Scheduler Concepts

Overview

Welcome to Enterprise Scheduler! This tutorial will introduce and guide you through the features and functions of the world's premier network scheduling solution, Enteprise Scheduler.

Each chapter in this guide builds upon the foundation of previous lessons. If this is your first time working with production scheduling software, or your first time using Enteprise Scheduler, you should work through each chapter consecutively. More experienced users might want to move directly to the lesson in which they are interested; however, many exercises utilize the results of previous exercises.

As you progress through the tutorial, more advanced and detailed topics are covered. By the end of this manual, you will be familiar with most of the major features of Enteprise Scheduler, and you will have the knowledge to create and maintain your own production schedules.

Prerequisites

Before using this tutorial, you must copy the sample files that are used in the tutorial to your machine. These files are in a **.tar** file on the Scheduler installation DVD.

To copy the files to your machine, follow these steps:

Step 1 Install the agent as root as described in the *Installation and Configuration Guide* before installing the tutorial sample files.

Step 2 The *.tar* file with the tutorial sample files is found on the DVD at **\agent\Unix\tutorial.tar.** Copy this file onto the agent machine to the / (root level) directory of the Unix agent you will run jobs from.

Step 3 Change the directory permissions:

```
chmod 777 *
```

Step 4 Unpack the *.tar* file with the following command:

```
tar -xvf tutorial.tar
```

The following sample files should be unpacked:

- UNIX_log.txt
- UNIX_Log2.txt
- UNIX_TEST_1.sh
- UNIX_TEST_2.sh
- UNIX_TEST_3.sh

- UNIX_TEST_4.sh
- UNIX_TEST_5.sh
- UNIX_TEST_JEVENT.sh

Step 5 Change the permissions for the unpacked tutorial sample files with the following command:

```
chmod 777 *
```

Step 6 Change the ownership of the files to group:

```
chgrp -R <name of group> *
```

Note You can identify which group you are a member of by entering:
id
This returns the name of the group you belong to.

Step 7 Change the ownership of the files to a owner:

```
chown -R <name of owner> *
```

Note The files can now be used with the lessons provided in this tutorial.
To complete the exercises in this tutorial, you need to:
Install Enteprise Scheduler in the default directory Scheduler (or the examples in this tutorial will not work properly.
Select the Super User option in your User definition.
Configure a default agent
Create and have available the work day calendar

System Configurations

Enteprise Scheduler is used to schedule and manage jobs on several different systems — even different operating systems — through a single operator station. This single point-of-control means that you have only one Tidal Web client running on your PC to manage jobs across multiple systems. Masters are the central point for scheduling jobs on associated agents. Enteprise Scheduler jobs can only run on licensed agents.

Masters

The master is the UNIX system on which you install the “brains” of your Enteprise Scheduler network. You interact with the master using the Tidal Web client. The master performs the requested service on a local or remote agent, and then returns updated information to the Tidal Web client which displays the results.

The master launches jobs on machines licensed as agents. An **agent** is software on another machine in the same network as the master, and runs jobs on behalf of the master. The Tidal Web clilent does not connect directly to the agent, but schedules and manages its jobs through the master.

Client Manager

Two main components of the Enterprise Scheduler architecture are the Master and Client Manager. Client Manager allows Enterprise Scheduler to achieve higher performance and scalability needs. The purpose of the Client Manager is to service requests from user initiated activities, such as through the Tidal Web Client, Tidal Transporter and from other external sources that utilize the Command Line Interface (CLI) or published Enterprise Scheduler Web services. Client Manager allows the Scheduler Master to focus more capacity on core scheduling needs related to job execution and job compilations, while the Client Manager addresses demands from such activities as RSS feeds and users viewing/configuring scheduling data and output. A single Client Manager is mandatory and additional Client Managers can be deployed to address additional performance needs.

Agents

The agent is any machine that runs jobs on behalf of a Enterprise Scheduler master. The master and the agent communicate with each other to execute jobs remotely. Multiple Enterprise Scheduler agents provide greater production reliability should the master become unavailable for some reason.

Job commands that run on the agent should be accessible to the agent machine on your network. They are scheduled on the master, and initiated on the agent by the master when schedule dependencies are satisfied.

Agents operate independently from the master. This allows continued processing of any work that is already sent by the master if either the master schedule or the common shared network becomes unavailable. The agent relays the results of the job it continued processing when either the network connection or the master is available again.

Job Definition

The job definition is central to job scheduling. It is a set of job rules that defines:

- Which command to run
- Where the command runs
- When to run the job
- How to handle dependencies
- Whether to issue actions based on pre-defined job events
- The job priority relative to other jobs

When you want to schedule a command to be executed, you use a job. Once a job is defined, you can keep the definition and run the job repetitively according to its specified calendar, or as needed.

Each job is assigned to only one command. The command can be an executable, a batch file (Windows only), a shell script, a command file or any other executable process. You can specify parameters to be passed to the command. This enables you to use one command in different ways, based upon the parameters that you pass to it.

For example, a job can back up files to tape, run a program to post transactions to a database or run a set of reports. In Enterprise Scheduler, you give each job a name, and, if the job is repetitive, a calendar by which it runs. You can also define dependencies that must be met before the command is executed. Using the calendar, Enterprise Scheduler automatically launches jobs each time they are scheduled to run, but only after all of their dependencies have been met.

Job Hierarchy

Jobs are built on a hierarchy of job and job group ownership. A **job group** is a container for a set of jobs, usually part of a common application or department. The job group has its own name and set of runtime instructions.

You can use job groups to submit jobs that either depend on each other, or should run together. For example, all the jobs in payroll can belong to a group called Payroll. The job group can provide default settings to all the child jobs that belong to it. Jobs and job groups are displayed in the **Jobs** pane. Job groups can save you the time it takes to set up job definitions because each job in the job group can inherit the characteristics of that job group. When you want to create several jobs with similar scheduling characteristics, you can define those jobs within a job group and set the scheduling characteristics in the job group definition. It is also possible to change scheduling characteristics at the job level even though the job belongs to a group.

For example, if a job group is defined to run every Friday, then every job in that job group is automatically defined to run on Friday. If one job in the job group must run on Saturday, then that one job can be changed to the proper run day without affecting the other jobs — as long as you disinherit the job group calendar and change the calendar from within that job.

The ultimate ownership of a job or job group belongs to either the user or a **workgroup**. A workgroup is a collection of users who can share access to the same jobs. Workgroups are displayed in the **Workgroups** pane.

Dependencies

Dependencies are prerequisite conditions that must be met before a job can run.

Date and Time Dependencies

The most common dependency is the date and time when Enteprise Scheduler executes a job.

For example, you can schedule a job to run every Tuesday after 6:00 pm, except on holidays when it is not to run. Date dependencies are built using calendars. Time dependencies are specified within a job's definition.

Job Dependencies

Jobs can also depend on other jobs reaching a particular status.

For example, you can run Job51 after Job101 and Job207 have reached the status of **Completed Normally**. During the job's life cycle, Enteprise Scheduler recognizes the current status of a job, such as:

Table 1-1 Job Status

Status	Description
Waiting on Dependencies	The job is waiting on Date, Time, Job, and/or File dependencies.
Waiting on Resources	The job is waiting for an execution slot. All Dependencies have been met.
Waiting on Operator	All the job's dependencies are met and the job is waiting for the operator to release it.

Table 1-1 Job Status

Status	Description
Active	The job is actively running in the Production Schedule.
Completed Normally	The job completed normally.
Completed Abnormally	The job completed abnormally.
Error Occurred	An internal error occurred which prevented the job from running.

File Dependencies

A job can also depend on the status of a file. The state, size, creation or modification date of the file can all be taken into consideration.

For example, you can run Job101 if the Unix file **/payroll/data/trandata**:

- Has been modified in the last twelve hours
- Has a file size greater than 1024KB

Variable Dependencies

A job can also depend on the value of a user-defined variable. Enteprise Scheduler has a repository of user-defined variables that can be updated or incremented either manually or through an action associated with a job event or system event.

For example, you can set a job to run when:

- Variable **Printer Online** is set to **Yes**. The **Printer Online** variable could be set by a job that changes printer settings and then issues an action changing the variable from **No** to **Yes**.
- Variable **Payroll Jobs** is incremented to **15** by another job that increments the variable each time it runs.

Calendars

Calendars are used to determine what days to run jobs. Calendars let you schedule jobs to run on a periodic yet intelligent basis.

For example, Labor Day in the United States is celebrated on the first Monday in September which falls on a different date each year. By defining Labor Day as the first Monday in September, you avoid the need to manually redefine it every year.

You can also define calendar groups which combine individual calendars.

For example, the **1st Half Holidays** calendar group can include the **New Year's Day**, **President's Day**, and **Memorial Day** calendars.

Job Instances

A job instance is a specific, scheduled run of a job definition (job) by Enteprise Scheduler. One job can create many instances.

For example, if a job is defined to run every Monday, Wednesday and Friday, then Enteprise Scheduler creates one instance for Monday, one for Wednesday, one for Friday, one for next instance of Monday and so on. These instances can be viewed in the Job Activity pane.

Jobs can enter the production schedule on a scheduled or unscheduled basis. For example, you may have some jobs you expect to run at the end of each month, and other jobs which you run only on demand.

Production Schedule

The **production schedule** is the timeline Enteprise Scheduler uses to manage instances. You control the span of time covered by the production schedule, typically between a few days and several weeks. Job instances are displayed in the **Job Activity** pane.

- Past job instances remain available for a user-defined period of time.
- Present job instances and their statuses (i.e. **Waiting**, **Active**, **Completed Normally**, etc.) are displayed in the **Job Activity** pane default view.
- Future job instances defined in the production span appear on the future dates in the **Job Activity** pane.

As time progresses, the production schedule is recorded, and automatically updates job instances for the defined number of days for the past, current and future runs. The concepts of time offsets and basing the production schedule times on agent time zones are explored in *Chapter 2: Understanding Offset Concepts in Production Scheduling*.

Master Status Display

Double-clicking the master status light at the bottom of the console brings up the master status display. Here are continually updated statistics related to Enteprise Scheduler components, such as connections and the master's status.

Events and Actions

Enteprise Scheduler monitors jobs throughout their life cycle for predefined **events** — such as when the job launches, when it completes, if it fails and many others. You configure an exception condition called an **event** to automatically respond when the event is detected by triggering an **action**.

When you configure a event, you specify:

- System conditions that will trigger the event
- One or more actions to take in response
- Jobs to which the event applies (for job events)
- A schedule of time intervals when the event is active (file, email and variable events)

Events can be internally generated by conditions within the system (job and system events) or they can be generated by conditions that are outside the system (file, email and variable events). To detect external conditions, you must create an event monitor to watch for those defined conditions.

Job events combine event triggers with actions such as stopping or restarting a job while in production.

Common **event triggers** include abnormal termination, excessive run time and failure to complete by a specific time. You can take the following types of **actions**:

- Send email messages
- Control a job instance in the Job Activity window
- Alert an operator to a job condition
- Send SNMP messages
- Launch an unscheduled job (new job action)
- Issue a log message
- Update a user defined variable

For example, you can define a job event that is triggered every time a job is cancelled by an operator. When a job cancellation occurs, you can have an email sent to you and a message sent to your SNMP management software noting this event.

A **system event** operates identically to a **job event**, except that the master originates the event rather than a job. System events define global conditions versus a job event defining conditions that affect jobs. For example, if an agent shuts down, a **system event** can be triggered to notify users of the problem.

An **email event** is the detection of a specified text string in an email that arrives at a designated email account on an designated Exchange server. An email monitor is created to watch for the specified email.

A **file event** is the detection of a file on an agent reaching a specified state. A file monitor is created to watch for a file the matches the specified conditions.

A **variable event** is the detection of a variable reaching a specified value, whether the variable is on a local or remote master. A variable monitor is created to watch for the variable to reach the desired value.

Queues

Queues let you optimize throughput and allocate system resources for scheduled and unscheduled jobs. The Enteprise Scheduler queue manager assigns jobs to queues when all their dependencies have been met, and decides when to launch jobs based upon the available system resource slots. The maximum number of slots available is determined either by the limit that you set in the system queue, the sum of each queue's limit or the sum of each licensed agent's job limit.

Queues can limit the number of jobs running on a computer or a network of computers at a given time.

- If the system is not running at its capacity, a job can run immediately provided that all of its dependencies are met.
- If the system is running at its capacity, the Enteprise Scheduler Queue Manager decides which jobs launch based on a priority structure that includes the following in order of importance:

Table 1-2 Queue Priorities

Status	Description
Queue priority levels	Jobs in active and open queues at higher priority levels run first.
Queue limits	Only jobs in queues not running at their allowable limit can be launched.
Agent job limits	Only jobs assigned to agents not running at their allowable limit can be launched.
Job priority levels	Jobs with the highest priority (assigned in the job definition) in the queue are run first.

- Queues are displayed in a hierarchy. Each item in the hierarchy is a queue and can contain jobs. You define the queue limit to set the number of jobs that can launch from any individual queue. You also define a priority for each queue.

Queue Filters

Jobs are directed to a queue based on the queue filters that you define. These filters describe the job properties that must exist for the queue manager to assign a job to a particular queue. Some examples of the queue filters that direct jobs to queues are:

- Job class
- Job name
- Job owner
- Job estimated runtime

Agent Lists

Enteprise Scheduler extends its capability for automatic job management through **agent lists**. An agent list describes a set of nodes on your network available to run jobs. Agent lists designate nodes as primary or alternate nodes for job submission, and allow jobs to be broadcast across all available nodes. Workload balancing algorithms can distribute jobs evenly among all available nodes.

Security Policies

Security policies restrict access to certain Enteprise Scheduler functions. The defined access rights can be saved as a security policy, and then assigned to one user or multiple users.

For example, there might be different sets of users who:

- Administer Enteprise Scheduler
- Create and schedule jobs for themselves and others
- Operate the job schedule

Using security policies, the users that create and schedule jobs can be restricted from modifying the schedules. Likewise, the operators can be restricted from creating jobs.

Enteprise Scheduler includes default security policy templates that can be modified to create your own security policies. Each user within the supplied working model has a defined set of Enteprise Scheduler functions. When all the default security policies are in use, all aspects of scheduling are covered and available.

The following table lists the system features available for each of the default security template:

Table 1-3 Scheduler Security Policies

Default Security Policy	Available System Features
Scheduler_Admin	The default for new installations. This includes all available functions.
Administrator	Configures users.
User	Creates, edits, and submits jobs. Creates workgroups and user-defined variables.

Table 1-3 Scheduler Security Policies

Default Security Policy	Available System Features
Scheduler	Edits and tests job schedules.
Operator	Runs and controls jobs. Responds to alerts that jobs may issue.
Inquiry	Views jobs and resources. Cannot perform modification.

Logs and Reports

Enteprise Scheduler includes a logging mechanism that keeps track of all user edits, job status information, and error messages. In the **Logs** pane, you can view, filter and search all messages for a specific time frame.

For example, if you want to see who modified **Job A** recently, you can go to the **Logs** pane, search on **Job A** and view all instances when the job was edited.

Enteprise Scheduler also supports numerous reports, such as

- Data displayed in every window
- Operator alerts and responses
- Job statuses
- Event history
- Dependency cross-references
- Production schedule summary

Note For troubleshooting issues deeper than those gathered in the operations logs, gather the logs located in the Log directory of the installation of each TES component (Master, ClientManager, FaultMonitor, and so on). The *.out* file is the output of the process, while the*.log* files are the logs generated.