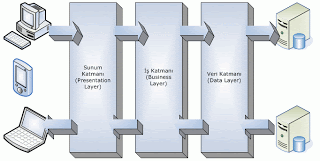
ABAP TEMEL BİLGİLER

ABAP(Advanced Business Application Programming ) yazılım şirketi SAP tarafından geliştirilmiş bir programlama dilidir. Java,C#…gibi yüksek seviyeli diller arasında yerini almaktadır. ABAP ın en temel kullanım amacı SAP üzerinde müşterinin isteklerine uygun uyarlamanın yapılmasıdır. Yani temel SAP yazılımının üzerine müşterinin istediği herhangi bir eklentinin yapılması gerektiğinde ABAP devreye girer. Bunun dışında ilk ve en çok kullanım amacı ise SAP sistemindeki verilerin istenilen biçimde raporlanarak sunulması,rapor programı yazılmasıdır. Bir rapor programında amaç kullanıcının sisteme girdiği veriler üzerinde işlem yapmaktır. En basit örnekle : bir çikolata şirketiniz olsun ve işlerinizi takip etmek için SAP kullannıyorsunuz. İşte belli iki tarih arasında hangi üründen nekadar sattığınızın raporu için ABAP ile raporlama programları yazılır. Toparalayacak olursak ABAP ile gerekli bilgileri veritabanından okuyan ve ekrana sunan report programlar, sisteme bilgi girilmesini sağlayan dialog programlar ve SAP ile diğer sistemler arası veri alışverişini sağlayan Interface’ler yazılabilir.

ABAP’ın önemli özelliklerinden biri çevirilebilen bir metin yapısının olması, bu sayede birden fazla dilde çalıştırılabilen programlar elde ediyoruz. Diğer bir özelliği ise OPEN-SQL yapısının olması ve sistem üzerinde direk program satırlarınızı yazarken veri tabanına erişim mümkün yani connection-stringsiz bir yapısı var. SAP sisteminin kendi içerisinde var olan veri tabanı tabloları var. Elbette bununla da sınırlı kalmıyor yani sadece kendi veri tabanını desteklemekle kalmıyor oracle, mssql gibi veri tabanlarınıda destekliyor. Yani her şekilde şirketinize uyum sağlıyor. Bir diğer özelliği her geçen gün gelişen yapısının eski kodlara zarar vermemesi. Bu da bize nesneye dayalı bir programlama(OOP) yapısına sahip olduğunu gösteriyor.

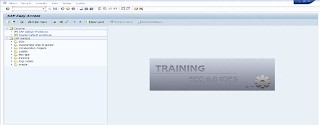
Temelde bildiğimiz 3 katmanlı mimari yapısı. SAP R/3,3 katmandan oluşan bir client/server mimarisidir. Sırası ile Database Layer(veritabanı katmanı), Application Layer(uygulama katmanı) ve Presentation Layer (sunum katmanı).



-Database Layer : Bu katmanda işletmenin bilgileri, girilen veriler bunun yanısıra metadata yani kullnılan veritabanı yapısını tarif eden metaveriler saklanır. Verilerle ilgli işlemler için SQL kullanılır ve daha öncede değindiğimiz gibi piyasadaki pek çok veritabanını destekler.

-Application Layer : Bu katmanda SAP tarafından geliştirilmiş uygulamalar, ABAP tarafında yazılmış programlar çalışır. Yani veritabanı üzerine yerleştirilmiş uygulamalar bu katmanda çalışır diyebiliriz.

-Presentation Layer : Kullanıcı tarafından verilerin girildiği, veriler üzerinde işlem yapılabildiği katman bu katmandır. SAP GUI(graphical user ınterface) kullanıcı arayüzümüzdür. En basit hali ile aşağıdaki şekilde karşımıza çıkar.



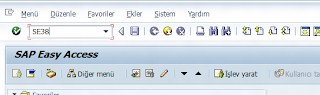
ABAP PROGRAMLAMA DİLİNE GİRİŞ

Geliştirmeleri yaptığımız ortam ABAP Workbench.Bu çalışma ortamını tam anlamıyla kullanmak için SAP sistemine erişiminizin olması gerekmektedir.Yani gerçek uygulamalar için bir SAP danışmanlık şirketinin üyesi olmanız gerek.Ancak bu erişime sahip değilseniz ve ABAP programlama dilini öğrenmek istiyorsanız

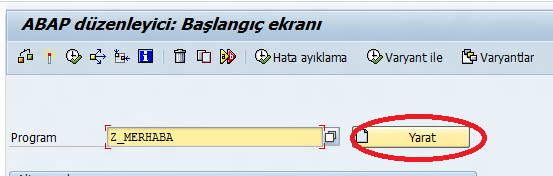
http://www.sdn.sap.com/irj/scn/downloads adresinden bilgisayınızla uyumlu deneme versiyonlarını indirip kurabilirsiniz.İçinde ABAP workbench i taşıyan en rahat sistem Mini SAP Basis dir.Ancak en kapsamlı olan SAP Netweaver Application Server deneme versiyonu.

ABAP Workbench : ABAP kod editörü(ABAP editör), fonksiyon modül editörü(function builder), kullanıcı ekran oluşturucu(screen painter), veri türü sözlüğü(data dictionary) ve sınıf oluşturucudan(class builder) oluşur. SAP pek çok modülü kapsayan, ayrıntısı oldukca fazla olan bir sistemdir. Bu sistem içerisinde ilgili alanlara erişebilmek için transaction kodları dediğimiz kısa yollar mevcuttur. ABAP ın bütün parçalarına eriştiğimiz transaction kodu : **SE80** dir.Sadece kod geliştirme ortamına erişmek için kullandığımız kod ise **SE38** dir. Şimdi ABAP da kod yazacağımız ortamı tanıyıp ardından programlama klasiği olan **ekrana ‘merhaba dünya’ yazan programcığı gerçekleştirelim** :)

**1-**İlk adımımız SE38 transaction kodu ile ABAP editörüne geçiş yapmak.

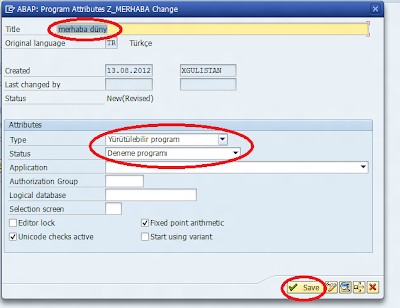


**2-**Bizi yazacağımız programa dair özellikleri vermemiz bildirmemizin istendiği ekran karşılayacak.

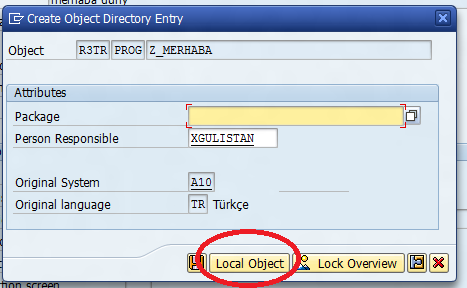


Programımıza İsmini veriyoruz. Burada bilmemiz gereken şey ABAP da sisteme dahil olmayan yani **programcı tarafından extra olarak yazılmış programlara isim verilirken ‘Z’ yada ‘Y ‘ ile başlanıyor olması. Bu durum ABAP’a özel bir özellik, kural haline gelmiş bir durum.** Programa ‘Z\_MERHABA’ ismini veriyoruz ve yarat seçeneğine tıklıyoruz.

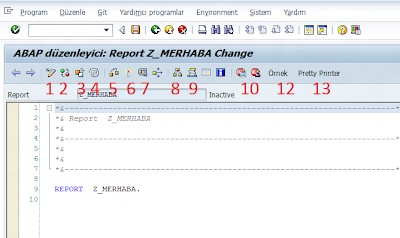
**3-**Şimdi ise programa dahil özelliklerin belirleneceği bir alan karşımıza çıkıyor. İlk alanda programla ilgili açıklamayı belirtmemiz gerekiyor. Bunu daha sonra yazdıklarımızla karışmaması ya da bizden başka okuyacak olanlara anlaşılma sağlaması için yapıyoruz. İkinci alanda programın tipi devreye giriyor ilk uygulamamız olacağı için ‘yürütülebilir programı’ seçiyoruz. Hemen altında programın statüsü var. Burada ‘deneme programı’nı seçiyoruz ve ‘save‘ diyoruz.



**4-**Bir sonraki adımda paket seçimi yapmamız gerekiyor.Ancak şuan yazacağımız program herhangi bir pakete dahil edilemeyeceği için ‘**local object**’ i seçiyor ve artık kodları yazacağımız editör ortamımıza ulaşıyoruz.



**5-**ABAP editöründeyiz artık. Kodlarımızı burada yazıp derleyeceğiz. İlk olarak notepad’i andıran kelime işlemci arayüzü bizi karşılar. Programın iilk satırları ABAP ediötürü tarafından otomatik olarak eklenir. Bu satırlar yorum satırıdır ve programın adını belirler. Bir sonraki adıma geçmeden ortamı tanıyalım.



\* 1-Düzenleme-görüntüleme modu arası geçiş tuşu (programı daha sonra tekrar çalışmak için kaydettiğimiz zaman değişiklik yapabilmek için düzenleme moduna geçmemiz gerekiyor yoksa kodlarımız üzerinde değişiklik yapmaya imkanımız olmuyor)

\* 2- Etkin olan-olmayan sürümler arası geçiş tuşu,

\* 3- Programın içinden veya dışından başka bir nesneye erişim tuşu,

\* 4- Enhancement kodlamaları,

\* 5- Syntax kontolü(varsa hata hangi satırsa onu bulur),

\* 6- Programın aktifleştirilmesi. (her program çalışmadan önce aktifleştirilmeli aksi halde hatalara sebep oluyor) ve bütün ABAP programları ilk yazıldığında inaktiftir.

\*7- Run.Programın yürütülmesi gerçekleştiriliyor(kısayolu F8),

\*8- Nesne görüntüleme tuşu(oluşturduğumuz nesneleri görüntüler),

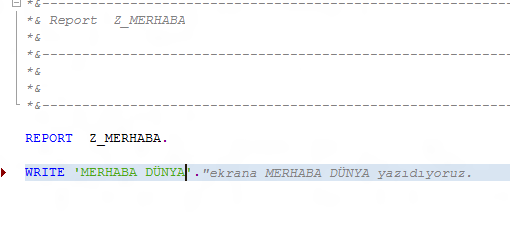
\*9- Dolaşma pencelesi ,

\*10- Programın içine kontrol noktası konulması (yani break point koyarak programımızı adım adım işletmemzi ve varsa hataları bulmamzı sağlıyor)

\*12-Programın içerisine dışarıdan aktarma yapmamıza yarayan tuş.Bu tuşla sorgu,fonksiyon kontrol mekanizması oluşturp ekleyebilriz.(örneğin daha önce yazdığımız bir programı yeni program içinde kullanmak için)

\*13-Kodlama yaparken güzel görünmesini,büyük küçük harf ayrımını ve blokları düzenlemeyi sağlayan tuş.

**6-** İlk programcığı yazmak için artık hazırız. ABAP da string(metin) tipinde bir çıktı alabilmek için “**WRITE**” komutunu kullanıyoruz. ABAP da her satırın bitimine ‘.'(nokta) koyuyoruz. -Bu sytax kurallarından birtanesi.- Ayrıca yorum satırı eklemekde mümkün. Bunun içinde tırnak işaretini ‘ ” ‘ kullanıyoruz. Ayrıca ABAP da büyük/küçük harf duyarlılığı yoktur.



**7-**Active edip run dediğimizde metnimiz program sonucu olarak gösterilmiş olacaktır.

