# Infor CloudSuite Business APS Concepts Guide

**10/30/15**

## Important Notices



## Trademark Acknowledgements



## Publication Information

Release: Infor CloudSuite Business 9.00.x

Publication date: October 30, 2015

# Contents

**About This Guide** . . . . . 5

Intended audience . . . . . 5

Contacting Infor . . . . . 5

**Chapter 1 Plan and Schedule Data Model** . . . . . 7

Database Overview . . . . . 7

Orders . . . . . 8

Database table: ORDER . . . . . 8

Items . . . . . 8

Database table for Scheduler: PART . . . . . 8

Batches . . . . . 9

Database table for Scheduler: BATCH . . . . . 9

Routings and Operations . . . . . 9

Database tables: JOBSTEP, JS19VR, and PROCPLN . . . . . 9

Resources . . . . . 9

Database table: RESRC . . . . . 9

Resource Groups . . . . . 10

Database table: RGRP, RGRPMBR . . . . . 10

Infinite Resource Groups . . . . . 10

Materials . . . . . 11

Database table: MATL . . . . . 11

Representing Component Items . . . . . 11

Shifts . . . . . 12

Database table: SHIFT . . . . . 12

Shift Exceptions . . . . . 12

Database table: SHIFTEXDI . . . . . 12

Holidays . . . . . 13

Database table: CAL . . . . . 13

Delivery Schedules . . . 13
Database table: MATLDELV . . . 13
**Chapter 2 About the Planner . . . 15**
Database . . . 15
Running the Planner . . . 15
Order Promising/Get CTP . . . 16
Plan Regeneration . . . 16
Output Data . . . 16
Basic Planning Considerations . . . 17
Forecasting . . . 17
Resource Collisions . . . 18
Material Components . . . 18
Dynamic Reallocation by Order Priority . . . 18
Realistic Order Completion Dates . . . 18
Material Scrap Factor/Shrinkage . . . 19
How the Planner Plans an Order . . . 19
Pull Planning . . . 19
Push Planning . . . 20
About Blocked Demands . . . 21
Example Plan . . . 21
Gantt Chart View . . . 22
Order Details View . . . 23
Defining Item Lead Times . . . 24
Standard Lead Time . . . 24
Expedited Lead Time . . . 24
Handling Temporary Vendor Lead Time Problems . . . 25
Using Item Acquisition Schedules . . . 26
Defining the Acquisition Schedule . . . 26
Assigning the Acquisition Schedule to an Item . . . 26
Using Item Availability Constraints . . . 27
About Usage of Phantom Item Inventory . . . 27
If the Phantom is an End Item . . . 27
About Phantom Inventory Usage . . . 27
Using Supply Tolerance . . . 28
Exception Messages . . . 29
General Notes . . . 29

Defining Order Priorities . . . . . . . . . . 29
Optimizing Usage of Purchased Item Supplies . . . . . . . . . . 29
Generating Purchase Order Exceptions . . . . . . . . . . 30
Replenishing Safety Stock . . . . . . . . . . 30
Safety Stock for Manufactured Items . . . . . . . . . . 30
Avoiding Too-large, Too-early Replenishment Quantities . . . . . . . . . . 30
Safety Stock for Purchased Items . . . . . . . . . . 31
Safety Stock for Transferred Items . . . . . . . . . . 31
Cross-referencing Orders . . . . . . . . . . 32
Using Pull-Up Orders . . . . . . . . . . 33
Defining a Time Fence for Inventory/Supply Usage . . . . . . . . . . 33
Planning Passes . . . . . . . . . . 34
Enforcing the Time Fence at the Expense of the Due Date . . . . . . . . . . 35
Time Fence Messages . . . . . . . . . . 35
Optimizing Usage of Manufactured Item Supplies . . . . . . . . . . 36
Using Overlapping Operations . . . . . . . . . . 36
Using Minimum and Maximum Lot Sizes . . . . . . . . . . 37
Using Maximum Order Lot Size . . . . . . . . . . 37
Using Order Minimum/Multiple . . . . . . . . . . 37
Defining the Earliest Start Time for Planning an Order . . . . . . . . . . 38
Using Alternate Items . . . . . . . . . . 38
Managing Resource Availability and Interruptions . . . . . . . . . . 38
How the Break Rule Affects Reallocation . . . . . . . . . . 39
Influence on Resource Allocation . . . . . . . . . . 39
**Chapter 3 About the Scheduler . . . . . . . . . . 41**
How Event-based Scheduling Works . . . . . . . . . . 41
About the Scheduler Trace Log . . . . . . . . . . 42
Event Processing Example . . . . . . . . . . 42
Using Scheduler Rules . . . . . . . . . . 45
Controlling Simultaneous Order Start Dates . . . . . . . . . . 46
Order Release Rules . . . . . . . . . . 46
Using Sequencing and Selection Rules . . . . . . . . . . 46
Resource Request Queues . . . . . . . . . . 46
Sequencing the Queue . . . . . . . . . . 47
Selecting Loads from the Queue . . . . . . . . . . 47
Sequencing and Selection Rules Example . . . . . . . . . . 47

Sequencing Rules . . . 54
Selection Rules . . . 56
Improving Speed of Selection Processing . . . 59
Using Values and Attributes . . . 59
Using Resource Group Member Selection Rules . . . 60
Using the Default Select in Sequence Rule . . . 60
Using Complex Rules . . . 60
Member Selection Example . . . 61
Resource Group Member Selection Rules . . . 66
Managing Resource Availability and Interruptions . . . 66
Handling Shift Overrun Situations . . . 67
Reducing Resource Availability . . . 68
Increasing Resource Availability . . . 69
Resource Reallocation During Interruptions . . . 69
Defining Operation Setup Time . . . 71
Defining the Setup Resource Group . . . 71
Defining When to Apply Setup Time . . . 71
Defining the Amount of Setup Time . . . 72
Creating a Setup Lookup Table . . . 72
Setup Time Example . . . 72
Using Overlapping Operations . . . 74
Split Processing . . . 74
Load Splitting with Uneven Quantities . . . 74
Multiple Resources in Resource Group . . . 75
Accumulation Processing . . . 75
Overlapping Operations Example . . . 75
Splitting a Load at a Single Operation . . . 76
Splitting Load Example . . . 77
Running the Scheduler . . . 77
Output Data . . . 78
Summary Performance Data . . . 78

# About This Guide

This manual provides instructions for customizing the Scheduler, i.e. creating user-defined rules. It begins with a discussion of how to write User Defined Rules with logic not included in the standard system and then presents several sections in reference format which describe the system modeling constructs used to customize the Scheduler.

There are 8 different rule types that can be user defined. In the drop down list on the form you will see something like “User Defined Allocation Rule 12, etc. That will indicate the user defined rule numbers that you can create. Following is a table which contains each type of rule, where it is found on the Product Name (variable) forms, what numbers are allowed, and how it fits into the Scheduler’s logic flow. Following that table is a figure which shows how a job progresses through the system from the Scheduler’s rule point of view.

## Intended audience

This information is intended for advanced APS users.

## Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at http://www.infor.com/inforxtreme.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

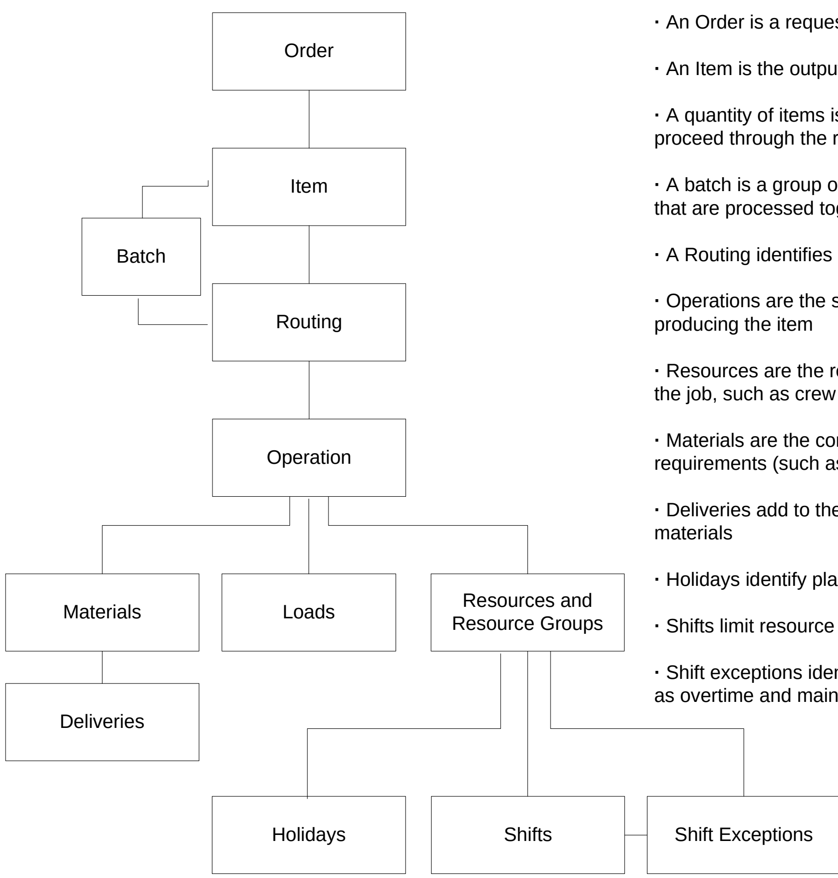
# Chapter 1: Plan and Schedule Data Model

1

This chapter introduces the hierarchy of data items the APS Planner and Scheduler use when creating synchronized plans and schedules.

## Database Overview

The diagram below illustrates the data hierarchy in the SQL Server database used by both the Planner and the Scheduler.




- An Order is a request for item production
- An Item is the output of the production
- A quantity of items is divided into Loads, which proceed through the routing
- A batch is a group of potentially disimilar items that are processed together
- A Routing identifies how to make the item
- Operations are the steps in the process of producing the item
- Resources are the requirements for performing the job, such as crew or machines
- Materials are the consumable production requirements (such as nuts and bolts)
- Deliveries add to the inventory level of materials
- Holidays identify plant shutdowns and holidays
- Shifts limit resource availability
- Shift exceptions identify special situations such as overtime and maintenance

Each item in the diagram is described in detail in the following sections of this chapter.

# Orders

## Database table: ORDER

Orders authorize production at a manufacturing facility for a single level of the bill of material. You can release an order for production based on a scheduled release date.

There are three order types—*new*, *in-process*, and *unconfirmed*.

The ORDER.STATUSCD code specifies the order type:

- **New**: represents an order released into the manufacturing system at a specific date and time.
- **In-process**: represents an order released into the manufacturing system before the start date (ALTSCHED.STARTDATE) of the Scheduler run.
- **Unconfirmed**: represents an order scheduled for release at a specific date and time. However, APS releases these orders only if the ALTSCHD.IGNUNCOMFG field is set to "N."

The APS Planner uses only the New order type.

# Items

## Database table for Scheduler: PART

Database table for Planner: MATL

The item component specifies the characteristics of the item being manufactured. An item is referenced by an order. In the Scheduler, items differ from materials (defined in the MATL table) in that materials generally represent completed subcomponents. In the Planner, both items and materials are defined in the MATL table.

An item can represent:

- finished product
- component item assembled later
- raw purchased items
- component item used as a material

# Batches

## Database table for Scheduler: BATCH

The batch definition component specifies how potentially disimilar items are combined together to be processed as a unit, or load. Examples of such processes are ovens or paint booths.

# Routings and Operations

## Database tables: JOBSTEP, JS19VR, and PROCPLN

A routing (also called a process plan) directs the manufacturing of an order.  Each routing consists of a sequence of one or more operations (also called jobsteps) needed to produce a *load* of items.  Each operation indicates the processing time, as well as the necessary machine, operator, storage space, and material.

When a load arrives at an operation, the system allocates resources or resource groups to—or frees them from—the load (based on how you defined the operation).

Each operation references a specific routing.  The first operation for a routing is indicated in the PROCPLN table.  After the first operation, each operation names the next operation (in the JOBSTEP.NEXTJSID column).

A routing is designated on an item’s order definition or, if defaulted on the order, on the item definition.

# Resources

## Database table: RESRC

As a load of items proceeds through the routing steps, it uses resources. Resources represent required elements of the manufacturing process such as machines, operators, and fixtures. Resources are the physical elements used to manufacture items, and usually constrain production flow.

# Resource Groups

## Database table: RGRP, RGRPMBR

A resource group allows you to classify resources that can perform the same function as a group, without member resources losing their individual identities. For example, you could define a resource group called MILLS to include resources MILL1, MILL2, and MILL_AUTO.

The system can use a resource group when a load on an operation needs a resource from a group of resources, but does not need a specific resource in the group. The load needs to use any available resource. The system selects a resource from the resource group and gives it to the load. The member resources may be working different shifts, have different current setups, or have different processing rates.

When defining a resource group, you provide this information:

- RGRP.RGID: a unique identifier that other components can reference.
- RGRP.ALLOCRL: a rule to use for selecting a particular resource member from among other members (Scheduler only).
- RGRPMBR.RESID: the member resources of the group.
- other options, discussed later.

The group members are resources you must define in the RESRC table and reference in RGRPMBR table. You can define a resource as a member of more than one resource group. Such a resource may be allocated and freed directly even though it is in one or more resource groups.

## Infinite Resource Groups

You can define a resource group to have infinite capacity. If a resource group is defined as infinite, the system does not constrain the resource group's member resources.

To define a resource group as infinite, use the RGRP.INFINITEFG field (for the Scheduler) or the RGRP.INFCAP field (for the Planner). Both the Planner and Scheduler programs consider infinite resources to be infinite only while the resources are on a working shift.

# Materials

## Database table: MATL

Materials represent inventories of items that are not actively processing. Materials generally represent subcomponents that continue processing under a different item and/or order number. Some example uses of materials are:

- Pieces of bar stock cut to size for downstream processing
- Electronic circuitry added to blanked and punched ceramic boards
- Co- or by-products

Materials are defined for each classification of consumable inventory. A material may represent a broad classification of items, such as a “subassembly” required by numerous orders. Materials can also represent specific classifications required only by a particular order.

The characteristics of materials remain constant throughout the Scheduler run except for the current quantity in inventory. When processing loads on the material interaction operations, the Scheduler increments or decrements the material’s current quantity. Before starting the run, you specify the initial quantity in the MATL.QUONHAND field.

**Note:** The Planner uses the MATL table for both end items and material subcomponents.

## Representing Component Items

You can use materials as the interface between the routings of two or more orders. For example, you can separate an order for a finished item into orders for one or more component items and an order for the finished item. You would create a material to represent each component item. You would then define a routing to create each type of component item.

The component item routing would add to its component item material as the component items are completed. The routing for the finished item would remove from the component item materials as necessary to make the finished item. If the finished item routing needs any component item material that is not currently available, it will wait until the material becomes available.

# Shifts

## Database table: SHIFT

As described earlier, a shift specifies the set of shift intervals (time periods) used to determine when resources on that shift are “up” or available to work on operations.

You must specify shifts in terms of weeks. This weekly pattern repeats for all weeks in the Scheduler horizon. A week starts Sunday morning at 00:00:00 and runs until Saturday at midnight. All periods within a week not specified with an interval period are assigned as not working, or down. Daily cycles do not automatically repeat. You must define each daily interval separately.

Resources may use more than one defined shift, and the shifts can overlap. The Scheduler considers the resource available during any of its shifts including an overlap period. More than one resource can use the same shift. If you do not specify a shift, the resource is available for work 24 hours per day, seven days per week.

When a shift reaches the end of an up interval, resources on that shift become unavailable. By default, a resource stops processing when its shift ends.

**Note:** In the Scheduler, when a resource’s shift ends, the load with the allocated resource uses the resource’s Allocation setting to determine whether to wait for the same resource or to try for other resources. See “Resource Reallocation During Interruptions” on page 69 for more information.

Use shifts when the manufacturing equipment or operator follows a regular pattern of up and down periods that make them only available to perform operations during up periods. Two example situations that might require shifts are:

- An operator group that works on the day shift only.
- A machine that requires operators goes down during the change between first and second shift, and between the end of second shift and the start of first shift.

# Shift Exceptions

## Database table: SHIFTEXDI

At times you must make exceptions to your basic shift patterns to handle special situations. Shift exceptions handle many special cases such as:

- working overtime
- working on a holiday
- reducing the length of a shift

- canceling all work on a shift

**Note:** Shift exceptions are applicable only for resources that are on one or more shifts.

# Holidays

## Database table: CAL

Use the holiday component to specify periods of scheduled down-time for all resources that are on one or more shifts.  Holidays address production down-time such as holidays, vacations, and plant shutdowns for inventory.  Periods not specified under the holiday period operate as under normal shifts.

To define the down period, specify the STARTDATE and ENDDATE in the CAL table.  The down period may span more than one day.

# Delivery Schedules

## Database table: MATLDELV

Use a delivery schedule to specify a time-based add-to-material event.  A delivery schedule can represent periodic delivery of raw material from an outside source, or arrival of components from an upstream process that isn't included in your model (typically a purchased item).

The system adds each delivery amount to the material at delivery time, if sufficient capacity is available.  Otherwise, the delivery rule (MATL.DELVRULE) specified on the material determines how to handle the over-capacity delivery.  The system maintains a queue of delivery requests separate from the “add-to” request queue for loads.  The delivery priority (MATL.DELPRI) field on the material defines the order in which to consider the delivery queue in relation to the normal add-to-material queue.

# Chapter 2: About the Planner

2

By comparing an order against a long-term plan for your shop floor, the Planner provides real-time projections of when you can satisfy the order. For example, during the order entry process, the Planner can consider the new order against other existing orders and the capacity of your shop floor and return a realistic completion date.

The long-term plan includes all current demands, supply orders, on-hand inventory, the routings and bills of material for all end items and components, your shifts, and resource information.

This chapter describes the considerations the Planner uses when generating a plan.

## Database

The Planner runs against an in-memory database, which your ERP system populates by entering data in corresponding SQL Server database fields.

The planer database resides in memory, providing real-time order promising capabilities. This structure provides increased efficiency and allows faster responses to disruptions in the plan. And because the database is memory-resident, most plan regenerations complete very quickly (depending on the size of your database).

After the Planner generates plan data, it transfers the output data back into the SQL tables.

## Running the Planner

You can run the Planner in two situations:

- Order Promising during order entry
- Full plan regeneration

## Order Promising/Get CTP

When you create a new demand in your ERP system, whether a customer order, estimate, forecast, transfer order, etc., you can test its viability against the plan using the Get CTP function. This function incrementally plans the demand in a temporary "test" copy of your planner database and calculates the projected completion date.

For example, when an order-entry user accepts a new customer order, he can use the Get CTP function and receive a projected date for the new order.

## Plan Regeneration

On a periodic basis, you regenerate the plan to resynchronize all demands with up-to-date on-hand inventory, planned supply (such as purchase orders), and resource availability information. In most situations, you run the plan regeneration for a long horizon, such as one or two months. If you are using Scheduling, you will likely run the Scheduler with a shorter horizon, such as one or two days. The plan regeneration generates planned orders and exception messages. You can then respond to the exception messages and firm the planned orders into actual job orders, purchase orders, and other types of orders.

## Output Data

During a plan regeneration, the Planner populates the following SQL Server tables:

| Table | Contents |
| --- | --- |
| ALTPLAN | The date and time the Planner was last run. |
| DOWNPLAN | A record for each period of time any resource was idle, and the reason for the idleness. |
| INVPLAN | A record for each supply and demand that affects material inventory. |
| JOBPLAN | Information about each operation in the plan (such as the actual duration of each operation). |
| MATLPLAN | Information about each item occurrence in the plan (such as the start and end date of the item's manufacture and the order that caused the manufacture or purchase). |
| MSLPLAN | Information about the starting inventory level and safety stock replenishment date for each item. |

| Table | Contents |
|---|---|
| ORDPLAN | The projected order completion date the Planner calculated for each order. |
| POEXCEPT | Exception messages for orders that need to be moved in or out. |
| RESPLAN | Information about every resource used in the plan (such as the start and end date of the resource usage). |
| TODEMAND | A record for every order transferred from a remote site (after a multi-site global plan regeneration). |
| TOSUPPLY | A record for every order transferred to a remote site (after a multi-site global plan regeneration). |

# Basic Planning Considerations

The Planner plans one line item per order, one order at a time. It plans the end item completely before considering the next order. This means the Planner plans each operation of each order recursively through its bill of material, considering all the necessary resource availability and material requirements. When the Planner plans an order, it reserves material and resource capacity. Therefore, it must plan the next order using the material and capacity that remains unreserved.

## Forecasting

The Planner plans forecasts as orders according to your defined order priority. When the Planner begins planning the forecast, it first allocates any available on-hand inventory and planned supplies (within the supply usage tolerance), further reducing the forecast quantity to be planned. Keep in mind that other demands, such as customer orders, will probably have priority over forecasts for using on-hand inventory and planned supplies, depending on how your order priority is set.

The Planner then generates a planned order to satisfy the remaining forecast quantity. New incoming demands can use this forecast’s supply only after the planning run is complete, when you incrementally plan them during an order promise/CTP process.

If a demand consumes a forecast, any subsequent order promise/CTP process you perform on that demand will consider this consumption of forecast supply in the calculation of the availability date.

## Resource Collisions

When planning operations in a routing, the Planner avoids situations in which two operations need the same resource for the same time period. When the Planner encounters such a resource collision, it adjusts the planned time of the lower-priority order to avoid the collision and to ensure a realistic plan.

This behavior is somewhat different than the Scheduler, which uses resource queues and resource-selection rules to deal with simultaneous requests for the same resource.

## Material Components

Material components required in the production process are either purchased or manufactured items. The Planner plans material components based on the time the components are actually needed (considering the assembly time). Materials are required at the start of the operation or the start of the job based on the setting of the ALTPLAN.SSFLAG bit9.

## Dynamic Reallocation by Order Priority

The Planner temporarily allocates all finished goods inventory, planned supplies, and WIP until the last possible moment in the planning cycle. Consequently, when you run the plan regeneration, the Planner treats all finished inventory, supply, and work-in-process inventory as unreserved, and reallocates it according to the ranking of the order’s priority type defined in the OPRULE table.

Each priority type, such as "Order Entry Demand" or "Forecast Demand," has an assigned priority value. Order types with lower priority numbers consume available capacity before order types with higher priority numbers.

## Realistic Order Completion Dates

The Planner calculates realistic order completion dates. During order entry, the user creates a new order and enters a request date to determine availability. The Planner determines whether the request date can be met and provides a "projected date" (also known as a capable-to-promise (CTP) date). This projected date is the earliest date your facility can make these items without disrupting the current set of planned orders.

The order-entry user can then set the official due date to the Projected date, to the original customer-requested date, or to some other date. If the chosen due date is earlier than the projected date, the Planner projects the order to be completed late. When you regenerate the plan, this date may change as higher priority orders consume capacity and material before lower priority orders.

The following terms are used to describe order dates in APS documentation.

- Entry Date (ORDER.ARIVDATE). The date the order is entered in the APS system. This date is for record keeping; the Planner does not typically use it.
- Request Date (ORDER.REQUDATE). The date the customer requested the order to be completed. You can specify this date when you enter the order, or you can leave it blank.
- Projected Date (ORDPLAN.CALCDATE). The capable-to-promise date the Planner currently projects the order will complete, based on current factory conditions.
- Due Date (ORDER.DUEDATE). The date promised to the customer that the order will be complete. This is the due date of the order. Once set, the Planner always attempts to complete an order by its due date.

## Material Scrap Factor/Shrinkage

The APS Planner calculates planned order starting quantities to compensate for any scrap/shrinkage that results from jobs. For example, if you need 100 widgets, and the job that produces the widgets creates 10% scrap/shrinkage, the APS Planner increases the starting quantity by 10%, to 110, so that the final quantity will exactly equal the required quantity. In cases where jobs are split into serarate demand and supply orders, the supply portion will be adjusted down by the shrink amount.

# How the Planner Plans an Order

When the Planner plans an order, it first searches backward from the due date to allocate the needed quantity. This is referred to as pull planning. If the plan cannot find sufficient on-hand inventory/ planned supplies, or projects a start date for a planned order that is in the past, the order is late. The Planner instead searches forward (or push plans) from the current date.

**Note:** The Planner considers the Time Fence and Pull-up Order options during the planning process; therefore, it may perform up to eight planning passes (for example, plan with Time Fence enabled; if that fails, plan with Time Fence disabled, and so on). These additional planning passes are described in “Defining a Time Fence for Inventory/Supply Usage” on page 33.

## Pull Planning

The Planner pull plans an order according to this sequence:

1 Allocate planned supplies of the end item to satisfy the demand quantity.

2 For any unsatisfied demand quantity, allocate on-hand inventory. If sufficient planned supplies are available or sufficient inventory is on hand to satisfy the order, the planning is complete.

3 If unsatisfied demand quantity remains, plan a manufacturing order or purchase order for the remainder. This step varies according to the type of item being planned:

- Unconstrained items (lead time=0): Unconstrained items require no further planning. An item is unconstrained if the MATL.FLAGS bit5 is enabled. The Planner plans the demand at the requested date, and assumes it to be satisfied.
- Purchased items: The Planner uses lead time to determine whether the item can be obtained in time. If lead time is within the request date and the current date, the Planner creates a planned purchase order at the request date, and planning is complete. If lead time extends into the past, the pull planning fails and the demand is push planned.
- Manufactured items: The Planner plans the operations through the routing, starting with the end item’s last operation, considering material and resource availability and WIP at each operation. All combinations of resources are considered until the fastest combination is found (unless you have limited the number of combinations through the ALTPLAN.PULLITERS parameter). The pull-planning fails if the Planner is unable to plan any given operation with enough time remaining to plan all remaining upstream operations. At any point, if enough WIP is found to satisfy demand, the pull is complete and successful.  After successfully pull-planning the current item through its routing, the Planner considers that item’s components. From this point, it recursively pull-plans each component. If any item’s plan extends into the past, all pull planning for the order fails and the Planner switches to push planning.

4 If the item has an order minimum value specified (MATL.ORDMIN), and the quantity the Planner plans to make or purchase will satisfy the entire demand quantity, it then frees the supply and inventory allocated in step 1.

## Push Planning

If the pull planning projects the order to be late, the Planner performs push planning.  Push planning involves planning forward from the current time out to a projected completion date.

The Planner push plans an order according to this sequence:

1 Calculate the end-item quantity needed. Allocate on-hand inventory to satisfy the end-item demand quantity.

2 Pass the remaining end-item demand quantity to the components, down to the lowest levels in the bill of material. Allocate on-hand inventory to satisfy the component demand quantity.

3 Plan the components in the lowest levels of the bill of material. For manufactured components, each operation in the component's route is planned in forward sequence (starting with the first operation). The result is the component’s earliest possible completion date, which will set its parent item's start date. For purchased components, set the component’s completion date to the current date + the component’s lead time.

4 Search for unallocated planned supplies, starting with the current date and searching out to the date calculated in the previous step. Allocate supplies only if the entire quantity can be satisfied.

5 When forward planning is finished, the end-item completion date becomes the demand's projected date.

6 The system may plan non-critical operations earlier than they are actually needed to meet the demand. To optimize the plan, the system runs a series of additional pull planning "iterations" starting from the demand's new projected date:

a. Perform the pull plan starting from the projected completion date. If this step projects a start date in the past, the system then pull plans the demand from the end of the plan horizon.

b. Perform one or more pull-planning iterations within a defined time window, dividing the window in half with each iteration until a feasible plan is found that is within the ALTPLAN.ITERDAYS (if this is a full plan regeneration) or ITERDAYSCTP (if this is a single-order/CTP plan) number of days of the need date. The beginning of the time window is always the demand's need date. The end of the time window will be either the projected completion date from the push plan or the end of the plan horizon (if the above-mentioned pull failed).

   For example, the first iteration pulls from the midpoint between the projected date and the need date. If that pull succeeds, the next iteration pulls from the midpoint between that new projected date and the need date. The process incrementally moves closer to the need date until it finds a plan that works (that is, a plan that doesn't calculate a start date in the past) and has a projected completion date that is within the specified days of the need date.

## About Blocked Demands

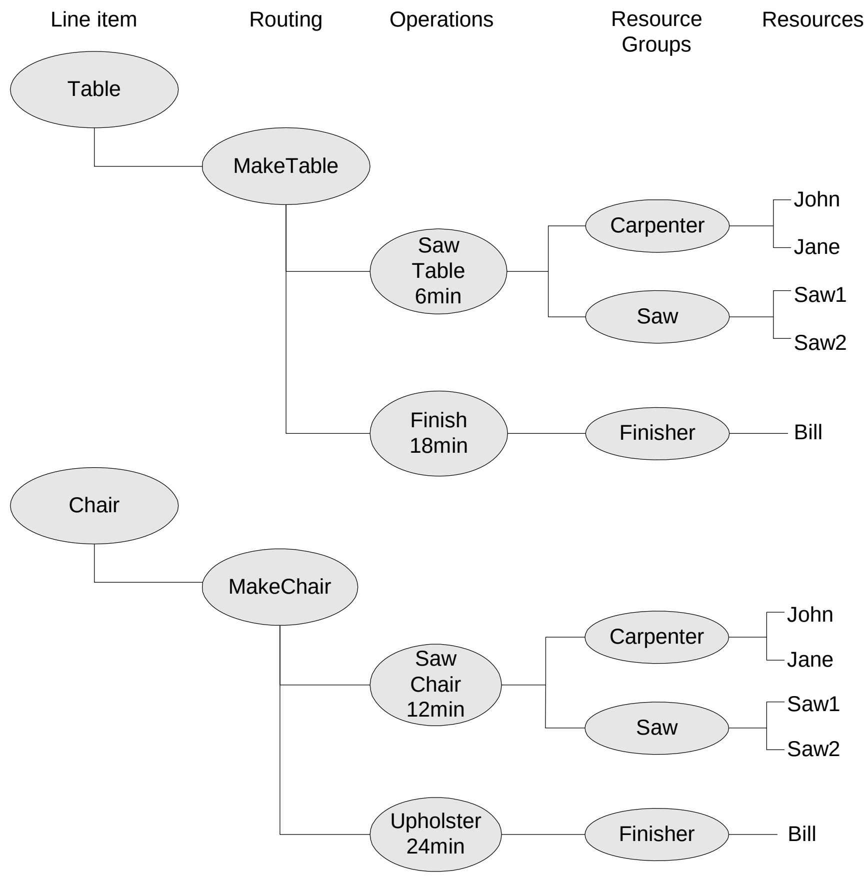
If the system is unable to push plan a demand in the time between the current date and the end of the plan horizon, the demand is displayed as "blocked," and cannot be planned.

# Example Plan

This simple example illustrates how the Planner processes a set of data and generates a plan from a set of orders. This example facility manufactures tables and chairs. Assume that the tables and chairs do not require any manufactured or purchased components.

| Item | Operation 1 | Operation 2 |
|---|---|---|
| Table | SawTable | Finish |
| Chair | SawChair | Upholster |

These operations each require certain resource groups.  They have cycle times of 6, 12, 18, and 24 minutes per item, as shown below.




Each resource has a shift that defines its available time. They are each available from 9:00 a.m. to 6:00 p.m. with a 1-hour break for lunch and maintenance. For simplicity, the shifts are defined as 9-5 instead of breaking an 8-5 shift for a 1-hour break.

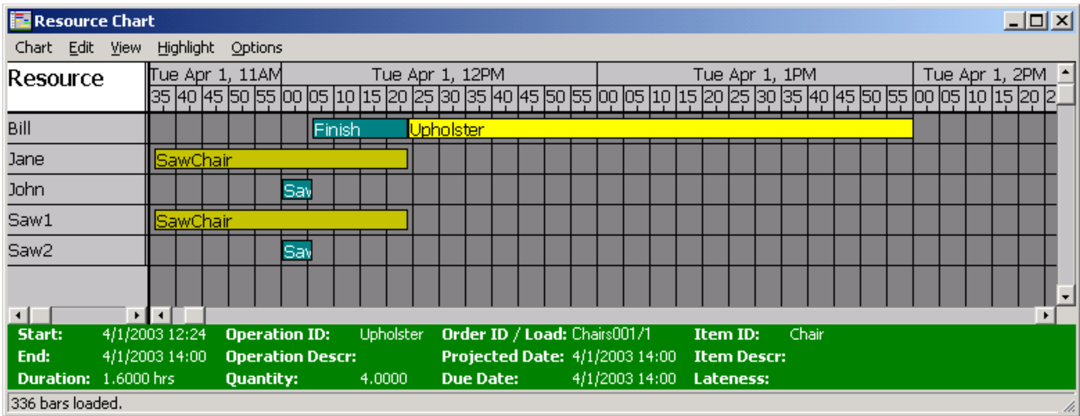
## Gantt Chart View

The screen below from the Gantt Chart screen in the APS Analyzer shows how the system allocated the resources to each operation. There are currently two orders in the system, each with a single line item:

| Order | Quantity | Due Date |
|---|---|---|
| Chairs001 | 4 | 4/1/2003 14:00 |
| Tables001 | 1 | 4/1/2003 14:00 |

The current date/time is 3/31/2003 at 9:00 a.m.

A pull plan for a 14:00 requested completion time promises 14:00 for order 1, as shown below. Operations are backward planned from the due date of 2:00 p.m. (14:00) on 4/1/2003. The order "Chairs001" completes just in time at 14:00. The order "Tables001" completes around 12:24, to avoid a resource collision at the finish area.




In summary, the system planned each order in reverse routing sequence, looking for available resources to be able to plan all the operations between the due date and the current date.

## Order Details View

In the Analyzer's Order Details view, you can view the details of how the system planned the two orders:

| Chairs001 | Start Date / Request Date | End Date / Usage Date | Quantity | Delay (Days) |
|---|---|---|---|---|
| Chair | 4/1/2003 11:36 | 4/1/2003 14:00 | 4.00 | |
| SawChair | 4/1/2003 11:36 | 4/1/2003 12:24 | 4.00 | |
| Carpenters | | | | |
| Saws | | | | |
| Upholster | 4/1/2003 12:24 | 4/1/2003 14:00 | 4.00 | |
| Finishers | | | | |

| Tables001 | Start Date / Request Date | End Date / Usage Date | Quantity | Delay (Days) |
|---|---|---|---|---|
| Table | 4/1/2003 12:00 | 4/1/2003 12:24 | 1.00 | |
| SawTable | 4/1/2003 12:00 | 4/1/2003 12:06 | 1.00 | |
| Carpenters | | | | |
| Saws | | | | |
| Finish | 4/1/2003 12:06 | 4/1/2003 12:24 | 1.00 | |
| Finishers | | | | |

# Defining Item Lead Times

When planning an order for a purchased item, the Planner calculates lead time from a combination of fixed, variable, and expedited lead time values.

## Standard Lead Time

You define the standard fixed and variable lead time for an item in the MATL table, in the FLEADTIME and VLEADTIME fields. FLEADTIME is the fixed lead time for the item (in hours), regardless of the order quantity. VLEADTIME is a lead time value (in hours) that is multiplied by the ordered quantity of this item. The Planner adds these two values together to determine the item's lead time on the order.

For example, if the order quantity is 100, the FLEADTIME=48 and VLEADTIME=4, the calculation is performed like this: 48 + (100 * 4) = 448 hours.

## Expedited Lead Time

You can specify a shorter lead time value for situations that require expedited processing. The Planner always uses the standard lead times on the first pull-planning run. If the first run fails to plan to a date because of insufficient capacity, inventory, or time, the second and/or third planning runs will use expedited lead times (if specified). There are two ways to specify expedited lead time:

- as a global reduction to the standard lead times
- as an item-specific, alternate set of fixed and variable lead time values.

You can specify the global lead-time reduction value for fixed lead time in ALTPLAN.FEXPLTIME. Specify the global reduction value for variable lead time in ALTPLAN.VEXPLTIME. The Planner applies these values only if the ALTPLAN.SSFLAGS field sets the Use Expedited Lead Time flag AND the expedited lead time value(s) for the item in the MATL table are blank. In this case, the Planner reduces the standard FLEADTIME by the value in FEXPLTIME and/or the standard VLEADTIME by the value in VEXPLTIME (to a lead time value 0 or higher).

For example, consider these values:

- ALTPLAN.SSFLAGS: Any integer value that sets bit5 (enable global expedited lead times).
- MATL.FLAGS: Any integer value that turns off bit9 (disable item-level expedited lead times).
- ORDER.ORDSIZE: 100
- MATL.FLEADTIME: 48
- MATL.VLEADTIME: 4
- ALTPLAN.FEXPLTIME: 2
- ALTPLAN.VEXPLTIME: 1

The calculation is performed like this: 48 – 2 + (100 * (4 – 1)) = 346 hours

You can specify item-specific expedited fixed lead time in MATL.FEXPLTIME. Specify expedited variable lead time in MATL.VEXPLTIME. The Planner applies these values only if the MATL.FLAGS field for the item sets the Use Expedited Lead Time flag.  In this case, the Planner uses the expedited lead time values in place of the standard lead time values.

For example, consider these values:

- MATL.FLAGS: Any integer value that sets bit9.
- ORDER.ORDSIZE: 100
- MATL.FLEADTIME: 48
- MATL.VLEADTIME: 4
- MATL.FEXPLTIME: 12
- MATL.VEXPLTIME: 2

The calculation is performed like this: 12 + (100 * 2) = 212 hours

# Handling Temporary Vendor Lead Time Problems

Your vendors may have temporary problems meeting their standard lead times for certain items. To ensure the Planner creates the planned order with the appropriate lead time in this situation, you can define an "earliest receipt date."

This date temporarily overrides the item's standard and expedited lead times. When the Planner creates a planned order for this item, it determines that the item can be received either by the current time plus lead time or by the earliest receipt date, whichever is later. The earliest receipt date overrides standard and expedited lead times as follows:

- Standard lead time: earliest receipt date overrides standard lead time only until the time between the current date and the earliest receipt date is within lead time (after which normal lead time planning resumes).
- Expedited lead time: if the earliest receipt date is between the standard lead time and the expedited lead time, it overrides the expedited lead time and becomes the "effective" expedited receipt date. Once the earliest receipt date is within the expedited lead time, the normal expedited lead time will be used.

You only need to define the earliest receipt date when a vendor lead time problem occurs, and you do not need to "reset" the date when the lead time situation returns to normal.

To define an earliest receipt date:

1 Open the Item Editor and select the purchased item for which to define an earliest receipt date.

2 Select the **Attributes** tab.

3 Click the **Add** button. The Add Attribute dialog displays.

4 In the Attribute ID field, enter **PODATE**.

5 In the Attribute Value field, enter the date that will be the earliest receipt date for this purchased item (for example, 3/11/2003).

6 Click **OK**.

# Using Item Acquisition Schedules

You can use an acquisition schedule to restrict the application of lead time for an item during a push or pull plan (for example, to represent the concept of business days). For example, the interval of the acquisition schedule could represent the time between an order being placed and the time it is satisfied, including paperwork, transit, unpacking, etc. During this interval, the demand does not consume any capacity.

You can specify an acquisition schedule for each purchased item and for each manufactured item-site combination (for manufactured items, this feature is useful for transferred items).

Any item can use the same acquisition schedule. For example, during pull-planning, the Planner starts at the item's need date and searches backward in time until it finds a lead-time (standard and expedited) amount of "working time" in the acquisition schedule's shift interval.

The Planner interprets the lead time for a purchased item (or transit time, for a manufactured item produced at another site) as working time on the acquisition schedule. A lead-time amount of working time from the item's need date offsets the time the purchase order or transfer order must be placed at the vendor or supply site.

For example, consider these conditions:

- An acquisition schedule is assigned to item 123, in which the interval is 12 p.m. to 5 p.m. on Saturdays.
- Item 123 has a fixed lead time of 0.0 and a variable lead time of 1 hour.
- There is an order for item 123 with a quantity of six, which is due in three weeks.

Given the shift interval of 12-5, one day per week, the Planner has 15 hours of working time within the three-week window. To determine the latest time the order can be placed to satisfy the requirement, the Planner looks three weeks into the future and then searches backward until it finds six hours of working time. The result is 4 p.m. on the second Saturday.

## Defining the Acquisition Schedule

You define the acquisition schedule as a shift in the SHIFT table. Define the schedule's interval in the SHIFT.SDAY/STIME and SHIFT.EDAY/ETIME fields. It is not necessary to have a resource working on this shift (an acquisition schedule is similar to a resource with infinite capacity being associated with the shift).

## Assigning the Acquisition Schedule to an Item

Assign an item to use the acquisition schedule by specifying the shift ID (SHIFT.SHIFTID) of the schedule in the item's MATL.SHIFTID field. If you do not specify an acquisition schedule for an item, the Planner calculates lead time normally, as calendar time.

# Using Item Availability Constraints

When you set up your item database in the MATL table, you can specify whether the availability of a given item will be a constraining factor in the generation of a plan.  The MATL.FLAGS field controls item constraints (such as "unconstrained purchased item").

If no item constraints are enabled, the Planner reports the times (or "exceptions") at which projected component inventory is insufficient to support the plan. You can use this exception information to obtain or manufacture the items to meet the projected shortages.

If purchased item constraints are enabled, the Planner delays the operation until sufficient material is projected to be available—either until a receipt of sufficient quantity is planned, or until the lead time of the item, whichever is first.  For constrained manufactured items, the Planner plans production of the component through the bill of material.

# About Usage of Phantom Item Inventory

A phantom is a subassembly that is assembled only to be included in another assembly. A phantom represents an item that is physically built, but not stocked, before being used in the next step or level of manufacturing. An item is a phantom when bit4 of the FLAGS column is enabled in the item's MATL record. When you run the Planner, the system bypasses the phantom item and creates planned orders only for the material components in the phantom item's current bill of material.

The system plans phantom items differently depending on several conditions.

## If the Phantom is an End Item

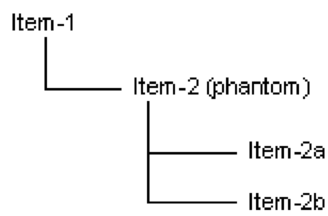
Phantoms are usually subassemblies. However, if a phantom is the finished "end" item, the system plans it normally, generating any necessary planned orders for it along with the planned orders for its components.

## About Phantom Inventory Usage

If a stocked quantity of a phantom is on hand, the system reduces the demand quantity for the phantom's material components by the amount of the phantom's on-hand quantity.

Example:

Consider the usage of the phantom item "Item-2" in this bill of material:




- There is a customer order for Item-1 for a quantity of 20.
- You have 5 of Item-2 on hand.

When you run the Planner, the system generates a planned order normally for Item-1. But before it blows through the phantom Item-2 to plan its components, it first reduces the component demand quantity by Item-2's on-hand quantity of 5. So, instead of planning a demand of 20 for Item-2a and Item-2b, it plans for 15.

# Using Supply Tolerance

When the Planner pull plans a demand, it begins by searching backward from the demand's due date for supplies that are planned to be available between the demand's due date and the current date. If the supplies available during that time do not satisfy the demand quantity, the system searches for on-hand inventory. If the demand is still not satisfied, the system can then use supply tolerance to search for supplies that are planned to be available AFTER the demand's due date.

You can define a default, global tolerance value in ALTPLAN.SUPPLYTOL or you can define an item-specific tolerance value in MATL.SUPPLYTOL. Define supply tolerance as a window of time, in terms of hours or days. The processing depends on whether you enter a positive or negative number:

- Positive number: the system searches for supplies starting from the demand's due date/time + the tolerance number of hours. For example, if the demand's due date is 3/5/2004 at 9:00, a value of 4 means the demand can use any supplies that are available between the current date/time and 3/5/2004 at 13:00.
- Negative number: the system searches for supplies starting from the demand's due date at 00:00 (midnight) + (24 hours * the positive tolerance value). For example, if the demand's due date is 3/5/2004 at 15:00, a value of -4 means the demand can use any supplies that are available up to 3/9/2004 at 00:00. In the same example, a value of -1 means the demand can use any supplies available up to 3/5/2004 at 24:00 (the same day).

A value of 0 entered in MATL.SUPPLYTOL bypasses the item-level tolerance; the system instead uses the global tolerance value.

## Exception Messages

When a tolerance supply is used, the "Move PO In" exception message is generated in the POEXCEPT table. You will need to move the supply to the date specified in this exception message.

## General Notes

- The system uses this parameter only if bit6 is enabled in the ALTPLAN.SSFLAGS column and bit8 is enabled in the MATL.FLAGS column for the item.
- The system uses a 24-hour day, 7-day week calendar when applying these values.
- If the system must use tolerance supplies, it uses as much of the tolerance supplies as possible before it uses any inventory or supplies available before the demand's due date.

# Defining Order Priorities

You can define order priority rules in the OPRULE table. Rules govern the capacity reallocation, such as "plan orders with earliest due date first" or "plan orders from xyz category first."

If two demands with the same order type request resources at the same time, the system allocates the capacity to the order with the earliest due date first.

For example, if an order suddenly appears directly from the CEO's desk with the request to ship as soon as possible, you can regenerate the plan, which causes it to allocate material and capacity according to the revised priorities.

# Optimizing Usage of Purchased Item Supplies

The Planner allocates supplies and inventory to fulfill demands based on order categories and priority. In situations involving complex bills of manufacture, this can result in a higher priority order reserving supply and inventory before a lower priority order, even if the lower priority order needs the quantity sooner. And if the lower priority order cannot satisfy the demand from supplies and inventory, the Planner purchases the unsatisfied quantity (earlier than you really need it).

To avoid this situation, set bit2 in ALTPLAN.SSFLAGS.

**Note:** This parameter only supports purchased items.

After the Planner pull-plans an order, it checks the use of supply and inventory by higher-priority orders that need items later than the current need. If any exist, the Planner switches their use so the lower-priority order uses the supply and plans a purchase order for the items the later order needs. The result is decreased planned inventory.

# Generating Purchase Order Exceptions

You can allow the Planner to allocate supply from a purchase orders to a demand even if the purchase order’s due date is later than the demand's need date. Enable this feature by enabling bit2 (Allow purchased item supply switching) and bit7 (Generate purchase order exceptions) in the ALTPLAN.SSFLAGS field. When the Planner allocates the supply, it assumes you will move in the due date for that purchase order record. An exception message will be generated in the POEXCEPT table.

# Replenishing Safety Stock

You define an item’s safety stock value in the MATL.MINCAP column. The system handles replenishment of safety stock differently based on whether the item is manufactured, purchased, or transferred, and based on whether bit2 and bit7 are enabled in ALTPLAN.SSFLAGS.

## Safety Stock for Manufactured Items

The system generates planned orders for manufactured items according to this process:

1. The system plans safety stock for each item in order by low level code (MATL.LOWLEVEL).
2. For each item, the system looks at the difference between the Safety Stock value (MATL.MINCAP) and the on-hand quantity of the item at the end of the Plan Horizon:
    - If on-hand is above safety stock, the system continues to the next item (in order by low level code and then by item ID), comparing on-hand to safety stock.
    - If on-hand is below safety stock, the system searches for the first date where inventory dips below safety stock. It creates a planned order to replenish safety stock, setting the planned order's due date to the date of that first shortage. The planned order quantity is set to the difference between the Safety Stock value and projected on-hand at the end of the Plan Horizon.

**Note:** The system generates only a single safety stock planned order for a given item (it does not generate a planned order for each instance where the on-hand level drops below safety stock).

## Avoiding Too-large, Too-early Replenishment Quantities

For manufactured items, it is possible that in some situations (when you have demands due far into the future that use up your near-term supplies), the system may generate a safety stock planned order that is due much earlier than the demands need it, leaving you with a large quantity of unnecessary supplies.

To avoid this problem, we recommend you set the Time Fence Rule to Accumulated Lead Time for the item on the Items form. By setting this Time Fence Rule, the later demands will not use the near-term supplies and the system will create planned orders due closer to when the demands are actually due. See “Defining a Time Fence for Inventory/Supply Usage” on page 33 for information about defining a time fence.

## Safety Stock for Purchased Items

Safety stock planning for purchased items depends on whether you select the Generate Purchase Order Exceptions and Purchased Supply Switching parameters on the Planning Parameters form (enable these features by enabling bit2 and bit7 in the ALTPLAN.SSFLAGS field). We recommend you always select both of these parameters; safety stock planning is not as accurate when either of these parameters is not selected.

After the system finishes planning all the demands normally, according to order priority, it performs a second, time-phased planning run. At the current date and at every date where there is a demand for the item, the system compares the on-hand quantity with the safety stock value:

1. If on-hand is not below safety stock, the system continues planning normally according to the Purchased Supply Switching/Generate Purchase Order Exceptions functionality.
2. If on-hand is below safety stock, the system searches for existing planned supplies to satisfy the demand.
3. If sufficient planned supplies are available to satisfy the entire demand quantity, the system allocates the supplies to the demand instead of allocating on-hand inventory. Safety stock is preserved without requiring a safety stock planned order.
4. If sufficient supplies are NOT available, the system generates a new planned order at the earliest possible date to cover the safety stock shortage and the demand quantity. The date at which on-hand first dipped below safety stock determines the due date and quantity of the safety stock planned order:
    - If the date of the first shortage is before the current date + standard (or expedited) lead time: the due date of the planned order is set to the current date + standard/expedited lead time. The planned order quantity is set to the difference between safety stock and the projected on-hand at the current date + lead time.
    - If the date of the first safety stock shortage is later than the current date + standard (or expedited) lead time: the due date of the planned order is set to the date of the first shortage. The planned order quantity is set to the difference between safety stock and the projected on-hand at the date of the first shortage.

**Note:** The system generates only a single safety stock planned order for a given item (it does not generate a planned order for each instance where the on-hand level drops below safety stock).

## Safety Stock for Transferred Items

If a Supply Site is defined for the transferred item, the system plans safety stock like it does for manufactured items.

If a Supply Site is NOT defined for the transferred item, the system plans safety stock like it does for purchased items.

# Cross-referencing Orders

You can cross-reference (or "peg") an order to another order. For example, you might cross-reference a demand to a supply or a demand to a replenishment demand. To represent a cross-reference relationship and to ensure that the Planner allocates the supply only to the demand to which it is pegged, you must set up the database records as shown in these example relationships:

| Relationship | Demand or BOM Item Record | Supply or Replenishment Demand Record |
|---|---|---|
| Demand xref'd to a supply | ORDER.REFORDERID should contain the ID from MATLDELV.ORDERID | Enable the XREF flag (bit19) in MATLDELV.FLAGS |
| Demand xref'd to a replenishment demand | ORDER.REFORDERID should contain the ID from ORDER.ORDERID | Enable the XREF flag (bit19) in ORDER.FLAGS |
| BOM item xref'd to a supply | PBOMMATLS.REFORDERID should contain the ID from MATLDELV.ORDERID | Enable the XREF flag (bit19) in MATLDELV.FLAGS |
| BOM item xref'd to a replenishment demand | PBOMMATLS.REFORDERID should contain the ID from ORDER.ORDERID | Enable the XREF flag (bit19) in ORDER.FLAGS |

**Note:** You cannot cross-reference a supply to multiple demands (a cross-reference is a 1-to-1 relationship between a demand and a supply).

**Note:** When the Planner plans a cross-referenced demand and supply, and there is not enough supply quantity to satisfy the demand, the system allocates as much of the supply as is available and generates an exception message to alert you of the shortage.

**Note:** If the demand's due date is earlier than the cross-referenced supply's due date, the Planner generates the exception message "Move In Receipt."

# Using Pull-Up Orders

Define an order as a pull-up order to specify that, when the order is planned, the system should attempt to plan the order line to meet its Request Date (that is, it pull plans from the Request Date). If that plan is not successful, the system attempts to meet the order line's Due Date. If it can achieve the Due Date, the system incrementally attempts to improve the plan between the Request Date and Due Date. If the system cannot plan the order line at least by the Due Date, it push plans the line item from the current date out to the as-soon-as-possible date.

You define an order as a pull-up in the ORDER.FLAGS field (enter the appropriate integer value to set bit11). This flag affects all line items on the order.  To force a particular order line to ignore this field, set the order line's Due Date equal to its Request Date.

**Note:** The Pull Up Order option does not override the normal order priority hierarchy. That is, this order line will not be pulled up at the expense of higher-priority demands.

# Defining a Time Fence for Inventory/Supply Usage

In some situations, orders due in the future may consume inventory and supply, causing shorter-term (CTP) orders to be promised later than necessary. You can specify a time fence inside which Planner should allocate inventory and supply to longer-term orders. Any orders that are due outside this time fence will not use available inventory or supply (thus leaving it available for shorter-term needs).

The quantity of inventory to reserve is defined in the ALTPLAN.TFMULT field. The time fence rule is defined in the MATL.TFRULE field. Rule values are as follows:

- 0=No time fence.
- 1=Use lead time: the length of the fence is defined by the end item’s lead time.
- 2=Use accumulated lead time: the length of the fence is defined by the accumulated lead time for the end item + the lead times for all its components.
- 3=Use specific value: the length of the fence is defined by the value in the TFVALUE field.

# Planning Passes

The planning algorithm considers the time fence as follows:

| Planning Pass | Description |
| --- | --- |
| A | Pull plan from the demand's Due date and use Time Fence. If the demand is a pull-up order, pull plan from the Request date.<br>If pass A is successful, plan the next item.<br>If pass A fails, perform pass B. |
| B | Pull plan from the demand's Due date and ignore Time Fence. If the demand is a pull-up order, pull plan from the Request date.<br>If pass B is successful, plan the next item.<br>If pass B fails and demand is a pull-up order, perform pass C.<br>If pass B fails and demand is not a pull-up order, perform pass E. |
| C | (Pull-up orders only.) Pull plan from the Due date and use Time Fence.<br>If pass C is successful, perform additional pull-planning iterations between the Due date and Request date to try to project a date that is closer to the Request date.<br>If pass C fails, perform pass D. |
| D | (Pull-up orders only.) Pull plan from the Due date and ignore Time Fence.<br>If pass D is successful, perform additional pull-planning iterations between the Due date and Request date to try to project a date that is closer to the Request date.<br>If pass D fails, perform pass E. |
| E | Push plan from the current date (Time Fence is ignored during a push).<br>If pass E is successful (that is, if the system was able to plan a projected completion date within the Plan Horizon), perform pass F.<br>If pass E fails, perform pass G. |
| F | Pull plan again from the projected date calculated by the push plan and use Time Fence.<br>If pass F is successful, perform iterative pull-planning using the projected date from the push as the end of the iterations time window.<br>If pass F fails, perform pass G. |

| Planning Pass | Description |
| --- | --- |
| G | Pull plan from the end of the Plan Horizon and ignore Time Fence.<br>If pass G is successful, perform iterative pull-planning using the end of the Plan Horizon as the end of the iterations time window.<br>If pass G fails, mark the item as "Blocked" and begin planning the next item. |
| iterative pull planning | Perform one or more pull-planning iterations within a defined time window, dividing the window in half with each iteration until a feasible plan is found that is within the ALTPLAN.ITERDAYS or ITERDAYSCTP number of days of the need date. The beginning of the time window is always the demand's need date. The end of the time window will be either the projected completion date from pass E or the end of the Plan Horizon.<br>For example, the first iteration pulls from the midpoint between the projected date and the need date. If that pull succeeds, the next iteration pulls from the midpoint between that new projected date and the need date. The process incrementally moves closer to the need date until it finds a plan that works (that is, doesn't calculate a start date in the past) and has a projected completion date that is within the specified days of the need date. |

## Enforcing the Time Fence at the Expense of the Due Date

If you are more concerned with preserving inventory and supply than with meeting the due date, you can disable pass B so the Planner always uses the Time Fence. To disable this pass, enter an integer value to enable bit8 in ALTPLAN.SSFLAGS.

## Time Fence Messages

To allow the system to log TRACE messages with time fence information, enable bit16 in the ALTPLAN.DEBUGLVL field. The system logs TRACE messages each time inventory or supply is protected by a time fence, such as:

TIMEFENCE: order  xxx

TIMEFENCE: item   yyy

TIMEFENCE: time fence  021227.080000

TIMEFENCE: quantity usable  50.0

TIMEFENCE: quantity protected 20.0

# Optimizing Usage of Manufactured Item Supplies

When a parent item is defined with an Order Maximum value, and its component items are defined with Order Minimum values, the system may create excess planned orders. For these items, you can enable Manufactured Supply Switching to allow the system to reallocate those excess supplies appropriately and not generate excess planned orders.

Because this extra processing may slow performance of the Planner regeneration, you should enable it only for items that are likely to create excess planned orders. For example, if the system produces numerous excess planned supplies that you cannot explain, the problem may be caused by this condition, and you should consider using this parameter. Otherwise, do not enable it.

This parameter works only during a plan regeneration. It applies only to manufactured items. To apply the switching, enable the parameter in the ALTPLAN.SSFLAGS field (bit3) AND on each MATL record (FLAGS field, bit10) you want to affect. Disabling the parameter in the ALTPLAN.SSFLAGS field disables it for all items.

# Using Overlapping Operations

You can configure the Planner to allow an operation to start before the previous operation has completed. The JS19VR.OLTYPE and JS19VR.OLVALUE fields define how the Planner calculates the start time of the overlapping operation.

During a planning run, the Planner calculates the time horizon in which to schedule an operation in the plan (the JOBPLAN.STARTDATE/ENDDATE fields). On a push-planning run, the Planner checks the previous operation's OLTYPE field for a value other than 0.

**Note:** On a pull-planning run, the Planner checks the current operation's overlap settings rather than the previous operation's settings.

If OLTYPE is enabled, the Planner uses the rules in the OLTYPE and OLVALUE fields to calculate the number of hours after the start of the first operation to start the second operation. This calculation is based on the OLTYPE rules shown below:

| OLTYPE | Rule |
|---|---|
| 0 | No overlap |
| 1 | Start overlap window after setup and cycle time for OLVALUE items has completed |
| 2 | Start after OLVALUE percent of setup + cycle time have completed |
| 3 | Start after OLVALUE percent of cycle time has completed |
| 4 | Start after OLVALUE hours of setup + cycle time have completed |
| 5 | Start after OLVALUE hours of cycle time have completed |

The Planner moves back the second operation's horizon to reflect this overlap. It then makes further adjustments to the horizon based on the following circumstances:

- Considers that the first operation may split across breaks for other operations (the cross-breaks rule is defined in the JS19VR.CRSBRKRL field) and makes the necessary adjustment.
- Ensures the second operation does not end before the first operation. If the difference between the second operation's new start date/time and the first operation’s end date is greater than the total duration of the second operation, the Planner adjusts the horizon again so the second operation cannot finish before the first operation.

If you run the Scheduler after running the Planner, the Scheduler performs its overlap processing based on the same OLTYPE codes. For more information, see Chapter 2, "About the Scheduler."

# Using Minimum and Maximum Lot Sizes

You can set up items to be produced in minimum and maximum lot quantities in a production run. The maximum lot size parameter is considered before the minimum and multiple lot size parameters.

## Using Maximum Order Lot Size

The MATL.ORDMAX parameter allows you to produce items in a maximum lot size. This parameter applies only to manufactured end items and does not apply to component items. When planning an end item, the Planner breaks the order quantity into multiple batches of the size specified in the ORDMAX parameter. If the ORDMAX value is less than the ORDMIN value, the Planner sets the batch size equal to ORDMIN. The last batch will contain the remainder of the original quantity.

For example, if the ORDMAX value is 1000, and the order is for 100,002 pieces of item # 502, the Planner plans the order as if it were for 99 line items of 1000 items each and one line item of 1002 items.

The Planner treats each lot as a separate demand. For each lot, at each level of the bill of material, the Planner considers planned supplies, inventory, scrap/shrinkage, and ORDMIN/ORDMULT. After planning these demands, the Planner recombines the output so it appears as a single demand again.

## Using Order Minimum/Multiple

The MATL.ORDMIN and ORDMULT parameters allow you to produce items in a minimum lot size (ORDMIN) and in a specific multiple (ORDMULT). Minimum and multiple are lot sizes for manufactured items and are the supply order sizes recommended for purchased items. A value of 0 or 1 indicates that the system does not enforce minimum or multiple.

For example, a manufactured item has an order minimum of 10 and a multiple of 5. A demand for a quantity of 12 generates a planned order. Instead of generating the planned order for a quantity of 12, the Planner plans a quantity of 15 because the multiple is 5. Other demands may then use the excess quantity of 3.

## Defining the Earliest Start Time for Planning an Order

You can define the earliest time the Planner can start planning a new order (an order with a DUEDATE of 0 and a CALCDATE date of 0). When the Planner plans the order, it does not allow any processing, such as using resource capacity, until after the specified time. The Planner will not plan on using the time between the specified earliest start time and the current system time.

There are two versions of this parameter; both are fields in the ALTPLAN table:

- NOSTIME: Specify the earliest start time as a specific time.
- NOSOFFSET: Specify the earliest start time as an offset from the current system time. For example, a value of 4 means "four hours from now."

The resulting start time/date is then reflected in the ATLPLAN.STARTDATE field for the planned order.

# Using Alternate Items

The Planner can consider all items in an alternate item group (PBOMMATLS.ALTID) and select the item that can be obtained the fastest. Because the Planner is pull planning, the fastest item is the one that can be started the latest.

Alternate items are those which have the same Alternate Group ID and Merge To point in a Planner BOM’s list of materials. On the first pull, if the Use Latest Pull flag (ALTPLAN.SSFLAGS bit4) is enabled, all alternates will be considered and the latest starting one will be used. If Use Latest Pull is disabled, each alternate will be planned in sequence. The first one that can be planned successfully (either pull planned or pushed within the plan horizon) will be chosen. If an item is available in on-hand inventory, the Planner selects that item.

# Managing Resource Availability and Interruptions

By reallocating resources, the Planner can fill in periods of downtime caused by the originally allocated resource's off-shift or busy periods. If the Reallocate field (JOBSTEP.FLAGS=4) is enabled on the operation, the system can "substitute in" a different resource from the same resource group to work the interrupted operation. The Reallocate field must also be selected on the resource group record (RGRP.REALLOCFG).

The new substitute resource will be used until it becomes unavailable, at which point the system can switch back to the original resource (if available) or select a different resource.

The system uses these criteria when choosing a substitute resource from the resource group.

- The system does not substitute a resource that is an invalid pair with any other resources allocated at this operation.
- The system substitutes the resource that has the most on-shift time or time not allocated to other jobs, that provides the best coverage of the down time caused by the interruption.
- If the operation's Break Rule is set to Shifts (JS19VR.CRSBRKRL=0), the system does not substitute a resource that is busy working on a different job directly after the off-shift period it is filling.

EXAMPLE: Suppose resource A is off-shift from 12:00-13:00, interrupting the operation. Resource B is on-shift from 12:00-20:00, but is scheduled to work on another job from 14:00-15:00. If the system inserted resource B to substitute for resource A, resource B would only work on the operation from 12:00-14:00 and then the operation would be suspended until resource B went off-shift at 20:00 (because of the operation's Break Rule being set to Shifts). The system avoids this situation by selecting a substitute resource that does not have a "busy" period directly after resource A's off-shift period.

## How the Break Rule Affects Reallocation

The operation's Break Rule (JS19VR.CRSBRKRL) affects how the Planner can reallocate resources to interrupted operations. The table below describes the Break Rule options and how they affect reallocation.

| Break Rule | Description |
|---|---|
| 0 (Cross shifts only) | The operation can only be planned within a block of time in which the only interruptions are off-shift periods. If a given block of time is interrupted by the resource working on other jobs, the operation will not be planned in that block of time. |
| 1 (Cross shifts and other orders) | The operation can be planned within a block of time that is interrupted either by off-shift periods or by other demands already planned. |
| 2 (No breaks) | There must be a large enough block of uninterrupted time on the associated resources for the operation to be completed (if the Planner cannot find an unbroken block of time, the job will be blocked). |

## Influence on Resource Allocation

The Planner’s ability to reallocate resources is closely tied to the way it chooses a resource combination to plan an operation. An operation may specify several resource groups that are required to perform the operation. Each resource group may contain multiple member resources. One or more resources may be required from each resource group that is assigned to the operation. When planning the operation, the Planner evaluates every combination (provided the

ALTPLAN.PULLITERS parameter is set to 0, as recommended) of those potential resources and chooses the combination that can plan the operation the fastest. Evaluation criteria include a resource's on-shift time and current allocation to other jobs.

The reallocation feature influences the Planner’s criteria of evaluating these resource combinations. If the resource group's Reallocate field is selected, the Planner considers the added availability of all the other candidate resources of that group.

# Chapter 3: About the Scheduler

3

The Scheduler sequences operations for a resource based on your job start and end dates, while considering sequence dependencies and resource-to-resource optimizations.

You can run the schedule over any period of time—a shift, day, week, month, etc. However, you will usually run it on a daily basis. The Scheduler projects job performance, suggests operation sequences, and reports changes in inventory levels. It also generates summary performance information you can use in reports and graphs.

A typical process of using the Scheduler consists of these steps:

1. Enter your real-world data (jobs, items, routings, operations, shifts, resources, etc.).
2. Define operation sequencing rules, thresholds, time horizons, and other options.
3. Run the Scheduler. The Scheduler performs a simulation of how your schedule would perform, given the input data and conditions and rules you define.
4. View the output and identify the problems. After the Scheduler completes a run, it populates the Scheduler output tables with projections of order performance, operation sequences at work centers, and changes in inventory levels. It also populates several tables with information you can use in reports and graphs. See Chapter 1, "SQL Server Schema," in the *APS Database Reference*, for descriptions of the Scheduler output tables.
5. Solve any scheduling problems. For example, add overtime for the machine or reprioritize the jobs.
6. Rerun the Scheduler.
7. Print a dispatch list and give it to your shop floor.

This chapter describes the database input components you can use to create a model of your manufacturing operation, from which you can generate a sequenced schedule.

## How Event-based Scheduling Works

The Scheduler creates a simulation of what will happen given all the conditions and data you provide to it. When running this simulation, the Scheduler views all events that happen at a given time. Then it moves forward to the next point in time where any events occur.

After the scheduling run is complete, the Scheduler generates output data (including a set of statistics) that summarize how well your simulated schedule performed. For example, if an operation was idle for an hour, the summary information helps you determine why (for example, because the required resource was busy). You can then adjust your data to make the schedule more efficient, such as adjusting the release date of an earlier job to free the resource faster. When the schedule results are acceptable, you can print a dispatch list to give to your shop floor personnel to execute the schedule.

## About the Scheduler Trace Log

The TRACELOG table records all events the Scheduler processes (when the ALTSCHED.TRACELVL field is set to 2, the "Extensive" detail level). In your day-to-day Scheduler activities, this log is useful only for troubleshooting. However, it serves as a good step-by-step illustration of how the Scheduler addresses each event in a schedule. Therefore, the events for the examples in this document are illustrated in the Trace log format.

**Note:** Because the Scheduler’s core program is used by other Infor simulation products, some of the Trace text is not relevant to the Scheduler. This irrelevant text has been omitted in the examples in this document.

## Event Processing Example

In this simple example, observe the Scheduler’s view of the events.  Assume these conditions:

- Item BK-110 has a routing with one operation, Oper110.
- The operation Oper110 requires a resource from resource group ST1.
- ST1 contains one resource, Crew1. The resource is on-shift for the time illustrated in the example.
- Operation Oper110 has a run time of 15 minutes. Run time (also called "step time") is the time to complete a load on the operation. Depending on the operation’s Step Time Rule (JOBSTEP.STEPEXPRL), this could represent various times, including a fixed time or the item cycle time (that is, time per piece). In this example, the run time represents the item cycle time.
- Job110-000 for BK-110 is released on 2/27/2002 at 8:00 a.m. for a quantity of 3.
- The schedule starts on 2/27/2002 at 8:00 a.m.

The Scheduler processes this data in these events:

| Event | Description |
|---|---|
| SCHEDULE TIME ADVANCE TO 02/27/2002 8:00:00 | The Scheduler moves to the time the first events occur (the schedule itself may have been set to start earlier). |
| 02/27/2002 8:00:00<br>START OF ORDER ARRIVAL EVENT FOR ORDER Job110-000<br>RELEASING LOADS FOR ORDER Job110-000<br>CREATING LOAD 1, SIZE 3<br>OPERATION Oper110, SELECTED AS FIRST | The job is released and the Scheduler places the job quantity into a unit called a load. In most situations, the load contains the entire job quantity (although the job quantity can be broken into several loads). The first job operation in the routing is identified. |
| 02/27/2002 8:00:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110, LOAD 1 OF ORDER Job110-000<br>SCHEDULING END OF MOVE FOR OPERATION AT<br>02/27/2002 8:00:00 | If any Move time had been defined for this operation, the Scheduler would have calculated its duration here. In this example, the operation is not defined with any Move time. |
| 02/27/2002 8:00:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110, LOAD 1 OF ORDER Job110-000<br>REQUESTING RESOURCE GROUP ST1<br>REQUESTING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION NOT COMPLETE | The Scheduler forces the first operation event to delay allocating the resource until the resource can check its request queue for any pending requests. A request is entered into the resource’s request queue. (Request queues are described in detail later.)<br>This delay covers the possibility that a higher-priority operation arrives at the same time as a lower-priority operation, allowing the more critical operation to allocate the resource. |

| Event | Description |
| --- | --- |
| 02/27/2002 8:00:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>PROCESSING OPERATION Oper110, LOAD 1 OF ORDER Job110-000<br>ALLOCATING 1 UNITS FROM RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION COMPLETE<br>SETUP NOT NECESSARY OR OF ZERO DURATION<br>SCHEDULING END OF SERVICE FOR OPERATION AT<br>02/27/2002 8:45:00 | The Crew1 resource checks its request queue for any pending requests and finds the request from OperBK001's load.<br>The Scheduler allocates the Crew1 resource from the ST1 resource group and calculates the time when the operation will be complete. |
| SCHEDULE TIME ADVANCE TO 02/27/2002 8:45:00 | The Scheduler moves to the time when the next events occur. |
| 02/27/2002 8:45:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110, LOAD 1 OF ORDER Job110-000<br>FREEING 1 UNITS OF RESOURCE GROUP ST1<br>FREEING RESOURCE Crew1 FROM GROUP ST1<br>RESOURCE Crew1 IS NOW IDLE<br>SCHEDULING END OF COOL FOR OPERATION AT<br>02/27/2002 8:45:00 | The operation completes and releases ("frees") the Crew1 resource.<br>The Crew1 resource is now idle and available for any other operations to allocate it.<br>If any Finish time (that is, cooling time) had been defined for this operation, the Scheduler would have calculated its duration here. In this example, there is no additional post-operation processing. |
| 02/27/2002 8:45:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>REQUEST QUEUE EMPTY | When a resource is freed, it checks whether any other operations have requested it while it was busy. In this example, there are no pending requests for the resource. |

| Event | Description |
|---|---|
| 02/27/2002 8:45:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110, LOAD 1 OF ORDER Job110-000<br>END OF PROCESS FOUND SELECTING NEXT OPERATION<br>END OF PROCESS FOR LOAD 1 OF ORDER Job110-000<br>0 LOADS OUTSTANDING<br>ORDER Job110-000 COMPLETE | Each operation specifies the next operation in the routing. In this example, the Scheduler finds no next operation, so it completes the job. |
| SCHEDULE HALTED BECAUSE THERE WERE NO MORE ORDERS TO PROCESS<br>SCHEDULE END AT 02/27/2002 8:45:00 | The Schedule is now complete. |

# Using Scheduler Rules

The Scheduler uses these types of rules to allocate resources for working on an operation:

- Order Release Rule (ALTSCHED.ORDRELRL): Controls how the Scheduler processes orders when two or more orders are released at the same date and time. By default, orders with the lowest priority number are processed first.
- Sequencing Rule (RESRC.SEQRL): Sequences the waiting requests in a resource’s request queue. For example, "loads for operations with the earliest due dates are listed first." By default, the rule is First In First Out (FIFO).
- Selection Rule (RESRC.SELRL): Controls how the resource selects the next waiting request from the queue. For example, "select the load with the least setup time based on the first downstream operation." By default, no selection rule is defined and requests are selected in the sequence they appear in the request queue.
- Resource Group Member Selection Rule (RGRP.ALLOCRL): Controls how the Scheduler selects a resource from a resource group to allocate to a load. For example, "select the resource that has been idle the longest." By default, the resource listed first in the resource group record is selected first.

**Note:** When a load arrives at an operation, the Scheduler requests all required resources in the required resource group, forcing each to check its request queue only after all loads scheduled at the same time have arrived. This prevents a lower-priority load from allocating a resource because it arrives when the resource is available, even though a load with higher priority is scheduled to arrive at the same time.

The Scheduler rule types are described in more detail in the next sections.

# Controlling Simultaneous Order Start Dates

In the previous examples, the orders have different start dates/times. In reality, multiple orders may have the same start date and time. The Order Release Rule controls the sequence that the Scheduler releases these simultaneous demands.

## Order Release Rules

| Code | Rule | Description |
| --- | --- | --- |
| 0 | High Priority | Orders with the highest priority value. |
| 1 | Low Priority (default) | Orders with the lowest priority value. |
| 2 | Earliest Due Date | Orders with the earliest due date. |
| 3 | Fewest Operations | Orders with the fewest operations. |
| 4 | Least Processing Time | Orders with the smallest total of setup, run, and move time. |
| 5 | Least Slack | Orders with the smallest remaining time until the due date. |
| 6 | Least Avg. Slack Per Operations | Orders with the least average slack per operations (remaining time until the due date / number of operations). |
| 7 | Least Avg. Slack Per Processing Time | Orders with the least average slack per processing time (remaining time until the due date / processing time). |
| 8 | Smallest Quantity | Orders with the smallest quantities. |
| 9 | Largest Quantity | Orders with the largest quantities. |

# Using Sequencing and Selection Rules

## Resource Request Queues

Loads of items compete for the use of resources. Each resource has its own request queue to hold outstanding requests from loads. During operation processing, if the Scheduler cannot allocate a member of a required resource group to a load, it enters a request in the request queue of each resource group member. When a resource in that group is free, the Scheduler allocates it to the waiting load and removes that load’s requests from the other member resources.

If the load requires multiple resources from the group, the Scheduler does not allocate any resources to the load until the required number of resources from that group is available.

If the load requires resources from multiple resource groups, the Schedule does not allocate any resources to the load until the required number of resources from each group is available.

## Sequencing the Queue

If multiple requests are in a resource's request queue, the Scheduler sequences the requests based on the *sequencing rule* you specify for the resource.

By default, a particular resource's sequencing rule is not defined and the Scheduler uses the "global" sequencing rule, which applies to all resources. The default global sequencing rule is First In First Out, and therefore places the first request to arrive at the resource higher in the queue.

For example, using the default global sequencing rule and assuming load XYZ arrives first, BCD second, and EFG last, the three loads would be sequenced in a request queue as follows:

| | |
|---|---|
| **1** | LOAD XYZ |
| **2** | LOAD BCD |
| **3** | LOAD EFG |

Sequencing rules defined for individual resources override the global rule.

## Selecting Loads from the Queue

When a resource becomes available, the Scheduler uses the *selection rule* to determine which requests to examine next from the request queue. By default, no selection rule is defined. That is, the first request in the request queue is selected when the resource becomes available.

| | |
|---|---|
| **1** | LOAD XYZ |
| **2** | LOAD BCD |
| **3** | LOAD EFG |

In most situations, you can use the default sequencing and selection rules. When tailoring your production model further, consider keeping the configuration simple by leaving one rule at the default value and defining your particular production situation using the other rule. For example, use the default FIFO sequencing rule and define selection rules as needed.

## Sequencing and Selection Rules Example

In this example, assume these conditions:

- Item BK-110 has a routing with one operation, Oper110.
- Operation Oper110 requires a resource from resource group ST1.
- Item CD-210 has a routing with one operation, Oper210.
- Operation Oper210 requires a resource from resource group ST1.
- Item EF-310 has a routing with one operation, Oper310.

- Operation Oper310 requires a resource from ST1.
- ST1 contains one resource, Crew1. The resource is on-shift for the time illustrated in the example.
- Crew1's sequencing rule is "Smallest Load Size." Requests for this resource that have the smallest quantity will be moved in the queue ahead of requests with larger quantities. The selection rule is not defined.
- Each operation has a run time of 6 minutes.
- The Scheduler Rule on each operation is Per Item.
- Job110-000 for BK-110 is released on 3/01/2002 at 9:00 a.m. for a quantity of 3.
- Job210-000 for CD-210 is released at 9:06 a.m. for a quantity of 10.
- Job310-000 for EF-310 is released at 9:08 a.m. for a quantity of 5.
- The schedule starts on 3/01/2002 at 8:00 a.m.

The events for this example would be represented in the Trace report as shown below. Note that operation Oper210's request for resource Crew1 will appear in the request queue before Oper310's request.

| Event | Description |
|---|---|
| SCHEDULE TIME ADVANCE TO 03/01/2002 9:00:00 | The Scheduler moves to the time the first events occur (the schedule itself may have been set to start earlier). |
| 03/01/2002 9:00:00<br>START OF ORDER ARRIVAL EVENT FOR ORDER Job110-000<br>RELEASING LOADS FOR ORDER Job110-000<br>CREATING LOAD 1, SIZE 3<br>OPERATION Oper110 SELECTED AS FIRST | Job110-000 is released and the Scheduler places the job quantity into a load. The first job operation in the routing is identified. |
| 03/01/2002 9:00:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>SCHEDULING END OF MOVE FOR OPERATION AT<br>03/01/2002 9:00:00 | |
| 03/01/2002 9:00:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>REQUESTING RESOURCE GROUP ST1<br>REQUESTING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION NOT COMPLETE | The Scheduler applies the resource allocation rule and enters a request for Oper110's load into Crew1's request queue. |

| Event | Description |
|---|---|
| 03/01/2002 9:00:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>ALLOCATING 1 UNITS FROM RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION COMPLETE<br>SETUP NOT NECESSARY OR OF ZERO DURATION<br>SCHEDULING END OF SERVICE FOR OPERATION AT<br>03/01/2002 9:18:00 | The Crew1 resource checks its request queue for any pending requests and finds the request from Oper110's load.<br>The Scheduler allocates the Crew1 resource from the ST1 resource group and calculates the time when the operation will be complete. |
| SCHEDULE TIME ADVANCE TO 03/01/2002 9:06:00 | The Scheduler moves to the time when the next events occur. |
| 03/01/2002 9:06:00<br>START OF ORDER ARRIVAL EVENT FOR ORDER Job210-000<br>RELEASING LOADS FOR ORDER Job210-000<br>CREATING LOAD 1, SIZE 10<br>OPERATION Oper210 SELECTED AS FIRST | Job210-000 is released and the Scheduler places the job quantity into a load. The first job operation in the routing is identified. |
| 03/01/2002 9:06:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>SCHEDULING END OF MOVE FOR OPERATION AT<br>03/01/2002 9:06:00 | |

| Event | Description |
|---|---|
| 03/01/2002 9:06:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>REQUESTING RESOURCE GROUP ST1<br>REQUESTING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION NOT COMPLETE | The Scheduler applies the resource allocation rule and enters a request for Oper210's load into Crew1's request queue. |
| 03/01/2002 9:06:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>RESOURCE NO LONGER FREE | The Crew1 resource is still busy with Oper110's load, so cannot be allocated to Oper210 yet. |
| SCHEDULE TIME ADVANCE TO 03/01/2002 9:08:00 | The Scheduler moves to the time when the next events occur. |
| 03/01/2002 9:08:00<br>START OF ORDER ARRIVAL EVENT FOR ORDER Job310-000<br>RELEASING LOADS FOR ORDER Job310-000<br>CREATING LOAD 1, SIZE 10<br>OPERATION Oper310 SELECTED AS FIRST | Job310-000 is released and the Scheduler places the job quantity into a load. The first job operation in the routing is identified. |
| 03/01/2002 9:08:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper310,<br>LOAD 1 OF ORDER Job310-000<br>SCHEDULING END OF MOVE FOR OPERATION AT<br>03/01/2002 9:08:00 | |
| 03/01/2002 9:08:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper310,<br>LOAD 1 OF ORDER Job310-000<br>REQUESTING RESOURCE GROUP ST1<br>REQUESTING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION NOT COMPLETE | The Scheduler applies the resource allocation rule and enters a request for Oper310's load into Crew1's request queue.<br>The resource's sequencing rule is "Smallest Load Size," and Oper310 has a smaller quantity that Oper210. Therefore, although this request arrived later than Oper210, the request for Oper310 is placed ahead of Oper210 in the queue.<br>NOTE: If the default FIFO sequencing rule had been defined, Oper210 would have remained first in the queue. |

| Event | Description |
|---|---|
| 03/01/2002 9:08:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>RESOURCE NO LONGER FREE | The Crew1 resource is still busy with Oper110, so cannot be allocated to Oper210 or Oper310 yet. |
| SCHEDULE TIME ADVANCE TO 03/01/2002 9:18:00 | The Scheduler moves to the time when the next events occur. |
| 03/01/2002 9:18:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>FREEING 1 UNITS OF RESOURCE GROUP ST1<br>FREEING RESOURCE Crew1 FROM GROUP ST1<br>RESOURCE Crew1 IS NOW IDLE<br>SCHEDULING END OF COOL FOR OPERATION AT 03/01/2002 9:18:00 | Oper110 completes and frees the Crew1 resource.<br>The Crew1 resource is now idle and available for other operations to allocate it. |
| 03/01/2002 9:18:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>PROCESSING OPERATION Oper310,<br>LOAD 1 OF ORDER Job310-000<br>ALLOCATING 1 UNITS FROM RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION COMPLETE<br>SETUP NOT NECESSARY OR OF ZERO DURATION<br>SCHEDULING END OF SERVICE FOR OPERATION AT 03/01/2002 9:48:00 | When Oper110 frees resource Crew1, a Resource Free Check event is triggered.<br>The Scheduler checks Crew1's request queue for any pending requests. It finds the request from the loads for Oper210 and Oper310. Crew1's selection rule is the default "no selection rule," so the Scheduler selects the first request listed at the top of the queue. Oper310 is listed first in the queue, so the Scheduler allocates the resource to Oper310's load.<br>Oper310's request is removed from the resource's request queue.<br>The Scheduler calculates the time when the operation will be complete. |

| Event | Description |
|---|---|
| 03/01/2002 9:18:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>END OF PROCESS FOUND SELECTING NEXT OPERATION<br>END OF PROCESS FOR LOAD 1 OF ORDER Job110-000<br>0 LOADS OUTSTANDING<br>ORDER Job110-000 COMPLETE | The Scheduler finds no next operation specified for Oper110, so it completes Job110-000. |
| SCHEDULE TIME ADVANCE TO 03/01/2002 9:48:00 | The Scheduler moves to the time when the next events occur. |
| 03/01/2002 9:48:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper310,<br>LOAD 1 OF ORDER Job310-000<br>FREEING 1 UNITS OF RESOURCE GROUP ST1<br>FREEING RESOURCE Crew1 FROM GROUP ST1<br>RESOURCE Crew1 IS NOW IDLE<br>SCHEDULING END OF COOL FOR OPERATION AT<br>03/01/2002 9:48:00 | Oper310 completes and frees the Crew1 resource.<br>The Crew1 resource is now idle and available for other loads to allocate it. |

| Event | Description |
|---|---|
| 03/01/2002 9:48:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>ALLOCATING 1 UNITS FROM RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION COMPLETE<br>SETUP NOT NECESSARY OR OF ZERO DURATION<br>SCHEDULING END OF SERVICE FOR OPERATION AT<br>03/01/2002 10:48:00 | When Oper310 frees resource Crew1, a Resource Free Check event is triggered.<br>The Scheduler checks Crew1's request queue for any pending requests. It finds the request from Oper210's load and allocates the resource.<br>Oper210's request is removed from the resource's request queue.<br>The Scheduler calculates the time when the operation will be complete. |
| 03/01/2002 9:48:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper310,<br>LOAD 1 OF ORDER Job310-000<br>END OF PROCESS FOUND SELECTING NEXT OPERATION<br>END OF PROCESS FOR LOAD 1 OF ORDER Job310-000<br>0 LOADS OUTSTANDING<br>ORDER Job310-000 COMPLETE | The Scheduler finds no next operation specified for Oper310, so it completes Job310-000. |
| SCHEDULE TIME ADVANCE TO 03/01/2002 10:48:00 | The Scheduler moves to the time when the next events occur. |

| Event | Description |
|---|---|
| 03/01/2002 10:48:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>FREEING 1 UNITS OF RESOURCE GROUP ST1<br>FREEING RESOURCE Crew1 FROM GROUP ST1<br>RESOURCE Crew1 IS NOW IDLE<br>SCHEDULING END OF COOL FOR OPERATION AT<br>03/01/2002 10:48:00 | Oper210 completes and frees the Crew1 resource.<br>The Crew1 resource is now idle and available for other loads to allocate it. |
| 03/01/2002 10:48:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>REQUEST QUEUE EMPTY | There are no more requests in Crew1's request queue. |
| 03/01/2002 10:48:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>END OF PROCESS FOUND SELECTING NEXT OPERATION<br>END OF PROCESS FOR LOAD 1 OF ORDER Job210-000<br>0 LOADS OUTSTANDING<br>ORDER Job210-000 COMPLETE | The Scheduler finds no next operation specified for Oper210, so it completes Job210-000. |
| SCHEDULE HALTED BECAUSE THERE WERE NO MORE ORDERS TO PROCESS<br>SCHEDULE END AT 03/01/2002 10:48:00 | The Schedule is now complete. |

## Sequencing Rules

You can define the sequencing rule for an individual resource in the RESRC.SEQRL field or you can define global sequencing rules in the ALTSCHED.GLBSEQRL field. Global rules apply to any resource that does not have a sequencing rule.

| Code | Rule | Description |
|---|---|---|
| 0 | Global | Default to the Global Sequencing Rule |
| 1 | FIFO | Loads that arrive first. |

| Code | Rule | Description |
| --- | --- | --- |
| 2 | LIFO | Loads that arrive last. |
| 3 | High Priority | Loads with the higher priority. Load priority is determined by the SQL field LSTATUS.PRIORITY. |
| 4 | Low Priority | Loads with the lower priority. |
| 5 | Earliest Due Date | Orders with the earliest due date. |
| 6 | Earliest Release Date | Orders with the earliest start date. |
| 7 | Short Current Operation | Loads with the shortest time for the current operation. |
| 8 | Long Current Operation | Loads with the longest time for the current operation. |
| 9 | Long Any Operation | Loads with the longest time for any subsequent operation. |
| 10 | Least Number of Operations | Loads with the least number of remaining operations. |
| 11 | Least Process Time | Loads with the least estimated remaining processing time. |
| 12 | Least Static Slack | Loads with the least remaining time to due date. |
| 13 | Least Average Static Slack Per Remaining Operations | Loads with the least average time to due date per remaining operations (static slack / remaining no. of operations). |
| 14 | Least Average Static Slack Per Remaining Processing Time | Loads with the least average time to due date per remaining processing time (static slack / remaining process time). |
| 15 | Least Dynamic Slack | Loads with the least remaining time to due date minus the remaining processing time. |
| 16 | Least Average Dynamic Slack Per Remaining Operations | Dynamic slack / number of remaining operations. |

| Code | Rule | Description |
|---|---|---|
| 17 | Least Average Dynamic Slack Per Remaining Processing Time | Dynamic slack / remaining processing time. |
| 18 | Adjusted Dynamic Slack | Adjust dynamic slack to more heavily weight the loads in danger of becoming late (if dynamic slack >= 0, then divide by remaining processing time; else multiply by remaining processing time). |
| 19 | Small Load | Smallest load size. |
| 20 | Large Load | Largest load size. |
| 21 | High value of a specified attribute | |
| 22 | Low value of a specified attribute | |
| 23 | Due date minus the remaining processing time | |
| 24-39 | User Defined | |

## Selection Rules

You can define the selection rule for an individual resource in the RESRC.SELRL field.

| Code | Rule | Description |
|---|---|---|
| 0 | Sequencing Rule | No selection rule; use the Sequence Rule. This is the default selection rule. |
| 1 | Dynamic Sequencing Rule | Use the Sequence Rule dynamically. The request queue is resorted based on the sequencing rule every time the resource removes a request from the queue. This rule works only with the sequencing rules that are based on dynamic and static slack. |
| 2 | Minimum Setup Time | Select the request for the load that minimizes the setup time for the next operation in the routing. |

| Code | Rule | Description |
|---|---|---|
| 3 | Reserve for Order | Consider requests for loads from the same job the machine is currently set up for, if any such loads exist. If none exist, the Sequence Rule is applied dynamically and the entire list is considered. |
| 4 | First Load | Consider only the first load ranked by the Sequence Rule. If the first load cannot allocate the resource, do not try the next load in the queue. |
| 5 | Dynamic First Load | Consider only the first load by applying the Sequence Rule dynamically. If the first load cannot allocate the resource, do not try the next load in the queue. This rule works only with the sequencing rules that are based on dynamic and static slack. |
| 6 | Minimum Setup First Load | Consider only the first load by applying minimum setup. If the first load cannot allocate the resource, do not try the next load in the queue. |
| 7 | First n Loads | Consider only the first n loads, where n is specified in Selection Value. |
| 8 | Dynamic First n Loads | Consider only the first n loads by applying the Sequence Rule dynamically, where n is specified in Selection Value. |
| 9 | Minimum Setup First n Loads | Consider only the first n loads by applying minimum setup, where n is specified in Selection Value. |
| 10 | Threshold | Apply the Sequence Rule dynamically, and only consider requests with values less than or equal to Selection Value. Note: If the Sequence Rule for the request queue is a “high to low” value type rule (for example, Highest Attribute Value, High Priority, etc.), consider only requests with values greater than Selection Value. |
| 11 | Priority/ Slack/ Minimum Setup | Select the request with the highest load priority. If tied, select the request with the smallest critical dynamic slack. Dynamic slack is critical if it is less than the threshold specified for the resource in Selection Value. If tied, select the request with minimum downstream setup. |
| 12 | Change-over Volume Loads/ Minimum Setup | After allocating n loads under the same setup, select the non-zero setup time request. If there are no non-zero setup time requests, zero setup time requests will be selected until the next non-zero setup time load arrives with minimum downstream setup, where n is specified in Selection Value. |

| Code | Rule | Description |
|---|---|---|
| 13 | Change-over Volume Loads/ Longest Waiting | After allocating n loads under the same setup, select the non-zero setup time request with the longest waiting time, where n is specified in Selection Value. |
| 14 | Change-over Volume Items/ Minimum Setup | After processing n items under the same setup, select the non-zero setup time request with minimum downstream setup, where n is specified in Selection Value. If there are no non-zero setup time requests, zero setup time requests will be selected until the next non-zero setup time load arrives. |
| 15 | Change-over Volume Items/ Longest Waiting | After processing n items under the same setup, select the non-zero setup time request with the longest waiting time, where n is specified in Selection Value. |
| 16 | Change-over Processing Time/ Minimum Setup | After spending n hours in processing loads under the same setup, where n is specified in Selection Value, select the non-zero setup time request with minimum downstream setup time. Note: Processing time does not include the setup time incurred for the first load. |
| 17 | Change-over Processing Time/Longest Waiting | After spending n hours in processing loads under the same setup, where n is specified in Selection Value, select the non-zero setup time request that has been waiting the longest. Note: Processing time does not include the setup time incurred for the first load. |
| 18 | Change-over Elapsed Time/ Minimum Setup | After n hours have elapsed under the same setup, where n is specified in Selection Value, select the non-zero setup time request with minimum downstream setup time. Note: Processing time does not include the setup time incurred for the first load. |
| 19 | Change-over Elapsed Time/Longest Waiting | After n hours have elapsed under the same setup, where n is specified in Selection Value, select the non-zero setup time request that has been waiting the longest. Note: Processing time does not include the setup time incurred for the first load. |
| 20 | Interrupted/ Dynamic Sequencing | Use the sequencing rule dynamically, but put requests for interrupted loads at the front of the request queue. Furthermore, if an interrupted load was being processed by the “selecting” resource, its request is placed at the front of the queue. |

| Code | Rule | Description |
|---|---|---|
| 21 | Tiered Selection | Select based on ranking rules for each of three levels, or tiers. Jobs is sequenced based on the first tier rule, then ties at the first tier are broken by the second tier rule, then ties at the second tier are broken by the third tier rule, and finally ties at the third tier are broken by the sequencing rule. |
| 22-39 | User Defined | |

## Improving Speed of Selection Processing

The selection rules First Load (code 4), First n Loads (code 7), Minimum Setup First Load (code 6), Minimum Setup First n Loads (code 9), or Threshold (code 10) will consider only a portion of the possible load requests in the request queue.  You can use these rules to speed up selection processing on resource (also add-to-material and remove-from-material as applicable) request queues containing a large number of load requests.  However, be aware that potential side effects exist when using these rules.

For example, situations can occur where the resource becomes free, sorts the request queue using the selection rule, fails to make an allocation, goes idle, and remains idle until one of the following occurs:

- A load processes an operation where it needs the resource.
- The resource goes off-shift and then back on-shift.

The resource remains idle for the rest of the Scheduler run because another load never needs the resource and the resource is not on a shift.  This leads to a related issue.  Each time you use the selection rules First Load and First n Loads, load requests at the top of the request queue may “block” load requests further down in the queue.  For example, this will occur if:

- First Load is the selection rule for resource RES1.
- The first load in the RES1 request queue requires resources RES1 and RES2.
- RES1 is free, but RES2 is not.

Resource RES1 will remain idle until the system can either allocate RES2 to the waiting load, or another load processes an operation where it needs RES1, or RES1 goes off shift and then back on shift.

## Using Values and Attributes

You can use values (RESRC.SELVALUE) with selection rules, and you can use attributes (RESRC.SEQATRID) with sequencing rules.  Realize that the Scheduler will use the default value (0.0) or the default attribute (no attribute) if you specify a rule requiring a value or an attribute.  For rules using values, the Scheduler accepts 0.0, but for rules using attributes, you must specify a valid load attribute name.

# Using Resource Group Member Selection Rules

A resource group contains one or more resources that perform the same type of work. For example, the "ChairNailing" resource group might contain three crew resources that are people who know how to nail the legs on a chair.

When defining an operation, you can specify that the Scheduler should allocate one or more members of a resource group to the load being processed at the operation. When you define a resource group, you specify a member selection rule that guides the allocation of members to a load being processed at an operation.

**Note:** The Scheduler allows allocation of resources from a single resource group to the same load on different operations. Resources become free in the order they were initially allocated.

## Using the Default Select in Sequence Rule

When the load of items arrives at the operation, the load must allocate one of the resources from the resource group required at that operation. By default, the Scheduler selects the first resource listed in the group definition. If another operation needs a resource from the same group and the first resource is busy, the Scheduler selects the next resource in the list, and so on. This default rule is called Select in Sequence.

If you use Select in Sequence for resource groups, you should define the resource group members in order of preferred usage. The Select in Sequence rule implies select member 1, then 2, and so on, with the last member being least preferred.

## Using Complex Rules

Although Select in Sequence may suit your needs, you can define more complex member selection rules to model a variety of situations. For example, you might want to distribute work across the resources in a group (with the default rule, the first resource listed may be most frequently selected). Use the **Longest Idle** rule to select the resource that has had the longest continuous idle period prior to the current request. Or use the **Cyclic** rule to select each resource in turn cyclically, starting with the next resource listed after the previously selected one.

The Index rule allows you to model parallel lines or processing cells. If you have multiple processing lines or cells with the same set of resources, and you want to stay within the same line/cell after you enter it, do the following:

1. Define resource groups for each resource type within the lines/cells. For example, you can model a set of four processing cells, each with a drill, mill, and lathe, by using resource groups as follows:
2. Define three groups (DRILLS, MILLS, and LATHES), each with a capacity of four units.
3. Make sure the resource order in the groups reflects the cell order. For example, make DRILL1, MILL1, and LATHE1 the first entry in the resource groups DRILLS, MILLS, and LATHES, respectively.

4 Define an attribute in the RESATTR table and initialize its ATTVALUE to zero. Reference that attribute’s name in the ALLOCATR field of all resource groups (in this example DRILLS, MILLS and LATHES). The Scheduler updates the attribute automatically based on the index of first member allocated.

## Member Selection Example

In this example, assume these conditions:

- Item BK-110 has a routing with one operation, Oper110.
- Item CD-210 has a routing with one operation, Oper210.
- The operations for both items need resource group ST1.
- ST1 contains 2 resources: Crew1 and Crew2. These resources are both on-shift during the times illustrated in this example.
- ST1's allocation rule is the "Longest Idle" rule. That is, allocate the resource that has had the longest continuous idle period prior to the request.
- Operation Oper110 has a run time of 15 minutes and its Step Time Rule is Item Cycle Time (JOBSTEP.STEPEXPRL=1).
- Operation Oper210 has a run time of 12 minutes and its Step Time Rule is Item Cycle Time.
- Job110-000 for BK-110 is released at 9:00 a.m. for a quantity of 5.
- Job210-000 for CD-210 is released at 9:05 a.m. for a quantity of 5.

The events for this example would be represented in the Trace report as shown below.

| Event | Description |
|---|---|
| SCHEDULE TIME ADVANCE TO 03/01/2002 9:00:00 | The Scheduler moves to the time the first events occur (the schedule itself may have been set to start earlier). |
| 03/01/2002 9:00:00<br>START OF ORDER ARRIVAL EVENT FOR ORDER Job110-000<br>RELEASING LOADS FOR ORDER Job110-000<br>CREATING LOAD 1, SIZE 5<br>OPERATION Oper110 SELECTED AS FIRST | Job110-000 is released and the Scheduler places the job quantity into a load. The first job operation in the routing is identified. |
| 03/01/2002 9:00:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>SCHEDULING END OF MOVE FOR OPERATION AT<br>03/01/2002 9:00:00 | |

| Event | Description |
| --- | --- |
| 03/01/2002 9:00:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>REQUESTING RESOURCE GROUP ST1<br>REQUESTING RESOURCE Crew1 FROM GROUP ST1<br>REQUESTING RESOURCE Crew2 FROM GROUP ST1<br>ALLOCATION NOT COMPLETE | The Scheduler applies the resource allocation rule and enters a request for Oper110 into Crew1's request queue. |
| 03/01/2002 9:00:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>ALLOCATING 1 UNITS FROM RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew1 FROM GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew2 FROM GROUP ST1<br>ALLOCATING RESOURCE Crew1 FROM GROUP ST1<br>ALLOCATION COMPLETE<br>SETUP NOT NECESSARY OR OF ZERO DURATION<br>SCHEDULING END OF SERVICE FOR OPERATION AT 03/01/2002 10:15:00 | The Crew1 resource checks its request queue for any pending requests and finds the request from Oper110.<br>The Scheduler allocates the Crew1 resource from the ST1 resource group and calculates the time when the operation will be complete. |
| 03/01/2002 9:00:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew2<br>REQUEST QUEUE EMPTY | The Crew2 resource checks for any pending requests; there are none. |
| SCHEDULE TIME ADVANCE TO 03/01/2002 10:15:00 | The Scheduler moves to the time when the next events occur. |

| Event | Description |
|---|---|
| 03/01/2002 10:15:00<br>START OF ORDER ARRIVAL EVENT FOR ORDER Job210-000<br>RELEASING LOADS FOR ORDER Job210-000<br>CREATING LOAD 1, SIZE 5<br>OPERATION Oper210 SELECTED AS FIRST | Job210-000 is released and the Scheduler places the job quantity into a load. The first job operation in the routing is identified. |
| 03/01/2002 10:15:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>SCHEDULING END OF MOVE FOR OPERATION AT 03/01/2002 10:15:00 | |
| 03/01/2002 10:15:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>FREEING 1 UNITS OF RESOURCE GROUP ST1<br>FREEING RESOURCE Crew1 FROM GROUP ST1<br>Resource Crew1 IS NOW IDLE<br>SCHEDULING END OF COOL FOR OPERATION AT 03/01/2002 10:15:00 | Oper110 completes and frees the Crew1 resource.<br>The Crew1 resource is now idle and available for other operations to allocate it. |
| 03/01/2002 10:15:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>REQUEST QUEUE EMPTY | The Crew1 resource checks for any pending requests; there are none. |
| 03/01/2002 10:15:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>REQUESTING RESOURCE GROUP ST1<br>REQUESTING RESOURCE Crew1 FROM GROUP ST1<br>REQUESTING RESOURCE Crew2 FROM GROUP ST1<br>ALLOCATION NOT COMPLETE | The Scheduler applies the resource allocation rule and enters a request for Oper210 into Crew1's request queue and Crew2's request queue. |

| Event | Description |
|---|---|
| 03/01/2002 10:15:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper110,<br>LOAD 1 OF ORDER Job110-000<br>END OF PROCESS FOUND SELECTING NEXT OPERATION<br>END OF PROCESS FOR LOAD 1 OF ORDER Job110-000<br>0 LOADS OUTSTANDING<br>ORDER Job110-000 COMPLETE | The Scheduler finds no next operation specified for Oper110, so it completes Job110-000. |
| 03/01/2002 10:15:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew1<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>ALLOCATING 1 UNITS FROM RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew1 FROM GROUP ST1<br>CANCELING REQUEST FOR RESOURCE Crew2 FROM GROUP ST1<br>ALLOCATING RESOURCE Crew2 FROM GROUP ST1<br>ALLOCATION COMPLETE<br>SETUP NOT NECESSARY OR OF ZERO DURATION<br>SCHEDULING END OF SERVICE FOR OPERATION AT<br>03/01/2002 11:15:00 | The Scheduler checks for pending requests in the request queues of both resources in the resource group. Both resources have pending requests and both resources are free. The member selection rule is Longest Idle. Therefore, because Crew2 has been idle up to this point, the Scheduler allocates Crew2 to the waiting Oper210 load.<br>NOTE: If the member selection rule had been the default "Select in Sequence" rule, Crew1 would have been allocated because it appears first in the resource group record.<br>Oper210’s request is removed from both resources’ request queues.<br>The Scheduler calculates the time when Oper210 will be complete. |
| 03/01/2002 10:15:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew2<br>REQUEST QUEUE EMPTY | |
| SCHEDULE TIME ADVANCE TO 03/01/2002 11:15:00 | The Scheduler moves to the time when the next events occur. |

| Event | Description |
| --- | --- |
| 03/01/2002 11:15:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>FREEING 1 UNITS OF RESOURCE GROUP ST1<br>FREEING RESOURCE Crew2 FROM GROUP ST1<br>RESOURCE Crew2 IS NOW IDLE<br>SCHEDULING END OF COOL FOR OPERATION AT 03/01/2002 11:15:00 | Oper210 completes and frees the Crew2 resource.<br>The Crew2 resource is now idle and available for other operations to allocate it. |
| 03/01/2002 11:15:00<br>START OF RESOURCE FREE CHECK EVENT FOR RESOURCE Crew2<br>REQUEST QUEUE EMPTY | |
| 03/01/2002 11:15:00<br>START OF OPERATION EVENT<br>PROCESSING OPERATION Oper210,<br>LOAD 1 OF ORDER Job210-000<br>END OF PROCESS FOUND SELECTING NEXT OPERATION<br>END OF PROCESS FOR LOAD 1 OF ORDER Job210-000<br>0 LOADS OUTSTANDING<br>ORDER Job210-000 COMPLETE | The Scheduler finds no next operation specified for Oper210, so it completes Job210-000. |
| SCHEDULE HALTED BECAUSE THERE WERE NO MORE ORDERS TO PROCESS<br>SCHEDULE END AT 03/01/2002 11:15:00 | The Schedule is now complete. |

## Resource Group Member Selection Rules

You can define the member selection rule for a resource group in the RGRP.ALLOCRL field.

| Code | Rule | Description |
|---|---|---|
| 0 | Cyclic | Select the number of resources cyclically starting with the next resource listed after the previously selected one, and select the appropriate number of available members/ resources. |
| 1 | Index | Select the resource with index equal to the number in the Allocation Rule Attribute field. If that resource is unavailable, the system waits for it. Note: A resource group member's "index" comes from the order resources are listed in the group definition. |
| 2 | Least Mean Utilization | Select the number of resources that have the least mean utilization. Ties are broken based on the order that resource group members are listed. |
| 3 | Longest Idle | Select the resource that has had the longest continuous idle period prior to this request. |
| 4 | Minimum Setup Time | Select the number of resources that have the minimum setup time for the first downstream operation. Ties are broken based on the order the resource group members are listed. |
| 5 | Random Selection | Select randomly the number of required resources. |
| 6 | Reserved for Order | Select the number of resources that are reserved for this order. If the required number is not available, available resources are chosen based on least current utilization. |
| 7 | Select in Sequence | Select the required number of resources, searching in the sequence that the resource group members are listed in the group definition (always start at the top of the list of resources). |
| 8-39 | User Defined. | |

# Managing Resource Availability and Interruptions

By default, a resource is available for work 24 hours per day, seven days per week. Three components allow you to refine the availability of your resources according to your plant shifts and work schedules:

- Shifts: assigned to resources to make the resource available for specified time intervals.
- Shift Exceptions: assigned to resources to define periods of down time or overtime within a defined shift.
- Holidays: downtime for all resources that are assigned to any shift.

## Handling Shift Overrun Situations

If a job operation must be completed before the resource goes off-shift, you can set the Must Complete field on the shift interval record (SHIFT.MUSTCOMPFG) and on the resource (RESRC.MUSTCOMPFG) to Y. If the resource cannot finish the entire operation within this interval, the resource will wait to start the operation in the next shift interval where it can finish the entire operation (or where the SHIFT.MUSTCOMPFG field is set to N). The Scheduler will try to schedule another operation, in selection rule priority, that can be completed within the available time.

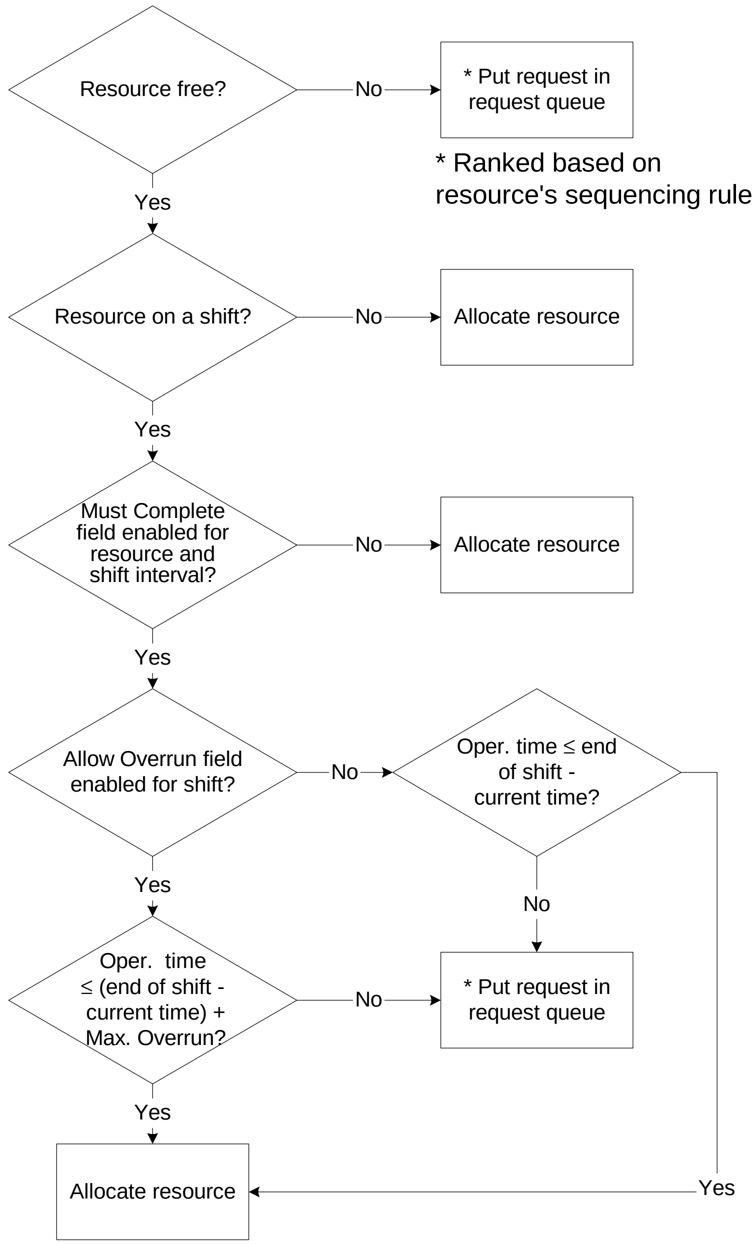
**Note:** For the Must Complete option to have an effect, it must be set to Y on both the resource record AND on the shift interval record.

When you use the Must Complete option, you can define a Maximum Overrrun (RESRC.MAXORUN) value to allow the resource an extra block of time to finish the operation, and allow it to start within the Must Complete shift interval. Think of this option as a "grace period" for Must Complete shift intervals.

For example, Operation 123 takes 3.25 hours to complete, and is released at 2:00. Operation 456 takes 2 hours to complete, and is released at 2:01. Both operations require resource ABC. Resource ABC has 3 hours of remaining on-shift time, and the Must Complete option is enabled. Therefore, Operation 123 does not allocate the resource and instead enters a request in the resource’s request queue. Operation 456 will allocate the resource because it can be completed within the available time.

**Note:** If you had defined the Maximum Overrun for .25 hours, Operation 123 would have allocated the resource, because the resource would have worked the extra .25 hours to complete the operation. The Allow Overrun option (SHIFT.OVERRUNFG) must be set to Y on the shift record for Maximum Overrun to function.

The diagram below illustrates how the Scheduler’s resource allocation decisions are affected by these options.




# Reducing Resource Availability

When defining a resource, you specify shifts when the resource is available. You can apply a shift exception to one of those shifts to shorten the shift and reduce resource availability.

With a shift exception, you can specify the portion of a shift when resources should not be available (SHIFTEXDI.WORKFG=N).

Consider this example with two shifts—SHIFT1 and SHIFT2. SHIFT1 runs from 07:00 to 15:00 Monday through Friday, and SHIFT2 runs from 15:00 to 23:00 Monday through Friday. Resources RES1 and RES2 are both available during these shifts.

If you want to cancel SHIFT2 for RES1 during the first two weeks of March (but leave it for RES2), you can define a shift exception, SHEX1, that:

- runs from 00:00 March 1 to 24:00 March 14
- has the WORKFG field set to N
- references RES1
- references SHIFT2.

This exception makes RES1 unavailable for SHIFT2 during this period.

**Note:** If you default the shift, the shift exception applies to all shifts worked by the resource. In the example, this means SHIFT1 and SHIFT2 for RES1.

## Increasing Resource Availability

You can also use a shift exception to increase resource availability (SHIFTEXDI.WORKFG=Y). To do so, define the shift exception to apply to both a shift not associated with the resource, and to the resource. This extends the resource's shift time and increases its availability.

For example, consider two shifts: SHIFT1 and SHIFT2. SHIFT1 runs from 07:00 to 15:00 Monday through Friday, and SHIFT2 runs from 15:00 to 23:00 Monday through Friday. Resource RES1 is available during these shifts. To add a third shift for RES1 during the first two weeks of March, define SHIFT3 to run from 23:00 to 07:00 Sunday through Friday, and define a shift exception, SHEX2, that:

- runs from 00:00 March 1 to 24:00 March 14
- references SHIFT3
- references RES1
- has the WORKFG field set to Y.

This makes RES1 available for SHIFT3 during this period.

## Resource Reallocation During Interruptions

When a load of items on a job is being processed, it usually completes an operation and moves on to the next one in the routing. However, three situations can interrupt the load from being processed at an operation:

- End of a shift
- Start of a shift exception
- Start of a holiday period

Regardless of the cause, three things happen when the load is interrupted:

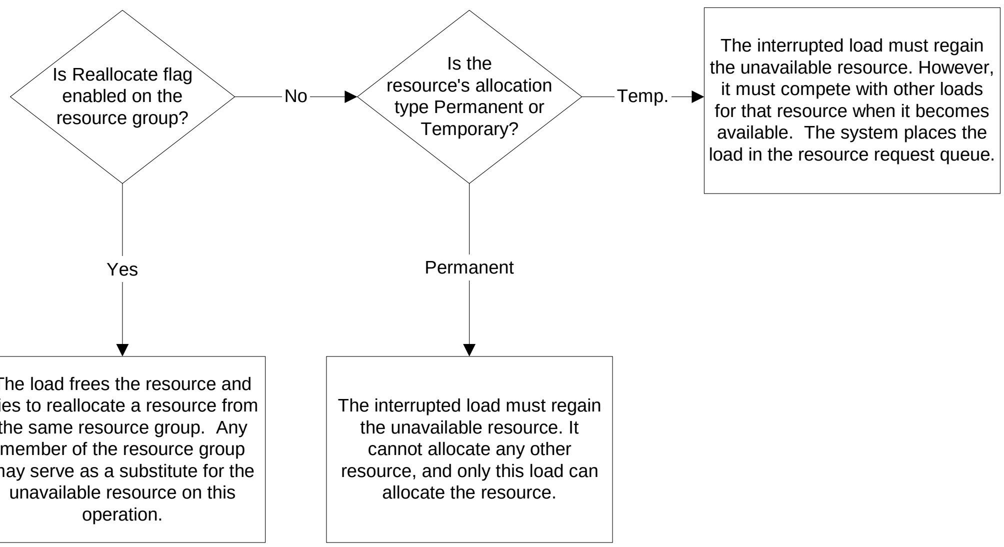
1 Operation processing for the load stops.

2 The resources that have gone off-shift become unavailable.

3 All other resources allocated to the load remain allocated.

The system keeps track of each unavailable resource for the load. Once interrupted at an operation, a load must wait to resume operation processing until the system can again allocate all of the load's needed resources.

The Reallocate option on the Resource Group (RGRP.REALLOCFG) and the Allocation status on the Resource (RESRC.ALLOCCD) allow you to control the situation of losing a resource due the conditions described above. The diagram below illustrates the combinations of the two options:




- Reallocate selected: The load frees the resource and tries to reallocate a resource from the same resource group. Any member of the resource group may serve as a substitute for the unavailable resource on this operation. The Scheduler does not consider the permanent/temporary state of the resource if Reallocate is selected on the resource group.
- Reallocate not selected: The load can wait until that specific resource is available. The setting of the original resource's Allocation option determines whether the load reserves that resource or must compete with other loads to reallocate it.
  - Allocation=Permanent: To resume processing at the operation, the interrupted load must regain the unavailable resource. It cannot allocate any other resource, and only this load can allocate the resource. Reallocation takes place when the resource becomes available, whether reallocation of any other needed resources succeeds or fails.
  - Allocation=Temporary: The interrupted load must regain the unavailable resource to resume operation processing. No other resource can substitute. The interrupted load must compete with other loads for the unavailable resource when it becomes available. The system places the load in the resource's request queue. Reallocation occurs only when the needed resource becomes available and when the interrupted load's request is selected.

**Note:** If one or more of the needed resources falls in the "no reallocation" category, reallocation is not possible until those resources become available again.

You will usually enable the Reallocate field for groups that contain crew resource types working on different shifts, but not for machine resource types.

# Defining Operation Setup Time

Setup time is the portion of the operation spent preparing the resource for performing the operation (such as tooling, proofing, or any other work not proportional to the units produced). The Scheduler factors the setup time into the duration of the operation.

Depending on your type of business, resources may always need to be set up. Or, they may need to be set up only when changing from working on one type of item to another. Likewise, the setup time could be a fixed value or may vary depending on the type of item.

## Defining the Setup Resource Group

On each operation, in the Setup Resource Group field (JS19VR.RGID), define a resource group from which to allocate the resource to be setup. The Scheduler will use the resource allocated from this group to determine the setup time based on the criteria you define (setup by changes in item, changes in setup group, etc.).

If you leave the JS19VR.RGID field blank, the operation incurs no setup time.

## Defining When to Apply Setup Time

To specify the situations in which the operation requires the resource to be set up, select the appropriate **Setup Rule** on the operation record (JS19VR.WHENRL). Two example values are shown below; see the *APS Database Reference* for a complete list.

- Always: this operation always needs the resource to be set up, regardless of the previous item the resource was working on.
- Basis: this operation needs the resource to be set up only when the previous operation using the resource was working on another item or Setup Group (group of related items).

When the Setup Rule is set to Basis, select the appropriate **Setup Basis** on the operation record (JS19VR.BASEDCD). Two example values are shown below; see the *APS Database Reference* for a complete list.

- Item: apply the setup time when the previous operation using the resource was working on a different item than this operation.
- Setup Group: apply the setup time when the previous operation was working on an item that is a member of a different Setup Group than the item this operation is working on. You define setup groups on the Setup Groups form. For example, you might group all red bike frames in a "RED"

setup group and blue bike frames in a "BLUE" group. The setup time could be based on the time needed to set up the resource from painting red frames to painting blue frames.

**Note:** Each operation can have only one setup basis, but you may define many setup basis configurations. For example, the bike assembly operation could have a setup basis of "Type of Handle Bar," and the bike frame operation could have a "Color" basis.

## Defining the Amount of Setup Time

You can define a fixed setup time or use a calculated value. You can select the **Setup Time Rule** on the operation record (JS19VR.STIMEXPRL) to represent either situation. Two example values are shown below; see the *APS Database Reference* for a complete list.

- Fixed: apply the value from the Setup Hrs field on the operation.
- Setup Lookup Table: look up a value from a table that defines setup time based on changing the resource’s setup from working on one type of item to working on another. See “Creating a Setup Lookup Table below for more information.

## Creating a Setup Lookup Table

Define setup records in the LOOKUP table to define the required setup time for changing a resource's setup from working on one type of item to another. You define a relationship between two items or groups of items and a corresponding setup value. When the Scheduler encounters this matrix referenced on an operation, it applies the corresponding setup time to the operation.

Just as you can define the Setup Basis on the operation for specifying when to apply setup time, you define the Setup Basis for the lookup table to apply its setup time. For example, a setup lookup table can be based on the item changing or the setup group changing.

## Setup Time Example

The following example illustrates how the Scheduler applies setup time for three different operations producing three different items. Assume these conditions:

- There are three items, each with a single operation. All three operations require the same resource.
  - Adult Bike: Member of setup group Adults. Setup time 2.5 hours.
  - Kids Bike, Blue: Member of setup group Kids. Setup time 0.5 hours.
  - Kids Bike, Red: Member of setup group Kids. Setup time 1.5 hours.

- The Setup Lookup table "Adults-to-Kids" determines setup time when the setup group changes as follows.

| From | To Adults | To Kids |
|---|---|---|
| Unknown | 5.0 | 6.0 |
| Adults | 0.2 | 1.0 |
| Kids | 2.0 | 0.3 |

> **Caution:** The "Unknown" row accounts for the situation where there is no operation scheduled prior to the current one.

The table below shows the Setup Rule and Setup Time rule for each operation and how the setup time is applied based on these rules. Items are listed in the sequence in which they arrive at the resource.

| Setup Rule: | Always | Basis | Basis | Always | Basis | Basis |
|---|---|---|---|---|---|---|
| **Setup Basis:** | **Item** | **Item** | **Group** | **Group** | **Item** | **Group** |
| **Setup Time Rule:** | **Fixed** | **Fixed** | **Fixed** | **Lookup table** | **Lookup table** | **Lookup table** |
| **Lookup Table:** | **N/A** | **N/A** | **N/A** | **Adults-to-Kids** | **Adults-to-Kids** | **Adults-to-Kids** |
| | Ø | Ø | Ø | Ø | Ø | Ø |
| Kids Bike, Red | 1.5 | 1.5 | 1.5 | 6.0 | 6.0 | 6.0 |
| Kids Bike, Blue | 0.5 | 0.5 | 0 | 0.3 | 0.3 | 0 |
| Kids Bike, Blue | 0.5 | 0 | 0 | 0.3 | 0 | 0 |
| Adult Bike | 2.5 | 2.5 | 2.5 | 2.0 | 2.0 | 2.0 |
| Kids Bike, Blue | 0.5 | 0.5 | 0.5 | 1.0 | 1.0 | 1.0 |

# Using Overlapping Operations

The Scheduler can split a load of items into smaller loads and allow completed loads to proceed to an overlapping operation without waiting for the original load's quantity to be completed. This feature allows for parallel processing of the original load of items across multiple operations. For example, you can define overlap hours on an operation to specify when to start the operation that follows the current one (for example, an Overlap Value of 0.5 means "start the next operation 30 minutes after this operation starts").

The JS19VR.OLTYPE and JS19VR.OLVALUE fields determine how the Scheduler calculates the load size and the start and end times for the resources. The sequencing of resources within the operations is recorded in the RESSCHD table.

## Split Processing

If an operation is defined to allow overlapping, prior to starting operation processing, the Scheduler creates any new loads and places the appropriate number of items on each load. The Scheduler generates split load sizes in terms of a number of items or a number of loads, based on the OLTYPE code and OLVALUE in the JS19VR record:

| OLTYPE | Calculation |
|---|---|
| 0 | No overlap |
| 1 | load size=OLVALUE |
| 2 | # loads=100 ÷ OLVALUE |
| 3 | same as 2 |
| 4 | # loads=JOBSTEP.STEPEXP ÷ OLVALUE (Fixed)<br>load size=OLVALUE ÷ JOBSTEP.STEPEXP (Per item) |
| 5 | same as 4 |
| 6 | parallel - no waiting |
| 7 | parallel - wait for all loads |

### Load Splitting with Uneven Quantities

The job quantity may not be evenly divisible by the split size. If this occurs, the Scheduler processes the uneven remainder of items by creating a new load.

For example, assume the job quantity is 100 and the load split size is 25. The number of whole loads is determined by dividing 100 by 25, which results in four loads of 25 each. But if the load split size were 30, there will be 3 whole loads of 30 and a remainder of 10. The Scheduler places the remaining items in a fourth, partial load of 10.

## Multiple Resources in Resource Group

If a resource group contains multiple resources, the loads can potentially allocate all resources and begin processing at the operation concurrently. For example, if you have 2 loads moving through Operation 10, where the required resource group contains 2 resources, each load can allocate one of the two resources (according to the selection rules).

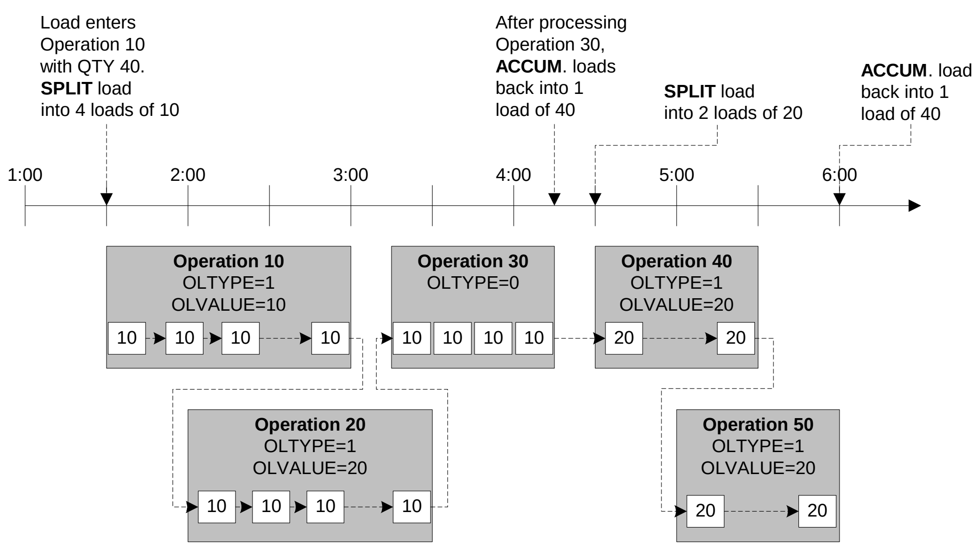
A load being split may hold resources at the time it arrives to the split phase. The Scheduler attempts to evenly distribute the resources held between the original load and the newly created loads. If there are excess units of a resource held, they are associated with the last load released from the operation. If there are not enough units of a resource to be spread evenly among the loads, the last load receives all of the units.

# Accumulation Processing

When the Scheduler encounters an operation with OLTYPE=0 after processing one or more operations for which OLTYPE≠0, it processes the operation that has OLTYPE=0 and then accumulates the original load back together. During the accumulation, the Scheduler adds a load to the accumulating quantity for its particular order. When the number of arriving items for a particular order is greater than or equal to the initial order size, the accumulation is considered complete.

# Overlapping Operations Example

The diagram below illustrates a typical overlapping operation situation.



1 The load arrives at operation 10 with a quantity of 40 items. Operation 10 is defined with OLTYPE=1, which allows overlap. The OLVALUE of 10 causes the Scheduler to split the load into loads of 10 (the load of 40 items becomes 4 loads of 10 items).

**Note:** The Scheduler keeps the loads divided into the load size calculated at the beginning of any series of overlapping operations.

2 The 4 loads compete for the resources in operation 10 according to the defined resource selection rules. One of the loads allocates the required resource. The resource does not begin processing the next load until it completes the first one (however, if the resource groups contained multiple resources, the other loads would allocate the available resources and begin processing).

3 When the load is completed in operation 10, it immediately moves to operation 20. Note that the OLVALUE for operation 20 is different than for operation 10. The Scheduler ignores the OLVALUE in all overlapped operations and keeps the loads divided into the multiple specified in the first operation in the series.

4 The Scheduler keeps the loads divided until after it completes operation 30. Then it accumulates them back into one load of 40.

5 The load arrives at operation 40, which is defined with OLTYPE=1. The Scheduler splits the load into two loads of 20 items, according to the OLVALUE of 20.

6 When the first load of 20 items is completed, it immediately moves to operation 50.

# Splitting a Load at a Single Operation

You can split the processing of a single operation across multiple resources. For example, if you have an operation that grinds 100 items, and you have five grinders, you could specify that the Scheduler break the load up into 5 groups of 20. You specify the load split size in the JS19VR.SPLITSIZE parameter.

When a load arrives at an operation that can be split, the Scheduler processes the operation as follows:

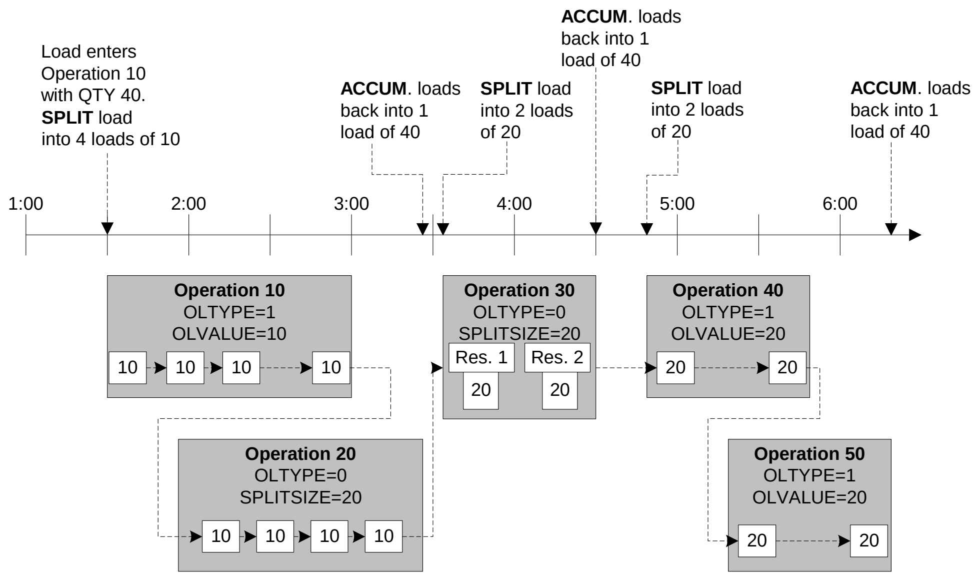
1 The original load quantity is broken into loads of the specified split size. The last load may be partial, depending on the quantity.

2 The loads are processed using the resource selection logic described in "Assigning Resources and Materials to Loads." The resource selection process balances resource utilization and delivery time requirements. All resources in the associated resource group have the potential of selecting each load.

3 After all loads have completed the operation, the Scheduler recombines them into one load.

## Splitting Load Example

The diagram below illustrates overlapping operations adjacent to an operation defined with a split size.




1 The processing at operations 10 and 20 proceeds the same as with the previous overlapping operations example. Note that the SPLITSIZE parameter is defined in operation 20; the Scheduler ignores it because the operation is overlapped.

**Note:** The Scheduler ignores load splitting when it is defined within a series of overlapped operations.

2 When the load arrives at operation 30, the Scheduler splits the load into two loads of 20 items according to the SPLITSIZE setting. The two resources in operation 30 each process one load simultaneously.

3 After operation 30 completes both loads, the Scheduler accumulates them back into one load of 40 items. The load proceeds to operation 40 (see the previous overlapping operations example for description).

# Running the Scheduler

When you integrate APS into your ERP system, you will decide on a method (such as a menu or button) for your users to start a Scheduler run. To run the Scheduler, your user interface must call the crun_scheduler API from the APS Server. This API call is described in detail in Chapter 3, "Using Functional APIs," in the *APS Integration Guide*.

# Output Data

During a Scheduler run, APS populates the following SQL Server tables:

| Table | Contents |
| --- | --- |
| BATTIME | A record for each batch which stores the time the batch was formed and how long it processed. |
| BATWAIT | A record for each order which was part of a batch which stores the time the order started waiting and how long it waited. |
| DOWN | A record for each period of time any resource is projected to be idle, and the reason for the idleness. |
| JOB | Information about allocations (operations scheduled) and loads the operations have generated. |
| MATSCHD | Information about jobs that affect material inventory. |
| ORDIND | Used internally by the Scheduler to join tables. |
| RESLOAD | Information about a resource's load for each planning period (such as the number of hours a resource is loaded). |
| RESSCHD | Information about the operations the Scheduler has scheduled and the resources those operations have allocated. This data includes the start and end times (sequencing information). |
| RGLOAD | Information about a resource group's load for each planning period. |

# Summary Performance Data

In addition to the output tables listed above, the Scheduler populates several output tables that provide general performance data about the schedule.

| Table | Contents |
| --- | --- |
| ALTSUM | Performance statistics for each alternative. |
| BATSUM | Summary information for each batch definition. |
| LOADPERF | Performance statistics for each load. |
| LOADSUM | Summary information for each load. |
| MATLDELOUT | Summary information about material deliveries. |
| MATSHPOUT | Summary information about material shipments. |
| MATSUM | Summary information about manufactured materials. |
| ORDPERF | Performance statistics about order completion. |
| ORDSUM | Summary information about orders (such as average lateness, processing time, makespan, etc.). |
| PARTSUM | Summary information about end-item completion (similar to ORDSUM). |
| RESSUM | Summary information about resource production (such as average breakdown times, queue length, number of jobs processed, etc.). |
| RGRPSUM | Load profile for resource groups. |

# Index


## A

Acquisition schedules, defining 2-26

Allocating resource temporarily 2-18

Alternate items, specifying 2-38

ALTPLAN table 2-16

ALTSUM table 3-78

## B

BATSUM table 3-78

BATTIME table 3-78

BATWAIT Table 3-78

## C

CAL table 1-13

Capable-to-promise (CTP) date 2-18

Component parts 1-11

Constraints 2-27

crun_scheduler 3-77

CTP date 2-18

## D

Database Overview 1-7

Database tables

- ALTPLAN 2-16
- ALTSUM 3-78
- CAL 1-13
- DOWN 3-78
- DOWNPLAN 2-16
- INVPLAN 2-16
- JOB 3-78
- JOBPLAN 2-16
- JOBSTEP 1-9
- JS19VR 1-9
- LOADPERF 3-78
- LOADSUM 3-78
- MATL 1-11, 2-27
- MATLDELOUT 3-78
- MATLDELV 1-13
- MATLPLAN 2-16
- MATSCHD 3-78
- MATSHPOUT 3-78
- MATSUM 3-78
- OPRULE 2-29
- ORDER 1-8
- ORDIND 3-78
- ORDPERF 3-78
- ORDPLAN 2-17
- ORDSUM 3-78
- PART 1-8
- PARTSUM 3-78
- PROCPLN 1-9
- RESLOAD 3-78
- RESPLAN 2-17
- RESRC 1-9
- RESSCHD 3-78
- RESSUM 3-78
- RGLOAD 3-78
- RGRP 1-10
- RGRPMBR 1-10
- RGRPSUM 3-78
- SHIFT 1-12
- SHIFTEXDI 1-12
- TODEMAND 2-17
- TOSUPPLY 2-17

Delivery rule 1-13

Delivery schedule 1-13
DOWN table 3-78
DOWNPLAN table 2-16
Due date for orders 2-19


**E**

Earliest start time, defining for orders 2-38
Entry date for orders 2-19
Example plan 2-21
Exceptions, to the plan 2-27
Expediting, lead time 2-24


**F**

Functional APIs
crun_planner 2-18
crun_scheduler 3-77


**H**

Holidays, resources 1-13


**I**

In-process, orders 1-8
Interrupted loads, in Scheduler 3-70
INVPLAN table 2-16
Item constraints 2-27
Items 1-8


**J**

JOB table 3-78
JOBPLAN table 2-16
JOBSTEP table 1-9
JS19VR table 1-9


**L**

Lead times, about 2-24
LOADPERF table 3-78
LOADSUM table 3-78
Lot sizes
maximum 2-37
minimum 2-37


**M**

Materials 1-11
manufactured 2-37
purchased 2-37
MATL table 1-11, 2-27
MATLDELOUT table 3-78
MATLDELV table 1-13
MATLPLAN table 2-16
MATSCHD table 3-78
MATSHPOUT table 3-78
MATSUM table 3-78
Maximum order lot size
for Planner 2-37


**N**

NOSOFFSET 2-38
NOSTIME 2-38


**O**

OLTYPE calculation 2-36
OLTYPE code 3-74
Operations 1-9
overlapping 2-36
OPRULE table 2-29
Optimizing, mfg. item usage 2-36
ORDER table 1-8
Orders
due date 2-19
entry date 2-19
new (scheduled) 1-8
projected date 2-18, 2-21
pull-up 2-33
request date 2-19
scheduled 1-8

triggering an order release 1-8
types of 1-8
ORDIND table 3-78
ORDPERF table 3-78
ORDPLAN table 2-17
ORDSUM table 3-78
Overlapping operations 2-36
Overtime 1-12

**P**

PART table 1-8
PARTSUM table 3-78
PROCPLN table 1-9
Projected completion date 2-18, 2-21
Pull-up orders 2-33
Purchased item supply switching 2-29
Push planning
differences from pull planning 2-20

**R**

Reallocation 3-71
Request date for orders 2-19
RESLOAD table 3-78
resource collisions 2-18
Resource groups 1-10
Resources 1-9
increasing availability 3-69
rules for sequencing 3-47
shifts 1-12
work schedule 1-13
RESPLAN table 2-17
RESRC table 1-9
RESSCHD table 3-78
RESSUM table 3-78
RGLOAD table 3-78
RGRP table 1-10
RGRPSUM table 3-78
Routings 1-9
Rules
Delivery 1-13
index, notes on use 3-60
selection of loads 3-47
sequencing resources and materials 3-47

**S**

Scheduler
collecting performance data 3-78
output 3-78
Scrap Factor 2-19
Selection rules
definition of 3-47
using values with 3-59
Sequencing rules
for resources and materials 3-47
using attributes with 3-59
Shift exceptions 1-12
example of 3-69
SHIFT table 1-12
SHIFTEXDI table 1-12
Shifts 1-12
modified by shift exceptions 1-12
resources 1-12
resources on 1-12
Shrinkage 2-19
soft-pegging inventory, APS 2-18
Supply switching, enabling 2-29

**T**

Time fence, for inventory and supply consumption 2-33

TODEMAND table 2-17

TOSUPPLY table 2-17

## U

Unconfirmed orders 1-8