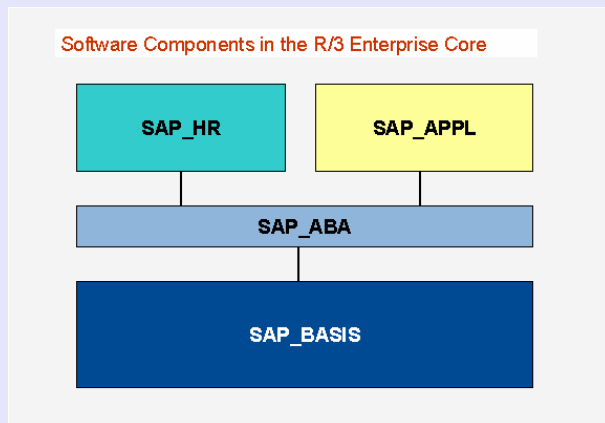


1. OVERVIEW



The software component hierarchy

SAP_BASIS - SAP_ABA - SAP_HR
- SAP_APPL

▲그림1-1. software component hierarchy

1-1. PACKAGE

Package는 프로그램과 같은 오브젝트들을 기능별로 관리하기 위하여 도입된 개념으로 다음과 같은 기능을 한다.

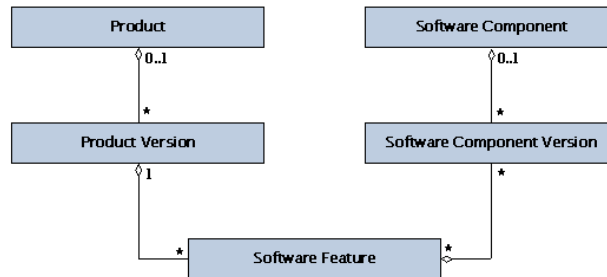
- 1) 오브젝트(프로그램등)의 구조화(structure)
- 2) 오브젝트 기능을 모듈화(encapsulate, modularize)
- 3) 오브젝트를 기능별로 구별(classify)

* 모듈별로 개발패키지를 각각 생성하여 사용한다.

예를들어 생산 모듈은 ZPP

회계 모듈은 ZC0

Software component	Description
SAP_BASIS	SAP Web Application Server (formerly: SAP Basis)
SAP_ABA	Cross-application components
SAP_HR	Human resources
SAP_APPL	Logistics and Accounting
HOME	This software component contains packages with objects that are not local, but that are not delivered to customers
LOCAL	Contains packages that contain only local objects



▲그림1-2. Software component 구조

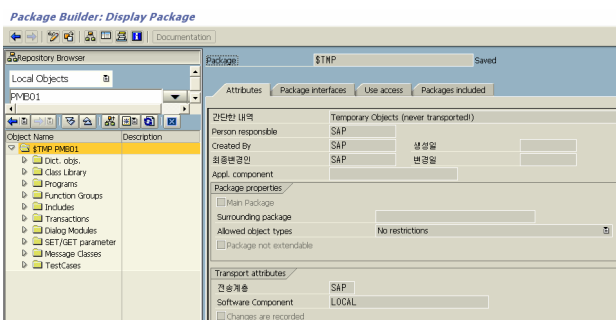
1-2. SOFTWARE COMPONENT

- Package들을 모아 한 set로 관리하는 그룹이다.
 - 오브젝트는 Package에 귀속되고, Package는 Software component에 할당되어 Customer에게 인도된다.
- 예를들어, LOCAL Component는 자동전송을 하지 않으므로, CTS를 생성하지 않는다.
다음장 참고

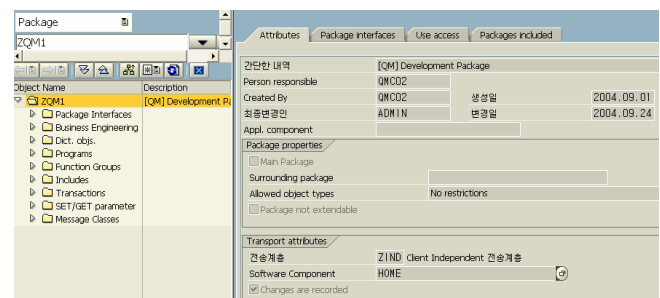
1-3. Software Component 사용의 목적?

- Software Component는 버전별로 관리되며, SAP UPGRADE 또는 Patch를 적용 할수 있도록 설계되어 있다.
- SXMB_IFR 트랜잭션을 통해 생성함

[그림1-3]은 Local Object로 프로그램 개발 테스트용으로 만들어지는 패키지이다. 이 패키지는 Softwar Component가 LOCAL 이며, “Changes are recorded”에서 CTS가 생성되지 않음을 확인할 수 있다. [그림1-4]의 Package는 “HOME” Software Component에 속하며 변화관리를 통해 CTS 가 생성이 된다.

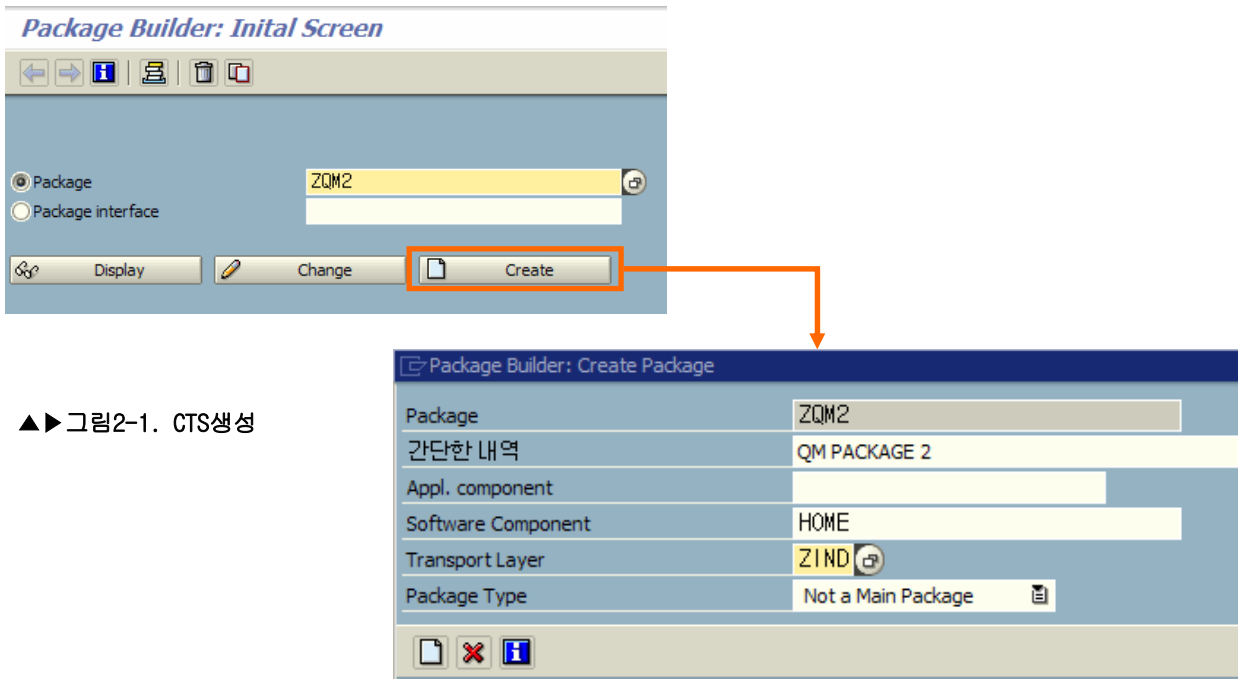


▲그림1-3. Software component 예



▲그림1-4. Software component 예

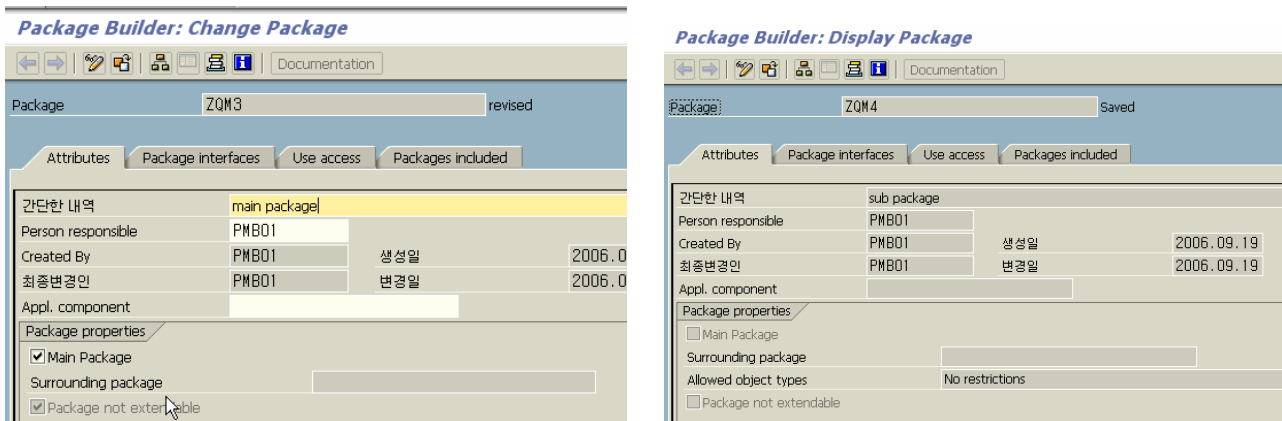
2. 패키지 생성



▲▶ 그림2-1. CTS생성

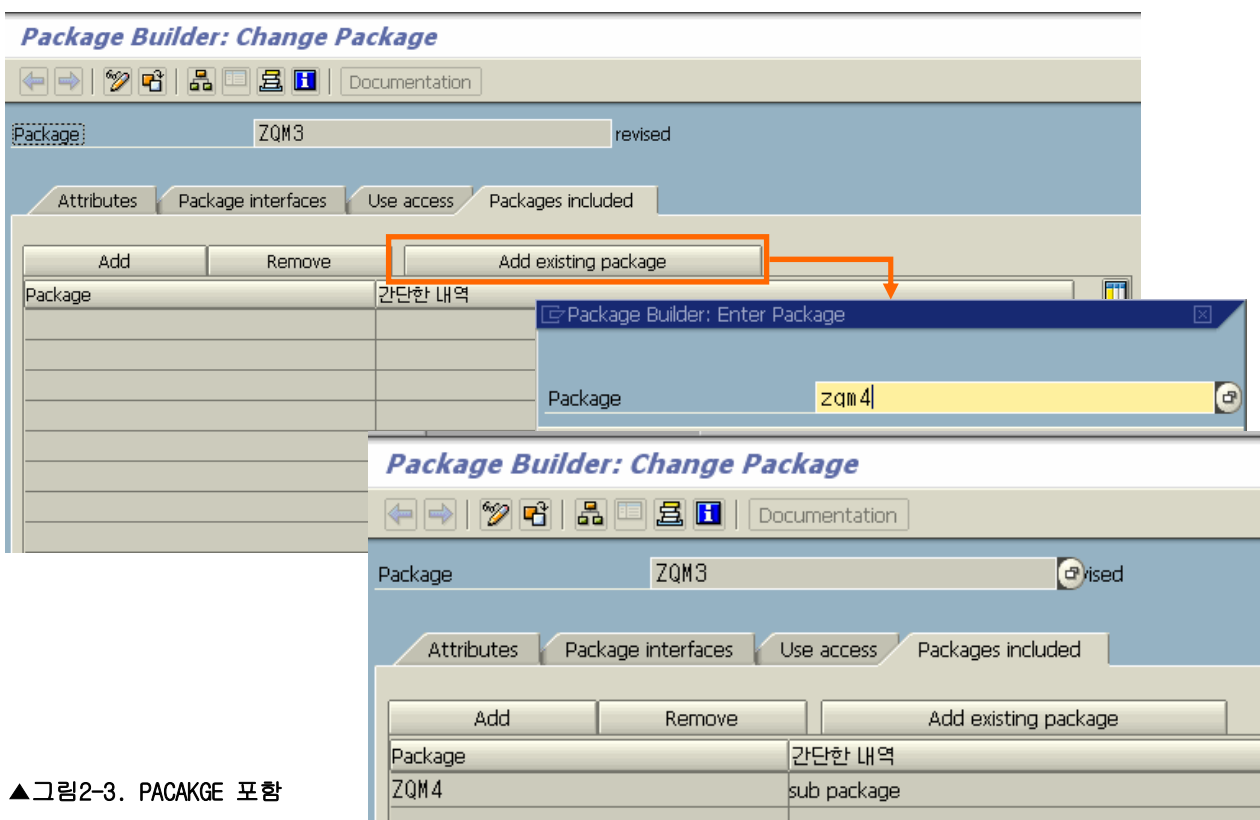
1. 패키지명을 입력하고 생성버튼을 클릭한다.
(transaction SE21 or transaction SPACKAGE)

※ 패키지명은 30자이내로 구성하되, Y 또는 Z로 시작하도록 한다.
A~S, X~U는 SAP standard 시스템에서 사용하도록 예약되어 있다.
2. Appl.Componet는 필수사항이 아님.
3. Software Component는 Customizing으로 제공된 'HOME' 고객개발을 선택한다.
이외 componet는 SAP에서 사용한다.
4. Transport Layer은 'ZIND'를 선택한다, 클라이언트 Independent임을 의미한다.
5. Package Type
'Not a Main Package' 선택
Main Package와 Sub Package를 생성하여
M.P에 S.P를 추가할수 있다.

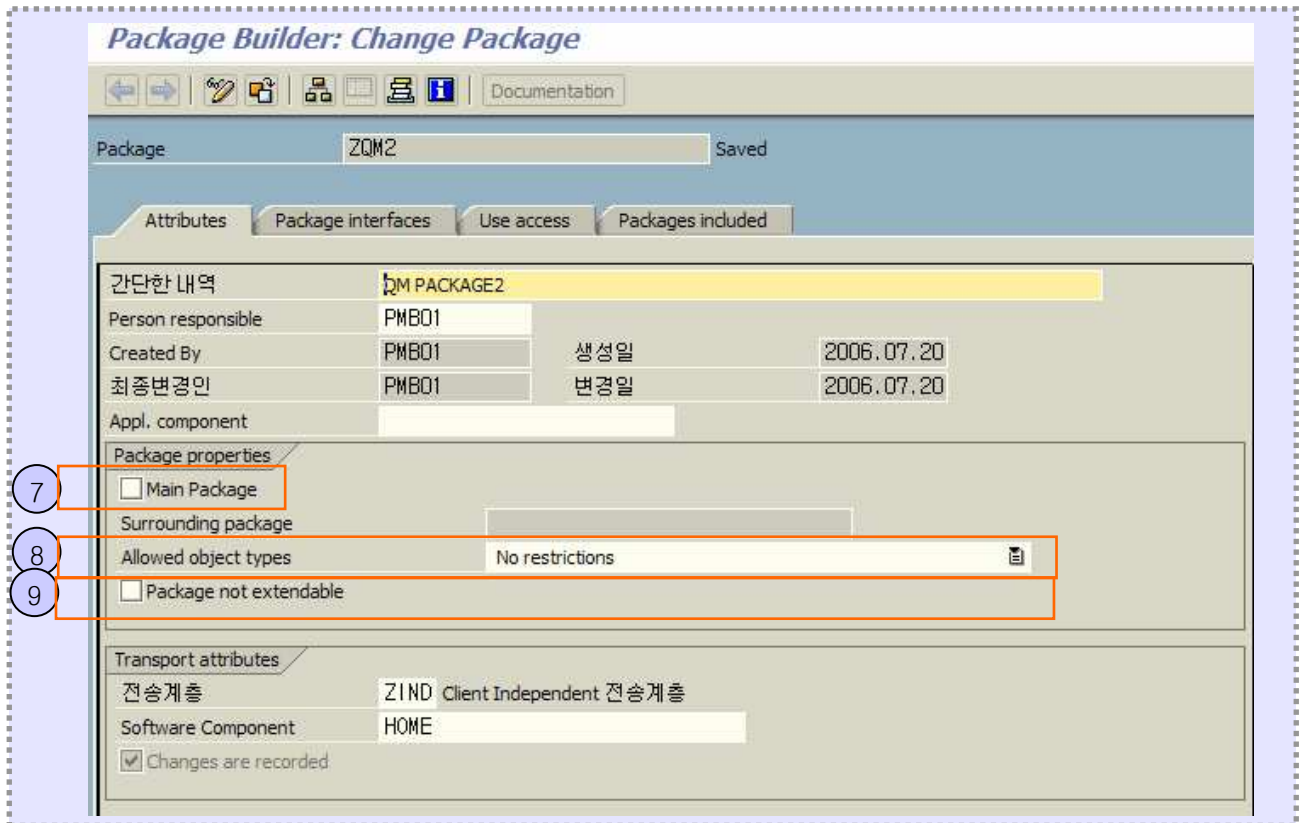


▲그림2-2. PACAKGE 속성

6. 패키지는 Main Package, Sub Package로 구분되어 Main은 sub를 포함 할 수 있다. [그림2-2]에서 ZQM3, ZQM4는 Main Package, SUB Package 이며 [그림2-3]과 같이 Main 패키지에 Sub 패키지를 추가할 수 있다. 위와 같이 package를 Module화 하여 관리할 수 있다.

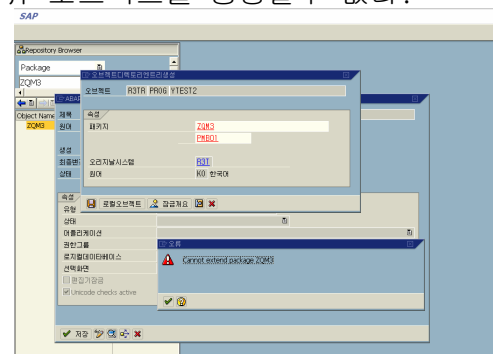


▲그림2-3. PACAKGE 포함

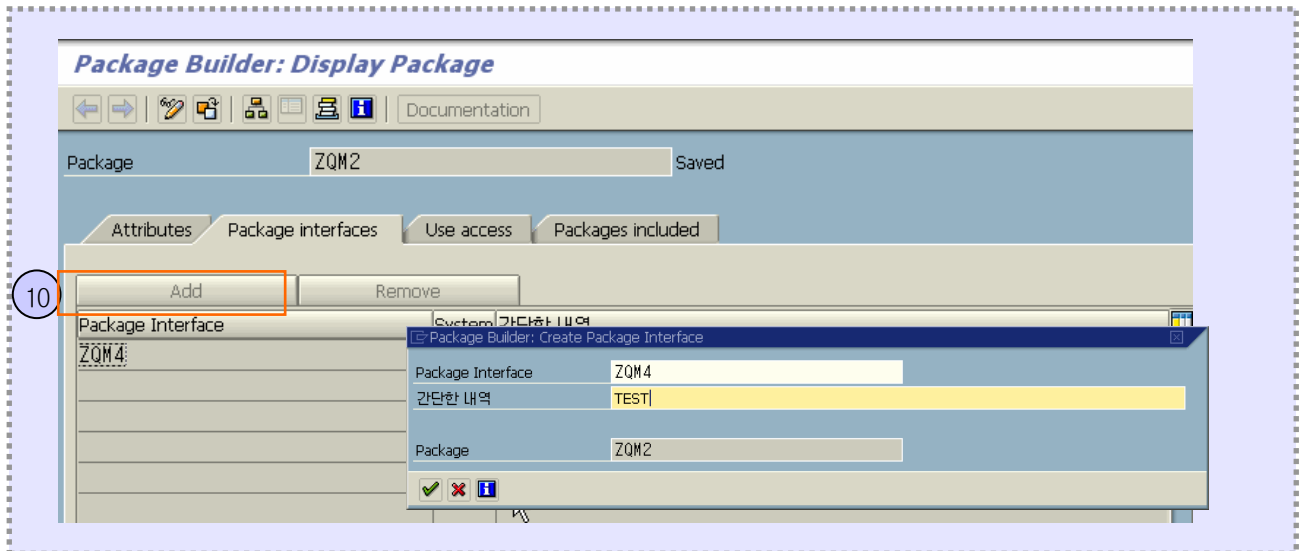


▲그림2-4. PACKAGE 속성

7. Main, User(Sub) package 구분자
8. Package 내에 포함할수 있는 오브젝트 타입을 지정한다.
(프로그램, 테이블, 평션등)
No restrionts를 선택하면 제한이 없다.
9. Package not extndable이 체크되면, 패키지내에 신규 오브젝트를 생성할수 없다.
Main package의 기본속성으로 변경할 수 없다.
즉, Main package는 user package만 추가할수 있다.
user packages는 다른 user package를 포함할 수 있다.



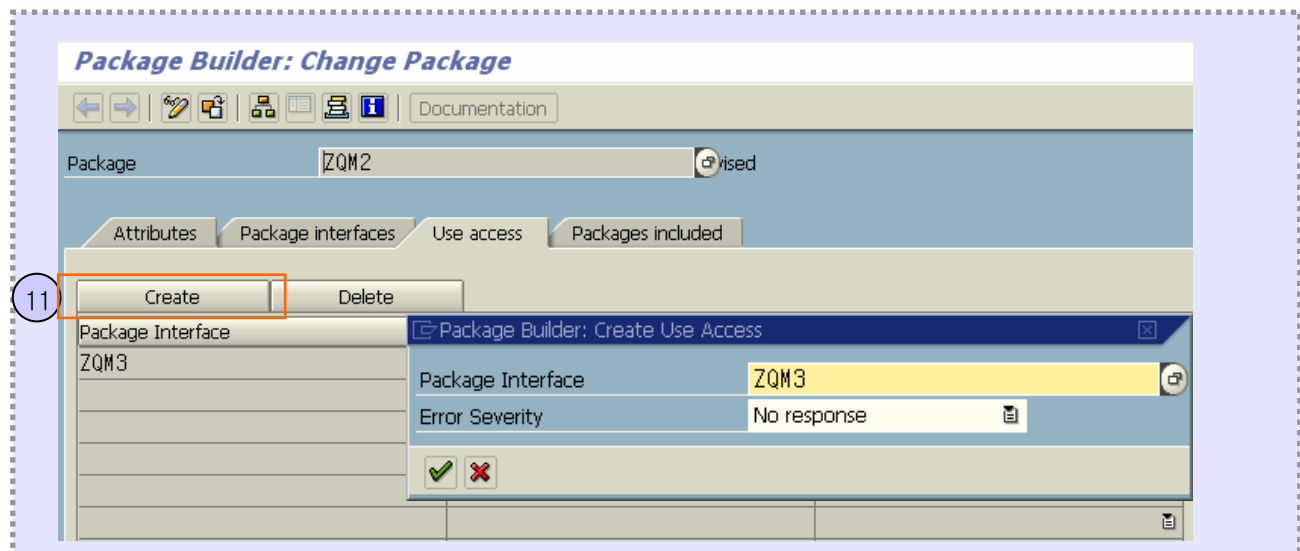
▲그림2-5. Package not extendable



▲그림2-5. PACKAGE Interface 추가

10. ZQM2 PACKAGE에서 사용될 인터페이스를 생성하여 추가할 수 있다.

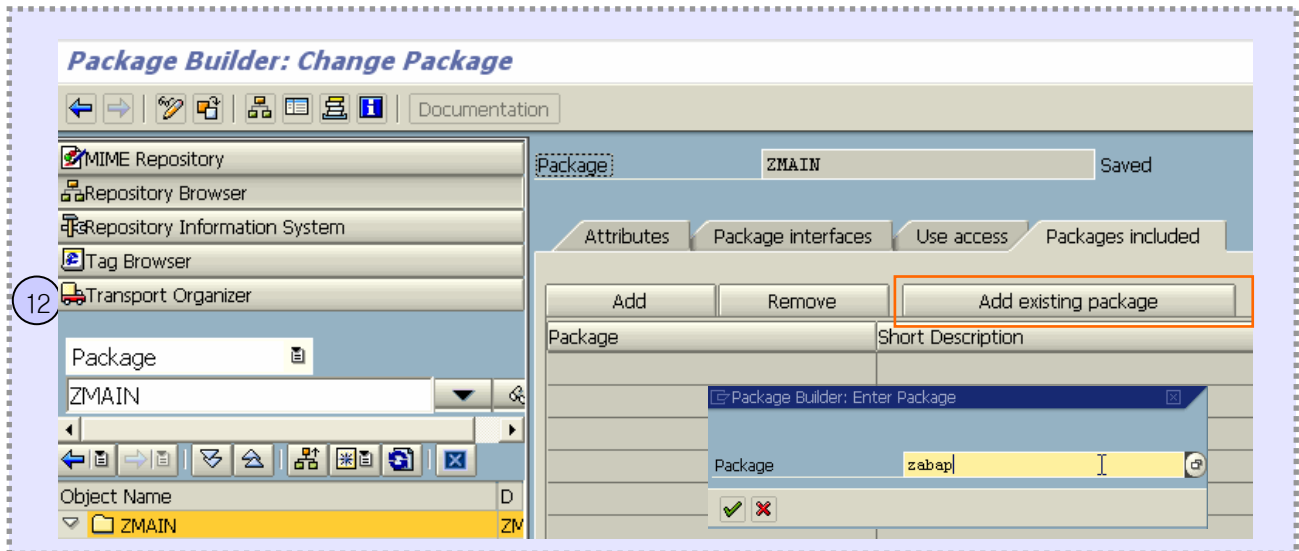
* 패키지 인터페이스는 챕터 후반에서 다룬다.



▲그림2-6. USE ACCESS

11. USE ACCESS

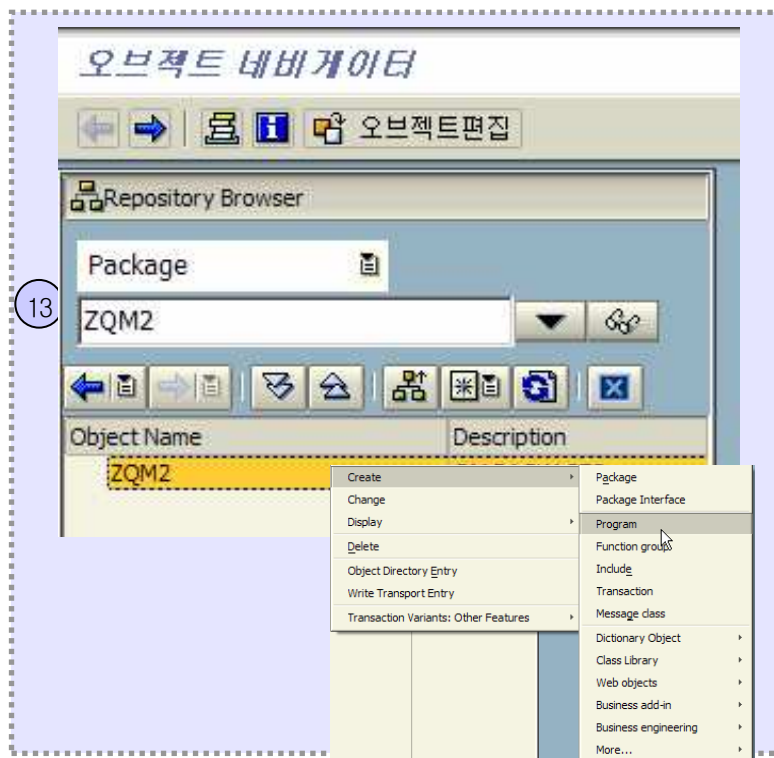
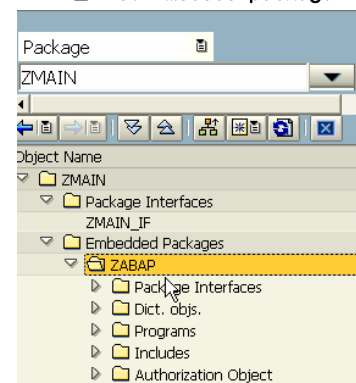
- USE ACCESS를 설정하지 않으면, User package가 모든 element를 이용할 수 없다.



▲그림2-7. USE ACCESS

12. Add exsting package를 통해 동일 레벨의 user package를 추가 할 수 있다.
[그림2-8]에서 추가된 user package는 embeded package로 소속된다.

▼ 그림2-8. Embeded package



13. [그림2-9] 생성된 패키지에 오브젝트를 추가하는 화면이다.

◀ 그림2-9. USE ACCESS

VALUE	Meaning
NONE	Default value The system does not respond when you use interface elements(true permission).
INFO	If you use interface elements, the system displays an Information message when you perform an extended program check (transaction: SLIN).
WARN	If you use interface elements, the system displays an Warning message when you perform an extended program check (transaction: SLIN).
ERRO	If you use interface elements, the system displays an Error message when you perform an extended program check (transaction: SLIN).
OBSO	After performing an extended program check, the system characterizes the interface elements as obsolete - that is, they should no longer be used.

▲표 USE ACCESS 권한

3. Package Interface

Package interface는 실제 존재하는 패키지가 아니라 패키지간에 인터페이스가 가능하도록 한 가상의 패키지이다.

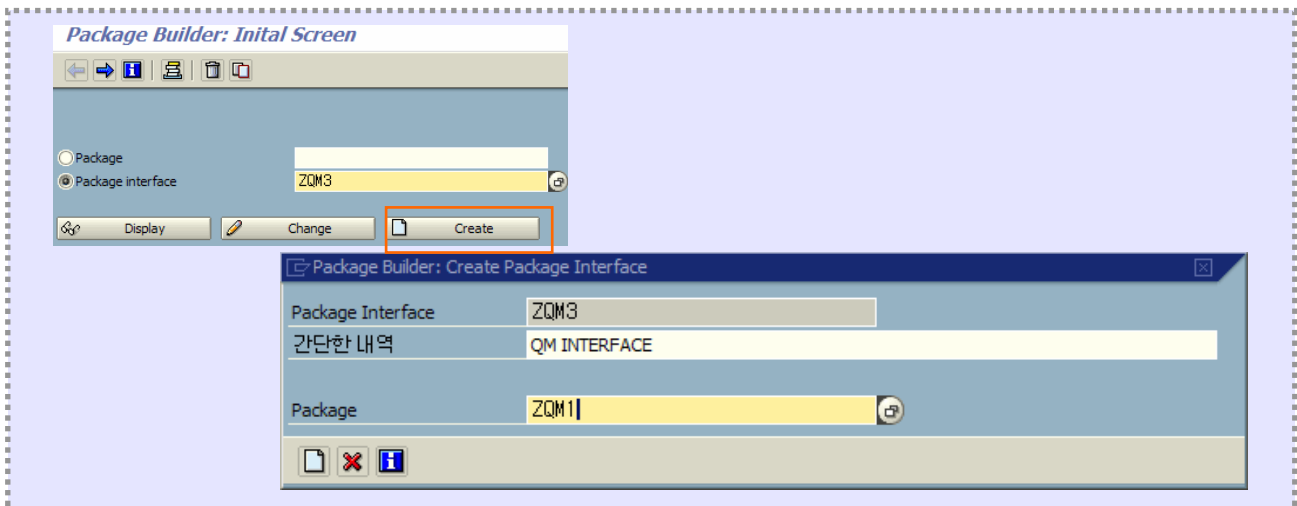
왜 필요한가?

QM모듈안에는 밀접한 연관성을 가진 오브젝트 (프로그램, 평션) 를 기능별 가상의 패키지로 관리하고 싶을 경우가 있다.

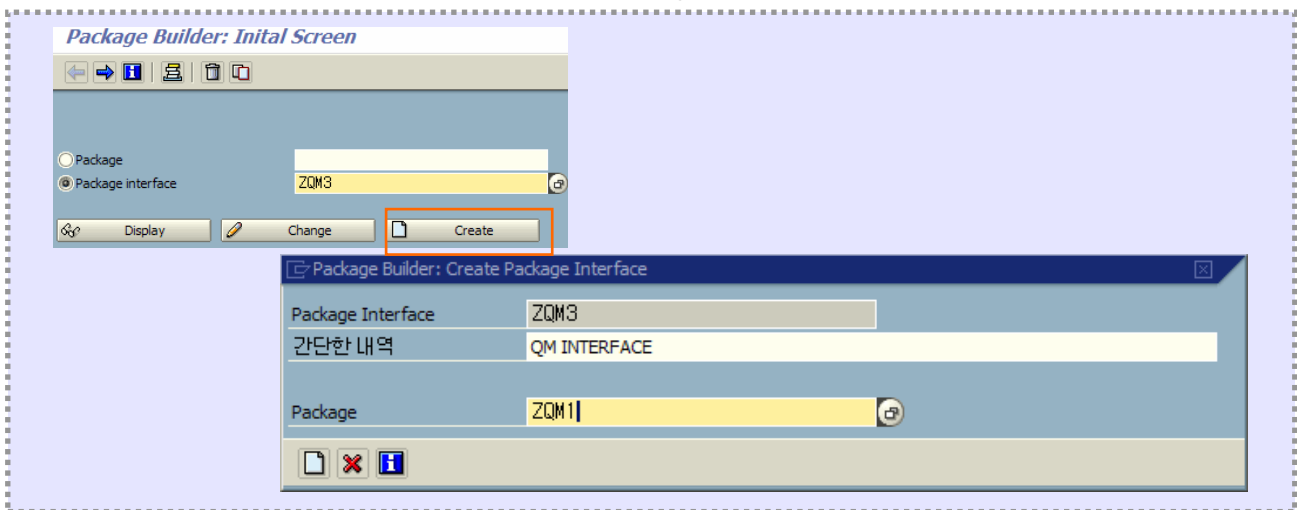
이 경우 가상의 패키지를 생성하여, 오브젝트를 추가할 수 있다.

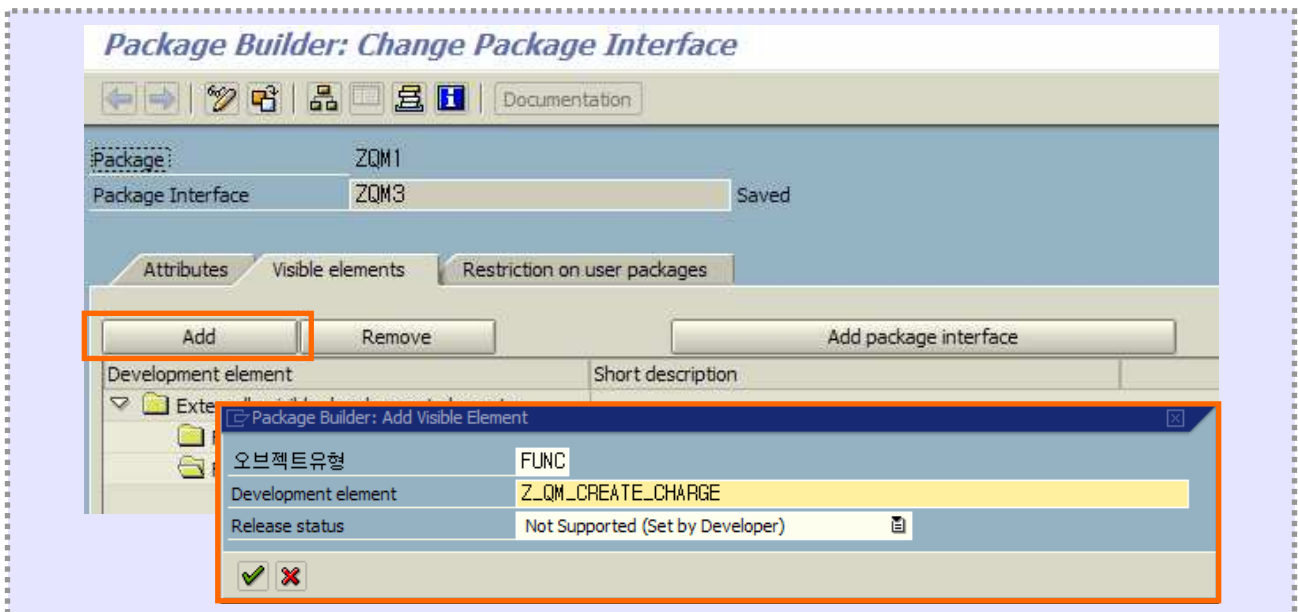
예를들면) 고객불만과 관련된 프로그램과 함수가 여러 개 존재할 경우 고객불만 패키지 인터페이스를 생성하여, 관련 프로그램 리스트를 관리할수 있다.

1. [그림3-1]에서 ZQM3, ZQM4 이라는 패키지 인터페이스를 생성해보자.

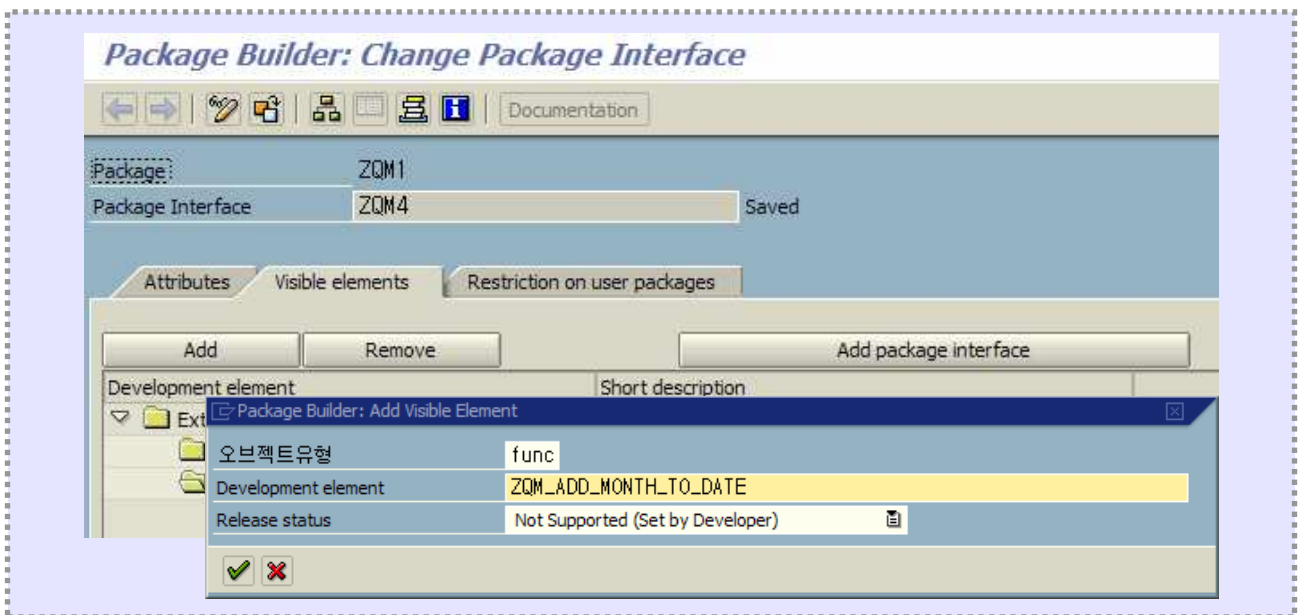


▲▼그림3-1. Package Interface



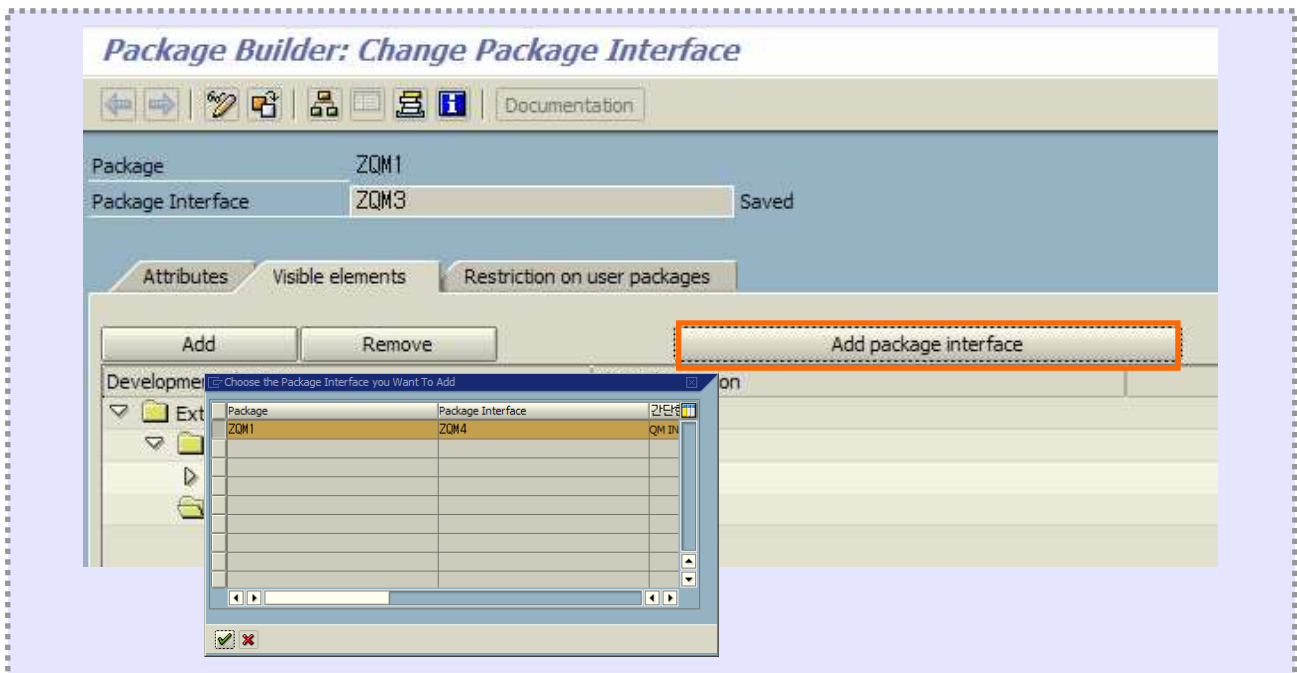


▲▼그림3-2. element 추가



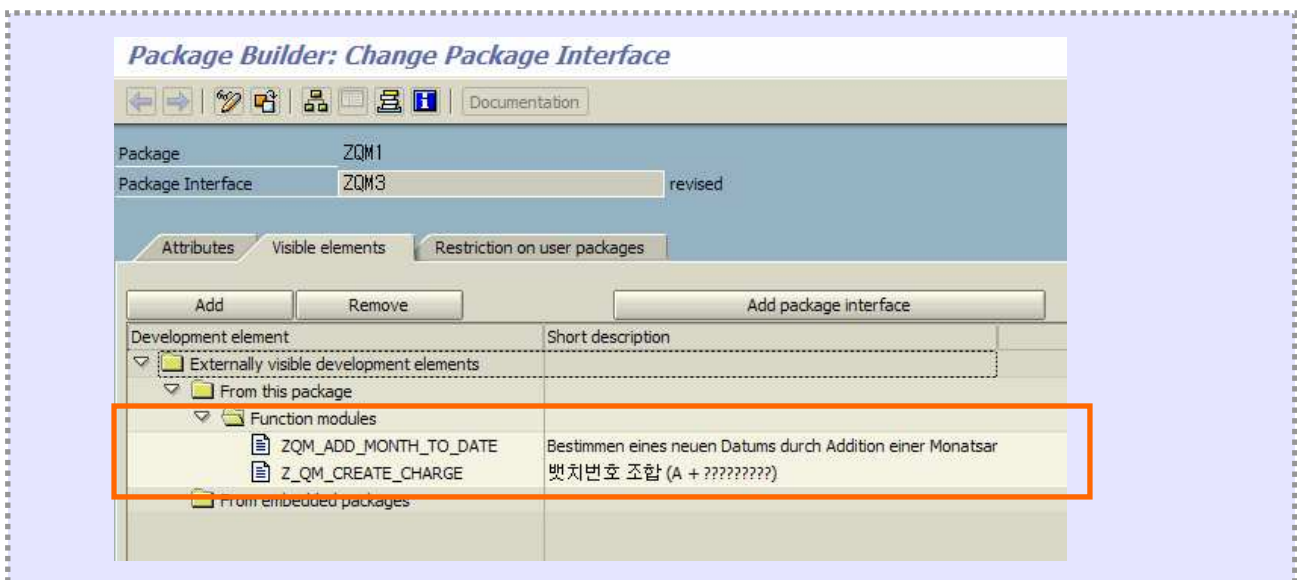
2. Visible elements TAB 선택 Add 버튼 클릭하여 ZQM1 패키지의 함수를 추가해준다.
오브젝트 유형은 FUNC 선택, Development element는 zqm1 패키지의 함수를 입력한다.

* ZQM4 패키지 인터페이스에는 동일한작업을 수행하며, ZQM_ADD_MONTH_TO_DATE평션을 추가한다.

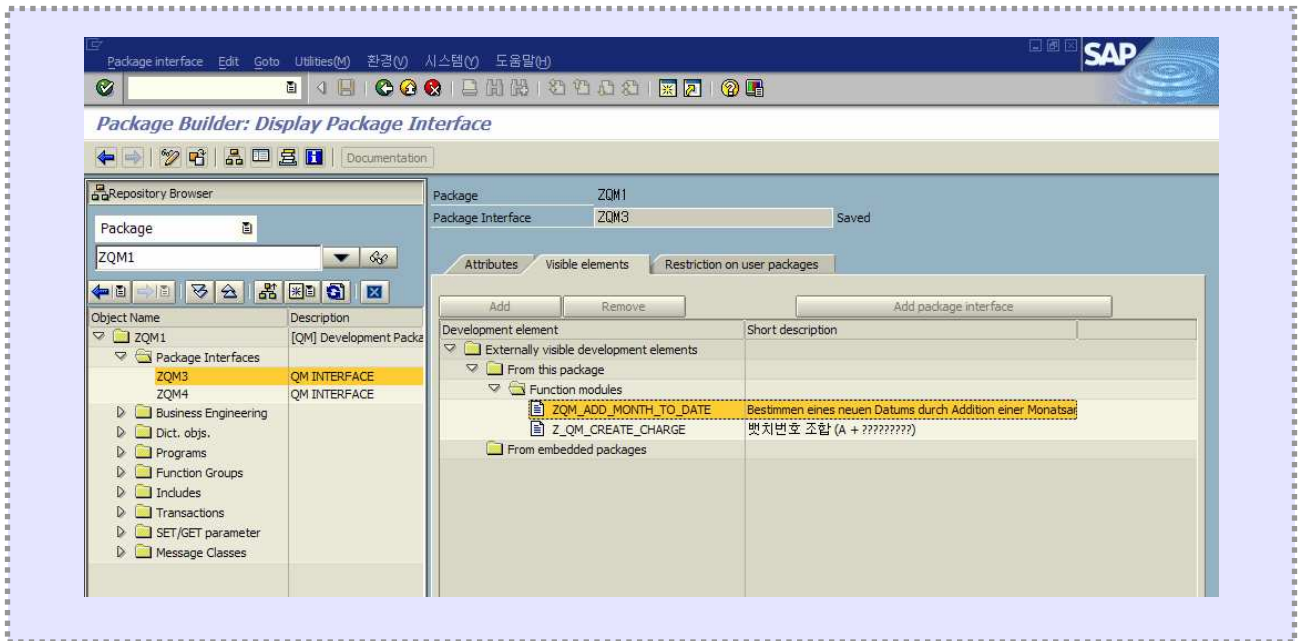


▲그림3-3. Add package Interface

3. ZQM3 패키지 인터페이스에 ZQM4 인터페이스를 추가한다. [그림3-3]
4. FUNCTION 모듈에 2개의 함수가 추가되었음을 확인할 수 있다. [그림3-4]



▲그림3-4. Add package Interface 조회



▲그림3-5. Add package Interface

5. ZQM1 패키지에서 ZQM3 패키지 인터페이스를 통해, 추가된 오브젝트(함수)를 조회 및 수정할수 있다.