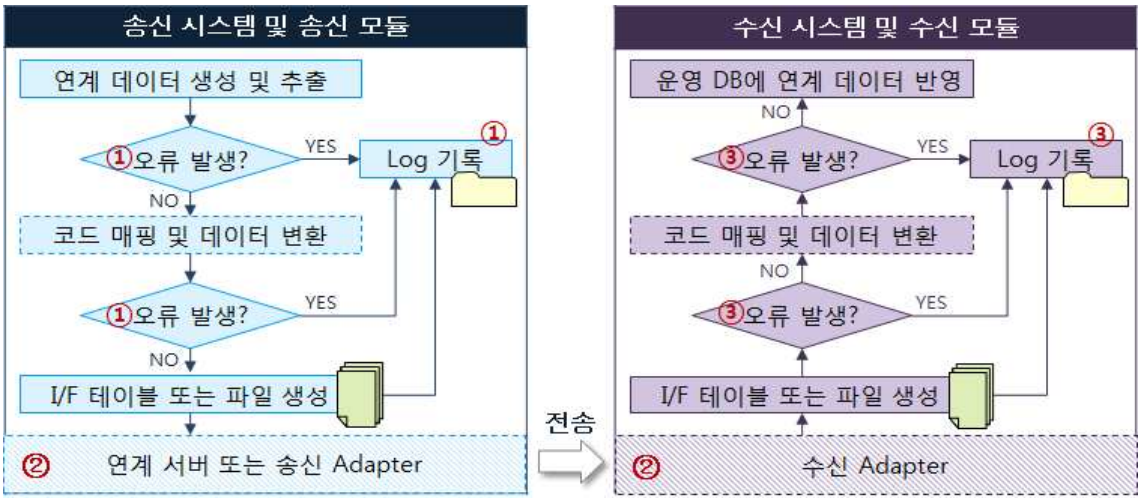
수행 tip

- 요구 사항 분석서의 송수신 주기에 대한 요구 사항을 식별하여 연계 메커니즘(Mechanism) 정의 과정에서 참고한다.
- 요구 사항 분석서에서 연계 요구 사항은 송수신 주기, 연계 프로그램 구현 방식 등 연계 메커니즘(Mechanism) 정의 및 연계 아키텍처 설계 시 결정 사항에 대한 기준이 된다.
- 연계(인터페이스) 테이블(파일) 설계 단위는 하나의 트랜잭션(Transaction)으로 처리되어야 하는 정보의 그룹이다.

2-2. 연계 장애 및 오류처리 구현

학습 목표	• 개발하고자 하는 응용 소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리 방안을 정의할 수 있다.

필요 지식 /



[그림 2-5] 연계 메커니즘에서 장애 및 오류 발생 시점과 기록(Logging)

연계 메커니즘에서 장애 및 오류가 발생할 수 있는 구간은 ① 송신 시스템에서 연계 데이터를 추출하여 생성하고 연계(인터페이스) 테이블(파일)에 등록하는 과정, ② 데이터가 실제 송수신 되는 과정, ③ 수신 시스템에서 연계(인터페이스) 테이블(파일)의 연계 데이터를 운영 DB에 반영하는 과정의 세 곳이다.

송신 시스템에서 생성된 연계 데이터가 수신 시스템에 잘 반영되기 위해서는 ① ~ ③ 구간에서 장애 및 오류가 발생하는지를 확인하고, 모니터링한다.

실시간으로 장애나 오류 발생을 인지하고 확인하여 장애 원인에 따른 대처를 함으로써 송신 시스템, 수신 시스템의 운영과 서비스가 정상적으로 동작하도록 보장하고 최적의 서비스 상태를 유지한다.

1 연계 메커니즘 구간별 장애 및 오류 모니터링 현황

연계 메커니즘 구간별 장애 및 오류 발생 시점, 오류 로그 기록 주체 등 상세 사항은 '<표 2-12> 연계 메커니즘 구간별 장애 및 오류 발생 시점과 기록(Logging) 주체'와 같다.

<표 2-12> 연계 메커니즘 구간별 장애 및 오류 발생 시점과 기록(Logging) 주체

구간	오류 발생 시점	오류 기록 테이블(파일)	오류 로그 기록 주체
송신 시스템	데이터 생성 및 추출 코드 매핑 및 데이터 전환 연계(인터페이스) 테이블(파일) 등록	- 구현한 연계 프로그램에서 정의한 로그 및 오류 로그 테이블(파일) - 연계(인터페이스) 테이블의 일부 관리 용도 항목	별도로 구현한 송신용 연계 (응용) 프로그램
연계 서버 (중계 서버 ·엔진)	송신 시스템 연계(인터페이스) 테이블의 연계 데이터 로드(Load) 및 전송 형식으로 변환 시 송수신 시 수신 받은 전송 형식을 수신 시스템 연계(인터페이스) 테이블(파일)의 형식으로 변환 및 로드(Load) 시	- 연계 서버(엔진)에서 설정된 로그 및 오류 로그 테이블(파일)	연계 서버(엔진) 또는 중계 서버(엔진)
수신 시스템	연계(인터페이스) 테이블(파일)의 연계 데이터 로드(Load) 코드 매핑(Mapping) 및 데이터 변환 운영 DB에 반영	- 구현한 연계 프로그램에서 정의한 로그 및 오류 로그 테이블(파일) - 연계(인터페이스) 테이블의 일부 관리 용도 항목	별도로 구현한 수신용 연계 (응용) 프로그램

연계 메커니즘에서 코드 매핑 및 데이터 전환 작업을 연계(중계) 서버에서 처리하도록 정의한 경우, 코드 매핑 및 데이터 전환에 대한 오류 발생과 처리는 연계(중계) 서버로 이동된다.

2 장애 및 오류 유형과 처리 방안

1. 장애 및 오류 유형

장애나 오류 유형은 ① 연계 시스템(서버 또는 엔진)의 오류, ② 송신 시스템의 연계 프로그램(운영 DB에서 연계 데이터를 생성 및 추출, 코드 및 데이터 변환 등을 수행하도록 별도로 구현한 응용 프로그램) 또는 수신 시스템의 연계 프로그램(운영 DB에 데이터를 반영하고, 코드 및 데이터 변환 등을 수행하도록 별도로 구현한 응용 프로그램) 오류, ③ 연계 데이터 자체의 오류로 분류할 수 있다.

2. 장애 및 오류 처리 방안

연계 서버 엔진에서 기록하는 로그와 송수신 시스템의 연계 프로그램에서 기록하는 로그를 확인하여 상세 오류 원인을 분석한 후 분석 결과에 따른 해결 방안을 결정하여 수행한다.

예를 들어, 연계 서버(엔진)의 장애일 경우 연계 서버 엔진에서 기록되는 로그(Log) 파일의

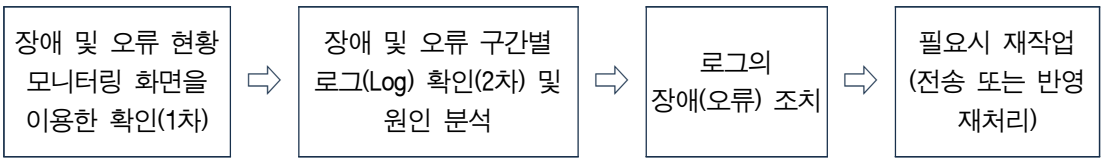
내용을 확인하여 연계 서버(엔진) 정상적 실행 여부, 연계 서버의 연계 데이터의 전송 형식으로의 변환 과정, 송수신 대상 시스템의 정상적 연결 여부(송수신 시스템의 IP 및 Port 접근 가능 여부 등), 송수신 연계(인터페이스) 테이블(파일) 접근 권한 등을 순차적으로 점검한다.

연계 서버(엔진)의 실행 여부를 점검 결과 시스템 작업 등으로 인해 연계 서버가 기동중이지 않은 경우는 연계 서버(엔진)를 기동한다.

<표 2-13> 장애 및 오류 유형

오류 유형		설명	사례	처리 방안
연계 시스템		연계 서버의 실행 여부를 비롯하여 송수신, 전송 형식 변환 등 서버의 기능과 관련된 장애 및 오류	연계 서버 미기동, 송수신 시스템의 아이피(IP) 및 포트(Port) 접속 불가	연계 서버(엔진)의 로그를 확인하여 원인 분석 후 처리
연계 응용 프로그램	송신 시스템 연계 프로그램	데이터 추출을 위한 DB 접근 시 권한 불충분, 데이터 변환 시 익셉션(Exception) 미처리 등 연계 프로그램 구현상의 오류	등록되지 않은 코드로 매핑 불가	송수신 시스템 연계 프로그램에서 기록하는 로그(Log)를 확인하여 원인 분석 후 결과에 따른 처리, 처리 이후 데이터 전송이나 반영 재작업
	수신 시스템 연계 프로그램	운영 DB에 반영하기 위한 DB 접근 권한 불충분, 데이터 변환 및 반영 시 익셉션(Exception) 미처리 등 연계 프로그램 구현상의 오류	등록된 데이터가 존재하지 않음 ⇒ 등록되지 않은 데이터에 대한 수정(Update) 처리	
연계 데이터 오류		송신 시스템에서 추출된 연계 데이터가 유효하지 않은 값으로 인한 오류	유효하지 않은 일자	송신 시스템 연계 프로그램에서 기록하는 Log를 확인하여 데이터 보정 후 재전송

3 장애 및 오류의 확인과 처리 절차



[그림 2-6] 장애 및 오류의 확인과 처리 절차

연계 시스템에서는 연계 서버(엔진)과 연계 응용 프로그램에서 기록한 장애 및 오류 로그(Log)를 확인하고, 검색할 수 있는 장애 및 오류 현황 모니터링 화면을 제공한다.

1. 장애 및 오류 확인 절차

운영자는 장애 및 오류 현황 모니터링 화면을 이용하여 오류 원인 및 발생 현황을 1차 확인한다. 시스템에서 제공되는 장애 및 오류 현황 모니터링 화면을 통해 확인이 불가할 경우, 엔진이나 응용 프로그램에서 기록하는 장애 및 오류 로그(Log) 테이블(파일)의 내용을

2차 확인하여 원인을 분석한다.

2. 장애 및 오류 처리 절차

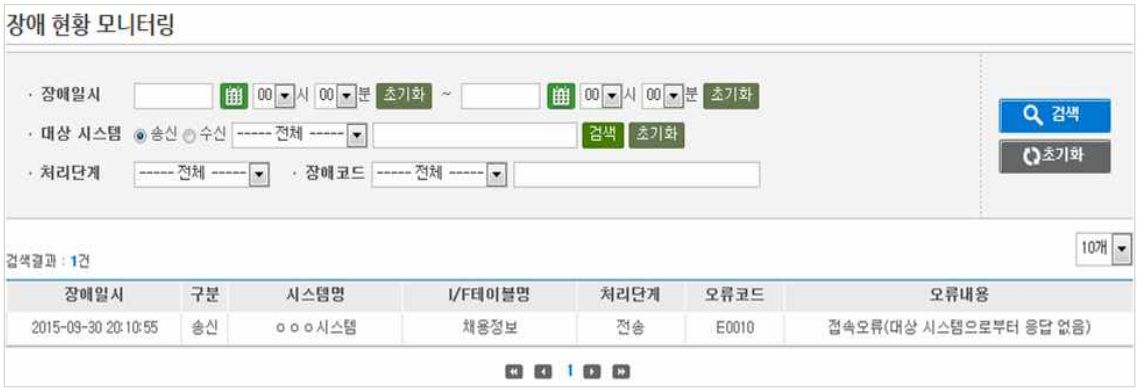
분석된 결과에 따른 대응 조치를 수행한 후, 장애 및 오류로 인해 전송되지 않거나 운영 DB에 반영되지 않은 연계 데이터의 재작업 여부를 결정한다. 재작업이 필요할 경우 송신 시스템의 경우에는 연계 데이터 생성 및 추출 작업을 재처리하고, 수신 시스템의 경우 운영 DB에 연계 데이터를 반영하기 위해 연계 응용 프로그램을 재실행하면서 장애 및 오류 조치 완료 여부를 확인한다.

3. 장애 및 오류 현황 모니터링 화면

연계 서버(엔진)에서 기록하는 장애 로그 테이블(파일) 및 장애 로그 테이블(파일) 내용을 확인하는 모니터링 화면은 연계 솔루션에서 기본적으로 포함되어 제공된다.

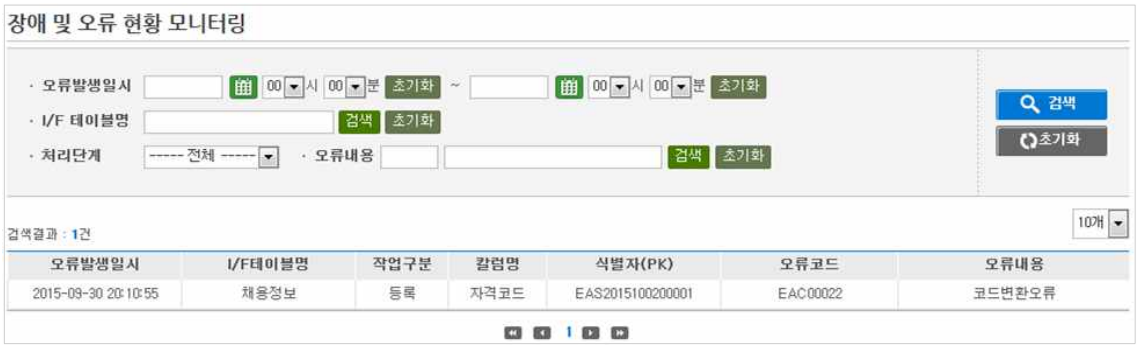
송수신 시스템의 연계 응용 프로그램에서 기록하는 장애 및 오류 로그(Log) 테이블(파일)과 장애 및 오류 로그(Log) 테이블(파일) 내용을 확인하기 위한 모니터링 화면은 별도로 구현해야 한다.

송신 시스템과 수신 시스템의 장애 및 오류 현황 모니터링 화면은 검색하고자 하는 조건과 확인하고자 하는 결과에 따라 각각 구현한다.



[그림 2-7] 장애 현황 모니터링 화면 - 연계 서버(엔진)

[그림 2-8] 장애 및 오류 현황 모니터링 화면 - 연계(인터페이스) 테이블(파일) 단위



[그림 2-9] 장애 및 오류 현황 모니터링 화면 - 연계(인터페이스) 데이터 단위

4. 장애 및 오류 로그(Log)

장애 및 오류 현황 모니터링 시 참조되는 장애 및 오류 로그 내용은 연계 서버(엔진)의 시스템 전체에 해당하는 오류 로그 파일과 송수신 시스템의 연계 응용 프로그램에서 기록하는 시스템 일부에 해당하는 오류 로그 파일로 구분할 수 있다.

송수신 시스템의 연계 응용 프로그램에서 기록하는 오류 로그 테이블(파일)은 연계(인터페이스) 테이블(파일) 단위의 오류 로그와 연계 데이터(행 또는 Row) 단위의 오류 로그 파일로 세분화된다.

<표 2-14> 장애 및 오류 로그(Log) 테이블(파일) 분류

로그분류	기록 단위	영향 범위	로그 파일명
시스템	연계 서버(엔진) 기능 전체	시스템 전체	- 연계 서버(엔진)에서 설정된 로그 및 오류 로그 테이블(파일)
	특정 송수신 시스템 또는 연계(인터페이스)	장애가 발생한 시스템 일부분	
연계 응응 프로그램	연계 인터페이스	오류가 발생한 연계(인터페이스) 테이블(파일) 일부분	- 구현한 연계 프로그램에서 정의한 로그 및 오류 로그 테이블(파일) - 연계(인터페이스) 테이블의 일부 관리 용도 항목
	행 또는 로우(Row)의 연계 데이터	연계(인터페이스) 테이블(파일)의 일부 행(Row) 단위 데이터	

장애 및 오류 로그(Log) 테이블(파일)의 상세 내용은 다음의 '④ 장애 및 오류의 정의와 설계'를 참고한다.

[2015-09-24 14:33:101339] [E00003] [redacted] 접근 오류, Network Access Request에 대한 Response Timeout=100 Sec 3회

[2015-09-24 14:50:073155] [E00908] 송신 Directory Path /usr/send/data No Privilege

[2015-09-24 14:55:281222] [E01262] [redacted] File not exists, Server can't create [redacted] error handling and null pointer exception

[그림 2-10] 시스템 오류 로그(Log) 파일 내용

[2015-09-27 10:01:061211] [EAI00001] 식별자(채용 공고 번호)=EAS2015100200006, row, 변경 구분(CHG_TYPE)=C(등록), Duplication(중복 오류), 등록하려는 채용 공고 번호가 존재합니다.

[2015-09-27 10:01:061243] [EAD0003] 식별자(채용 공고 번호)=EAS2015092900177, 학교 코드(SCHL_CODE)=SCH06945, 변경 구분(CHG_TYPE)=C(등록), 코드 변환 오류, 등록되지 않은 학교 코드입니다.

[2015-09-27 10:01:070001] [EAD0600] 식별자(채용공고번호)=EAS2015100100001, 채용 공고명(EMPL_NOTC_NM)=[redacted] 채용 공고, 변경 구분(CHG_TYPE)=U(수정), Length Exceed Exception, 길이가 초과하였습니다.

[그림 2-11] 송수신 연계 응용 프로그램에서 기록하는 오류 로그(Log) 파일 내용

④ 장애 및 오류의 정의와 설계

장애 및 오류 현황을 기록하고 확인하기 위해서 ① 장애 및 오류 관리 대상(범위), ② 관리 대상의 장애 및 오류 코드와 메시지, ③ 장애 및 오류 코드와 메시지 관리 방식, ④ 장애 및 오류를 기록하는 방식을 정의하고 설계한다.

1. 장애 및 오류 관리 대상(범위)

연계 서버(엔진)에서 관리하는 장애 및 오류는 연계 서버(엔진)의 실행 여부, 네트워크 접근 가능 여부, 송수신 폴더 및 테이블의 권한 여부 등 시스템 전반적인 사항에 주로 해당하며 적용하는 연계 솔루션에 의해 결정된다.

적용하는 연계 솔루션에 의해 결정되는 연계 서버(엔진)에서 관리하는 장애 및 오류는 기록하는 방식(테이블 또는 파일)을 제외하고는 제공하는 기능을 변경하거나 추가 설계하여 적용할 수 없다.

따라서, 장애 및 오류를 기록하고 관리하는 방식의 설계는 송수신 시스템의 연계 응용 프로그램에 의한 관리 대상 장애 및 오류 범위로 한정한다.

2. 장애 및 오류 코드와 메시지 정의

송신 시스템에서 운영 DB로부터 연계 데이터를 추출하고, 코드 및 데이터를 다른 형식으로 전환하여 연계(인터페이스) 테이블(파일)에 등록하는 과정에서 발생 가능한 오류를 정의한다.

식별한 오류 내용을 분류하여 주제별로 분류하고, 각 오류 내용에 코드를 부여한다. 오류 코드는 규칙에 따라 부여한다. 오류 내용을 사용자 및 운영자가 이해하기 용이하도록 보완 작성한다.

<표 2-15> 오류 코드 부여 규칙

A B C ###
1) 2) 3) 4)

1)	장애 및 오류 여부 구분 코드(1자리)	- Error의 'E' 사용
2)	오류가 발생하는 위치 구분(1자리)	- 연계 서버(엔진), 시스템: S - 연계 응용 프로그램: A
3)	오류 유형 또는 분류 구분(1자리)	- 데이터 형식 관련 오류: F - 데이터 길이 관련 오류: L - 코드 관련 오류: C - 데이터 관련 오류(중복 또는 미등록 등): D - 필수 입력 관련 오류: M - 접근 권한 등 시스템 관련 오류: S
4)	일련번호 (3자리)	오류 유형 및 분류별로 일련번호 부여
예시	EAD001: 연계 응용 프로그램에서 처리하는 중복된 PK 값으로 인한 데이터 처리 오류 이미 등록된 ○○○○○(이)가 존재합니다.	

<표 2-16> 오류 코드 정의

오류 분류	오류 코드	오류 메시지	오류 원인 및 내용
데이터 (PK 오류)	EAD001	등록된 ○○○○○이(가) 존재합니다.	변경 구분 값이 I(추가)이나 동일한 식별자(PK)로 등록된 ○○○○○이(가) 존재하는 경우
	EAD002	수정할 ○○○○○이(가) 존재하지 않습니다.	변경 구분 값이 U(수정)이나 동일한 식별자(PK)로 등록된 ○○○○○이(가) 없는 경우
	EAD003	삭제할 ○○○○○이(가) 존재하지 않습니다.	변경 구분 값이 D(삭제)이나 동일한 식별자(PK로 등록된 ○○○○○이(가)가 없는 경우)
	:	:	:
필수	EAM001	○○○○○은(는) 필수 입력 항목입니다.	Not Null 칼럼의 값이 스페이스를 제거한 경우 Empty일 경우
길이	EAL001	○○○○○은(는) 허용 길이를 초과하였습니다.	수신 시스템의 코드 길이를 초과한 코드인 경우
코드	EAC001	등록된 코드가 없습니다.	송수신 시스템 간의 변환되어할 코드가 등록되어 있지 않을 때
	:	:	:
시스템	:	:	:
	ESS015	연계(인터페이스) 테이블 ○○○○○의 접근 권한이 없습니다.	연계 응용 프로그램이 실행되는 사용자(user)의 연계(인터페이스) 권한이 충분하지 않을 경우
	ESS016	데이터 암·복호화 처리를 위한 라이브러리를 참조할 수 없습니다.	연계 데이터 중 암·복호화 처리 대상이 존재하나, 암·복호화 처리 API나 컴포넌트가 설치되지 않았거나 또는 Path에서 확인이 안되는 경우
:	:	:	:

연계 응용 프로그램에서 정의한 오류 코드 및 메시지(Message, 내용)를 일관되게 참조할 수 있도록 관리하는 방식으로는 테이블 관리 방식과 파일 관리 방식이 있다.

<표 2-17> 오류 코드 및 메시지 관리 방식 분류

관리 유형	사용 환경	예시
테이블	- 검색 및 참조, 생성 등 관리 용이 - 관리 대상 오류 코드 및 메시지가 많을 경우 (성능 우수) - DB 접근이 가능할 경우만 사용 가능	오류 코드 / 메시지 내용 EAD001 / 등록된 ○○○○○이(가) 존재합니다. : / :
파일	- 관리 대상 오류 코드 및 메시지가 적을 경우 - DB에 접근할 수 없을 경우에도 오류 코드 및 메시지를 참조해야 할 경우	[error message.properties 파일] EAD001=등록된 ○○○○○이(가) 존재합니다.

3. 오류 로그 테이블(파일) 설계

(1) 오류 로그 테이블(파일) 설계 방식

오류 로그 테이블(파일)은 기록 단위에 따라 연계(인터페이스) 테이블(파일)에 대한 로그(Log)와 연계 데이터(Row 또는 행)에 대한 로그(Log)로 설계할 수 있다.

오류 로그 테이블(파일)은 연계 메커니즘 송수신 과정의 전반적인 진행 사항을 모니터링하기 위한 로그(Log)를 기록하는 테이블(파일)과 분리하여 설계할 수도 있고 또는 통합하여 설계할 수도 있다.

※ 이후 연계 메커니즘 송수신 과정의 전반적인 진행 사항을 모니터링하기 위한 로그(Log)를 '로그(Log)' 로, 오류 내용을 확인하기 위한 기록은 '오류 로그(Error Log)' 로 명명함

(2) 오류 로그 테이블(파일)의 관리 정보

오류 로그 테이블(파일)을 로그(Log) 테이블(파일)과 분리 설계 또는 통합 설계하는 방식과는 관계없이 오류 데이터를 추적하여, 오류 원인을 정확하게 분석하고, 분석 결과에 따라 보완 후 재처리할 수 있도록 관련 정보를 관리한다.

【 오류 로그 테이블(파일)의 필수적 관리 및 설계 정보 】

① 오류 발생한 데이터가 포함된 로우(Row, 행)의 식별값(주 식별자, 일반적으로 Primary Key)

② 오류가 발생한 항목(칼럼)

③ 오류 코드

④ 오류 내용 등

① ~ ④ 외에도 처리 일시나 송수신 일시와 구분하여 오류 발생 일시, 연계 데이터의 담당자 정보 등 재작업이나 오류 발생의 선후 관계 또는 시간적 순서의 추적을 용이하게 하기 위한 부가적인 정보 등을 관리할 수 있다.

(3) 오류 로그 테이블(파일)의 설계 항목 및 방법

오류 로그 테이블(파일) 설계 시 상세 항목은 '1-2. 연계 데이터 식별 및 표준화 3항 연계 데이터 식별자와 변경 구분 추가 중 관리를 위한 정보' 를 참고한다.

오류 로그 테이블(파일) 설계 결과 산출물은 테이블 정의서 또는 파일 레이아웃(Layout) 정의서를 활용하여 작성한다.

오류 로그 테이블(파일)이 로그 테이블(파일)과 통합하여 설계하는 경우는 '2-1. 연계 메커니즘 정의 3항 인터페이스(I/F) 테이블 또는 파일 생성' 을 참고한다.

<표 2-18> 오류 로그 테이블(파일) 정의서 - 송신 시스템, 연계(인터페이스) 테이블(파일) 단위, 로그와 분리 설계

테이블 ID	TB_TBL_ERR_LOG	테이블명	연계 단위 오류 로그
데이터베이스명	연계 DB	테이블 스페이스명	IF_DATA_TS010
트리거 구성	없음		
테이블 설명	연계(인터페이스) 테이블 단위 오류 로그를 관리한다.		

초기 건수	증가량 (일)	보관 주기	최대 건수	용량	비고
0	20	월	3,000	50MB	최대 건수는 [일일 전송 주기 X 연계 (인터페이스) 수]로 산정함

칼럼명	칼럼 ID	타입 및 길이	Not Null	PK	IDX	제약 조건
생성 일시	CRE_DTTM	Date	Y	Y	IDX_TBL_ERR_LOG_1	
인터페이스명	IF_NM	VARCHAR (500)	Y	Y	IDX_TBL_ERR_LOG_1	
오류 발생 일시	ERR_OCCR_DTTM	Date	Y		IDX_TBL_ERR_LOG_2	
처리 단계	PROC_STEP	VARCHAR (2)	Y			공통 코드 (COMM_CD_10)
오류 코드	ERR_CD	VARCHAR (5)	Y			공통 코드 (COMM_CD_90)
오류 내용	ERR_CONT	VARCHAR (4,000)				
오류 건수	ERR_CNT	Number (10)	Y			
송신 일시	SEND_DTTM	Date	Y			
:	:	:	:	:	:	:

<표 2-19> 오류 로그(Log) 테이블(파일) 정의서 - 수신 시스템, 연계 데이터 단위, 로그와 분리 설계

테이블 ID	TB_DATA_ERR_LOG	테이블명	데이터 단위 오류 로그
데이터베이스명	연계 DB	테이블 스페이스명	IF_DATA_TS010
트리거 구성	없음		
테이블 설명	연계 데이터 단위 오류 로그를 관리한다.		

초기 건수	증가량 (일)	보관 주기	최대 건수	용량	비고
0	20	월	500,000	200MB	최대 건수는 일일 전송량으로 산정함

칼럼명	칼럼 ID	타입 및 길이	Not Null	PK	IDX	제약 조건
생성 일시	CRE_DTTM	Date	Y	Y	IDX_DATA_ERR_LOG_1	
인터페이스명	IF_NM	VARCHAR (500)	Y	Y	IDX_DATA_ERR_LOG_1	
오류 발생 일시	ERR_OCCR_DTTM	Date	Y	Y	IDX_DATA_ERR_LOG_2	
오류 데이터 PK	ERR_DATA_PK	VARCHAR (100)	Y	Y	IDX_DATA_ERR_LOG_1	
작업 구분	WORK_TYPE	VARCHAR (1)	Y			공통 코드 (COMM_CD_03)
오류 항목	ERR_DATA_COL	VARCHAR (30)	Y	Y	IDX_DATA_ERR_LOG_1	
처리 단계	PROC_STEP	VARCHAR (2)	Y			공통 코드 (COMM_CD_10)
오류 코드	ERR_CD	VARCHAR (5)	Y			공통 코드 (COMM_CD_90)
오류 내용	ERR_CONT	VARCHAR (4,000)				
이전 데이터 PK	PREV_DATA_PK	VARCHAR (100)				
송신 일시	SEND_DTTM	Date	Y			
:	:	:	:	:	:	:

수행 내용 / 연계 장애 및 오류 처리 구현하기

재료 · 자료

- 코드 정의서(송신 시스템, 수신 시스템)
- 송수신 절차도 또는 연계 메커니즘 정의서, 연계 아키텍처 설계서
- 연계(인터페이스) 정의서
- 연계(인터페이스) 명세서
- 인터페이스(I/F: Interface) 테이블(파일 Layout) 정의서
- 오류 코드 정의서
- 로그 테이블(파일 Layout) 정의서
- 오류 로그 테이블(파일 Layout) 정의서
- 화면 설계서(사용자 인터페이스 설계서)

기기(장비 · 공구)

- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

1 송수신 절차 또는 연계 메커니즘 정의서를 활용하여 오류 발생 기록 시점 및 기록 단위, 기록 방식 등을 결정하여 연계 메커니즘 정의서의 구현 방식과 메커니즘 결정 사항을 추가 작성한다.('<표 2-11> 연계 메커니즘 정의서(설계서) 양식' 참고)

1. '2-1. 연계 메커니즘 정의' 에서 작성한 연계 메커니즘 정의서를 참고하여, 송신 시스템과 수신 시스템 각각에서 아래 사항을 결정한다.

   일반적으로 송신 시스템과 수신 시스템의 오류 발생 기록 단위나 방식 등은 동일하게 설계하나 송수신 시스템의 운영 현황이나 연계 요구 사항에 따라 송수신 시스템을 상이하게 설계할 수 있다.

<표 2-20> 장애 및 오류 처리 구현을 위한 결정 사항

결정해야 할 사항	고려 요소
오류 발생 기록 시점	‘2-1. 연계 메커니즘 정의’ 에서 설계한 연계 메커니즘의 처리 단계별 오류 발생 기록 ① 송신 시스템: 운영 DB에서 연계 데이터 추출 및 생성 시점 ② 송신 시스템과 수신 시스템 중 결정: 결정한 코드 매핑이나 데이터 변환 시스템, 시점 적용 ③ 수신 시스템: 운영 DB에 데이터 반영 시점
오류 기록 단위	① 로그(Log) 테이블과 통합 vs 로그(Log) 테이블과 분리하여 설계 ② 연계(인터페이스) 테이블(파일) 전체 vs 개별 연계 데이터 ⇨ 트랜잭션(Transaction)과 연관된 사항
오류 기록 방식	테이블 vs 파일

2. 결정 사항은 ‘2-1. 연계 메커니즘 정의’ 에서 작성한 연계 메커니즘 정의서의 구현 방식과 결정 사항에 추가한다.

2 송수신 시스템의 발생 가능한 오류 내용을 식별하여 오류 코드를 정의 및 설계한다.

1. 송신 시스템과 수신 시스템의 연계 응용 프로그램에서 발생 가능한 장애 및 오류를 식별한다. (‘<표 2-16> 오류 코드 정의’ 참고)
2. ‘<표 2-15> 오류 코드 부여 규칙’ 을 참고하여 송신 시스템과 수신 시스템의 오류 코드를 ‘<표 2-16> 오류 코드 정의’ 의 형식으로 작성한다.
3. ‘<표 2-17> 오류 코드 및 메시지 관리 방식 분류’ 를 참고하여 송신 시스템과 수신 시스템의 오류 코드 관리 방식을 결정하고, ‘2-1. 연계 메커니즘 정의’ 에서 작성한 연계 메커니즘 정의서의 구현 방식과 결정 사항에 작성한다.

3 1에서 결정한 오류 로그 기록 방식에 따라 오류 로그 테이블(파일)을 정의 및 설계한다.

1. 송수신 시스템에서 관리할 연계(인터페이스) 단위와 연계 데이터 단위의 오류 관리 정보를 식별 및 정의한다.
   예) 오류 발생 일시, 오류 코드, 이전 데이터 주 식별자(PK: Primary Key) 등
2. 식별된 오류 관리 정보 중 공통 코드 관리 대상은 송수신 시스템의 코드 정의서에 코드를 부여하여 추가한다.
3. ‘<표 2-18> 오류 로그(Log) 테이블(파일) 정의서 – 송신 시스템, 연계(인터페이스) 테이블(파일) 단위, 로그와 분리 설계’ 와 ‘<표 2-19> 오류 로그(Log) 테이블(파일) 정의서 – 수신 시스템, 연계 데이터 단위, 로그와 분리 설계’ 를 참고하여 오류 로그(Log) 테이블(파일) 정의서를 작성한다.

4 2, 3에서 설계한 오류 코드 및 오류 로그(Log) 테이블을 활용하여 송수신 시스템의 오류 현황을 모니터링하는 화면을 설계한다.

화면 설계 시 아래의 예시 화면을 참고하고, '<표 2-21> 화면(사용자 인터페이스) 설계서 양식' 을 활용하여 작성한다.

【 예시 화면 】

① [그림 2-3] 송신 시스템 로그(Log) 테이블(파일)에 활용한 송신 현황 모니터링 화면

② [그림 2-4] 수신 시스템 로그(Log) 테이블(파일)에 활용한 수신 현황 모니터링 화면

③ [그림 2-8] 장애 및 오류 현황 모니터링 화면 - 연계(인터페이스) 테이블(파일) 단위

④ [그림 2-9] 장애 및 오류 현황 모니터링 화면 - 연계(인터페이스) 데이터 단위

<표 2-21> 화면(사용자 인터페이스) 설계서 양식

화면 ID		화면명	
화면 유형		메뉴 경로	
화면 개요			

화면

입출력 항목				
항목명 (한글명)	컨트롤명 (영문명)	타입 및 길이	속성	Validation Check
※ 속성 I: Input, O: Output, R: Read Only, E: Editable, H: Hidden				
처리 내용				
기술적 고려 사항				

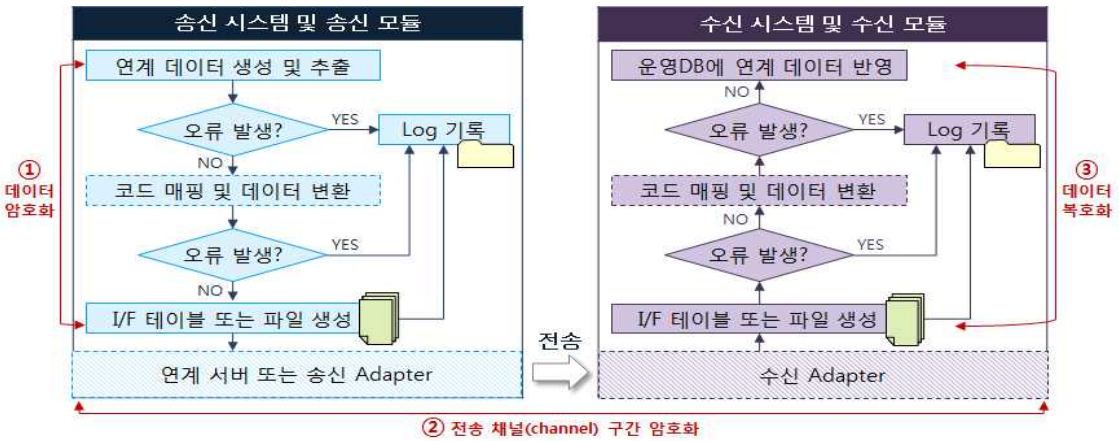
수행 tip

- 오류 기록 단위와 기록 방식 및 오류 코드는 연계 시스템과 연계 요구 사항에 대한 이해를 기반으로 사용자와 운영자 관점에서 정의한다.
- 오류 코드는 적용한 연계 솔루션의 제공 기능과 연계 데이터에 대한 이해가 우선되어야 정확하고 적합하게 정의할 수 있다.

2-3. 연계 데이터 보안 적용

학습 목표	• 응용 소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.

필요 지식 /



[그림 2-12] 연계 메커니즘에서 연계 데이터 보안

연계 데이터 보안은 전송 구간에서 암·복호화와 데이터의 암·복호화로 구현한다.

1 전송 구간(채널, Channel) 보안

전송 구간 암호화, 즉 채널 암호화는 네트워크에서 데이터가 전송되는 형식, 패킷(Packet)의 암·복호화로 네트워크에서 비인가자 또는 악의적인 사용자가 전송 데이터, 패킷을 가로채더라도 쉽게 전송 데이터의 내용을 파악하지 못하도록 한다.

전송 구간 암호화는 범위가 네트워크이므로 전송되는 전체 데이터에 적용된다.

1. 전송 구간 암호화

전공 구간 암호화를 위해서는 전송 구간 암호화를 지원하는 가상 사설망(VPN: Virtual Private Network) 또는 유사 솔루션이나 연계 솔루션을 적용해야 한다. 채널 보안 솔루션, 가상 사설망 또는 연계 솔루션에서 지원하는 보안 기능으로서 솔루션에 따라서 암·복호화 적용 여부와 적용 암호화 알고리즘을 선택 가능하며 별도로 구현하거나 변경할 수 있는 기능은 없다.

2. 전용선과 가상 사설망(VPN: Virtual Private Network)

전송 구간의 보안을 위해서는 송수신 시스템 간에 송수신 시스템만을 위한 개별 전송 회선인 전용선 설치와 병행하여 전송 구간 암호화 솔루션을 적용할 수 있다.

송수신 시스템 간의 거리 구간, 송수신 시스템의 네트워크 상이(인터넷 vs 인트라넷), 비용 등으로 인해 전용선 설치 및 이용이 어려울 경우 공중망의 회선을 사설망처럼 이용하는 가상 사설망을 활용한다. 가상 사설망에는 에스 에스 엘 브이 피 엔(SSL VPN), 아이피 섹 브이 피 엔(IPSec VPN) 등이 있으며 환경과 데이터의 중요성 등에 따라 선정하여 적용한다.

2 데이터 보안

연계 데이터 보안은 연계 데이터를 송신 시스템 운영 DB에서 연계 데이터를 추출하여 연계 솔루션에 의해 송신하기 전과 수신 시스템 연계(인터페이스) 테이블(파일)로부터 운영 DB에 반영하는 과정에서 데이터 암·복호화를 적용하는 방식이다.

데이터 보안은 암·복호화 적용 대상, 암호화 알고리즘, 암·복호화 적용을 위한 환경 설정을 설계하고 구현해야 한다.

1. 연계 메커니즘에서 데이터 암·복호화 처리 절차

송신 시스템에서는 운영 DB에서 연계 데이터를 추출하여 보안 대상 속성(칼럼)을 선정한 암호화 알고리즘으로 암호화한 후 연계(인터페이스) 테이블(파일)에 등록한다.

수신 시스템에서는 연계(인터페이스) 테이블(파일)에서 연계 데이터를 읽어서 암호화한 칼럼이 있을 경우 복호화하여 운영 DB에 반영한다.

송수신 시스템 간에 데이터 보안을 위해서는 연계 데이터 중 암·복호화 적용 속성(칼럼), 적용 암호화 알고리즘과 암호화 키(Key) 등을 협의하고 공유해야 한다.

송수신 시스템에는 암·복호화를 위한 동일한 암호화 알고리즘 라이브러리(Library)가 설치된다.

2. 암·복호화 적용 대상 선정

연계 데이터 암·복호화는 연계 메커니즘에서 송신 시스템의 암호화 과정, 수신 시스템의 복호화 과정이 추가됨으로 인해 암·복호화 속성(칼럼, Column) 수와 연계(인터페이스) 테이블(파일) 수가 증가할수록 시스템 성능은 비례하여 저하된다. 따라서, 적정 수준의 보안 적용 대상 칼럼(Column) 수, 연계(인터페이스) 테이블(파일) 수를 결정해야 한다.

<표 2-22> 보안 적용 대상 중요 정보

분류	적용 및 관련 법령	대상 정보	
개인 식별	『개인 정보 보호법』, 『정보 통신망법(정보 통신망 이용 촉진 및 정보보호 등에 관한 법률)』	주민 번호, 운전 면허 번호, 여권 번호, 장애인 관리 번호, 보훈 관리 번호	이름, 주소, 이메일, 휴대 전화 번호, 전화 번호 등
금융	『전자 금융 거래법』, 전자 금융 감독 규정(금융 감독원)	계좌 번호, 신용 카드 번호, 보험 계약 번호, 증권 번호 등	
:	:	:	

연계 데이터의 중요도에 따라 암・복호화 적용 속성(칼럼)을 선정한다.

중요도는 송수신 시스템에서 정의한 기준에 따라 상이할 수 있지만, 어떤 시스템에도 공통적인 보안 대상은 『개인 정보 보호법』에 의한 개인 식별 정보이다.

개인 정보는 살아 있는 개인을 식별할 수 있는 정보로, 두 가지 이상 정보가 결합되어 개인을 식별할 수 있다면 결합되는 정보들도 개인 정보로 분류될 수 있다.

중요도에 따라 필수 보안 적용 대상 개인 정보는 주민 번호, 운전 면허 번호, 장애인 관리 번호, 보훈 관리 번호 등이 해당된다. 개인 정보 외에 신용 카드 번호, 계좌 번호, 증권 번호 등 금융 관련 정보도 중요한 보안 대상 정보이다.

3. 암호화 알고리즘

암호화 알고리즘은 기준에 따라 다양하게 분류할 수 있다.

암호화 방향 분류로는 단방향과 양방향 암호화 알고리즘이 있다. 단방향 암호화 알고리즘은 암호화는 가능하나 복호화는 불가능한 알고리즘으로 비밀번호 등의 정보에 적용한다.

송수신 시스템 간의 동일 암호화 키(Key) 사용 여부에 따라 대칭 키 알고리즘과 비대칭 키 알고리즘으로 분류할 수 있다. 대칭 키 암호화 알고리즘은 송수신 시스템 간에 동일한 암호화 키(Key)를 공유하는 방식이며, 비대칭 키는 인증서 적용 시 공개 키 인프라스트럭처(PKI: Public Key Infrastructure)와 같이 송수신 시스템 간에 상이한 암호화 키(Key)를 사용하는 방식이다.

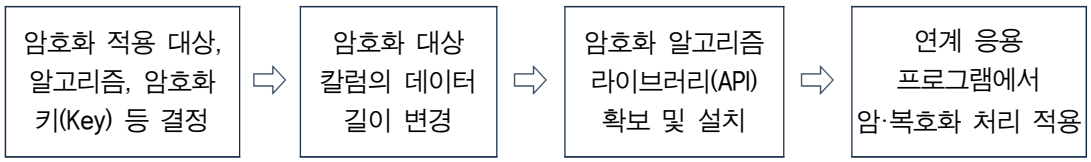
<표 2-23> 주요 양방향 대칭 키 암호화 알고리즘 유형

<table>
<tr><th>구분</th><th>알고리즘</th><th>상세 설명</th></tr>
<tr><td rowspan="2">국내</td><td>SEED-128,
SEED-256,</td><td>- 1999년 한국인터넷진흥원에서 개발된 블록 암호화 알고리즘
- 사용하는 암호화 키 길이에 따라 시드-128(SEED-128)과 시드-256(SEED-256)이 있음</td></tr>
<tr><td>ARIA</td><td>- Academy(학계), Research Institute(연구 기관), Agency(정부 기관)
- 경량화 및 하드웨어 환경에 최적화되도록 국가정보원에서 산학연과 같이 개발한 블록 암호화 알고리즘
- 암호화 블록 크기는 128비트이며, 암호화 키 길이에 따라 ARIA 128, 192, 256으로 분류됨</td></tr>
<tr><td rowspan="2">국외</td><td>AES128,
AES192,
AES256</td><td>- Advanced Encryption Standard
- 미국 표준 기술 연구소(NIS)에 의해 개발된 128비트의 블록 크기(암호화 처리 단위) 대칭형 알고리즘
- 사용하는 키 길이에 따라 128, 192, 256이 있음
<table>
<tr><th>구분</th><th>키 길이(단위: 비트)</th><th>블록 크기(단위: 비트)</th><th>라운드 수</th></tr>
<tr><td>AES-128</td><td>4</td><td>4</td><td>10</td></tr>
<tr><td>AES-192</td><td>6</td><td>4</td><td>12</td></tr>
<tr><td>AES-256</td><td>8</td><td>4</td><td>14</td></tr>
</table></td></tr>
<tr><td>DES,
3DES(Triple DES)</td><td>- Data Encryption Standard
- 1972년 미국 NIST 전신에 의해 개발된 64비트 암호화 단위와 56비트 암호화 키를 사용하는 비대칭키 암호화 알고리즘
- 쓰리 디 이 에스(3-DES), 트리플 디 이 에스(Triple DES)는 각 데이터 블록에 DES 알고리즘을 세 번 적용하여 보안을 강화한 암호화 알고리즘</td></tr>
</table>

일반적으로 연계 시에는 송수신 시스템 간에 양방향의 대칭 키 방식 암호화 알고리즘을 적용한다.

암·복호화 처리를 위해서는 송수신 시스템 간에 암호화 알고리즘을 선정하고, 결정한 암호화 알고리즘에서 사용할 암호화 키(Key)를 공유해야 한다.

4. 송수신 시스템 연계 응용 프로그램 구현 시 암호화 알고리즘 적용



[그림 2-13] 송수신 시스템 연계 응용 프로그램 구현 시 암호화 알고리즘 적용

암호화 적용 대상 속성(칼럼), 암호화 알고리즘, 암·복호화 시 사용할 암호화 키(Key) 등이 결정되면 결정 사항은 '2-1. 연계 메커니즘 정의' 에서 작성한 연계 메커니즘 정의서에 반영한다.

(1) 암호화 적용 대상 칼럼의 길이

'1-2. 연계 데이터 식별 및 표준화' 에서 작성한 연계(인터페이스) 정의서와 명세서, '2-2. 연계 메커니즘 정의' 에서 작성한 연계(인터페이스) 테이블(파일) 정의서의 암호화 적용 대상 칼럼 길이를 충분히 늘려 준다. 암호화된 값은 적용하는 암호화 알고리즘과 암호화 대상 속성(칼럼)의 길이에 따라 상이하지만 암호화 전의 평문 길이보다는 커진다.

예를 들면, 주민 번호 속성(칼럼)의 경우 평문(암호화를 하지 않은 데이터)의 길이는 13자리이나 암호화된 주민 번호는 암호화 전의 평문 길이보다 길어지므로 적용하는 암호화 알고리즘에 따라 50자리에서 500자리로 변경한다. 변경한 길이는 연계(인터페이스) 테이블(파일) 정의서와 명세서에 반영하여 현행화 한다.

(2) 암·복호화 알고리즘 적용 방식

암호화 알고리즘은 연계 솔루션이나 DB 암호화 솔루션 등 운영하는 송수신 시스템에서 사용하고 있는 암호화 알고리즘 또는 제공하는 암호화 알고리즘을 적용할 수 있다.

또는 시드(SEED), 아리아(ARIA)와 같이 국내 개발용 알고리즘을 별도로 적용하고자 할 경우에는 한국인터넷진흥원의 홈페이지에서 알고리즘 신청서를 작성하여 승인이 되면 제공받을 수 있다.

구글에서 제공하는 오픈 소스 암호화 에이 피 아이(API)인 Key Czar 등으로 적용하는 방법도 있다.

(3) 암·복호화 처리를 위한 시스템 준비 및 확인 사항

암·복호화 처리를 위해서는 결정한 알고리즘이 적용된 암호화 API(Application Programming Interface)와 복호화 API가 패키징(Packaging)된 라이브러리(Library)를 확보하여 연계 시스템(송수신 시스템, 중계 시스템 등)에 설치해야 한다.

연계 솔루션에서 제공하는 알고리즘이거나 이미 운영 시스템에서 적용되고 있는 암호화 알고리즘이 아닌, 별도로 확보한 암호화 알고리즘인 경우에는 확보한 라이브러리(Library, API)를 연계 시스템의 라이브러리(Library, API) 저장소 또는 폴더에 설치한다. 그럼으로써 연계 응용 프로그램에서 암·복호화 처리 시 암호화 API(Application Programming Interface)나 복호화 API를 참고할 수 있다.

(4) 응용 애플리케이션에서 암·복호화 API 참조 방법과 암호화 공통 컴포넌트

암호화 알고리즘 라이브러리에서 제공하는 암호화 에이 피 아이(API)와 복호화 에이 피 아이(API)를 접근하는 암호화 처리 공통 모듈과 복호화 처리 공통 모듈을 개발하여 연계 응용 프로그램에서 암·복호화 처리 시 참조한다. 공통모듈의 활용으로 송수신 시스템 간에 공유된 암호화 키(key)의 노출과 변경 시 수정 부분을 최소화할 수 있다.

수행 내용 / 연계 데이터 보안 적용하기

재료 · 자료

- 송수신 절차도 또는 연계 메커니즘 정의서, 연계 아키텍처 설계서
- 연계(인터페이스) 정의서
- 연계(인터페이스) 명세서
- 인터페이스(I/F: Interface) 테이블(파일 Layout) 정의서
- 암호화 라이브러리(API: Application Programming Interface)

기기(장비 · 공구)

- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

1 암호화 대상 속성(칼럼), 암호화 알고리즘, 암호화 키 등을 결정하여 '2-1. 연계 메커니즘 정의' 에서 작성한 연계 메커니즘 정의서에 구현 방식과 메커니즘 결정 사항을 추가 작성한다. ('<표 2-11> 연계 메커니즘 정의서(설계서) 양식' 참고)

<표 2-24> 데이터 암호화 적용을 위한 결정 사항

<table>
<tr><th>결정해야 할 사항</th><th>고려 요소</th></tr>
<tr><td>암호화 대상 속성(칼럼)</td><td>- 『개인 정보 보호법』 또는 금융 관련 정보의 속성(칼럼) 암호화
- ○○○○○ 연계(인터페이스) 테이블(파일)의 △△△△△ 칼럼
<table>
<tr><th>연계 테이블명</th><th>속성(칼럼)명</th><th>데이터 타입 및 길이</th><th>암호화 적용 여부</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>√</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>암호화 알고리즘</td><td>- 연계 솔루션에서 제공하거나 송수신 시스템에서 이미 적용되고 있는지 암호화 알고리즘이 있는지 확인 ⇨ 있을 경우, 제공되거나 적용되고 있는 암호화 알고리즘 적용(보안성, 성능 등)을 검토하여 결정
- 제공되거나 적용하고 있는 암호화 알고리즘이 없을 경우 보안성, 성능, 유지 보수성 등을 비교·분석하여 암호화 알고리즘 선정</td></tr>
<tr><td>암호화 키(Key)</td><td>- 성능에 민감하지 않을 경우 암호화 키는 결정한 암호화 알고리즘에서 지원하는 가장 최대 길이의 암호화 키 길이 적용</td></tr>
</table>

2 '1-2. 연계 데이터 식별 및 표준화' 에서 작성한 연계(인터페이스) 정의서 및 명세서와 '2-1. 연계 메커니즘 정의' 에서 작성한 연계(인터페이스) 테이블(파일 Layout) 정의서 중 1에서 선정한 암호화 대상 속성(칼럼)의 데이터 길이를 수정한다.

암호화 대상 속성(칼럼)의 데이터 길이는 1에서 선정한 암호화 알고리즘과 암호화 대상 속성(칼럼)의 암호화 전 평문 길이를 고려하여 조정한다.

【 3. 내외부 연계 모듈 구현하기 수행 이후 수행 내용 】

아래의 3, 4는 연계 환경 구축 및 연계 프로그램 구현 후 적용할 수 있는 부분이므로, '3. 내외부 연계 모듈 구현하기' 학습 이후에 수행하거나, '3-1. 연계 모듈 구현 환경 구성 및 개발' 학습 시 수행한다.

3 연계 환경에 1에서 선정한 암호화 알고리즘 라이브러리(API)를 확보하여 설치한다.

구축된 송수신 시스템 각각의 연계 시스템(중계 시스템 포함)에서 참조하는 라이브러리 저장소 또는 폴더에 확보된 암호화 알고리즘 라이브러리(API)를 설치(배치)한다.

연계 솔루션에서 제공되는 암호화 알고리즘이나 이미 운영 시스템에서 사용 중인 암호화 알고리즘을 적용할 경우에는 암호화 알고리즘을 설치할 필요가 없으며, 어떤 위치에 설치되어 있는지만 확인한다.

4 연계 응용 프로그램에서 선정된 암호화 적용 대상 속성(칼럼)을 암·복호화한다.

1. 송신 시스템의 연계 응용 프로그램에서 연계 데이터를 추출하여 연계(인터페이스) 테이블(파일)에 등록하는 부분에 암호화 적용 대상 속성(칼럼)은 암호화 API를 호출하도록 적용한다.
   암호화 API 호출이 정상적으로 실행되는지, 실행 결과 암호화된 값이 리턴(return) 되는지 확인한다.
2. 수신 시스템의 연계 응용 프로그램에서 연계(인터페이스) 테이블(파일)의 데이터를 운영 DB에 반영하는 부분에 암호화 적용 대상 속성(칼럼)은 복호화 API를 호출하도록 적용한다.
   복호화 API 호출이 정상적으로 실행되는지, 실행결과 복호화된 평문이 리턴(return) 되는지 확인한다.
3. 수신 시스템 운영 DB에 별도로 적용되는 암호화 대상 칼럼 및 암호화 알고리즘이 있을 경우, 송신 시스템 암호화 알고리즘으로 복호화하고, 수신 시스템 운영 DB 암호화 알고리즘으로 복호화하여 저장한다.

수행 주체	송신 시스템		수신 시스템		수신 시스템
수행 활동	암호화		복호화		암호화
적용 알고리즘	송수신 시스템간 공유된 암호화 알고리즘	⇨	송수신 시스템 간 공유된 암호화 알고리즘	⇨	수신 시스템 운영 DB의 암호화 알고리즘

[그림 2-14] 수신 시스템 운영 DB에 별도의 암호화 알고리즘이 적용된 경우

수행 tip

- 송수신 시스템 간의 암호화 알고리즘에서 사용하는 암호화 키(key) 관리를 위해서는 암호화 API와 복호화 API를 공통 컴포넌트로 구현하여 제공한다.
- 암호화 대상 속성(칼럼)을 식별 시 보안과 트래이드 오프(Trade-off) 관계에 있는 성능도 함께 확인해야 한다.
- 수신 시스템의 운영 DB에 연계 데이터를 반영하기 위한 연계 응용 프로그램에서는 운영 DB에 반영하는 부분에서 운영 DB에 적용된 암호화 대상 속성(칼럼)과 암호화 여부, 암호화 알고리즘 등을 확인하여 처리해야 한다.

학습 2 교수 · 학습 방법

교수 방법

- 연계 메커니즘과 연계 방식, 연계 메커니즘의 각 구성 요소별 기능 및 역할에 대해 개념도로 도식화하여 설명한다.
- 연계 메커니즘에서 연계 방식에 따라 연계 솔루션(엔진)에서 제공되는 기능과 별도로 구현해야 하는 기능을 제시하고, 적용하는 연계 솔루션(Solution)과 구현하는 연계 응용 프로그램의 연관 관계를 설명한다.
- 연계 솔루션은 실제 상용 제품 설명을 추가함으로써 학생들이 실무에서 활용되는 소프트웨어와 공통적으로 제공하는 서비스, 주로 적용하는 기능을 학습할 수 있도록 유도하며, 공개 소스 소프트웨어 사례도 제시한다.
- 연계 메커니즘 정의 시 결정되어야 하는 연계 데이터 생성 및 추출, 코드 매핑 및 데이터 변환, 연계(인터페이스) 테이블(파일) 설계, 로그(Log) 기록, 운영 DB에 연계 데이터 반영 방식 등은 도식화와 사례를 활용하여 설명한다.
- 실무에서 작성되었던 산출물인 연계 요구 사항 분석서, 연계 아키텍처 설계서, 송수신 시스템 운영 환경 구성도 등을 활용하여 연계 방식, 연계 주기 등을 결정하고, 연계 메커니즘에 반영하는 과정을 설명한다.
- 연계 메커니즘 정의서, 연계(인터페이스) 테이블((파일 Layout) 정의서, 오류 로그 테이블(파일) 등의 산출물 작성 및 활용 방법 설명 전에 데이터베이스 및 데이터 모델링의 개념과 구성 요소, 테이블의 개념과 구조, 데이터 등에 대한 학생들의 이해 · 지식 정도를 확인한다.
- 데이터베이스, 데이터 모델링에 대한 개념이 없거나 부족한 경우, 기본적인 사항에 대한 설명을 선행하여 진행한다.
- 실제 프로젝트에서 작성된 연계 메커니즘 정의서, 연계(인터페이스) 테이블((파일 Layout) 정의서, 오류 로그 테이블(파일) 등 관련 자료를 제시하고, 각 자료의 필요성과 활용 방법 등을 설명한다.
- 연계 메커니즘 정의 과정 및 산출물 작성 방법을 사례를 제시하고 설명한다.
- 연계 송수신 현황, 오류 현황을 모니터링하기 위한 정보 생성 방법, 정보 관리 방식, 정보 확인 방식 등을 사례를 들어 설명한다.

- 연계 송수신 현황, 오류 현황 모니터링 방식을 설명 시 화면 설계에 대한 학생들의 이해 및 지식 수준을 확인한다.
- 연계 데이터 보안 학습 내용 진행에 선행하여 데이터의 중요성을 판단할 수 있는 여러 가지 기준을 설명한다.
- 연계 메커니즘에서 보안이 적용되는 구간과 적용 구간별 방식에 대해 사례를 들어 설명한다.
- 연계 데이터 보안을 위한 대상 선정, 암·복호화 처리 등 보안 적용절차, 적용 절차별 실제 수행하는 작업 등을 설명한다.

학습 방법

- 연계 메커니즘 정의 과정에서 수행해야 할 작업, 과정별 수행 작업의 필요성, 작업 결과 작성되는 산출물에 대해 숙지한다.
- 수행 내용에서 제시된 시스템 및 운영 환경, 연계 요구 사항 등 주어진 조건에 적합한 연계 방식 및 연계 주기를 결정하고 연계 메커니즘을 정의하는 실습을 한다. 실습 수행 결과로 연계 메커니즘 정의서를 편집기를 활용하여 작성한다.
- 연계 메커니즘 정의서의 작성 시점, 작성 목적, 작성 항목, 활용 방법 등을 이해하고 연계 메커니즘 정의서를 작성하는 실습을 한다.
- 연계(인터페이스) 테이블(파일 Layout) 정의서의 작성 시점, 작성 목적, 작성 항목, 활용 방법 등을 이해하고, 연계(인터페이스) 테이블(파일 Layout) 정의서를 작성하는 실습을 한다.
- 로그 테이블(파일 Layout) 정의서 및 오류 로그 테이블(파일 Layout) 정의서의 관리 정보, 데이터 생성 시점, 활용 방법 등을 이해하고 로그 테이블(파일 Layout) 정의서 및 오류 로그 테이블(파일 Layout) 정의서를 작성하는 실습을 한다.
- 활용 목적에 따라 로그 테이블(파일 Layout) 정보 및 오류 로그 테이블(파일 Layout) 정보를 활용할 수 있는 화면을 편집기 또는 에이치 티 엠 엘(HTML: Hyper Text Markup Language) 에디터(Editor)를 사용하여 설계한다.
- 데이터의 중요도에 따라 보안 대상 연계 데이터 항목을 선정하고, 제시된 환경과 시스템 등의 특성과 제약 조건을 적용하여 데이터 중요도 기준을 정의한다.
- 정의한 보안 적용 대상 연계 데이터의 보안 적용 방식과 절차를 이해하고, 연계 메커니즘 정의서에 작성한다.

학습 2 평 가

평가 준거

- 평가자는 학습자가 수행 준거 및 평가 시 고려 사항에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.

- 평가자는 다음 사항을 평가한다.

학습 내용	평가 항목	성취수준		
		상	중	하
연계 메커니즘 정의	- 개발하고자 하는 응용 소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다.			
	- 개발하고자 하는 응용 소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구 사항을 고려하여 연계 주기를 정의할 수 있다.			
연계 장애 및 오류 처리 구현	- 개발하고자 하는 응용 소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리 방안을 정의할 수 있다.			
연계 데이터 보안 적용	- 응용 소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.			

평가 방법

- 평가자 체크 리스트

학습 내용	평가 항목	성취수준		
		상	중	하
연계 메커니즘 정의	- 연계 메커니즘의 이해			
	- 연계 메커니즘의 구성 요소별 기능 및 역할 이해			
	- 연계 방식과 연계 주기를 반영한 연계 메커니즘 정의서 작성			
	- 연계(인터페이스) 테이블(파일 Layout) 정의서, 로그테이블(파일 Layout) 정의서의 작성과 활용 방법 이해 및 응용			
연계 장애 및 오류 처리 구현	- 장애 및 오류 처리 절차 및 필요 요소의 이해			
	- 오류 로그 테이블(파일 Layout) 정의서의 작성과 활용 방법 이해 및 응용			
	- 오류 처리를 위한 정보의 활용 및 응용			
연계 데이터 보안 적용	- 연계 데이터 보안 적용 과정과 방법의 이해 및 응용			
	- 전송 구간 암호화와 데이터 암호화의 차이 구분			
	- 보안 적용 대상 데이터 식별 및 정의			
	- 암호화 알고리즘의 선정과 암·복호화 처리 과정을 연계 메커니즘에 반영			

● 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
연계 메커니즘 정의	- 연계 메커니즘의 이해 및 정의			
	- 연계 메커니즘의 구성 요소별 기능의 이해 및 응용			
	- 연계 메커니즘 정의서의 완전성, 결정 사항의 적합성			
	- 연계(인터페이스) 테이블(파일 Layout) 정의서, 로그 테이블(파일 Layout) 정의서 작성의 완전성, 일치성			
연계 장애 및 오류 처리 구현	- 장애, 오류 처리 절차와 필요 요소 이해 및 정의			
	- 오류 로그 테이블(파일 Layout) 정의서 작성의 완전성, 적합성, 일치성			
	- 오류 현황 모니터링 화면 설계의 적합성, 사용자 편의성, 활용 적합성, 오류 정보 추적의 완전성			
연계 데이터 보안 적용	- 연계 데이터 보안 적용 과정과 방법의 이해 및 응용			
	- 보안 적용 데이터 식별의 적합성, 식별 근거의 타당성			
	- 암호화 알고리즘 선정의 적합성, 선정 근거의 타당성			
	- 연계 메커니즘에 연계 데이터 보안 과정 적용의 완전성, 일관성, 적합성			

피 드 백

1. 체크리스트(Checklist)를 통한 관찰
   - 연계 메커니즘 정의서 작성 실습 과정에서 연계 메커니즘 개념, 연계 메커니즘 구성 요소별 기능 및 역할의 이해, 연계 방식과 연계 주기를 반영하여 연계 메커니즘 정의서를 작성 및 활용 방법에 대한 이해를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 오류 로그 테이블(파일 Layout) 정의서 작성과 오류 현황 모니터링 화면 설계 실습 과정에서 장애와 오류 처리를 구분하고, 오류 처리를 위한 관리 정보, 생성 시점, 관리 방식에 대한 이해, 오류 로그 테이블(파일 Layout)의 정보를 활용하는 방법에 대한 이해와 응용 정도를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 보안 적용 데이터 대상과 적용 암호화 알고리즘을 선정하는 실습 과정에서 연계 메커니즘의 보안 적용 절차, 보안 적용 대상 데이터 식별, 암호화 알고리즘 선정 및 적용에 대한 이해, 연계 메커니즘 정의서에 반영하는 방법, 반영 시점에 대한 이해를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.

2. 포트폴리오(Portfolio)
   - 제출한 연계 메커니즘 정의서의 충분성, 완전성, 적합성, 일관성, 구체성을 평가한 후에 필수 작성 항목 포함 여부 점검, 정의 및 설계 내용의 구체성 정도, 연계 메커니즘 구성 요소별 결정 사항에 대한 역할 대비 적합성, 결정 사항의 근거 타당성, 시스템 운영 환경과 연계 요구 사항 대비 연계 방식 및 연계 주기의 적합성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 제출한 오류 로그 테이블(파일 Layout) 정의서의 충분성, 완전성, 적합성, 일관성, 구체성을 평가한 후에 장애 및 오류 현황 모니터링 목적 달성을 위해 설계한 관리 정보의 적합성, 충분성, 구체성 및 일관성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 제출한 오류 현황 모니터링 화면 설계의 충분성, 완전성, 적합성, 일관성, 구체성을 평가한 후에 화면에서 필요 정보의 노출성, 오류 데이터 추적의 적합성, 구체성 및 일관성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 연계 데이터 보안 적용 학습 후에 보안 적용 사항을 반영하여 제출한 연계 메커니즘 정의서의 보안 적용 대상 및 암호화 알고리즘 선정의 충분성, 적합성과 연계 메커니즘 정의서 반영의 적합성, 완전성, 구체성 등을 평가한 후에 필수 보안 대상 항목 포함 여부 점검, 보안 적용 대상 및 암호화 알고리즘 선정 시 근거의 타당성, 연계 메커니즘 반영 내용의 구체화 정도 등 주요 사항에 대하여 표시하여 돌려준다.

학습 1	연계 데이터 구성하기
학습 2	연계 메커니즘 구성하기

학습 3 내외부 연계 모듈 구현하기

3-1. 연계 모듈 구현 환경 구성 및 개발

학습 목표	• 구성된 연계 메커니즘에 대한 명세서를 참조하여 연계 모듈 구현을 위한 논리적, 물리적 환경을 준비할 수 있다. • 구성된 연계 메커니즘에 대한 명세서를 참조하여 외부 시스템과 연계 모듈을 구현할 수 있다.

필요 지식 /

연계 모듈 구현 환경은 '2. 연계 메커니즘 구현하기'에서 설계된 연계 메커니즘에 의해 결정된다. 연계 메커니즘 구성 요소별 고려 요소와 결정에 따라 연계 메커니즘이 정의되고, 설계된 연계 메커니즘에 따라 연계 시스템 환경 및 모듈 구현 환경을 구성한다.

연계 메커니즘 정의와 고려 요소 및 결정 사항, 연계 메커니즘 정의서는 '2-1. 연계 메커니즘 정의'를 참고한다.

【 학습 모듈의 연계 구현 환경 】

다양한 연계 모듈 구현 환경 중 이번 학습 모듈에서는

① 데이터베이스 관리 시스템(DBMS: Database Management System) 오브젝트(Object)인 트리거(Trigger)와 연계(인터페이스) 테이블의 연계 모듈 구현 환경

② 데이터 추출을 위한 배치 응용 프로그램과 연계(인터페이스) 파일의 연계 모듈 구현 환경 구성에 대해 설명한다.

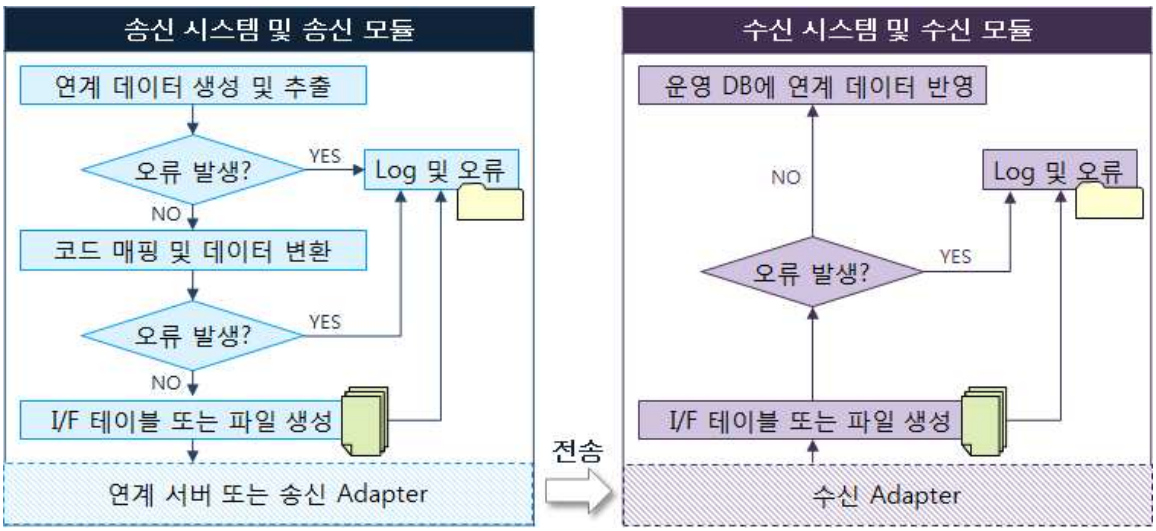
이후 학습 내용부터는 ①번 트리거(Trigger)와 연계(인터페이스) 테이블의 실시간 처리 연계 모듈 구현 환경은 EAI/ESB(Enterprise Application Integration)/(Enterprise Service Bus) 방식으로 명명한다. ②번 배치(Batch) 응용 프로그램(Program)과 연계(인터페이스) 파일의 주기적 처리 연계 모듈 구현 환경은 웹서비스(Web Service) 방식으로 축약하여 명명한다.

①의 구현 환경을 EAI/ESB 방식으로, ②의 구현 환경을 웹 서비스(Web Service) 방식으로 축약하여 명명한 기준은 각각이 실제 EAI/ESB 방식과 웹 서비스(Web Service) 방식에 근접하기 때문이다.

‘<표 3-1> 학습 내용의 연계 모듈 구현 환경을 위한 결정 사항’ 에서 ①, ②로 결정된 사항에 의한 연계 메커니즘은 ‘[그림 3-1] 연계 모듈 구현 환경을 위한 결정 사항을 적용한 연계 메커니즘’ 과 같다.

<표 3-1> 학습 내용의 연계 모듈 구현 환경을 위한 결정 사항

연계 데이터 추출 및 생성 방식		연계(인터페이스) 방식		오류 로그 관리 방식		운영 DB에 연계 데이터 적용 방식		코드 매핑 및 데이터 변환 위치	
DBMS 오브젝트(Object)	①	테이블	①	로그 및 연계(인터페이스) 테이블과 통합	①, ②	DBMS 오브젝트	①	송신 시스템	①, ②
추출 프로그램 구현(Batch)	②	파일	②	로그 및 연계(인터페이스) 테이블과 분리		추출 프로그램 구현(Batch)	②	수신 시스템	
운영 응용 프로그램에서 생성		-		-		-		중계 서버	
-								송수신 시스템에서 각각 변환	



[그림 3-1] <표 3-1> 연계 모듈 구현 환경을 위한 결정 사항을 적용한 연계 메커니즘

1 EAI/ESB 방식

EAI/ESB 방식은 ‘<표 3-1> 학습 내용의 연계 모듈 구현 환경을 위한 결정 사항’ 에서 ①로 결정된 사항을 적용하는 연계 방식 및 연계 모듈 구현 환경이다.

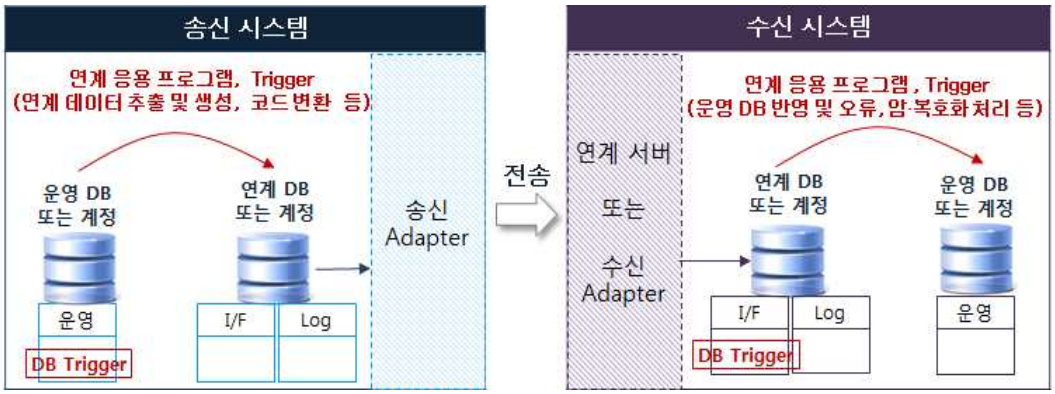
EAI/ESB 방식 연계 모듈 구현 환경은 ‘[그림 3-2] EAI/ESB 방식 연계 모듈 구현 환경’ 과 같다.

트리거(Trigger)를 자바(Java)와 같은 프로그램 언어(Program Language)를 사용한 응용 애플리케이션으로 구현하거나, 연계(인터페이스) 테이블을 연계(인터페이스) 파일로 변경할 수 있다.

1. EAI/ESB 솔루션 도입 시 연계 모듈 구현 환경 변경 사항

EAI/ESB 솔루션을 도입하여 연계 모듈 구현 환경을 구성할 경우, 연계(인터페이스) 데이터 처리를 위해 DB 테이블을 사용할 경우에는 연계 방식을 테이블 방식, 파일을 사용할 경우에는 파일 방식이라고 한다. 테이블 방식 또는 파일 방식과 무관하게 전체적인 연계 아키텍처 및 연계 모듈 구현 환경 구성은 동일하다.

EAI/ESB 솔루션에서는 데이터의 실제적인 전송과 연계(인터페이스) 테이블의 전송 형식으로 변환 및 전송 이력을 기록(Logging)하므로 전송이나 전송 이력을 기록(Logging)하는 기능을 구현할 필요는 없다. 연계 응용 애플리케이션에서 데이터 변환이나 운영 DB에 반영 시 발생하는 데이터 오류에 대한 사항만 기록(Logging)하는 기능을 구현하면 된다.



[그림 3-2] EAI/ESB 방식 연계 모듈 구현 환경 개념도

2. 연계 모듈 구현 환경 구축 절차

(1) 연계 DB 또는 계정 생성

송신 시스템과 수신 시스템에서 각각 연계를 위한 DB를 설치하고, DB 사용자 계정을 생성한다. 송신 시스템의 연계 사용자 계정명과 수신 시스템의 연계 사용자 계정명이 동일하지 않아도 된다. 기존에 연계를 위한 DB가 운영 중인 경우, 연계 목적에 따라 분리한 연계 사용자 계정을 생성한다.

운영 DB의 사용자와 연계를 위한 사용자는 생성되는 스키마(Schema)의 목적이 다르므로 물리적 및 논리적으로 분리하여 관리한다.

(2) 연계를 위한 테이블 생성

위의 (1)항에서 생성한 연계 DB 사용자 계정으로 연계를 위한 데이터 테이블인 연계(인터페이스) 테이블, 송수신 이력 및 오류 확인을 위한 로그 테이블, 코드 매핑 및 변환을 위한 매핑 테이블, 오류 처리를 위한 오류 코드 테이블 등을 생성한다.

【 연계를 위한 테이블 생성 위치 및 대상 】

연계를 위한 데이터 및 로그 테이블, 오류 코드 테이블은 송신 시스템과 수신 시스템 각각 생성한다. 테이블명이 동일하더라도 송신 시스템과 수신 시스템에서 관리하는 정보, 처리하는 정보가 상이하므로 테이블에 포함된 관리 정보를 확인한 후 송신 시스템과 수신 시스템에 적합하게 생성한다.

【 연계를 위한 테이블 생성 구조 】

생성할 테이블의 상세 구조는 '학습 1-2. 연계 데이터 식별 및 표준화' 에서 설계한 연계(인터페이스) 테이블(파일 Layout) 정의서 및 명세서' 와 '학습 2-1. 연계 메커니즘 정의' 에서 설계한 로그(Log) 테이블(파일 Layout) 정의서를, '학습 2-2. 연계 장애 및 오류 처리 구현' 에서 설계한 오류 코드 테이블 정의서 내용으로 생성한다.

향후 운영과 유지 보수를 위해서 테이블을 생성할 때는 데이터베이스 오브젝트(Object)별 명명 규칙에 따라 테이블명을 부여한다.

<표 3-2> 연계 DB 사용자 계정에 생성할 테이블

생성 위치	대상 테이블	포함 정보 및 용도	산출물명	참고 위치
송수신 시스템	인터페이스(I/F) 테이블	연계 데이터와 로우(Row) 또는 속성(칼럼) 단위 오류 확인을 위한 관리 정보	연계(인터페이스) 테이블 정의서	학습 2-1. 연계 메커니즘 정의
	로그 테이블	송수신 이력, 연계(인터페이스) 단위 오류 확인을 위한 관리 정보	로그 테이블(파일 Layout) 정의서	학습 2-1. 연계 메커니즘 정의, 학습 2-2. 연계 장애 및 오류 처리 구현
	오류코드 테이블	송수신 시스템에서 송수신이나 데이터 처리 시 발생한 오류 코드 및 내용	오류 코드 테이블(파일 Layout) 정의서	학습 2-2. 연계 장애 및 오류 처리 구현
	코드 테이블	진행 단계와 같은 공통 코드 정보(필요시 작성)	코드 정의서	학습 1-1. 연계 요구 사항 분석
송신 시스템	코드 매핑 테이블	송신 시스템의 코드를 수신 시스템의 코드로 변환하기 위한 매핑 정보	코드 매핑 내역서	학습 1-2. 연계 데이터 식별 및 표준화

(3) 연계 응용 프로그램 구현

연계 응용 프로그램은 송신 시스템과 수신 시스템에서 각각 구현한다.

【 송수신 시스템의 연계 응용 프로그램 구현 내용 】

송신 시스템의 연계 응용 프로그램은 운영 DB에서 데이터를 추출하고, 송신 시스템의 코드를 수신 시스템의 코드로 변환하는 작업을 처리한다.

수신 시스템의 연계 응용 프로그램은 연계(인터페이스) 테이블의 데이터에 대해 유효성(Validation)을 점검(Check)하고, 운영 DB에 반영하는 작업을 처리한다.

【 연계 응용 프로그램 구현 방식 】

학습 내용인 EAI/ESB 연계 방식에서 연계 응용 프로그램 구현 방식으로는 데이터베이스 관리 시스템(DBMS: Database Management System)의 오브젝트(Object)인 트리거(Trigger)를 활용한다. 트리거를 활용 시에는 자바(Java)와 같은 프로그램 언어(Program Language)로 구현된 암호화 알고리즘을 호출하는데 제한적이다. 연계 데이터 보안 적용을 위해 암호화 알고리즘을 적용할 경우에는 프로그램 언어(Program Language - Java, C 등)를 이용하여 연계 응용 프로그램을 구현한다.

트리거(Trigger)는 송신 시스템의 운영 DB의 테이블과 수신 시스템의 연계(인터페이스) 테이블을 대상으로 구현하며, 테이블 단위로 구현된다.

<표 3-3> 연계 응용 프로그램을 트리거(Trigger)로 구현 시 트리거(Trigger) 생성 위치

생성 위치	생성 테이블	처리 내용
송신 시스템	운영 DB의 테이블	운영 DB에서 데이터 추출, 코드 및 데이터 변환, 오류 및 데이터 유효성(Validation) 처리, 필요시 암호화 처리 등
수신 시스템	연계(인터페이스) 테이블	운영 DB에 데이터 반영, 오류 및 데이터 유효성(Validation) 처리, 필요시 복호화 처리

```
CREATE OR REPLACE TRIGGER [송신 응용 프로그램명]
AFTER INSERT OR UPDATE OR DELETE ON [연계 데이터를 추출 할 테이블명] FOR EACH ROW
  -- 연계 데이터를 추출할 테이블의 각 행에 insert, update, delete가 발생할 때마다 수행됨

DECLARE
  af_chg_code VARCHAR2(5);      -- 사용할 변수 선언

BEGIN
  IF INSERTING THEN             -- 추가된 경우
    -- 공통 코드 및 데이터 변환 처리
    SELECT [변환 후 코드값]
    INTO     af_chg_code
    FROM    [코드 매핑 테이블]
    WHERE  [변환 전 코드값] = :new.[코드 1 칼럼]

    -- 진행 단계별 수행 결과에 대한 Logging 처리, Log 테이블에 처리 결과 Insert
    INSERT INTO [로그(Log) 테이블명] ( [칼럼명 1], [칼럼명 2], ... [칼럼명 n] ) VALUES ( ... );


    -- 연계(인터페이스 테이블)에 insert 처리, 변경 구분을 'I' 로 Setting
    INSERT INTO [연계(인터페이스) 테이블명]
            ( [칼럼명 1 - 처리 일시], [칼럼명 2 - 변경 구분], ... [칼럼명 n] )
    VALUES ( SysDate, 'I' , af_chg_code, ... [:new.칼럼명] );

    -- 진행 단계별 수행 결과에 대한 Logging 처리, Log 테이블에 처리결과 Insert
  ELSIF DELETING THEN
    -- 공통 코드 및 데이터 변환 처리
    -- 연계(인터페이스 테이블)에 insert 처리, 변경 구분을 'D' 로 Setting
    -- 진행 단계별 수행 결과에 대한 Logging 처리, Log 테이블에 처리 결과 Insert
  ELSE
    IF UPDATING( '[코드 1 칼럼]' ) OR UPDATING( '[코드 2 칼럼]' ) THEN
      -- 연계 데이터 중 코드 칼럼이 변경된 경우
      -- 연계(인터페이스 테이블)에 insert 처리, 변경 구분을 'U' 로 Setting
      -- 진행 단계별 수행 결과에 대한 Logging 처리, Log 테이블에 처리 결과 Insert
    ELSIF UPDATING( '[텍스트 칼럼]' ) THEN -- 연계 데이터 중 텍스트 칼럼이 변경된 경우
    ELSIF UPDATING THEN                    -- 연계 데이터가 아닌 칼럼이 변경된 경우
    END IF;
  END IF;

  EXCEPTION                         -- 오류 처리
  WHEN NO_DATA_FOUND THEN       -- 등록된 코드에 매핑하는 코드가 없을 경우
    -- 오류 로그 테이블에 오류 코드 및 오류 내용 INSERT
  WHEN DUP_VAL_ON_INDEX THEN  -- 연계(인터페이스) 테이블 등록 시 동일한 PK의 데이터 존재함
    -- 오류 로그 테이블에 오류 코드 및 오류 내용 INSERT
END;
```

[그림 3-3] 트리거(Trigger)로 구현한 송신 시스템의 연계 응용 프로그램

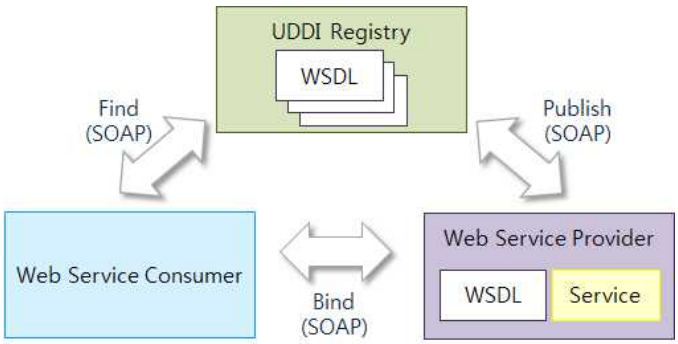
송신 시스템에서 연계 데이터를 추출하거나 수신 시스템에서 운영 DB에 반영할 시에는 연계(인터페이스) 테이블 간 데이터의 생성 순서 및 반영 순서, 즉 선후 관계를 확인하여 처리해야 한다.

예를 들어, 접수 테이블과 처리 결과 테이블의 내용이 연계될 경우에 발생의 선후 관계를 반영하여 접수 테이블의 데이터는 처리 결과 테이블의 데이터보다 선행하여 생성 및 적용해야 한다. 선후 관계가 반대로 처리된다면, 접수되지도 않은 처리 결과가 등록되거나 또는 처리 결과 테이블의 데이터는 반영되지 않고 오류로 처리될 수 있다.

2 웹 서비스(Web Service) 방식

1. 웹 서비스(Web Service) 기본 구조 및 구성 요소

웹 서비스(Web Service)란 네트워크에 분산되어 있는 정보를 서비스 형태로 개방하여 표준화된 방식으로 공유하는 기술로서 서비스 지향 아키텍처(SOA: Service Oriented Architecture) 개념을 실현하는 대표적인 기술이다.



[그림 3-4] 웹 서비스(Web Service) 기본 구조

(1) 솝(SOAP: Simple Object Access Protocol)

솝(SOAP)은 에이치 티 티 피(HTTP), 에이치 티 티 피 에스(HTTPS), 에스 엠 티 피(SMTP) 등을 사용하여 엑스 엠 엘(XML: Extensible Markup Language) 기반의 메시지를 네트워크 상에서 교환하는 프로토콜(Protocol)이다.

솝(SOAP)은 웹 서비스에서 메시지를 전달하는 기반으로 에이치 티 티 피(HTTP) 프로토콜(Protocol) 상에 솝 엔 벨 로프(SOAP Envelope), 헤더(Header), 바디(Body) 등이 추가된 엑스 엠 엘(XML) 문서로, 기본적인 송수신은 에이치 티 티 피(HTTP)를 통해서 이루어진다.

솝(SOAP)의 복잡하고 무거운 구조로 인해 요즘은 웹 서비스에서 솝(SOAP)보다는 레스트 풀(RESTful: REpresentational State Transfer) 프로토콜(Protocol)을 이용하기도 한다.

(2) 유 디 디 아이(UDDI: Universal Description, Discovery and Integration)

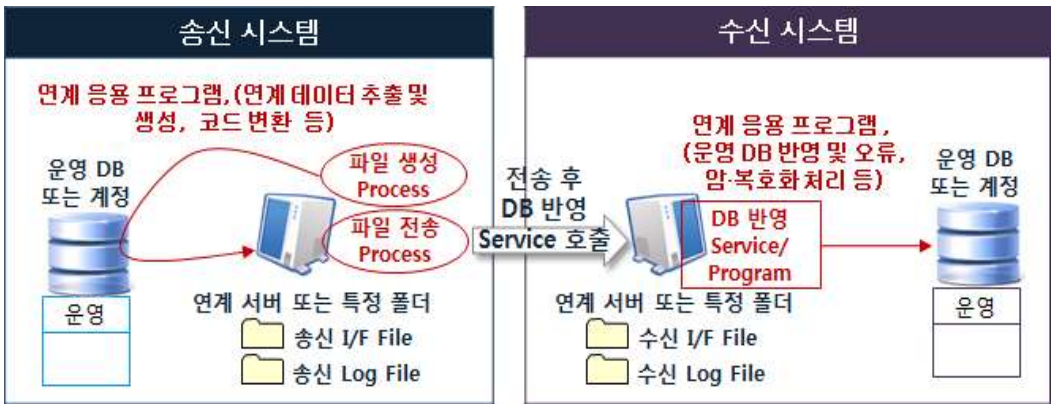
유 디 디 아이(UDDI)를 웹 서비스에 대한 정보인 더블유 에스 디 엘(WSDL)을 등록하고, 검색하기 위한 저장소로 공개적으로 접근, 검색이 가능한 레지스트리이다. 검색 엔진처럼 유 디 디 아이(UDDI)에서 웹 서비스 정보를 검색하여 사용한다. 알려진 송수신 시스템의 통합 구현에서는 서비스의 공개 및 검색과정이 필요 없으므로, 유 디 디 아이(UDDI)를 구축하지 않는다.

(3) 더블유 에스 디 엘(WSDL: Web Service Description Language)

더블유 에스 디 엘(WSDL)은 웹 서비스명, 웹 서비스 제공 위치, 웹 서비스 메시지 포맷(Format), 프로토콜(Protocol) 정보 등 웹 서비스에 대한 상세 정보를 기술한 파일로 엑스 엠 엘(XML) 형식으로 구현한다.

2. 연계 모듈 구현 환경 구축 절차

웹 서비스(Web Service) 방식은 EAI/ESB 방식과는 달리 실제 전송이나 전송 이력에 대한 기록(Logging) 및 모니터링 기능을 별도로 구현해야 한다.



[그림 3-5] 웹 서비스(Web Service) 방식 연계 모듈 구현 환경 개념도

(1) 송수신 파일 생성 위치 및 파일명 정의

송신 시스템에서는 송신 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일이 생성될 위치와 파일명을 정의한 후 디렉토리(Directory, 폴더)를 생성한다.

수신 시스템에서는 수신 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일이 생성될 위치와 파일명을 정의한 후 디렉토리(Directory, 폴더)를 생성한다.

파일 위치 및 파일명 명명 규칙에 대한 상세 내용은 '2-1. 연계 메커니즘 정의 중 4항 로그(Log) 기록' 을 참고한다.

<표 3-4> 송수신 파일 생성 위치 및 파일명

시스템	파일 생성 위치	파일명
송신 시스템	/IF/send/data/ 년도(YYYY)/월(MM)/일(DD) 예) /IF/send/data/2015/09/20	연계 파일명_처리 시점 시분(HH24MI).xml 예) EMPL_NOTC_INFO_1010.xml
	/IF/send/log /연도(YYYY)/월(MM)/일(DD) 예) /IF/send/log/2015/09/20	- 시스템 로그(시스템 Error Log 포함) : 구분자_년월일 시간(YYYYMMDDHH).log 예) system_log_2015092010.log - 연계 파일 단위 로그 : 연계 파일명_처리 시점 시분(HH24MI).log 예) EMPL_NOTC_INFO_1010.log 또는 LOG_EMPL_NOTC_INFO_1010.log
	/IF/send/error/ 연도(YYYY)/월(MM)/일(DD) 예) /IF/send/error/2015/09/20	연계 파일명_오류 발생 시점 시분(HH24MI).err 예) EMPL_NOTC_INFO_1030.err 또는 ERR_EMPL_NOTC_INFO_1030..log
수신 시스템	/IF/recv/data/ 년도(YYYY)/월(MM)/일(DD)	연계 파일명_송신 시스템 처리 시점 시분(HH24MI)_수신 시스템 처리 시점 시분(HH24MI).xml 예) EMPL_NOTC_INFO_1010_1015.xml
	/IF/recv/log/ 연도(YYYY)/월(MM)/일(DD)	- 시스템 로그(시스템 Error Log 포함) : 구분자_년월일 시간(YYYYMMDDHH).log 예) system_log_2015092010.log - 연계(인터페이스) 파일 단위 로그 : 연계(인터페이스) 송신 시스템 처리 시점 시분(HH24MI)_수신시스템 처리 시점 시분(HH24MI).log 예) EMPL_NOTC_INFO_1010_1015.log 또는 LOG_EMPL_NOTC_INFO_1010_1015.log
	/IF/recv/error/ 연도(YYYY)/월(MM)/일(DD)	연계 파일명_오류 발생 시점 시분(HH24MI).err 예) EMPL_NOTC_INFO_1030.err 또는 ERR_EMPL_NOTC_INFO_1030..log

(2) 송신 시스템 연계 응용 프로그램 구현

송신 시스템에서 운영 DB로부터 연계 데이터를 추출하여 송신 파일을 XML 형식으로 생성하는 자바 프로그램(Java Program)을 구현한다.

자바(Java)로 구현되는 연계 응용 프로그램에서는 송신 시스템의 운영 DB에 연결(Connection)을 생성하고, 운영 DB로부터 데이터를 추출(Select)하여, 코드 매핑 및 데이터 변환을 수행한 후 연계(인터페이스) 파일을 생성(File Write)한다.

송신용 연계 응용 프로그램에서 구현하는 내용은 '[그림 3-3] 트리거(Trigger)로 구현한 송신 시스템의 연계 응용 프로그램 예시'를 참고한다.

구현된 서비스는 주기적으로 실행하기 위해서는 운영체제(OS: Operating System)의 크론(Cron) 또는 데이터베이스 관리 시스템(DBMS: Database Management System)의 잡 스케줄러(Job Scheduler)에 등록한다.

(3) 파일 전송 및 수신 시스템의 DB 반영 서비스 호출 처리

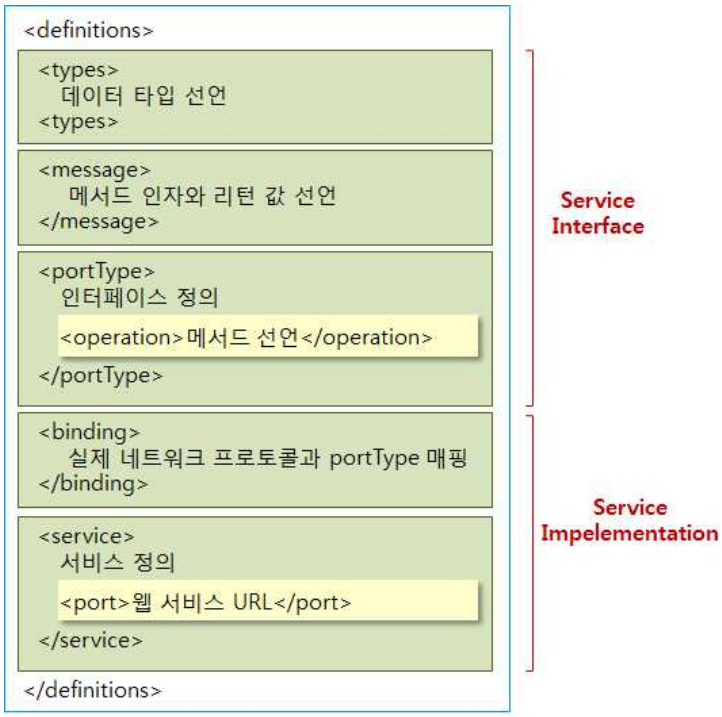
웹 서비스(Web Service) 방식에서 파일 전송은 HTTP 기반의 SOAP을 사용하여 송수신한다. 솝(SOAP) 대신 레스트 풀(RESTful) 프로토콜(Protocol)로 대체할 수 있다.

연계(인터페이스) 파일을 수신 시스템의 수신 위치로 전송 후 전송된 파일 내용을 수신 시스템 운영 DB에 반영하는 서비스를 실행한다. 수신 시스템의 서비스 실행을 위해서 서비스 위치, 서비스명 등 서비스에 대한 상세 내용이 포함된 서비스 명세서인 WSDL을 작성하여 송수신 시스템 간에 공유한다. WSDL에는 수신 시스템에 연계 파일을 전송 후, 연계 파일의 내용을 수신 시스템 운영 DB에 반영하는 서비스의 위치, 서비스명 등이 기술되어 있다.

솝(SOAP) 및 더블유 에스 디 엘(WSDL)은 엑스 엠 엘(XML) 형식으로 구현한다.

SOAP 기본 구조	HTTP 리퀘스트(Request)에 실린 SOAP 메시지
HTTP Header SOAP Envelope SOAP Header SOAP Body Attachment Attachment : Attachment	<SOAP-ENV:Envelope xmlns:SOAP-ENV= "http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle= "http://schemas.xmlsoap.org/soap/encoding/" > <SOAP-ENV:body> <m:ProcEmplNotcNm xmlns:m= " " > <emplNotcNo>2015092000017</emplNotcNo> <m:ProcEmplNotcNm> </SOAP-ENV:body> </SOAP-ENV:Envelope>
	HTTP 리스판스(Response)에 실린 SOAP 메시지
	<SOAP-ENV:Envelope xmlns:SOAP-ENV= "http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle= "http://schemas.xmlsoap.org/soap/encoding/" > <SOAP-ENV:body> <m:ProcEmplNotcNmResponse xmlns:m= " " > <emplNotcNm>[redacted] 채용 공고</emplNotcNm> <m:ProcEmplNotcNmResponse> </SOAP-ENV:body> </SOAP-ENV:Envelope>

[그림 3-6] SOAP의 기본 구조 및 메시지 구현



[그림 3-7] WSDL(Web Service Description Language) 기본 구조

(4) 수신 시스템의 연계 응용 프로그램 구현

더블유 에스 디 엘(WSDL)에 기술된 운영DB에 연계 데이터를 반영하는 서비스를 구현한다.

서비스 구현 방식은 송신의 연계 응용 프로그램과 동일하게 자바(Java)를 사용하며, 연계(인터페이스) 파일의 데이터를 수신 시스템의 운영 DB에 반영하고, 처리 과정을 기록(Logging)하는 내용으로 구현한다.

구현한 서비스는 수신 시스템의 서비스 호출이 가능한 위치(외부에서 웹 서비스 호출이 가능한 위치 및 URL)에 배치(Deployment)한다.

```
class [수신 연계 응용 프로그램명]  {
 public static void main(String[] args) {
   // 변수 선언
   String dbURL =  "jdbc:[dbms 제품명]:[driver명]@[운영 DB ip]:[운영 DB port]:[운영 DB instance명]" ;
   Connection Conn = null;
   PreparedStatement pstmt = null;
   String logString =  "" ;

   try {
       // 연계 파일, 로그(Log) 파일, 오류 로그 파일 Open 또는 Create
       // 연계 파일이 xml일 경우, DOM 방식으로 XML을 Parsing하여 Read
       File dataFile = new File( "[연계 파일 생성 위치]"  +  "[연계 파일명]" );
       DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
       DocumentBulder dBuilder = dbFactory.new DocumentBuilder();
       Document doc = dBuilder.parse(dataFile);

       BufferedWriter logWriter = new BufferedWriter(new FileWriter(
                   "[로그 파일 생성 위치]"  +  "[로그 파일명]" );

       // 연계(인터페이스) 파일 Read
       NodeList nList = doc.getElementsbyTagName( "[연계 데이터 행 단위 XML Tag명]" );

      // 진행 단계별 수행 결과에 대한 Logging 처리, Log 파일에 처리 결과 Write
      logString = "[진행 단계명] " + [진행 단계 변수명] + " [처리 결과] " + [처리 결과 변수명] +
                 " [처리 건수]  " + [처리 건수 변수명] + "  [오류 건수]  " + [오류 건수 변수명];
       logWriter.write(logString);

       // 운영 DB Connection
       Class.forName( "[DBMS 제품에 따른 JDBC Driver명]" );
       conn = DriverManager.getConnection(dbURL,  "[DB User ID]" ,  "[DB User PW]" );

       if( conn == null) {
           // 운영 DB 연결 실패, 오류 로그 처리
       } else {
           // 진행 단계별 수행 결과에 대한 Logging 처리, Log 파일에 처리 결과 Write

           // 운영 DB의 테이블에 데이터 Insert를 위한 SQL
           String sql =  "INSERT INTO [운영 DB 데이터 반영 대상 테이블명]"  +
                         " ( [칼럼명 1], [칼럼명 2], ..., [칼럼명 n]) "          +
                         " VALUES ( ?, ?, ..., ? ) " ;

           pstmt = conn.prepareStatement(sql);
```

```
            // 진행 단계별 수행 결과에 대한 Logging 처리, Log 파일에 처리 결과 Write

            for(int idx=0; idx < nList.getLength(); idx++) {        // 행(Row)의 수만큼 반복
                Node dataNode = nList.item(idx);

                if(dataNode.getNodeType == Node.ELEMENT_NODE) {
                    Element dataElement = (Element) dataNode;

                    // 진행 단계별 수행 결과에 대한 Logging 처리, Log 파일에 처리 결과 Write

                    // Insert SQL의 파라미터(?)를 연계 파일에서 Read한다.
                    // 행(Row)의 각 Column 값으로 Setting
                    pstmt.setString(1, dataElement.getAttribute( "[칼럼 단위 XML Tag명 1]" ));
                    pstmt.setString(2, dataElement.getAttribute( "[칼럼 단위 XML Tag명 2]" ));
                    :
                    pstmt.setInt(n, dataElement.getAttribute( "[칼럼 단위 XML Tag명 n]" ));

                    // 진행 단계별 수행 결과에 대한 Logging 처리, Log 파일에 처리 결과 Write

                    // 운영 DB의 테이블에 insert
                    int insertCnt = pstmt.executeUpdate();

                    // 진행 단계별 수행 결과에 대한 Logging 처리, Log 파일에 처리 결과 Write
                }
            }
        }
    } catch (SQLException sqle) {
        // 로그(Log) 파일에 진행 단계별 결과 Insert or Update
        // 로그(Log) 파일 또는 오류 로그 파일에 오류 코드 및 오류 내용 Insert or Update

    } catch (NullPointerException npe) {

    } catch (FileNotFoundException fnfe) {
        // 로그(Log) 파일에 진행 단계별 결과 Insert or Update
        // 로그(Log) 파일 또는 오류 로그 파일에 오류 코드 및 오류 내용 Insert or Update

    } catch (IOException ioe) {
    }
  }
}
```

[그림 3-8] 자바(Java)로 구현한 수신 시스템의 연계 응용 프로그램

수행 내용 / 연계 모듈 구현 환경 구성 및 개발하기

재료 · 자료

- 코드 정의서(송신 시스템, 수신 시스템)
- 송수신 절차도 또는 연계 메커니즘 정의서, 연계 아키텍처 설계서
- 인터페이스(I/F: Interface) 테이블(파일 Layout) 정의서
- 코드 매핑 내역서
- 오류 코드 정의서
- 로그 테이블(파일 Layout) 정의서
- 오류 로그 테이블(파일 Layout) 정의서
- 프로그램 설계서 또는 프로그램 명세서
- 송신용 연계 응용 프로그램, 수신용 연계 응용 프로그램
- 송수신 연계(인터페이스) 테이블(파일)별 데이터, 송수신 시스템별 코드 테이블의 데이터, 오류 코드 테이블의 데이터, 코드 매핑 테이블의 데이터

기기(장비 · 공구)

- DBMS 및 JDK 1.6 이상 설치가 완료된 컴퓨터
- DBMS는 송신 데이터베이스 사용자(DB User) <User 계정: senduser>, 수신 데이터베이스 사용자(DB User) <User 계정: recvuser> 생성하여 접속, 객체 생성, 객체 사용 등 권한 부여를 완료하고 DB가 서비스 되고 있는 상태

  ※ 데이터베이스 사용자(DB User) 계정의 비밀번호(Password)는 계정과 동일하게 지정함
  예) 데이터베이스 사용자(DB User) 계정/비밀번호(PWD, Password): senduser/senduser
- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

1 연계 메커니즘 정의서 또는 연계 아키텍처 설계서를 확인 및 점검한다.

1. '학습 2. 연계 메커니즘 구현하기' 에서 작성했던 연계 메커니즘 정의서 중 송수신 시스템에서의 연계 응용 프로그램 구현 방식, 연계(인터페이스) 데이터 생성 방식(테이블 vs 파일), 코드 매핑 및 데이터 변환 처리 위치(송신 시스템 vs 수신 시스템)을 확인한다.

2. 송수신 시스템 각각에서 연계 응용 프로그램 구현을 자바 프로그램(Java Program) 방식으로 연계 메커니즘 정의서를 수정 및 보완한다.

3. 연계(인터페이스) 데이터 생성은 파일 방식으로 연계 메커니즘 정의서를 수정 및 보완한다.

   ※ 연계 데이터 발생량이 대량일 경우에는 파일 방식보다는 테이블 방식이 성능이나 모니터링 측면에서 상대적으로 더 적합함

4. 코드 매핑 테이블 생성 위치를 송신 시스템의 운영 DB로 연계 메커니즘 정의서를 수정 및 보완한다.

5. 연계 응용 프로그램 구현과 연계(인터페이스) 데이터 생성 방식의 변경으로 영향받거나 연관된 부분(연계 파일 생성 위치, 로그 파일 생성 위치, 코드 매핑 처리 위치 등)이 있는지 확인하여 일관된 내용으로 수정 및 보완한다.

2 송신 시스템의 연계 모듈 구현 환경을 구축한다.

1. 송신용 데이터베이스 사용자(DB User)의 생성 여부, 송신용 데이터베이스 사용자(DB User) 및 비밀번호(Password)로 접근 가능 여부 등 데이터베이스 관리 시스템(DBMS: Database Management System) 환경을 점검한다.

2. 송신용 데이터베이스 사용자(DB User)로 채용 공고 테이블과 코드 테이블, 코드 매핑 테이블, 송신용 오류 코드 테이블을 생성하고 데이터를 입력한다.

   ※ 최소 데이터 등록 건 수 기준은 없으나, 실습이 원활히 진행될 수 있도록 최대한 많은 데이터를 입력해야 함, 데이터는 입력(Insert) 쿼리(SQL: Structured Query Language) 문을 이용하거나 골든(Golden), 토드(TOAD), 오렌지(Orange) 등의 데이터베이스 관리 시스템(DBMS: Database Management System) 서드 파티(3rd party) 툴을 사용하여 입력함

   '학습 1-1. 연계 요구 사항 분석' 에서 작성했던 '<표 1-5> 채용 공고(TB_EAS010) 테이블 정의서(송신 시스템)' 의 테이블을 생성하고, 생성한 테이블에 데이터를 입력한다.

   '학습 1-1. 연계 요구 사항 분석' 에서 작성했던 '<표 1-6> 사용자(TB_SYS010) 테이블 정의서(송신 시스템)' 의 테이블을 생성하고, 생성한 테이블에 데이터를 입력한다.

   '학습 1-1. 연계 요구 사항 분석' 에서 작성했던 '<표 1-7> 코드 정의서(송신 시스템)' 의 코드 테이블을 생성하고, 생성한 테이블에 데이터를 입력한다.

   '학습 1-2. 연계 데이터 식별 및 표준화의 수행 내용' 에서 작성했던 코드 매핑 테이블을 생성하고, 생성한 테이블에 데이터를 입력한다.

   '학습 2-2. 연계 장애 및 오류 처리 구현의 수행 내용' 에서 작성했던 오류 코드 테이블을 생성하고 데이터를 입력한다. 이때 오류 코드 및 내용은 송신과 코드 매핑

및 데이터 변환 처리 시 발생 가능한 오류 코드 및 내용으로 입력한다. 송신 시스템의 오류 코드 및 내용 입력 시 '학습 2-2 연계 장애 및 오류 처리 구현의 수행 내용' 에서 작성했던 오류 코드 정의서를 참고한다.

3. 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일이 위치할 송신 디렉토리를 생성한다.

'<표 3-4> 송수신 파일 생성 위치 및 파일명' 을 참고하여 송신 시스템의 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 생성 위치 및 파일명 부여 기준을 정의한다.

정의한 각 파일 생성 위치 및 파일명 부여 기준은 연계 메커니즘 정의서에 반영한다.

연계(인터페이스), 로그, 오류 로그 파일이 위치할 디렉토리(폴더)를 생성하고 접근권한을 읽기, 쓰기로 부여한다.

4. 송신 시스템의 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 정의서의 내용을 확인 및 점검한다.

'학습 2-1. 연계 메커니즘 정의의 수행 내용' 에서 작성했던 송신용 연계(인터페이스) 테이블(파일) 정의서 내용의 구조 및 각 항목의 데이터 타입, 길이 등을 확인한다.

'학습 2-1. 연계 메커니즘 정의의 수행 내용' 에서 작성했던 송신용 로그(Log) 테이블(파일) 정의서 내용의 구조 및 각 항목의 데이터 타입, 길이 등을 확인한다.

'학습 2-2. 연계 장애 및 오류 처리 구현의 수행 내용' 에서 작성했던 송신용 오류 로그 테이블(파일) 정의서 내용의 구조 및 각 항목의 데이터 타입, 길이 등을 확인한다.

3 수신 시스템의 연계 모듈 구현 환경을 구축한다.

1. 수신용 데이터베이스 사용자(DB User)의 생성 여부, 수신용 데이터베이스 사용자(DB User) 및 비밀번호(Password)로 접근 가능 여부 등 데이터베이스 관리 시스템(DBMS: Database Management System) 환경을 점검한다.

2. 수신용 데이터베이스 사용자(DB User)로 공고 정보 테이블과 코드 테이블, 수신용 오류 코드 테이블을 생성하고 데이터를 입력한다.

'학습 1-1. 연계 요구 사항 분석' 에서 작성했던 '<표 1-8> 공고 정보(NOTICE_INFO) 테이블 정의서(수신 시스템)' 의 테이블을 생성하고, 생성한 테이블에 데이터를 입력한다.

'학습 1-1. 연계 요구 사항 분석' 에서 작성했던 '<표 1-9> 코드 정의서(수신 시스템)' 의 코드 테이블을 생성하고, 생성한 테이블에 데이터를 입력한다.

'학습 2-2. 연계 장애 및 오류 처리 구현의 수행 내용' 에서 작성했던 오류 코드 테이블을 생성하고 데이터를 입력한다. 이때의 오류 코드 및 내용은 수신 및 DB 반영

시 발생 가능한 오류 코드 및 내용으로 입력한다. 수신 시스템의 오류 코드 및 내용 입력 시 '학습 2-2. 연계 장애 및 오류 처리 구현의 수행 내용' 에서 작성했던 오류 코드 정의서를 참고한다.

3. 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일이 위치할 수신 디렉토리를 생성한다.

'<표 3-4> 송수신 파일 생성 위치 및 파일명 예시' 를 참고하여 수신 시스템의 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 생성 위치 및 파일명 부여 기준을 정의한다.

정의한 각 파일 생성 위치 및 파일명 부여 기준은 연계 메커니즘 정의서에 반영한다.

연계(인터페이스), 로그, 오류 로그 파일이 위치할 디렉토리(Directory, 폴더)를 생성하고 접근 권한을 읽기, 쓰기로 부여한다.

4. 수신 시스템의 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 정의서의 내용을 확인 및 점검한다.

'학습 2-1. 연계 메커니즘 정의의 수행 내용' 에서 작성했던 수신용 연계(인터페이스) 테이블(파일) 정의서 내용의 구조 및 각 항목의 데이터 타입, 길이 등을 확인한다.

'학습 2-1. 연계 메커니즘 정의의 수행 내용' 에서 작성했던 수신용 로그(Log) 테이블(파일) 정의서 내용의 구조 및 각 항목의 데이터 타입, 길이 등을 확인한다.

'학습 2-2. 연계 장애 및 오류 처리 구현의 수행 내용' 에서 작성했던 수신용 오류 로그(Log) 테이블(파일) 정의서 내용의 구조 및 각 항목의 데이터 타입, 길이 등을 확인한다.

4 송신 시스템과 수신 시스템의 연계 응용 프로그램을 개발한다.

1. 컴퓨터(PC: Personal Computer)에서 자바 개발 도구(JDK: Java Development Kit) 설치 여부, 설치 버전 등 환경을 확인 및 점검한다.

[java -version] 명령어와 옵션을 통해 설치된 자바 개발 도구(JDK: Java Development Kit) 버전을 확인할 수 있으며, 설치 경로(path) 등 정상적 설치 여부와 설치 환경을 확인한다.

2. 송신 시스템의 연계 응용 프로그램 설계서를 작성한다.

개발할 송신 시스템 연계 응용 프로그램의 설계서를 '<표 3-5> 프로그램 설계서(명세서) 양식' 을 활용하여 작성한다.

프로그램 설계서는 프로그램 ID, 프로그램명, 인터페이스 파라미터 등을 비롯하여 상세 처리 로직을 수도(Pseudo) 코드 형태 또는 자바 언어(Java Language)의 문법으로

구체적이며 중요 내용을 누락 없이 작성한다.

① 송신용 데이터베이스 사용자(DB User) 계정으로 데이터베이스 연결(DB Connection) 생성
② 송신용 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 생성
③ 송신용 DB에서 채용 공고 테이블의 데이터 추출(Select)
④ 추출한 데이터 중 코드 데이터를 수신 시스템의 코드로 매핑 처리
⑤ 송신 시스템 채용 공고 테이블과 수신 시스템 공고 정보 테이블의 연계 데이터 중 데이터 타입, 데이터 길이가 상이한 경우 수신 시스템의 공고 정보 테이블 형식으로 변환 처리
⑥ 송신용 연계(인터페이스) 파일에 변환 완료된 데이터 로우(Row, 행) 단위로 기록(Write)
송신용 연계(인터페이스) 파일은 xml 형식이 아닌 구분자(Delimiter)를 콜론(:)으로 사용하는 텍스트(Text) 파일로 기록하며, 파일 확장자는 .data로 부여함
⑦ 데이터 추출, 코드 매핑, 변환, 연계(인터페이스) 파일 기록 등 처리 단계별 송신용 로그 파일에 처리 내용 및 결과 기록(Write)
⑧ 처리 단계별 오류 발생 시 송신용 오류 코드 테이블 참고하여 송신용 오류 로그 파일에 기록(Write)

[그림 3-9] 송신 시스템 연계 응용 프로그램 설계서의 주요 처리 내용

<표 3-5> 프로그램 설계서(명세서) 양식

1. 공통 사항

프로그램 ID		프로그램 유형	
프로그램명			
프로그램 목적 및 용도			
관련 프로그램			

2. 상세 사항

메서드(Method) ID		메서드(Method)명	
입력 파라미터 (타입/명)			
반환값(타입/명)			
관련 테이블			
처리 내용			

:

메서드(Method) ID		메서드(Method)명	
입력 파라미터 (타입/명)			
반환값(타입/명)			
관련 테이블			
처리 내용			

3. 수신 시스템의 연계 응용 프로그램 설계서를 작성한다.

개발할 수신시스템 연계 응용 프로그램의 설계서를 '<표 3-5> 프로그램 설계서(명세서) 양식'을 활용하여 작성한다.

프로그램 설계서는 프로그램 ID, 프로그램명, 인터페이스 파라미터 등을 비롯하여 상세 처리 로직을 수도(Pseudo) 코드 형태 또는 자바 언어(Java Language)의 문법으로 구체적이며 중요 내용을 누락 없이 작성한다.

① 수신용 데이터베이스 사용자(DB User) 계정으로 데이터베이스 연결(DB Connection) 생성
② 수신용 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 생성
③ 수신용 연계(인터페이스) 파일을 읽기 모드(Read Mode)로 열어서 한 행(Row)씩 로드(Load)
④ 파일에서 로드(Load)한 데이터를 구분자(Delimiter, :)를 활용하여 각 속성(컬럼)으로 분할
⑤ 각 속성의 데이터 유효성 점검(Validation Check)
예 : 엠프티 스트링(Empty String, 공백 문자열) 여부, 데이터 타입, 데이터 길이 등
⑥ 분할된 속성(컬럼) 중 코드가 존재할 경우, 수신용 코드 테이블에 존재하는 코드인지 확인
⑦ 행(Row)의 분할된 모든 속성(칼럼)이 데이터 유효성 점검(Validation Check)를 통과하여 유효할 경우 수신용 DB의 공고 정보 테이블에 변경 구분자에 따라 입력(Insert), 수정(Update), 삭제(Delete) 처리
⑧ 연계(인터페이스) 파일 읽기 모드로 생성, 데이터 Load, 각 속성(컬럼) 분할, 데이터 Validation Check 등 처리 단계별 수신용 로그 파일에 처리 내용 및 결과 기록(Write)
⑨ 처리 단계별 오류 발생 시 도는 데이터 유효성 점검(Validation Check) 시 유효하지 않은 값일 경우 수신용 오류 코드 테이블 참고하여 수신용 오류 로그 파일에 기록(Write)

[그림 3-10] 수신 시스템 연계 응용 프로그램 설계서의 주요 처리 내용

4. 작성한 송수신 연계 응용 프로그램 설계서의 내용을 자바 언어(Java Language)를 사용하여 단독 배치(Batch) 형식으로 개발한다.

개발된 자바(Java) 프로그램은 송신 시스템의 연계 응용 프로그램은 송신 디렉토리(Directory, 폴더)의 루트(root)에, 수신 시스템의 연계 응용 프로그램은 수신 디렉토리(Directory, 폴더)의 루트(root)에 배치한다.

5. 송수신 환경에서 개발한 연계 응용 프로그램을 각각 컴파일하고, 실행한다.

송신 시스템의 연계 응용 프로그램을 컴파일(Compile, javac 명령어 이용)한다.

성공적으로 컴파일된 송신 시스템의 연계 응용 프로그램을 실행하여 송신 디렉토리(Directory, 폴더) 위치에 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일 등이 생성되는지 확인한다.

※ 송신용 데이터베이스 사용자(DB User) 테이블의 데이터 추출에서 코드 및 데이터 변환 처리 후 연계(인터페이스) 파일에 기록 및 생성하는 과정까지 오류가 발생하지 않고, 정상적으로 처리되었을 경우 오류 로그 파일은 생성되지 않을 수 있음

수신 시스템의 연계 응용 프로그램을 컴파일(Compile, javac 명령어 이용)한다.

수신 디렉토리 중 연계(인터페이스) 파일 위치에 수신용 연계(인터페이스) 파일 구조대로 데이터를 입력하여 수신용 연계(인터페이스) 파일을 생성한다.

성공적으로 컴파일된(Compiled) 수신 시스템의 연계 응용 프로그램을 실행하여 수신

디렉토리(폴더)에 로그 파일, 오류 로그 파일 등이 생성되는지 확인한다.

※ 연계(인터페이스) 파일의 내용 로드(Load)에서 부터 데이터 유효성 점검(Validation Check), 수신용 데이터베이스 사용자(DB User) 계정의 공고 정보 테이블에 데이터를 반영하는 과정까지 오류가 발생하지 않고 정상적으로 처리되었을 경우 오류 로그 파일은 생성되지 않을 수 있음

수행 tip

- 데이터베이스 관리 시스템(DBMS: Database Management System)에 따른 에스 큐 엘(SQL: Structured Query Language, 쿼리) 문법을 확인 및 학습한다.
- 자바 언어(Java Language) 문법을 확인 및 학습한다.
- 자바(Java) 문법 중 파일(File) 처리, 데이터베이스 연결(DB Connection) 및 에스 큐 엘(SQL: Structured Query Language, 쿼리) 처리, 예외(Exception) 처리 등을 확인 및 학습한다.
- 데이터베이스 관리 시스템(DBMS: Database Management System)과 자바 언어(Java Language)에 따른 오류(Error, Exception 등) 코드 및 내용을 확인 및 학습한다.
- 자바(Java)의 컴파일 명령어(javac), 실행 명령어(java) 등 필요한 간단한 명령어들을 확인 및 학습한다.
- 프로그램(Program) 구현 시 처리 로직의 누락 예방 및 코드의 최적화를 위해서는 프로그램 설계서(명세서)를 상세하고 구체적으로 작성하며, 구현할 언어를 활용하여 작성한다.

3-2. 연계 테스트 및 검증

학습 목표	• 연계 모듈의 안정적인 작동 여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다. • 구현된 연계 모듈을 테스트할 수 있는 테스트 케이스를 작성하고, 단위 테스트를 수행하기 위한 테스트 조건을 명세화할 수 있다.

필요 지식 /

[그림 3-11] 연계 테스트의 대상 및 범위

구축된 연계 시스템과 연계 시스템의 구성 요소(송신 모듈, 수신 모듈, 연계 서버 및 엔진, 모니터링 현황 등)가 정상적으로 동작하는지 확인하고 검증하는 활동이 테스트이다.

연계 업무의 단위 테스트는 연계(인터페이스) 테스트와 데이터가 발생하고 활용되는 응용 애플리케이션 기능과의 통합 구조상 흐름을 테스트하는 통합 테스트로 단계적으로 수행한다.

【 연계(인터페이스) 테스트 구간 】

연계(인터페이스) 테스트는 구간을 상세화하여 송수신 시스템 간의 연계(인터페이스) 테이블(파일)간의 테스트를 수행하고, 정상적으로 데이터가 송수신되면 송신 시스템의 운영 DB에서 연계 데이터를 추출 및 생성하여 수신 시스템의 운영 DB에 반영하는 테스트로 순차적으로 진행한다.

일반적으로 연계(인터페이스) 테스트는 송신 시스템의 운영 DB에서 연계 데이터를 추출 및 생성하여 수신 시스템의 운영 DB에 반영하는 구간의 테스트를 의미한다.

<표 3-6> 단계별 테스트의 테스트 구간 및 범위

구분	데이터 흐름 및 처리 절차 (테스트 시 확인해야 할 사항)	위치	단위(인터페이스) 테스트 구간	통합 테스트 구간
송신 시스템	데이터 발생 및 운영 DB 저장	운영 응용 애플리케이션		
	연계 데이터 추출 및 생성	연계 응용 프로그램		
	연계(인터페이스) 테이블(파일) 생성	연계 응용 프로그램		
수신 시스템 연계 서버 또는 중계 서버	연계(인터페이스) 테이블(파일) 간 송수신	연계 서버(엔진) 또는 중계 서버		
	연계(인터페이스) 테이블(파일) 생성	연계 서버(엔진) 또는 중계 서버		
수신 시스템	연계(인터페이스) 테이블(파일)에서 연계 데이터 로드(Load)	연계 응용 프로그램		
	운영 DB에 반영	연계 응용 프로그램		
	반영된 데이터 검색 및 활용	운영 응용 애플리케이션		

1 연계 모듈 테스트 케이스(Testcase) 작성 및 명세화

연계(인터페이스) 테스트 구간에서의 데이터 및 프로세스 흐름에 따라 테스트 케이스(Testcase)를 작성한다. 연계(인터페이스) 테스트 케이스는 연계(인터페이스) 테이블(파일) 단위로 작성한다.

테스트 케이스(Testcase) 작성 시 가장 핵심적인 사항은 테스트 항목의 도출이다. 테스트할 대상 및 기능 등이 충분히(완전하게), 누락 없이, 일관성 있게, 구체적으로 식별되어야 한다.

1. 송수신용 연계 응용 프로그램의 단위 테스트 케이스(Testcase) 작성 및 명세화

연계(인터페이스) 테스트 환경은 실제 운영 환경과 동일하게 또는 유사하게 구축한다.

테스트 케이스는 송신 시스템과 수신 시스템에서 확인해야 할 사항을 각각 도출하며, 송수신 시스템 각각에서 단순 개별 데이터의 유효값을 체크(Check)하는 경우의 수와 데이터 간의 연관 관계를 체크(Check)하는 경우의 수로 식별하여 작성한다.

단순 개별 데이터 유효값은 코드일 경우, 코드 테이블에 등록된 값의 여부 또는 유효 데이터 타입(Type), 길이, 필수 입력 여부 등의 체크(Check)를 의미한다.

데이터 간의 상관 관계 체크는 특히, 수신 시스템에서 운영 DB에 반영하는 연계 응용 프로그램에서 등록되어 있지 않은 데이터에 대해서 수정이나 변경이 발생했거나 또는 참조하는 테이블에 데이터가 없는 경우에 데이터를 등록하는 경우의 점검이 해당된다.

<표 3-7> 송신용 연계 응용 프로그램의 단위 테스트 케이스(Testcase) 작성

단위 시험 ID	TC-IF-AS-001	단위 시험명	송신용 채용공고 연계 응용 프로그램		
테스트 일시	2015.10.02. 11:○○	시험자명	○○○		
		확인자명	△△△		
설명	운영 DB에서 채용정보 연계 데이터 추출, 코드 변환 및 연계(인터페이스) 파일 생성 등 확인				
관련 프로그램 ID(명)	채용 공고 등록, 수정, 삭제				
케이스 ID	시험 항목 및 처리 절차	입력 데이터	예상 결과	검증 방법	시험 결과
001	채용 공고 테이블에서 테스트 일자, 시간에 추가, 수정, 삭제된 정보 검색	2015.10.02. 11시	100건 조회	SELECT Count(*) FROM 채용공고 WHERE 변경일시 Like '201510021 1%'	
002	자격 코드를 수신 시스템의 자격 코드로 변환	2001, 청소년 상담사	8, 청소년 및 사회 복지	SELECT 코드, 코드명 FROM 코드매핑 WHERE 변환 전 코드 = 채용공고.자격코드	
:	:	:	:	:	
N	처리 일시의 연계(인터페이스) 파일 생성	처리 일시 (테스트 시점 시간)	테스트 시점의 일자 폴더에 send_테스트 시점 시분.data 파일 생성	송신용 연계 파일 생성 위치에 채용 정보 연계 파일 생성 여부와 크기 확인	

2. 연계(인터페이스) 테스트 케이스(Testcase) 작성 및 명세화

연계(인터페이스) 테스트 케이스(Testcase)는 송신용 연계 응용 프로그램과 수신용 연계 응용 프로그램의 기능 위주 결함을 확인하는 단위 테스트 케이스(Testcase)로 작성한다. 작성한 단위 테스트 케이스를 연계(인터페이스) 테이블(파일) 간 송수신 절차의 선후로 연결하여 흐름을 확인하는 내용으로 작성한다.

<표 3-8> 연계(인터페이스) 테스트 케이스 작성

단위 시험 ID	TC-IF-AS-100	단위 시험명	채용 공고 연계(인터페이스) 테스트		
테스트 일시	2015. 10. 05. 11:○○	시험자명	○○○		
		확인자명	△△△		
설명	송신 시스템 운영 DB의 채용 공고 변경(추가, 수정, 삭제) 데이터를 추출하여 수신 시스템으로 전송 후 수신 시스템 운영 DB의 공고 정보 테이블에 반영되는지를 확인				
관련 프로그램 ID(명)	송신용 채용 공고 연계 응용 프로그램, 수신용 공고 정보 연계 응용 프로그램				
케이스 ID	시험 항목 및 처리 절차	입력 데이터	예상 결과	검증 방법	시험 결과
001	시험일자 11시로 송신용 채용 공고 연계 응용 프로그램 실행하여 완료 후 결과 확인	2015. 10. 05. 11시	송신용 채용 공고 연계 파일 생성 파일명 : ○○○○ 파일 크기 : ○○ KB	파일 크기 확인 및 파일을 열어서 생성된 데이터 건수 확인	
002	○분 후 채용 공고 연계 파일 송수신	-	송신 완료, 송신용 채용 공고 연계 파일의 행(Row) 데이터별 송신 완료 Flag	연계 서버의 모니터링 화면에서 송신 결과 확인	
:	:	:	:	:	
N	수신용 공고 정보 연계 응용 프로그램 실행하여 완료 후 결과 확인	-	공고 정보 테이블에 공고 정보 연계 파일의 정상적 데이터 반영	SELECT Count(*) FROM 공고정보 WHERE 변경일시 Like '20151005%'	

2 연계 테스트 수행 및 검증

1. 연계(인터페이스) 테스트 환경 구축

연계(인터페이스) 테스트 환경은 실제 운영 환경과 동일하게 또는 유사하게 구축한다.

연계 서버(엔진), 송수신용 어댑터(Adapter) 또는 에이전트(Agent) 설치를 비롯하여 송수신 운영 DB, 송수신 운영 DB에서 연계 데이터 추출 및 반영에 필요한 테이블, 데이터와 송수신용 연계 응용 프로그램들을 설치 및 준비한다.

송수신 기관 간에 테스트 수행 전에 연계(테스트) 일정, 절차, 방법, 소요 시간, 테스트 환경, 환경 구축 기간 등을 협의하여 계획을 수립하고 테스트 환경을 구축한다.

<표 3-9> 연계(인터페이스) 테스트를 위한 테스트 환경 구축 점검 및 준비 사항

위치	점검 및 준비 사항		설치(준비) 완료 계획일	담당자명/ 확인자명	점검 결과
송신 시스템	연계 솔루션	Adapter(Agent) 설치 및 설정	○○.○○.○○	△△△	
		수신 기관과 연계할 IP 및 포트 허용 신청	○○.○○.○○	△△△	
	DB	운영 DB	○○.○○.○○	△△△	
	연계 대상 (운영) 테이블	채용 공고 테이블 및 데이터	○○.○○.○○	△△△	
		:	:	:	
	연계 테이블(파일)	채용 공고 테이블(파일) 및 데이터	○○.○○.○○	△△△	
		:	:	:	
	모니터링 및 관리 테이블(파일)	로그 테이블(파일)	○○.○○.○○	△△△	
		오류 로그 테이블(파일)	○○.○○.○○	△△△	
		오류 코드 테이블(파일) 및 데이터	○○.○○.○○	△△△	
	프로그램	채용 공고 연계 응용 프로그램	○○.○○.○○	△△△	
		:	:	:	
	:	:	:	:	
연계 서버 (중계 서버)	시스템 설치 및 환경 설정	엔진 설치 및 시스템 설정	○○.○○.○○	△△△	
		로그 기록용 DB 및 테이블	○○.○○.○○	△△△	
		로그 기록용 폴더 생성 및 권한 부여	○○.○○.○○	△△△	
		송수신시스템과 연계 시 사용할 아이피(IP) 및 포트(Port) 허용 신청	○○.○○.○○	△△△	
수신 시스템	연계 솔루션	Adapter(Agent) 설치 및 설정	○○.○○.○○	△△△	
		송신 기관과 연계할 아이피(IP) 및 포트 허용 신청	○○.○○.○○	△△△	
	DB	운영 DB	○○.○○.○○	△△△	
	반영 대상 (운영) 테이블	공고 정보 테이블 및 데이터	○○.○○.○○	△△△	
		:	:	:	
	연계 테이블 및 파일	공고 정보 테이블(파일) 및 데이터	○○.○○.○○	△△△	
		:	:	:	
	모니터링 및 관리 테이블(파일)	로그 테이블(파일)	○○.○○.○○	△△△	
		오류 로그 테이블(파일)	○○.○○.○○	△△△	
		오류 코드 테이블(파일) 및 데이터	○○.○○.○○	△△△	
	프로그램	공고 정보 연계 응용 프로그램	○○.○○.○○	△△△	
		:	:	:	
	:	:	:	:	

2. 연계(인터페이스) 테스트 수행

연계(인터페이스) 테스트 케이스(Testcase)의 시험 항목 및 처리 절차대로 실제 연계 응용 프로그램을 실행하며 테스트하고 결과를 확인한다.

송신용 연계 응용 프로그램과 수신용 연계 응용 프로그램의 단위 테스트를 수행한다.

단위 테스트가 오류 없이 수행 완료되면 작성한 테스트 케이스(Testcase)대로 데이터 추출, 송수신, 데이터 반영 과정의 연계(인터페이스) 테스트를 수행한다.

테스트명	단위 테스트		연계 테스트
테스트 대상	송수신 시스템의 연계 응용 프로그램	⇨	송신 시스템 연계 응용 프로그램, 연계 서버(엔진)의 데이터 송수신, 수신 시스템 연계 응용 프로그램
테스트 내용	기능 동작 여부 및 결함 여부 (데이터 추출, 코드 및 데이터 변환, 데이터 반영 등)		데이터의 흐름 및 처리 절차, 기능 정상적 동작 확인

[그림 3-12] 연계(인터페이스) 테스트 수행 순서

수행 위치	송신 시스템		연계(중계) 서버		수신 시스템
테스트 수행 절차	연계 응용 프로그램 실행 (수동 또는 자동)	⇨	연계 데이터 송수신	⇨	연계 응용 프로그램 (시스템에서) 실행

[그림 3-13] 연계(인터페이스) 테스트 수행 절차

3. 연계(인터페이스) 테스트 수행 결과 검증

연계(인터페이스) 테스트 케이스(Testcase)의 시험 항목 및 처리 절차 순서에 따라 테스트를 진행하면서 수행한 테스트 결과가 예상한 정상적인 결과인지를 확인하기 위해 검증을 수행한다.

테스트 결과를 검증하는 방법으로는 테스트하는 항목에 따라 ① 운영 DB 테이블의 건수를 카운트(Count)하는 방법, ② 실제 테이블이나 파일을 열어서 데이터를 확인하는 방법, ③ 파일 생성 위치의 파일 생성 여부와 파일 크기를 확인하는 방법, ④ 연계 서버(또는 연계 엔진)에서 제공하는 모니터링 화면의 내용을 확인하는 방법, ⑤ 시스템에서 기록하는 로그를 확인하는 방법 등이 있다.

통합 테스트에서는 송신 시스템에서 데이터를 생성하는 응용 애플리케이션의 기능을 이용하여 데이터를 생성하고, 수신 시스템의 운영 응용 애플리케이션 화면에서 송신 시스템에서 송신한 데이터가 유효한 값으로 검색이 된다면 정상적으로 데이터가 송수신되어 처리된 것으로 확인할 수 있다.

<표 3-10> 연계(인터페이스) 테스트 결과 검증 방법

구분	테스트 항목	검증 방법	검증 방법 예시
송신 시스템	운영 DB의 테이블에서 데이터 추출 건수	운영 DB의 테이블에서 조회된 건수와 추출 건수 비교	SELECT COUNT(*) FROM 운영 DB 테이블명 WHERE 추출 조건;
	코드 매핑 및 변환	코드 매핑 테이블의 변경 전 코드의 등록 여부 조회	SELECT 변경 후 코드 FROM 코드 매핑 테이블명 WHERE 변경 전 코드 = 운영 DB 테이블명.코드;
		변경 전 코드에 대한 변경 후 코드 조회	
	연계(인터페이스) 테이블(파일) 생성	테이블인 경우, 연계 DB 테이블에서 등록된 건 확인	SELECT COUNT(*) FROM 연계 DB 테이블명 WHERE 등록일시=처리일시;
		테이블인 경우, 연계 DB 테이블에서 변경 구분값(등록, 수정, 삭제)에 따른 등록 건수 확인	SELECT COUNT(*) FROM 연계 DB 테이블명 WHERE 등록일시=처리일시 AND 변경 구분= 'U' ;
		테이블인 경우, 연계 DB 테이블에서 특정 변경 후의 자격 코드값에 해당하는 등록 건수 확인	SELECT COUNT(*) FROM 연계 DB 테이블명 WHERE 등록일시=처리일시 AND 변경 구분= 'I' AND 자격 코드= '8' ;
		파일인 경우, 파일의 생성 위치 및 파일명, 파일 크기 등을 파일 시스템에서 직접 확인	ls 또는 ls -al 등 운영체제(OS)의 파일 관리 명령어
		생성된 파일을 열어서 생성된 데이터 건수(행 수) 등 확인	cat 생성된 파일명 또는 cat -b 생성된 파일명 등 OS의 파일 보기 명령어
연계 서버 (중계 서버)	송신 여부, 수신 여부	연계 서버(엔진)에서 제공하는 관리자 시스템에서 송수신 현황 모니터링 화면으로 확인	http://[redacted]/admin
		연계 서버(엔진)에서 기록하는 로그 파일 확인	cat 연계 서버(엔진)의 로그 파일명 또는 cat -b 연계 서버(엔진)의 로그 파일명 등
		수신 연계(인터페이스) 파일의 생성 여부 및 크기 확인	ls 또는 ls -al 등 운영체제(OS)의 파일 관리 명령어
수신 시스템	등록된 코드 여부	운영 DB 코드 테이블에서 연계(인터페이스) 테이블(파일)의 코드 등록 여부 확인	SELECT 코드명 FROM 운영 DB 코드테이블명 WHERE 코드 = 연계 테이블(파일) 행의 코드;
	유효한 데이터 타입, 길이 여부	수신용 연계 응용 프로그램에서 데이터 밸리데이션 체크(Data Validation Check), DBMS 컨스트레인트(Constraint), 운영 시스템 화면에서 확인	화면 위치 : Home > 채용 정보 관리 > 채용 정보 검색
	운영 DB의 테이블에 반영 건수	운영 DB 테이블에서 조회된 건수와 반영된 건수 비교	SELECT COUNT(*) FROM 운영 DB 테이블명 WHERE 변경일시 = 처리일시;
		운영 DB 테이블에서 특정 자격 코드값에 해당하는 반영 건수 확인	SELECT COUNT(*) FROM 운영 DB 테이블명 WHERE 변경일시 = 처리일시 AND 자격코드 = '8' ;

수행 내용 / 연계 테스트 및 검증하기

재료 · 자료

- 송신 데이터베이스 사용자(DB User) 계정 및 수신 데이터베이스 사용자(DB User) 계정이 생성된 데이터베이스 관리 시스템(DBMS: Database Management System) 설치 환경
- 프로그램 설계서 또는 프로그램 명세서
- '학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용' 에서 구현한 송신용 연계 응용 프로그램, 수신용 연계 응용 프로그램
- 송신 데이터베이스 사용자(DB User) 계정으로 생성한 채용 공고 테이블, 사용자 테이블, 코드 테이블, 코드 매핑 테이블, 오류 코드 테이블과 각 테이블의 데이터

  ※ '학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용' 대로 실습한 경우, 송신 데이터베이스 사용자(DB User) 계정으로 필요한 테이블 및 데이터는 생성된 상태이므로 테이블 생성이나 데이터 등록은 준비할 필요 없음
- 수신 데이터베이스 사용자(DB User) 계정으로 생성한 공고 정보 테이블, 코드 테이블, 오류 코드 테이블과 각 테이블의 데이터

  ※ '학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용' 대로 실습한 경우, 수신 데이터베이스 사용자(DB User) 계정으로 필요한 테이블 및 데이터는 생성된 상태이므로 테이블 생성이나 데이터 등록은 준비할 필요 없음
- 연계(인터페이스) 파일 및 로그 파일, 오류 로그 파일이 생성될 송수신 디렉토리(Directory, 폴더)

  ※ '학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용' 대로 실습한 경우, 송수신 디렉토리(Directory, 폴더)가 생성된 상태이므로 송수신 디렉토리(Directory, 폴더)를 생성할 필요 없음
- 단위 테스트 케이스 및 연계(인터페이스) 테스트 케이스(Testcase)

기기(장비 · 공구)

- 데이터베이스 관리 시스템(DBMS: Database Management System) 및 자바 개발 도구(JDK: Java Development Kit) 1.6 이상 설치가 완료된 컴퓨터
- 데이터베이스 관리 시스템(DBMS: Database Management System)은 송신 데이터베이스 사용자(DB User) <User 계정: senduser>, 수신 데이터베이스 사용자(DB User) <User 계정: recvuser> 생성하여 접속, 객체(Object) 생성, 객체(Object) 사용 등 권한 부여를 완료하고 DB가 서비스되고 있는 상태

  ※ 데이터베이스 사용자(DB User) 계정의 비밀번호(Password)는 계정과 동일하게 지정함
  예) 데이터베이스 사용자(DB User) 계정/비밀번호(PWD: Password): senduser/senduser
- 컴퓨터, 편집기(파워포인트, MS-Word, 한글 등)

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서

1 연계(인터페이스) 테스트 케이스(Testcase)를 작성 및 명세화 한다.

1. '<표 3-7> 송신용 연계 응용 프로그램의 단위 테스트 케이스 작성' 을 참고하여 송신용 연계 응용 프로그램 및 수신용 연계 응용 프로그램에 대한 단위 테스트 케이스(Testcase)를 각각 작성 및 명세화 한다.

   단위 테스트 케이스(Testcase)는 프로그램 단위의 기능상 결함을 확인할 수 있는 항목으로 도출하여, 연계(인터페이스) 단위로 작성한다.

   발생 가능한 오류에 대해서도 처리하고 있는지 확인할 수 있는 테스트 케이스(Testcase) 항목을 식별하여 명세화한다.

   테스트 케이스(Testcase)는 정상적인 기능 수행 외에도 처리 과정상 각 연계 데이터 항목이 가질 수 있는 데이터값의 유효 범위를 참고하여 오류 발생에 대한 처리 사항 체크(Check) 여부도 포함하여 작성한다.

2. '<표 3-8> 연계(인터페이스) 테스트 케이스 작성' 을 참고하여 연계(인터페이스) 테스트 케이스(Testcase)를 작성 및 명세화 한다.

   수신 시스템에서 데이터 추출, 송신용 연계(인터페이스) 테이블(파일) 및 데이터 생성, 송수신, 수신용 연계(인터페이스) 테이블(파일 및 데이터 생성, 송신 시스템 운영 DB의 테이블에 데이터 반영까지가 중단 없이 수행되는지를 확인할 수 있도록 테스트 케이스(Testcase) 항목을 식별하여 작성한다.

2 연계(인터페이스) 테스트 환경을 구축한다.

1. 송수신 기관 간의 사전 협의를 통해 테스트 일정(기간), 방법, 절차, 환경, 환경 구축 기간 등을 결정하고 연계(인터페이스) 테스트 환경을 구축한다.

   테스트 환경 및 테스트 수행을 위해 송수신 기관 간의 협의 시 일정 수립 및 준비사항에 '<표 3-9> 연계(인터페이스) 테스트를 위한 테스트 환경 구축 점검 및 준비 사항' 을 참고한다.

   송수신 기관 간의 협의된 결과에 따라, 연계 서버(중계 서버) 및 송수신 어댑터(Adapter) 또는 에이전트(agent) 설치, 연계를 위한 아이피(IP: Internet Protocol) 및 포트(Port) 허용 신청, 연계를 위한 DB 계정 및 테이블과 데이터 생성 등 테스트 환경 구축을 진행한다.

2. 연계(인터페이스) 테스트 환경을 '학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용'에서 구성한 환경으로 대체하여 환경을 점검 및 확인한다.

송신 시스템, 송신 시스템의 운영 DB, 수신 시스템, 수신 시스템의 운영 DB, 연계 서버(중계 서버) 등의 설치 및 환경 구축을 위해서는 여러 가지 제약 조건이 존재하므로 '학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용'의 환경으로 대체하여 구축한다.

'학습 3-1. 연계 모듈 구현 환경 구성 및 개발의 수행 내용'대로 실습한 경우, 이미 환경 구축이 완료된 상태이므로 추가적인 설치 및 구축 작업은 없으나 아래의 '연계(인터페이스) 테스트 환경 점검 항목'으로 구축되어 있는지 확인한다.

【 연계(인터페이스) 테스트 환경 점검 항목 】

① 송신 데이터베이스 사용자(DB User) 및 수신 데이터베이스 사용자(DB User) 생성

② 송신 데이터베이스 사용자(DB User)로 채용 공고 테이블, 사용자 테이블, 코드 테이블, 오류 코드 테이블, 코드 매핑 테이블 생성 및 각 테이블 데이터 입력

③ 수신 데이터베이스 사용자(DB User)로 공고 정보 테이블, 코드 테이블, 오류 코드 테이블 생성 및 각 테이블 데이터 입력

④ 송신 디렉토리(Directory, 폴더) 생성 및 권한 부여: 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일이 위치할 디렉토리(Directory, 폴더)

⑤ 수신 디렉토리(Directory, 폴더) 생성 및 권한 부여: 연계(인터페이스) 파일, 로그 파일, 오류 로그 파일이 위치할 디렉토리(Directory, 폴더)

3 1항에서 작성한 테스트 케이스로 테스트를 수행하며, 테스트 수행 결과를 검증한다.

1. 송신 시스템의 연계 응용 프로그램을 실행하여 단위 테스트를 수행하고, 결과를 검증한다.

송신 시스템의 연계 응용 프로그램을 실행하여 테스트 케이스(Testcase) 항목의 순서대로 테스트한다.

테스트 결과는 테스트 케이스(Testcase)의 테스트 케이스 항목별 검증 방법으로 검증한다.

2. 생성된 송신 시스템 연계(인터페이스) 파일을 수신 디렉토리의 연계(인터페이스) 파일 위치에 복사하여 이동시키고, 연계(인터페이스) 파일명을 수신용 연계(인터페이스) 파일명의 기준에 맞도록 수정한다.

※ 송신 시스템에서 생성하여 복사한 연계(인터페이스) 파일의 구조 또는 항목을 수신용 연계(인터페이스) 구조로 맞추기 위해 연계(인터페이스) 파일의 항목을 추가, 삭제함

3. 수신 시스템의 연계 응용 프로그램을 실행하여 단위 테스트를 수행하고, 결과를 검증한다.

수신 시스템의 연계 응용 프로그램을 실행하여 테스트 케이스(Testcase) 항목의 순서대로 테스트 한다.

테스트 결과는 테스트 케이스(Testcase)의 테스트 케이스 항목별 검증 방법으로 검증한다.

수행 tip

- 테스트 환경 구축 시 연계 메커니즘 정의서나 연계 아키텍처 설계서를 참고한다.
- 테스트 케이스 단위는 연계(인터페이스) 테이블이나 파일 단위로 작성한다.
- 테스트 케이스 항목은 정상적인 처리 흐름과 오류가 발생할 수 있는 경우의 수별 처리 흐름을 먼저 정의한 후 경우의 수별 테스트 절차를 참고하며 작성한다.

학습 3 교수 · 학습 방법

교수 방법

- 연계 메커니즘 정의서 또는 연계 아키텍처 설계서를 활용하여 연계 방식별 연계 모듈 구현 환경과 구성 요소, 환경 구축 방법 등을 설명한다.
- 연계 방식 중 EAI/ESB, 웹 서비스(Web Service) 연계 환경, 구성 요소, 구축 방법의 기본적인 사항을 설명하고, 비교 분석하여 차이점과 장단점을 제시한다.
- 송신용 연계 모듈과 수신용 연계 모듈의 처리 내용을 구현하기 위해 프로그램 설계서 작성 방법 및 작성 시점, 작성 항목, 활용 방법 등을 설명한다.
- 연계 메커니즘 정의서와 연계 방식, 연계 모듈의 프로그램 설계서에 따라 연계 모듈 구현 방법, 절차 및 구현 내용을 설명하고, 수행 내용에서 작성한 실습 결과물을 사례로 들어 분석 · 설계 단계 산출물과 개발 단계의 프로그램(Program) 소스(Source)인 스크립트(Script)와의 연관성을 설명하는 방향으로 진행한다.
- 연계(인터페이스) 테스트 범위 및 대상, 구간을 도식화하여 설명한다.
- 연계(인터페이스) 테스트를 위한 테스트 케이스 작성 방법, 목적, 항목, 순서 등을 실무에서 작성된 산출물 사례를 제시하며 설명한다.
- 단위 테스트와 연계(인터페이스) 테스트의 차이를 도식화하여 제시하고, 각 테스트 수행 절차를 설명한다.
- 테스트 환경 구축과 테스트를 위한 필수 구성 요소, 구성 요소별 기능 및 역할을 실제 테스트 환경 구축 사례를 활용하여 설명한다.
- 테스트 환경 구축을 위한 사전 준비 사항, 점검 사항, 수행 절차 등을 실제 테스트 환경 구축 사례를 활용하여 설명한다.
- 연계(인터페이스) 테스트 수행 절차와 진행 시 고려 사항, 제한 요건 등을 설명한다.
- 실제 프로젝트 수행 산출물인 단위 테스트 계획서와 결과서, 단위 테스트 케이스를 활용하여 단위 테스트 수행 절차와 순서, 항목, 검증 방법에 대해 설명한다.
- 실제 프로젝트 수행 산출물인 연계(인터페이스) 테스트 계획서와 결과서, 연계(인터페이스) 테스트 케이스(Testcase) 및 시나리오(Senario)를 활용하여 연계(인터페이스) 테스트 수행 절차와 순서, 항목, 검증 방법에 대해 설명한다.

학습 방법

- 다양한 연계 방식과 연계 방식에 따른 연계 시스템 환경, 연계 모듈, 연계 모듈 구현 환경 등을 파악한다.
- 연계 방식별 연계 모듈 구현 환경과 구성 요소, 구성 요소별 기능 및 역할, 구현 환경 구축 방법 등을 파악한다.
- EAI/ESB, 웹 서비스(Web Service) 연계 방식에 대한 연계 환경, 연계 시스템 구성 요소, 연계 시스템 구축 방법을 숙지한다.
- 연계 메커니즘 및 연계 시스템 구성 요소인 송신용 연계 모듈(연계 응용 프로그램), 수신용 연계 모듈(연계 응용 프로그램)의 처리 내용을 프로그램 명세서(설계서)로 작성한다.
- 송신용 연계 모듈(연계 응용 프로그램), 수신용 연계 모듈(연계 응용 프로그램)의 구현 방법, 절차를 이해하고, 프로그램 명세서(설계서)의 내용을 자바 프로그램 언어(Java Program Language)를 활용하여 구현한다.
- 연계(인터페이스) 테스트 과정에서 수행해야 할 작업, 테스트 과정별 수행해야 할 작업의 필요성, 작업 수행 결과 작성되는 산출물에 대해 숙지한다.
- 단위 테스트와 연계(인터페이스) 테스트의 차이를 숙지하고, 각 유형별 테스트 목적에 적합한 수행 절차, 방법을 테스트 케이스(Testcase) 및 시나리오(Senario)에 정의한다.
- 단위 테스트 케이스 작성 목적, 범위, 항목, 방법, 작성 시 주의 사항, 핵심 내용 등을 숙지하고, 워드(MS-Word) 또는 한글(HWP)과 같은 편집기를 활용하여 단위 테스트 케이스(Testcase)를 작성한다.
- 연계(인터페이스) 테스트 케이스 작성 목적, 범위, 항목, 방법, 작성 시 주의 사항, 핵심 내용 등을 이해하고 워드(MS-Word) 또는 한글(HWP)과 같은 편집기를 활용하여 연계(인터페이스) 테스트 케이스(Testcase) 및 시나리오(Senario)를 작성한다.
- 연계(인터페이스) 테스트 환경 구축을 위한 절차, 수행 시점, 점검 사항 등을 숙지한다.
- 수행 내용에서 제시한 시스템 환경에 따라 연계(인터페이스) 테스트 환경 구축 요소를 파악한다.
- 수행 내용의 실습 과정에서 송수신용 연계 응용 프로그램을 단위 테스트하고, 테스트 수행 결과를 검증한다.
- 수행 내용의 실습 과정에서 연계(인터페이스) 테스트를 수행하고, 테스트 결과를 검증한다.

학습 3 평 가

평가 준거

- 평가자는 학습자가 수행 준거 및 평가 시 고려 사항에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가한다.

학습 내용	평가 항목	성취수준		
		상	중	하
연계 모듈 구현 환경 구성 및 개발	- 구성된 연계 메커니즘에 대한 명세서를 참조하여 연계 모듈 구현을 위한 논리적, 물리적 환경을 준비할 수 있다.			
	- 구성된 연계 메커니즘에 대한 명세서를 참조하여 외부 시스템과 연계 모듈을 구현할 수 있다.			
연계 테스트 및 검증	- 연계 모듈의 안정적인 작동 여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다.			
	- 구현된 연계 모듈을 테스트할 수 있는 테스트 케이스(Testcase)를 작성하고, 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.			

평가 방법

- 평가자 체크리스트

학습 내용	평가 항목	성취수준		
		상	중	하
연계 모듈 구현 환경 구성 및 개발	- 연계 방식별 연계 환경 및 구성 요소별 기능과 역할 이해			
	- 송수신 연계 모듈의 역할과 처리 내용 이해			
	- 송수신 연계 모듈의 처리 내용을 프로그램 설계서로 작성			
	- 송수신 연계 모듈 구현을 위한 연계 환경 및 시스템 구성			
	- 송수신 연계 모듈의 프로그램 설계서 이해 및 프로그램으로 구현			
연계 테스트 및 검증	- 연계 테스트의 범위와 대상 이해 및 정의			
	- 단위 테스트와 연계 테스트의 차이 이해 및 구분			
	- 송수신 연계 모듈의 테스트 케이스(Testcase) 작성 및 명세화			
	- 연계 테스트 케이스(Testcase)·시나리오(Senario) 작성 및 명세화			
	- 송수신 연계 모듈 단위 테스트 및 연계 테스트 수행을 위한 연계 테스트 환경 구축과 응용 구성			
	- 송수신 연계 모듈의 단위 테스트 수행과 결과 검증			
	- 연계 테스트 수행과 결과 검증			

● 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
연계 모듈 구현 환경 구성 및 개발	- 송수신 연계 모듈 프로그램 설계서 작성의 구체성, 논리성, 적합성, 일치성, 완전성			
	- 작성한 송수신 연계 모듈의 프로그램 설계서에 따른 연계 모듈 구현			
	- 송수신 연계 모듈 구현의 완전성, 정확성, 논리성, 일치성, 현실성			
연계 테스트 및 검증	- 송수신 연계 모듈 단위 테스트 케이스(Testcase) 작성의 완전성, 적합성, 구체성, 충분성, 일치성			
	- 연계 테스트 케이스(Testcase) 작성의 완전성, 적합성, 구체성, 충분성, 일치성			
	- 송수신 연계 모듈 단위 테스트 수행과 결과 검증의 적합성, 구체성, 충분성, 정확성, 완전성, 단위 테스트 케이스(Testcase)와 실제 테스트 수행 내용의 일치성			
	- 연계 테스트 수행과 결과 검증의 적합성, 구체성, 충분성, 정확성, 완전성, 연계 테스트 케이스(Testcase)와 실제 테스트 수행 내용의 일치성			

피 드 백

1. 체크리스트(Checklist)를 통한 관찰
   - 송수신 연계 모듈 프로그램 설계서 작성 실습 과정에서 송수신 연계 모듈의 역할과 처리 내용의 이해, 송수신 연계 모듈의 처리 내용을 프로그램 설계서로 작성, 처리 내용의 응용 정도를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 송수신 연계 모듈 구현 환경 구성 실습 과정에서 연계 방식별 연계 환경 및 구성 요소별 기능과 역할에 대한 이해, 송수신 연계 모듈 구현을 위한 연계 환경 및 시스템 구성에 대한 이해와 응용 정도를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 송수신 연계 모듈을 프로그램으로 구현하는 실습 과정에서 송수신 연계 모듈의 프로그램 설계서에 대한 이해, 송수신 연계 모듈의 프로그램 설계서를 실제 프로그램으로 구현한 수준을 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 송수신 연계 모듈 단위 테스트 케이스(Testcase) 작성 실습 과정에서 연계 테스트의 범위와 대상 이해 및 정의, 단위 테스트와 연계 테스트의 차이 이해 및 구분, 송수신 연계 모듈의 테스트 케이스(Testcase) 작성 및 명세화 정도를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 연계 테스트 케이스(Testcase) 작성 실습 과정에서 연계 테스트의 범위와 대상 이해 및 정의, 단위 테스트와 연계 테스트의 차이 이해 및 구분, 연계 테스트 작성 및 명세화 정도를 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 송수신 연계 모듈 테스트 구성 실습 과정에서 송수신 연계 모듈 단위 테스트 및 연계 테스트 수행을 위한 연계 테스트 환경 구축 및 응용 구성을 평가한 후에 개선해야 할 사항 등을 정리하여 돌려준다.
   - 송수신 연계 모듈의 단위 테스트와 연계 테스트 실습 과정에서 송수신 연계 모듈의 테스트 수행 진행 정도, 송수신 연계 모듈 단위 테스트 결과 검증 수준, 연계 테스트 수행 진행 정도, 연계 테스트 결과 검증 수준 등을 평가한 후에 개선해야 할 사항들을 정리하여 돌려준다.

2. 포트폴리오(Portfolio)
   - 제출한 송수신 연계 모듈의 프로그램 설계서 작성 내용 구체성, 상세화, 논리성, 적합성, 완전성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 제출한 송수신 연계 모듈 프로그램의 구현 완전성, 기능 동작 정확성, 논리성, 일치성, 처리 내용 현실성, 작성한 송수신 연계 모듈의 프로그램 설계서와의 일치성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 제출한 송수신 연계 모듈 단위 테스트 케이스(Testcase)의 테스트 케이스 항목 완전성, 적합성, 구체성, 충분성, 일치성 및 송수신 연계 모듈 단위 테스트 수행과 결과 검증의 적합성, 구체성, 충분성, 정확성, 완전성, 단위 테스트 케이스(Testcase)와 실제 테스트 수행 내용의 일치성 등 주요 사항에 대하여 표시하여 돌려준다.
   - 제출한 연계 테스트 케이스(Testcase)의 테스트 케이스 항목 완전성, 적합성, 구체성, 충분성, 일치성 및 연계 테스트 수행과 결과 검증의 적합성, 구체성, 충분성, 정확성, 완전성, 연계 테스트 케이스(Testcase)와 실제 수행 내용과의 일치성 등 주요 사항에 대하여 표시하여 돌려준다.

연계 요구 사항 분석서

요구 사항 ID	요구 사항명	요구 사항 유형	요구 사항 설명	해결 방안	중요도	우선순위	난이도	확인 기준

코드 변환 정의서

송신 시스템 코드					수신 시스템 코드				
코드 그룹	코드 그룹명	코드 길이	코드	코드명	코드 그룹	코드 그룹명	코드 길이	코드	코드명

연계(인터페이스) 정의서

I/F 번호	송신		수신		인터페이스 방식	인터페이스 주기
	시스템명	인터페이스 ID/명	시스템명	인터페이스 ID/명		

연계(인터페이스) 명세서

I/F 번호			I/F명		
송신	I/F ID		수신	I/F ID	
	I/F명			I/F명	
주기 및 방식			DB 및 파일 형식	◦ DB ◦ FILE ( )	

송신						수신					
한글명	영문명	타입	길이	PK	Code 여부	한글명	영문명	타입	길이	PK	Code 여부
처리 내용											

NCS학습모듈 개발이력

발행일	2015년 12월 31일		
세분류명	응용SW엔지니어링(20010202)		
개발기관	한국소프트웨어기술진흥협회, 한국직업능력개발원		
집필진	강석진(이비스톰)* 김보운(이화여자대학교) 김홍진(LG CNS) 유은희 장현섭((주)커리텍) 주선태(T3Q) 진권기(이비스톰) 최재준	검토진	김승현(경희대학교) 엄기영(우리에프아이에스) 장온순(한국IT컨설팅) 조상욱(세종대학교) 조성호(삼성카드) *표시는 대표집필자임
발행일	2018년 12월 31일		
학습모듈명	통합 구현(LM2001020206_16v4)		
개발기관	한국직업능력개발원		

통합 구현(LM2001020206_16v4)

저작권자	교육부
연구기관	한국직업능력개발원
발행일	2018. 12. 31.

※ 이 학습모듈은 자격기본법 시행령(제8조 국가직무능력표준의 활용)에 의거하여 개발하였으며, NCS통합포털사이트(http://www.ncs.go.kr)에서 다운로드 할 수 있습니다.

National Competency Standards Learning Module

NCS
학습모듈

www.ncs.go.kr