

Logo Netsis ERP | İleri Üretim Planlama Script Desteđi Tanıtım Dokümanı ve Örnek Uyarlama



İçindekiler

İleri Üretim Planlama – Script Desteği.....	3
1. Hazırlık Süresi Tanımlama Ekranı – Script Desteği	3
2. Çizelgeleme Modelleme Aracı – Script ve IF/ELSE Bloğu	5
3. Genel Algoritma – Sıralama Kriterinde Script Desteği	6
4. Örnek Uyarlamalar	9
4.1. Örnek Uyarlama-1	12
4.2. Örnek Uyarlama-2	14
4.3. Örnek Uyarlama-3	15

İleri Üretim Planlama – Script Desteği

İleri üretim planlama uygulamasında 3 farklı bölümde VbScript programlama dilini kullanarak script yazmak ve yapılacak uyarılmalarda esneklik sağlamak mümkündür. Script özelliğinin desteklendiği bölümler aşağıdaki gibi listelenebilir:

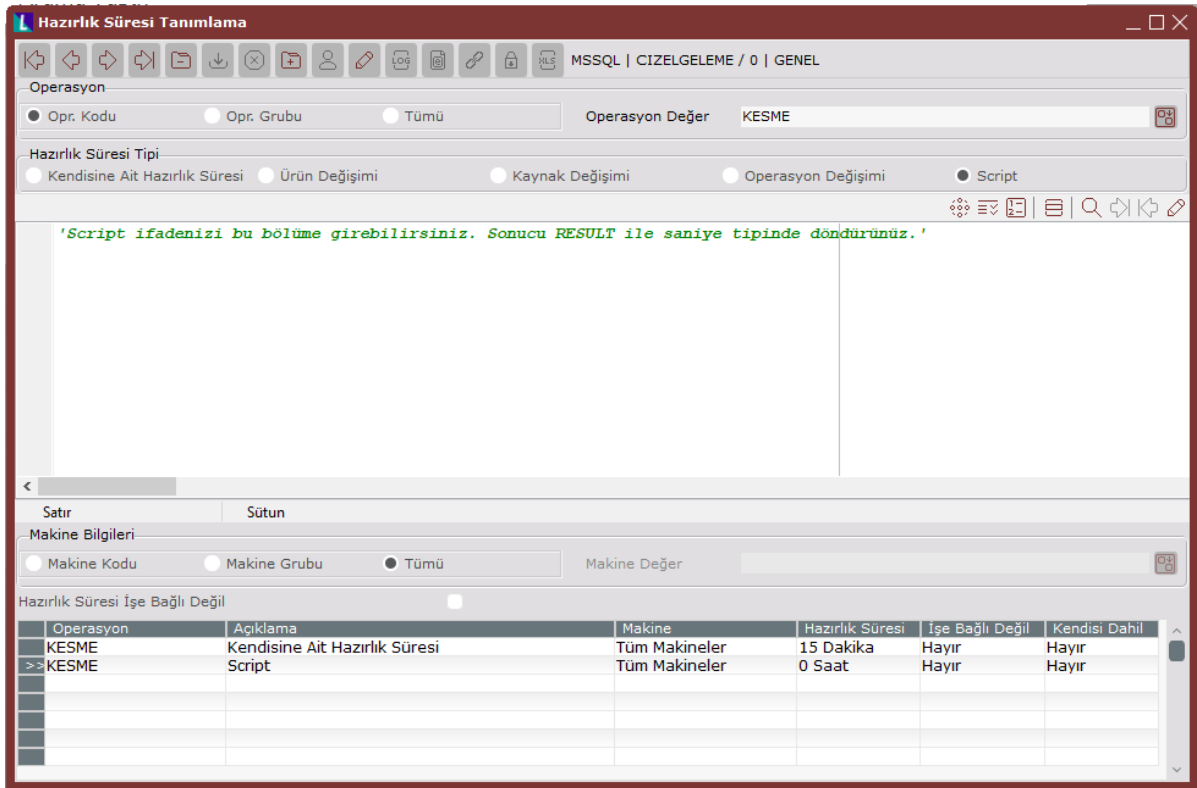
- Hazırlık Süresi Tanımlama
- Çizelgeleme Modelleme Aracı – Script ve IF/ELSE Bloğu
- Çizelgeleme Modelleme Aracı – Genel Algoritma için Sıralama Kriteri

1. Hazırlık Süresi Tanımlama Ekranı – Script Desteği

“MRP → Kayıt → İleri Üretim Çizelgeleme → Hazırlık Süresi Tanımlama” ekranında 5 farklı hazırlık süresi tipi bulunmaktadır:

- Kendisine Ait Hazırlık Süresi
- Ürün Değişimi
- Kaynak Değişimi
- Operasyon Değişimi
- Script

Script seçeneği seçildiğinde TEMELSET ekranlarında dinamik kodlamadan alışık olduğumuz script giriş ekranı açılacaktır. Bu bölüme yazılacak script ile özel iş mantıklarına uygun olacak şekilde hazırlık sürelerinin hesaplanması sağlanabilir ve çizelgeleme sırasında hesaplanan hazırlık süreleri kullanılabilir. (Bkz. Ekran Görüntüsü-1)



Hazırlık Süresi Tanımlama

MSSQL | CIZELGELEME / 0 | GENEL

Operasyon: Opr. Kodu Opr. Grubu Tümü | Operasyon Değer: KESME

Hazırlık Süresi Tipi: Kendisine Ait Hazırlık Süresi Ürün Değişimi Kaynak Değişimi Operasyon Değişimi Script

'Script ifadenizi bu bölüme girebilirsiniz. Sonucu RESULT ile saniye tipinde döndürünüz.'

Satır | Sütun

Makine Bilgileri: Makine Kodu Makine Grubu Tümü | Makine Değer:

Hazırlık Süresi İşe Bağlı Değil:

Operasyon	Açıklama	Makine	Hazırlık Süresi	İşe Bağlı Değil	Kendisi Dahil
KESME	Kendisine Ait Hazırlık Süresi	Tüm Makineler	15 Dakika	Hayır	Hayır
>KESME	Script	Tüm Makineler	0 Saat	Hayır	Hayır

Ekran Görüntüsü 1

Hazırlık süresi için yazılan script'in çizelgeleme sırasında genel algoritma tarafından dikkate alınması için "Çizelgeleme Modelleme Aracı" ekranındaki algoritma opsiyonlarında "Hazırlık Süresi Hesaplama Politikası" olarak aşağıdaki seçeneklerden biri seçilebilir:

- Hazırlık Sürelerinin Toplamı: Planlanacak iş emri için bütün hazırlık süresi tanımları değerlendirilir, birden fazla hazırlık süresi geliyorsa bunların toplamı alınır.
- Hazırlık Sürelerinin En Küçüğü: Planlanacak iş emri için bütün hazırlık süresi tanımları değerlendirilir, birden fazla hazırlık süresi geliyorsa bunların en küçüğü alınır.
- Hazırlık Sürelerinin En Büyüğü: Planlanacak iş emri için bütün hazırlık süresi tanımları değerlendirilir, birden fazla hazırlık süresi geliyorsa bunların en büyüğü alınır.
- Script Hazırlık Süresi: Planlanacak iş emri için sadece script tipindeki hazırlık süresi tanımları dikkate alınır.

Çizelgeleme sırasında boşta olan bir makine seçildikten sonra bu makineye yerleştirilecek iş listesinden seçim yapılmaktadır, bu aşamada her bir iş için hazırlık süresi hesaplanmaktadır, yani her bir iş planlanacağı zaman hazırlık süresi script'i tekrar tekrar çalışacak ve o andaki iş emri ve güncel durum için hesaplama yapılacaktır. Script üzerinden başka tiplerde tanımlanmış hazırlık sürelerine de erişim sağlanabilir, örneğin kaynak değişimi için tanımlanmış bir hazırlık süresi varsa bununla ilgili süreye script üzerinden erişim sağlanabilir.

Hazırlık süresi ekranında yazılacak script'in sonucunu döndürebilmek için RESULT değişkenine bulunan hazırlık süresini saniye cinsinden atamak gereklidir. Çizelgeleme sırasında RESULT değişkeninden dönecek sonuç hazırlık süresi olarak değerlendirilir.

Script yazılırken erişilmek istenen belli başlı özellikler için SETUPSCRIPTPARAMS nesnesi kullanılabilir, bu nesne üzerinden erişilebilecek detay bilgiler aşağıdaki gibidir:

Nesne / Özellik / Prosedür / Fonksiyon	Açıklama
SetupScriptParams.Machine	Çizelgelenecek iş için seçilmiş makine nesnesini döndürür. Makine nesnesi üzerinden makineye ait birçok detaya erişim sağlanabilir.
SetupScriptParams.Operation	Çizelgelenecek işin operasyon nesnesini döndürür. Operasyon nesnesi üzerinden operasyona ait birçok detaya erişim sağlanabilir.
SetupScriptParams.OperationChangeSetupTime	Çizelgelenecek iş için seçilen makineyi baz alarak makinede yapılan bir önceki operasyon ve şu anda yapılacak operasyon bilgilerini de göz önünde bulundurarak, "operasyon değişimi" tipinde uygun hazırlık süresi varsa saniye cinsinden değerini döndürür.
SetupScriptParams.OprOwnSetupTime	Çizelgelenecek işin operasyonu için "Kendisine Ait Hazırlık Süresi" tipinde uygun hazırlık süresi varsa saniye cinsinden değerini döndürür.
SetupScriptParams.PrevOperation	Çizelgelenecek iş için seçilmiş makinede yapılan bir önceki operasyonun nesnesini döndürür.

SetupScriptParams.PrevProdResources	Çizelgelenecek iş için seçilmiş olan makinede en son kullanılan kaynakların listesini döndürür.
SetupScriptParams.PrevProduct	Çizelgelenecek iş için seçilmiş makinede planlanmış bir önceki ürüne ait nesneyi döndürür. Ürün nesnesi üzerinden stok koduna ait birçok detaya erişim sağlanabilir.
SetupScriptParams.ProdResourceChangeSetupTime	Çizelgelenecek iş için seçilen makineyi baz alarak makinede kullanılan son kaynaklar ve şu anda kullanılması gereken kaynak bilgilerini de göz önünde bulundurarak, “kaynak değişimi” tipinde uygun hazırlık süresi varsa saniye cinsinden değerini döndürür.
SetupScriptParams.ProdResources	Çizelgelenecek iş için seçilmiş olan makinede kullanılması gereken kaynakların listesini döndürür.
SetupScriptParams.Product	Çizelgelenecek işe ait ürün nesnesini döndürür. Ürün nesnesi üzerinden stok koduna ait birçok detaya erişim sağlanabilir.
SetupScriptParams.ProductChangeSetupTime	Çizelgelenecek iş için seçilen makineyi baz alarak makinede yapılan son işi ve şu anda yapılacak iş bilgilerini de göz önünde bulundurarak, “ürün değişimi” tipinde uygun hazırlık süresi varsa saniye cinsinden değerini döndürür.

Yukarıdaki listede yer alan özelliklerin ne şekilde çalıştığını daha detaylı anlayabilmek için “Örnek Uyarılama-1” bölümü incelenebilir.

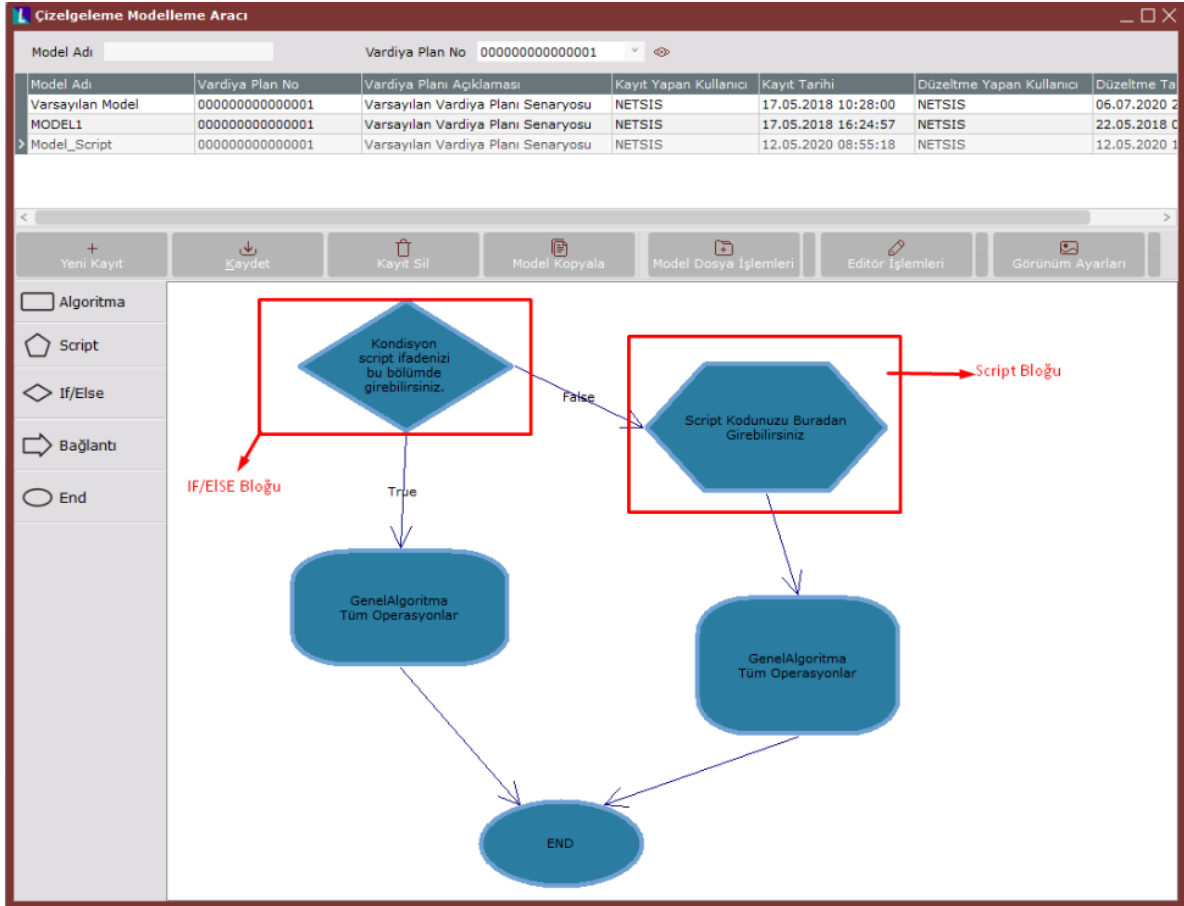
2. Çizelgeleme Modelleme Aracı – Script ve IF/ELSE Bloğu

“MRP → Kayıt → İleri Üretim Planlama → Çizelgeleme Modelleme Aracı” ekranında tanımlanan modellerde script veya IF/ELSE bloğunu kullanarak script yazmak mümkündür. (Bkz. Ekran Görüntüsü-2)

Modelleme aracında script bloğunu kullanarak özel iş mantıkları çalıştırılabilir. Örneğin; ilk istasyonda yapılacak işler çizelgelendikten sonra, ortaya çıkan sonuçları değerlendirip veri tabanı seviyesinde bir tabloya kayıt atılabilir veya ortaya çıkan sonuçlara göre bir hesaplama yapılarak bu hesaplama sonucu modelin ilerleyen kısımlarında kullanılabilir.

Benzer şekilde modelleme aracı üzerinde IF/ELSE bloğunda aynı şekilde script yazılabilir ve RESULT değişkenine “true” veya “false” değerleri atanarak modelin iki farklı yönden ilerlemesi sağlanabilir.

Modelleme aracındaki yukarıda bahsedilen bloklar üzerinde script yazılırken kullanılacak en önemli nesne SCHEDULEDATA ve NETSISCORE olarak söylenebilir, çünkü bu nesnelere üzerinden çizelgeleme sırasında kullanılan sabit veya anlık hesaplanan dinamik verilerin büyük kısmına erişmek mümkündür. Daha detaylı örnek için “Örnek Uyarılama-2” bölümü incelenebilir.



Ekran Görüntüsü 2

3. Genel Algoritma – Sıralama Kriterinde Script Desteği

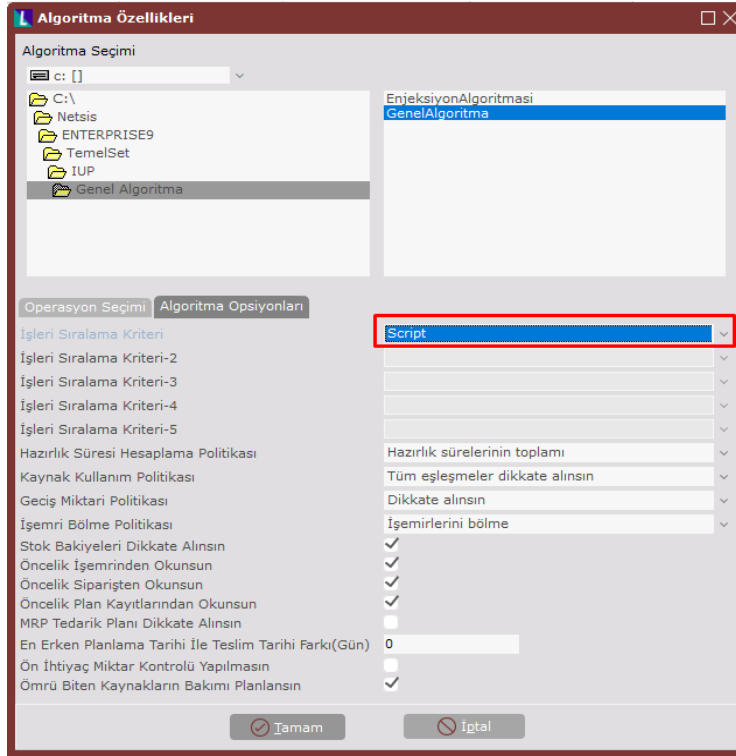
“MRP → Kayıt → İleri Üretim Planlama → Çizelgeleme Modelleme Aracı” ekranında kullanılan genel algoritma için “Sıralama Kriteri” olarak “Script” seçeneğini kullanmak ve özel iş mantıklarına göre iş seçimi yapabilmek mümkündür. Bu şekilde genel algoritmanın desteklediği ön tanımlı sıralama kriterlerinden bir kısmını kullanarak veya hiçbirini kullanmayarak tamamen ihtiyaca özel algoritmalar yazılabilir ve bu şekilde iş sıralaması yapılabilir. (Bkz. Ekran Görüntüsü-3)

Sıralama kriteri olarak “Script” seçildiğinde VbScript diliyle script yazabileceğimiz bir ekran açılacaktır. Gene algoritma çalışırken her bir iş seçimi sırasında tekrar tekrar yazılan script çalışacaktır ve belirtilen iş mantığına göre seçim yapılacaktır. Genel mantık olarak script bölümündeki işleyiş şu şekildedir: Genel algoritma çalışırken herhangi bir t tanında iş seçimi için script çalışacaktır. Script bölümüne o anda seçilebilecek bütün iş emirlerinin listesi gönderilecektir ve script’in sonunda bu listedeki işlerden birinin seçilmiş olması beklenmektedir, kullanıcı yazacağı script’in sonunda seçtiği iş emri bilgisini algoritmaya geri vermek zorundadır.

Algoritma opsiyonları bölümünde “Sıralama Kriteri” olarak sırasıyla değerlendirilecek 5 farklı kriter seçilebilir, eğer “Script” seçeneği kullanılacak ise son seçilen sıralama kriterinin “Script” olması gerekmektedir, script olarak seçilen sıralama kriterinden sonraki alanlar pasif hale gelmektedir. Örneğin sıralama kriterlerinin aşağıdaki gibi seçildiğini varsayalım:

- Sıralama Kriteri-1: “Teslim Tarihi (Küçükten Büyüğe)”
- Sıralama Kriteri-2: “Script”

Bu durumda iş seçimi sırasında seçilebilecek iş emirleri öncelikle teslim tarihine göre küçükten büyüğe sıralanacak ve o andaki işler içinden en küçük tarihli olan iş emirleri bulunacaktır. En küçük teslim tarihine sahip birden fazla iş varsa, bu işlerin listesi script bloğuna gönderilecek ve bu işlerin içinden birisinin script tarafından seçilmesi beklenecektir.



Ekran Görüntüsü 3

Sıralama kriteri olarak script yazılırken aşağıdaki nesnelere erişim sağlanabilir:

- NETSISCORE: Dinamik kodlama ortamında genel Netsis nesne ve özelliklerine erişim sağlar.
- SEARCHQUERY: Veri tabanı üzerinde sorgu çekmek için bu nesne kullanılabilir.
- SORTSCRIPTOBJECTLIST: Çizelgeleme sırasında script bölümüne gelen iş listesine ve işlerin belli başlı özelliklerine bu nesne üzerinden erişilebilir.
- SCHEDULEDATA: Çizelgeleme sırasında kullanılan sabit veya dinamik verilerin büyük bölümüne bu nesne üzerinden erişilebilir.

SORTSCRIPTOBJECTLIST üzerinden erişilebilecek belli başlı nesne ve özellikler aşağıdaki listede verilmektedir. Daha detaylı örnek için "Örnek Uyarılama-3" bölümü incelenebilir.

Nesne / Özellik / Prosedür / Fonksiyon	Açıklama
SortScriptObjectList.SortScriptCount	Ön ihtiyaçları karşılanmış çizelgelenebilecek aday iş emri sayısını döndürür.
SortScriptObjectList.GetSortScriptObject (pIndex)	Ön ihtiyaçları karşılanmış çizelgelenebilecek iş listesinden belli bir işi döndürür. PIndex parametresini alır, bu parametre iş listesinin kaçınıcı işini döndüreceğimizi belirler. Dönüş değeri olarak SortScriptObject nesnesi döndürür,

	bu nesne seçilecek iş emirlerinden birine denk gelmektedir.
SortScriptObjectList.SetSortScriptObject pIndex, PSortScriptObject	Çizelgelenecek iş listesinde bir işin listedeki sırasını belirlemek için kullanılır. PIndex parametresi işin kaçınıcı sıraya yerleştirileceğini belirtir, PSortScriptObject parametresi ise seçilen iş nesnesidir. Script sonunda seçilecek işi belirlemek için bu prosedür kullanılır ve seçilen işin liste sırası 0 olarak verilir.
SortScriptObject.Job.ProductionOrder	Aday iş emri nesnesini döndürür. Bu nesne üzerinden iş emri numarası vs. şeklinde iş emrinin bütün bilgilerine ulaşılabilir.
SortScriptObject.SetupTime	Aday iş emri seçilirse oluşacak hazırlık süresini döndürür (milisaniye)
SortScriptObject.JobPriority	Aday iş emrinin "Sıralama Önceliği" değeri
SortScriptObject.RemainingTime	Aday iş planlandıktan sonra işin planlanan bitiş tarihine bakılarak teslim tarihine ne kadarlık bir sürenin kaldığı bilgisini döndürür (saat)
SortScriptObject.RunTime	Aday işin seçilen makinedeki toplam üretim süresi (milisaniye)
SortScriptObject.RemainingUnScheduledOperationCount	Aday işin rotasındaki çizelgenmemiş operasyon sayısı
SortScriptObject.Tardiness	Aday iş planlandıktan sonra işin planlanan bitiş tarihi ve teslim tarihine bakılarak ne kadarlık gecikme olduğunu döndürür. (saat)
SortScriptObject.OriginalListIndex	Aday işin iş emri listesindeki orjinal sırasını döndürür.
SortScriptObject.Job.Product.UserDefinedFields.GetUserDefinedField(pIndex).Value	Aday işe ait stok kartındaki kullanıcı tanımlı sahaların değerini döndürür. PIndex parametresini alır, örneğin kullanıcı tanımlı numerik saha-1 (KULL1N) okunmak isteniyorsa, pIndex parametresi 0 olarak gönderilir.
SortScriptObject.Job.Product.AdditionalInformations.GetAdditionalInformationByFieldName(pFieldName).Value	İş listesindeki belli bir iş emri için stok kartına tanımlanmış saha-tablo eşleştirmelerine ait değeri döndürür. PIndex parametresi iş listesindeki seçilecek işin sırasını belirtirken, pFieldName parametresi ise saha-tablo eşleştirmesine ait alan ismini belirtmektedir.
SortScriptObject.Machine	Çizelgelenecek iş için seçilmiş makine nesnesini döndürür. Makine nesnesi üzerinden makineye ait birçok detaya erişim sağlanabilir.
SortScriptObject.Machine.OperationBeforeTime(CDate("2050-01-01"))	Çizelgelenecek iş için seçilmiş makinede yapılan bir önceki iş nesnesini döndürür.

SCHEDULEDATA nesnesi üzerinden erişilebilecek belli başlı nesne ve özellikler aşağıdaki gibidir:

Nesne / Özellik / Prosedür / Fonksiyon	Açıklama
ScheduleData.JobList	İleri üretim planlama ekranından seçilen ve çizelgelemeye dahil olan bütün işlerin listesini döndürür.
ScheduleData.JobList.GetJob(PIndex)	İş listesindeki belli bir sıradaki iş nesnesini döndürür. PIndex parametresini alır.
ScheduleData.MachineList	Çizelgeleme sırasında kullanılabilir makine listesini döndürür.
ScheduleData.MachineList.GetMachine(pIndex)	Makine listesindeki belli bir sıradaki makine nesnesini döndürür. PIndex parametresini alır.
ScheduleData.EditedJobOperationList	Sabitlenen iş listesini döndürür.
ScheduleData.WorkCenterList	İş istasyonu listesini döndürür.
ScheduleData.SpecialParameterList. GetSpecialParameterByGroupAndKeyCode().Value	Grup Kodu ve Anahtar bilgilerini parametre olarak alıp uygun bir çizelgeleme özel parametresi arar ve varsa bu özel parametrenin değerini döndürür. Uyarlama aşamasında kullanılabilir parametrik değerleri çizelgeleme özel parametresi olarak tanımlayabilir ve bu şekilde script içinde okuyabilirsiniz.
ScheduleData.ScheduleKeyValueDic	Key-Value yapısında kullanıcıya özel bir liste döndürür. Bu listeye modelleme aracındaki script veya If/Else bloğundan erişim sağlanabileceği gibi sıralama kriterinde yazılan script üzerinden de erişim sağlanabilir. Kullanıcı bu listeye istediği şekilde değerler atabilir ve bu değerlere daha sonradan erişim sağlayabilir.

4. Örnek Uyarlamalar

Örnek uyarlamaların anlatılacağı üretim sisteminin yapısı özet olarak şu şekilde anlatılabilir: Sistemde enjeksiyon operasyonu sonucunda meydana gelen iki farklı yarı mamul bulunmaktadır ve bu yarı mamullerin dışardan satın alınan bileşenler ile montaj edilmesi ve ardından paketlenmesi sonucunda iki farklı mamul meydana gelmektedir. Enjeksiyon operasyonu sırasında kullanılan kalıplar kaynak olarak tanımlanmıştır.

Üretim sistemine ait genel bilgiler aşağıdaki gibidir:

- Reçete Kayıtları
Sistem üzerinde MAMUL30 ve MAMUL40 koduyla üretimi ve satışı ve yapılan iki mamul kodu bulunmaktadır. Bu mamullere ait reçete bilgileri aşağıdaki gibidir:

MAMUL Seviye 1	BİLEŞEN Seviye 2	BİLEŞEN Seviye 3	MİKTAR
MAMUL30			1 Adet
	HAMMADDE1		1 Adet
	YARIMAMUL30		1 Adet
		HAMMADDE30	1 Adet

Ekran Görüntüsü 4

MAMUL Seviye 1	BİLEŞEN Seviye 2	BİLEŞEN Seviye 3	MİKTAR
MAMUL40			1 Adet
	HAMMADDE2		1 Adet
	YARIMAMUL40		2 Adet
		HAMMADDE40	1 Adet

Ekran Görüntüsü 5

- b) Bu mamullere ve yarı mamullere ait rota kayıtları ve operasyon detayları aşağıdaki gibidir. MAMUL30 ve MAMUL40 ürünleri için OP_MONTAJ operasyonu yapılırken, YARIMAMUL30 ve YARIMAMUL40 ürünleri için sırasıyla OP_MONTAJ ve OP_PAKETLEME operasyonları yapılmaktadır:

ROTA KODU	OPERASYON KODU	MAMUL/YARI MAMUL KODU
ROTA1		
	OP_ENJEKSİYON	YARIMAMUL30
		YARIMAMUL40

ROTA KODU	OPERASYON KODU	MAMUL/YARI MAMUL KODU
ROTA2		
	OP_MONTAJ	
	OP_PAKETLEME	
		MAMUL30
		MAMUL40

Ekran Görüntüsü 6

- c) Yukarıdaki mamul ve yarı mamuller için operasyonların yapılabileceği makine bilgileri ve 1 adetlik üretim için işlem süreleri "Operasyon-Makine Eşleştirme" ekranı üzerinden aşağıdaki gibi tanımlanmıştır:

STOK KODU	OPERASYON KODU	MAKİNE KODU	OPERASYON SÜRESİ
MAMUL30	OP_PAKETLEME	PAKETLEME_MAK1	90 Saniye
MAMUL30	OP_PAKETLEME	PAKETLEME_MAK2	120 Saniye
MAMUL30	OP_MONTAJ	MONTAJ_MAK1	150 Saniye
MAMUL40	OP_PAKETLEME	PAKETLEME_MAK1	120 Saniye
MAMUL40	OP_PAKETLEME	PAKETLEME_MAK2	150 Saniye
MAMUL40	OP_MONTAJ	MONTAJ_MAK1	200 Saniye
YARIMAMUL30	OP_ENJEKSİYON	ENJ_MAK1	15 Saniye
YARIMAMUL40	OP_ENJEKSİYON	ENJ_MAK1	20 Saniye

Ekran Görüntüsü 7

- d) OP_ENJEKSİYON operasyonu sırasında YARIMAMUL30 üretimi için KALIP30 ve YARIMAMUL40 üretimi için de KALIP40 kaynakları kullanılmaktadır. Bu tanımlamalar “Operasyon-Kaynak Eşleştirme” ekranı üzerinden yapılmıştır.
- e) Kalıp değişimleri için “Hazırlık Süresi Tanımlama Ekranı” üzerinden aşağıdaki gibi hazırlık süreleri tanımlanmıştır.

HAZIRLIK SÜRESİ TİPİ	ÖNCEKİ KAYNAK	SONRAKİ KAYNAK	HAZIRLIK SÜRESİ
Kaynak Değişimi	KALIP30	KALIP40	30 Dakika
Kaynak Değişimi	KALIP40	KALIP30	45 Dakika

Ekran Görüntüsü 8

- f) Yarı mamuller için hammadde tipi bilgisi stok kartı kayıtları ekranındaki “Alfa Sayısal Saha-2 (KULL2S)” üzerinde tutulmaktadır. Yarı mamul ve mamuller için çeper kalınlığı bilgisi ise stok kartı kayıtları ekranına eklenen “Çeper Kalınlık” isimli saha-tablo eşleştirmeleri sahasında tutulmaktadır (Saha-tablo eşleştirmelerindeki alan adı: KT_CeperKalınlık)
Bu bilgiler stok kodu bazında aşağıdaki gibidir:

STOK KODU	HAMMADE TİPİ	ÇEPER KALINLIĞI (mm)
YARIMAMUL30	Tip1	10
YARIMAMUL40	Tip2	15
MAMUL30		10
MAMUL40		15

Ekran Görüntüsü 9

- g) Çizelgeleme öncesinde sistemde MAMUL30 ve MAMUL40 için aşağıdaki gibi müşteri siparişlerini bulunmaktadır ve bu siparişler üzerinden MRP çalıştırılarak çıkan sonuçlara göre iş emirleri açılmıştır. Bu aşamada sistemde açık iş emri ve stok bakiyesi bulunmadığı varsayılmaktadır:

MÜŞTERİ SİPARİŞ NO	STOK KODU	MİKTAR	TESLİM TARİHİ
MS0000000000001	MAMUL30	200	15.12.2020
MS0000000000002	MAMUL30	100	18.12.2020
MS0000000000003	MAMUL30	300	20.12.2020
MS0000000000004	MAMUL40	200	16.12.2020
MS0000000000005	MAMUL40	200	20.12.2020
MS0000000000006	MAMUL40	300	24.12.2020

Ekran Görüntüsü 10

4.1. Örnek Uyarlama-1

Bu örnek uyarlama başlığında script tipindeki hazırlık süresi tanımıyla ilgili detaylara yer verilecektir.

Çizelgeleme sırasında OP_ENJEKSİYON operasyonu için kullanılacak hazırlık süreleri şu mantığa uygun olarak hesaplanacaktır:

Hammadde tipindeki değişimler için 20 dakika ve çeper kalınlığındaki değişimler için 30 dakikalık hazırlık süreleri oluşmaktadır. Hem hammadde tipi hem de çeper kalınlığında değişim bulunuyorsa iki süresinin toplamı $30+20 = 50$ dakikalık hazırlık süresi meydana gelmektedir. Ortaya çıkan bu hazırlık süresi, "Kaynak Değişimi" tipiyle "Hazırlık Süresi Tanımlama" ekranı üzerinden tanımlanmış hazırlık süresiyle karşılaştırılacak ve büyük olan hazırlık süresi seçilecektir.

Örnek script bloğu ve kod üzerindeki yorumlar aşağıdaki gibidir:

"Yorum-1: Hesaplanacak hazırlık süresine ilk olarak sıfır değeri atanıyor.

SetupTime = 0

"Yorum-2: Seçilen makine üzerinde daha önceden planlanmış bir iş bulunup bulunmadığını kontrol etmek için bir önceki iş bilgisi değişkene atanıyor.

Set Onceki_Is = SETUPSCRIPTPARAMS.PrevProduct

"Yorum-3: Seçilen makine üzerinde daha önceden planlanmış bir iş varsa, bir önceki işin stok koduna ait "Alfa Sayısal Saha-2 (Hammadde Tipi)" değeri ve saha-tablo eşleştirmelerinden "Çeper Kalınlık" değeri bulunmaktadır.

If (Not Onceki_Is Is Nothing) Then

Onceki_is_hammadde_tipi =
SETUPSCRIPTPARAMS.PrevProduct.UserDefinedFields.GetUserDefinedFieldByName("Hammadde Tipi").Value

Onceki_is_ceper =
SETUPSCRIPTPARAMS.PrevProduct.AdditionalInformations.GetAdditionalInformationByFieldName("KT_CeperKalinlik").Value

End If

"Yorum-4: Seçilen makine üzerinde planlanacak güncel stok koduna ait "Alfa Sayısal Saha-2 (Hammadde Tipi)" değeri ve saha-tablo eşleştirmelerinden "Çeper Kalınlık" değeri bulunmaktadır.

```
Guncel_is_hammadde_tipi =
SETUPSCRIPTPARAMS.Product.UserDefinedFields.GetUserDefinedFieldName("Hammadde
Tipi").Value
```

```
Guncel_is_ceper =
SETUPSCRIPTPARAMS.Product.AdditionalInformation.GetAdditionalInformationByFieldName(
"KT_CeperKalınlık").Value
```

"Yorum-5: Seçilen makinede bir önceki iş bilgisi bulunuyorsa hazırlık süresi hesaplanacaktır.

"Önceki işin hammadde tipi şu andaki planlanacak işin hammadde tipinden farklı ise 20 dakikalık (1200 saniye) hazırlık süresi eklenmektedir.

"Önceki işin çeper kalınlığı şu andaki planlanacak işin çeper kalınlığından farklı ise 30 dakikalık (1800 saniye) hazırlık süresi eklenmektedir.

```
If (Not Onceki_Is Is Nothing) Then
```

```
  If (Onceki_is_hammadde_tipi <> Guncel_is_hammadde_tipi) Then
```

```
    SetupTime = SetupTime + 1200
```

```
  End If
```

```
  If (Onceki_is_ceper <> Guncel_is_ceper) Then
```

```
    SetupTime = SetupTime + 1800
```

```
  End If
```

```
End If
```

"Yorum-6: "Kaynak değişimi" tipinde tanımlanan hazırlık süresi ile yukarıdaki hesaplamalarda bulunan hazırlık süresi karşılaştırılarak bunlardan büyük olanı alınıyor.

```
If SETUPSCRIPTPARAMS.ProdResourceChangeSetupTime > SetupTime Then
```

```
  SetupTime = SETUPSCRIPTPARAMS.ProdResourceChangeSetupTime
```

```
End If
```

"Yorum-7: Bulunan hazırlık süresi RESULT değişkenine atanarak algoritma tarafından kullanılması sağlanmaktadır.

```
RESULT = SetupTime
```

4.2. Örnek Uyarlama-2

Bu örnek uyarlama başlığında “Çizelgele Modelleme Aracı” ekranında script bloğunun kullanımına yönelik detaylara yer verilecektir.

Örnek sistemde üretim sahasına ait anlık kayıtların 3. parti bir yazılım üzerinde tutulduğu ve her makine üzerindeki devam eden son iş bilgisinin TB_URETIMKAYIT tablosuna kaydedildiği varsayılmaktadır. Bu bilgiler genel algoritma çalışmadan önce çizelge modelleme aracındaki script bloğunda sql sorgusuyla getirilecek ve daha sonrasında “işleri sıralama kriteri” bölümünde kullanmak amacıyla “ScheduleData.ScheduleKeyValueDic” listesinde saklanacaktır. (Not: Normal şartlarda böyle bir durum için üretim akış kaydı atılarak direkt entegrasyon sağlanabilir, sadece örnek olması açısından bu tarz bir varsayıma gidilmiştir)

Örnek script bloğu ve kod üzerindeki yorumlar aşağıdaki gibidir:

"Yorum-1: NetsisCore kütüphanesi kullanılarak veri tabanından sorgu çekmek amacıyla yeni bir query nesnesi oluşturuluyor.

```
Set rs = NetsisCore.NetLibDB.GetNewQuery
```

"Yorum-2: Makineler üzerinden devam eden işlerin tutulduğu TB_URETIMKAYIT tablosuna sorgu çekiliyor.

```
strSql = "SELECT MAKINE_INCKEYNO, SON_ISEMRINO FROM TB_URETIMKAYIT  
WITH(NOLOCK)"
```

```
rs.RecSql(strSql)
```

"Yorum-3: Tablodan dönen kayıtlar key = MAK_INCKEYNO ve value = SON_ISEMRINO olacak şekilde

```
While Not rs.EOF
```

```
    MAK_INCKEYNO = rs.fieldbyname("MAKINE_INCKEYNO").AsInteger
```

```
    SON_ISEMRINO = rs.fieldbyname("SON_ISEMRINO").AsString
```

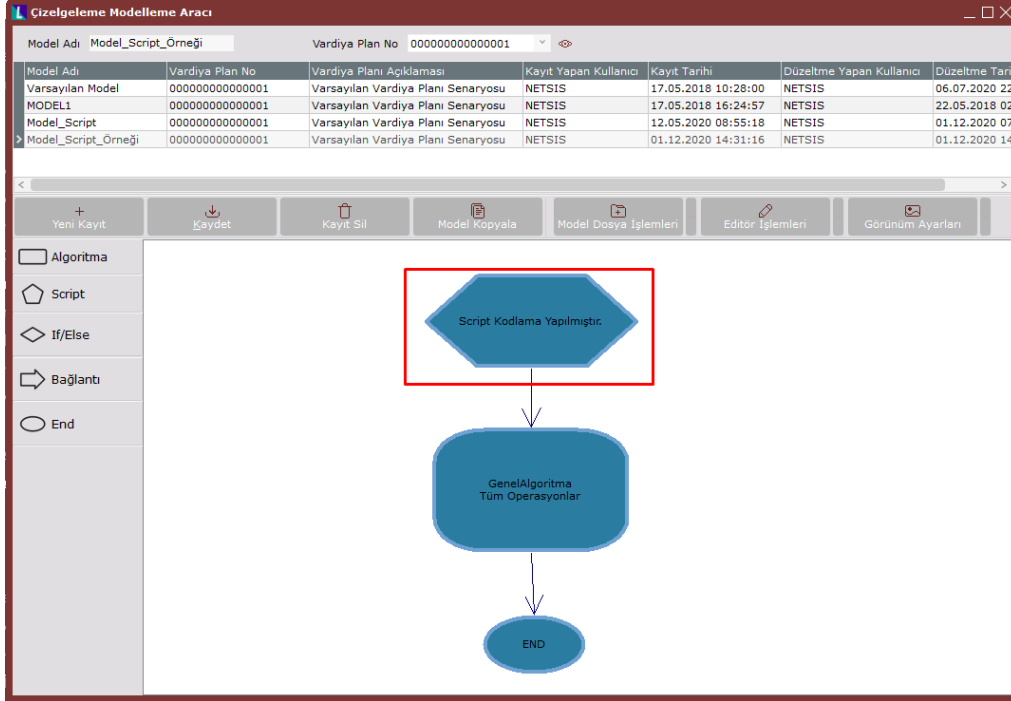
```
    Call ScheduleData.ScheduleKeyValueDic.AddOrSetValue(MAK_INCKEYNO,SON_ISEMRINO)
```

```
    rs.Next
```

```
Wend
```

```
Set rs = Nothing
```

Kullanılacak script bloğu “Çizelge Modelleme Aracı” ekranında algoritma bloğundan önce çalışacak şekilde yerleştirilmiştir ve bu durumda model tasarımı aşağıdaki gibi olmaktadır:



Ekran Görüntüsü 11

4.3. Örnek Uyarılama-3

Bu örnek uyarılama başlığında “Çizelge Modelleme Aracı” ekranında algoritma bloğu içinde “Sıralama Kriteri” olarak kullanılacak script detaylarına yer verilecektir.

Oluşturulan çizelgeleme modelindeki algoritma bloğunda genel algoritma kullanılmaktadır ve işleri sıralama kriteri olarak “Script” seçeneği seçilmiştir. Yazılacak script ile seçilecek işlerin sıralama mantığının aşağıdaki şekilde çalışması amaçlanmaktadır:

Öncelikle seçilen makine üzerinde son olarak devam eden bir iş olup olmadığını anlamak için Örnek Uyarılama-2’de doldurulan “ScheduleKeyValueDic” listesi kontrol edilecek ve son devam eden iş emrinin öncelikli olarak seçilmesi sağlanacaktır. Aksi durumda seçilebilecek işlerin içinde gecikmesi en yüksek olan işin öncelikli olarak seçilmesi sağlanacaktır. Eğer gecikmesi bulunan bir iş emri de yoksa, bu durumda makinede yapılan bir önceki iş ile aynı “Çeper Kalınlık” değerine sahip iş seçilmeye çalışılacaktır, eğer işlerin çeper kalınlıkları da eşit çıkarsa en küçük teslim tarihli iş seçilecektir.

Örnek script bloğu ve kod üzerindeki yorumlar aşağıdaki gibidir:

"Yorum-1: Seçilecek iş bilgisi SelectedIndex değişkeninde tutulacak, script sonunda karar vereceğimiz değişken bu oluyor.

SelectedIndex = 0

"Yorum-2: Çeper kalınlıkları eşit çıkan birden fazla iş olursa, bunların arasından teslim tarihi en küçük olan iş seçiliyor.

"Bu sebeple minimum teslim tarihi en başta büyük bir değer olarak atanıyor, sonrasında teslim tarihi küçük olan bir iş seçildikçe bu değişken de güncellenecek.

MinTeslimTarihi = "2050-01-01" " Çeper değişimi olmayan işler için kullanılacak.

MinTeslimTarihi2 = "2050-01-01" " Çeper değişimi olan işler için kullanılacak.

CeperDegisimiYok = false

"Yorum-3: Gecikmesi en yüksek olan işin öncelikli olarak seçilmesi sağlanacaktır. Bu sebeple maksimum gecikme bilgisi en başta 0 olarak atanıyor, iş seçimi yapıldıkça bu değişken de güncellenecek.

MaxTardiness = 0

GecikmesiOlanIsVar = false

"Yorum-4: Seçilen makine bilgisi bir değişkene atanıyor.

Set SelectedMachine = SORTSCRIPTOBJECTLIST.GetSortScriptObject(0).Machine

"Yorum-5: Seçilen makinede planlanmış bir önceki iş bilgisi bir değişkene atanıyor. Bir önceki işin "Çeper Kalınlık" bilgisine erişmek için kullanılacak.

Set PreviousOperationItem =

SORTSCRIPTOBJECTLIST.GetSortScriptObject(0).Machine.OperationBeforeTime(CDate("2050-01-01"))

"Yorum-6: Seçilen makinede en son devam eden iş bilgisi modeldeki script bloğunda ScheduleKeyValueDic listesine atılmıştı, bu bilgi alınıyor.

If SCHEDULEDATA.ScheduleKeyValueDic.ContainsKey(SelectedMachine.Code) Then

 Son_Devam_Eden_IsEmriNo =

 SCHEDULEDATA.ScheduleKeyValueDic.GetValue(SelectedMachine.Code)

End If

"Yorum-7: Planlanabilecek iş listesi dönülüyor ve bu işlerden seçilecek olana karar veriliyor.

```
For i = 0 To SORTSCRIPTOBJECTLIST.SortScriptCount - 1
```

"Yorum-8: Script sırasında daha yüksek performans elde edebilmek için ilgili iş nesnesinin aşağıdaki gibi ayrı bir değişkene (JobItem) set edilmesi önerilmektedir.

```
Set JobItem = SORTSCRIPTOBJECTLIST.GetSortScriptObject(i)
```

"Yorum-9: Değerlendirme sırasında kullanılacak bilgiler alınıyor.

```
IsEmriNo = JobItem.Job.ProductionOrder.Number "İş emri numarası
```

```
Teslim_Tarihi = JobItem.Job.ProductionOrder.DeliveryDate "Teslim tarihi
```

```
Tardiness = JobItem.Tardiness "İş çizelgelendikten sonra ne kadar geciktiği bilgisi (saat)
```

```
Guncel_is_ceperep =
```

```
JobItem.Job.Product.AdditionalInformations.GetAdditionalInformationByFieldName("KT_CeperKalinlik").Value "Değerlendirilen işin çeper kalınlığı
```

```
If (not PreviousOperationItem is Nothing) Then
```

```
    Onceki_is_ceperep =
```

```
    PreviousOperationItem.GetJob.Product.AdditionalInformations.GetAdditionalInformationByFieldname("KT_CeperKalinlik").Value "Bir önceki planlanan işin çeper kalınlığı
```

```
Else
```

```
    Onceki_is_ceperep = "" "Makinede daha önceden planlanan iş yoksa boş olarak atanıyor, bu durumda değerlendirmeye alınmayacak.
```

```
End If
```

"Yorum-10: Değerlendirilen iş emri seçilen makinedeki son devam eden iş emri ise direkt olarak seçilir ve döngüden çıkılır.

```
If IsEmriNo = Son_Devam_Eden_IsEmriNo Then
```

```
    SelectedIndex = i
```

```
    Exit For
```

```
End If
```

"Yorum-11: Değerlendirilen iş emrinin gecikmesi varsa ve bu gecikme diğer işlerin gecikmesinden büyük ise ilgili iş seçilir.

```
If (Tardiness > 0) And (Tardiness > MaxTardiness) Then
```

```
    SelectedIndex = i
```

```
    MaxTardiness = Tardiness
```

```
    GecikmesiOlanIsVar = true
```

```
End If
```

"Yorum-12: Eğer daha önceden değerlendirilen işler arasında gecikmesi olan bir iş yoksa, değerlendirilen işin çeper kalınlık değişimine bakılır. Çeper değişimi olmayan işler arasından en küçük teslim tarihli olan iş seçilecektir.

```
If (GecikmesiOlanIsVar = false) Then
```

```
    If ((Onceki_is_ceper = Guncel_is_ceper) Or Onceki_is_ceper = "") And (Teslim_Tarihi < MinTeslimTarihi) Then
```

```
        SelectedIndex = i
```

```
        MinTeslimTarihi = Teslim_Tarihi
```

```
        CeperDegisimiYok = true
```

```
    End If
```

"Yorum-13 = Eğer daha önceden değerlendirilen işler arasında çeper değişimi olmayan bir iş bulunamadıysa, değerlendirilen işin çeper değişimi olsa bile minimum teslim tarihine sahip iş olarak seçilmesi sağlanıyor.

```
If (CeperDegisimiYok = false) And (Teslim_Tarihi < MinTeslimTarihi2) Then
```

```
    SelectedIndex = i
```

```
    MinTeslimTarihi2 = Teslim_Tarihi
```

```
End If
```

```
End If
```

```
Next
```

"Yorum-14 = Seçilen iş bilgisinin tutulduğu SelectedIndex değişkeni algoritmaya geri döndürülüyor ve ilgili iş seçilmiş oluyor.

```
SORTSCRIPTOBJECTLIST.SetSortScriptObject
```

```
0,SORTSCRIPTOBJECTLIST.GetSortScriptObject(SelectedIndex) " İlk parametre işin listedeki sırası, ikinci parametre ilgili iş nesnesi
```