



# Tidal Workload Automation Tutorial for Windows and UNIX

Version 6.3.3

First Published: January 2018

[tidalautomation.com](http://tidalautomation.com)

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS. THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE PRODUCTS IN THIS MANUAL ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR STA GROUP REPRESENTATIVE FOR A COPY.

The implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. © 1981 Regents of the University of California. All rights reserved.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered uncontrolled copies and the original online version should be referred to for latest version.

© 2018 STA Group LLC. All rights reserved.

# Contents

Contents .....	3
Preface .....	7
Audience .....	7
Conventions .....	7
Related Documentation .....	8
Obtaining Documentation and Submitting a Service Request .....	8
Document Change History .....	9
Tidal Workload Automation Concepts .....	11
Overview .....	11
System Configurations .....	11
Master .....	11
Client Manager .....	11
Java Client .....	12
Agents .....	12
Job Definition .....	12
Job Hierarchy .....	12
Dependencies .....	13
Calendars .....	14
Job Instances .....	14
Production Schedule .....	14
Events and Actions .....	15
Queues .....	16
Agent Lists .....	16
Security Policies .....	16
Logs and Reports .....	17
Understanding Offset Concepts .....	19
Overview .....	19
Defining a Production Day .....	19
Positive Offset (Late Start) .....	20
Negative Offset (Early Start) .....	20
Scheduling Based on Agent Time Zone .....	21
Using a Positive Production Day Offset .....	22
Agent Running Ahead of the Master .....	22
Agent Running Behind the Master .....	23
Using a Negative Production Day Offset .....	24
Agent Running Ahead of the Master .....	24
Agent Running Behind the Master .....	25
Defining a Compile Offset .....	26
Sysval 150 .....	26

Creating Your First Job .....	27
Overview .....	27
Launching the TWA Web Client .....	27
Setting Default Job Parameters .....	27
Creating the Report Job Definition .....	27
Adding Your Job to the Production Schedule .....	29
Monitoring Your Job .....	29
Job Output Options .....	30
Specifying the Disposition of Job Output .....	30
Viewing Job Output .....	30
Using TWA Calendars .....	33
Overview .....	33
Building the First Workday Calendar .....	33
Building the Work Day Calendar .....	34
Building the Fiscal Week End Calendar .....	35
Alternate Method of Creating a Calendar .....	36
Building the Quarter End Calendars .....	36
Fiscal Quarter End .....	36
Calendar Quarter End .....	37
Scheduling the Report Job with a Calendar .....	38
Setting Up a Job Dependency .....	41
Overview .....	41
Defining a Job for Operator Release .....	42
Setting the Require Operator Release Option .....	42
Adding a Job with a Dependency .....	43
Monitoring Your Jobs .....	43
Viewing Dependencies from the Job Activity Pane .....	44
Releasing the Jobs .....	44
Scheduling Jobs in Groups .....	47
Overview .....	47
Job Group Hierarchy .....	48
Job Group Scheduling .....	48
Adding a Job Group .....	48
Creating Jobs that Belong to a Job Group .....	49
Creating the Ap31 Child Job .....	49
Creating the Ap32 Child Job with a Dependency .....	50
Scheduling the Job Group .....	51
Scheduling a Job that Depends on a Group .....	52
Adding a Job Group Dependency .....	52
Releasing a Job Group .....	53

Creating a File Dependency .....	55
Overview .....	55
Creating a Job with a File Dependency .....	56
Viewing the Job in Production .....	57
Creating a File to Satisfy the Dependency .....	57
Monitoring the Job in Production .....	58
Defining Actions Based on Job Conditions .....	59
Overview .....	59
Understanding Actions .....	59
Creating an Alert Action .....	60
Associating the Action to the Job .....	60
Changing the edi.bat file to Fail .....	61
Changing the Status of the Alert .....	62
Creating Variable Dependencies .....	63
Overview .....	63
Creating a Variable .....	63
Creating the Variable Update Action .....	64
Creating a Job Event for the Gather Report Job .....	64
Creating the Gather Orders Job Group .....	64
Creating a Repeating Job and Associating a Job Event .....	65
Creating a Job Dependent on the Gather Orders Variable .....	65
Scheduling and Monitoring the Gather Orders Jobs .....	66
Managing Jobs in the Production Schedule .....	69
Overview .....	69
Creating the Parent Job Group .....	69
Creating Three Child Jobs .....	70
Viewing the Jobs in the Job Activity Pane .....	71
Job Sorting .....	71
Removing a Job from the Production Schedule .....	71
Inserting a Job into the Production Schedule .....	71
Canceling a Job in the Production Schedule .....	72
Placing a Job on Hold .....	72
Removing the Job from Hold Status .....	72
Overriding a Dependency .....	73
Estimated and Excessive Durations .....	73
Managing Command Parameters .....	75
Overview .....	75
Passing Command Parameters .....	75

Specifying Command Parameters and Viewing the Batch File (Windows) .....	75
Tracking Methods .....	76
Using the Exit Code .....	76
Creating the Reporting Job Group .....	77
Setting the Report Writer 1 Job to Succeed .....	77
Setting the Report Writer 2 Job to Fail .....	77
Adding the Reporting Job Group to the Production Schedule .....	78
Setting Report Writer 2 to Complete Normally .....	78
Setting Up Queues .....	79
Overview .....	79
Setting Up the Inventory Weekly Job Group .....	80
Creating Jobs for the Inventory Weekly Job Group .....	81
Duplicating a Job Definition .....	81
Setting Up a Queue .....	82
Adding the Inventory Weekly Job Group to Production .....	83
Monitoring the Job Group in Production .....	83
Changing the Queue Limit .....	83
Resubmitting the Inventory Weekly Job Group .....	83
Using TWA Agents .....	85
Overview .....	85
Defining An Agent Connection .....	86
Adding a Job that Requires Operator Release .....	86
Adding an Agent Job Dependency .....	86
Releasing the Agent Jobs .....	87
Summary .....	87

# Preface

This guide provides step-by-step examples and instructions on how to use Tidal Workload Automation (TWA) 6.3.3.

## Audience

This guide is for the administrators, operators, and users of TWA.

## Conventions

This document uses the following conventions.

Conventions	Indication
<b>bold font</b>	Commands and keywords and user-entered text appear in <b>bold font</b> .
<i>italic font</i>	Document titles, new or emphasized terms, and arguments for which you supply values are in <i>italic font</i> .
[ ]	Elements in square brackets are optional.
{ x   y   z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

**Note:** Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

**Caution:** Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

**Warning:** IMPORTANT SAFETY INSTRUCTIONS

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

**SAVE THESE INSTRUCTIONS**

**Regulatory:** Provided for additional information and to comply with regulatory and customer requirements.

## Related Documentation

For a list of all Tidal Workload Automation guides, see the *Tidal Workload Automation Documentation Overview* of your release on [tidalautomation.com](http://tidalautomation.com) at:

<http://docs.tidalautomation.com/>

**Note:** We sometimes update the documentation after original publication. Therefore, you should also review the documentation on [tidalautomation.com](http://tidalautomation.com) for any updates.

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see *What's New in Tidal Product Documentation* at:

<http://docs.tidalautomation.com/rss>

Subscribe to *What's New in Tidal Product Documentation*, which lists all new and revised Tidal technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.



## Document Change History

The table below provides the revision history for the Tidal Workload Automation Tutorial Guide.

**Table 1**

Version Number	Issue Date	Reason for Change
6.2	June 2014	Updates for 6.2 release.
6.3	July 2016	<ul style="list-style-type: none"><li>■ Merged the Windows and UNIX tutorials into one guide.</li><li>■ Rebranded Cisco Tidal Enterprise Scheduler (TES) to Cisco Workload Automation (CWA).</li></ul>
6.3.3	January 2018	<ul style="list-style-type: none"><li>■ Rebranded Cisco Workload Automation (CWA) to Tidal Workload Automation (TWA).</li></ul>



# 1

## Tidal Workload Automation Concepts

### Overview

Welcome to Tidal Workload Automation (TWA). This tutorial introduces and guides you through the features and functions of the world's premier network scheduling solution, Tidal Workload Automation.

Each chapter in this guide builds upon the foundation of the previous chapters. If this is your first time working with production scheduling software, or your first time using Tidal Workload Automation, 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 Tidal Workload Automation, and you will have the knowledge to create and maintain your own production schedules.

**Note:** You can use this tutorial on either a Windows or UNIX platform. Where there are differences, platform-specific instructions are provided.

### System Configurations

Tidal Workload Automation 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 need only TWA Web client or Java Client on your PC to manage jobs across multiple systems. Masters are the central point for scheduling jobs on associated agents. Tidal Workload Automation jobs can only run on licensed agents.

### Master

The master is the Windows or UNIX system on which you install the “brains” of your Tidal Workload Automation network. You interact with the master using the TWA Web client. The master performs the requested service on a local or remote agent, and then returns updated information to the TWA 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 TWA Web client does not connect directly to the agent, but schedules and manages its jobs through the master.

### Client Manager

Two main components of the Tidal Workload Automation architecture are the Master and Client Manager. Client Manager allows TWA 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 TWA Web Client, TWA Transporter and from other external sources that utilize the Command Line Interface (CLI) or published TWA Web services. Client Manager allows the TWA 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.

## Java Client

Tidal Workload Automation Java Client is a thick client that gets installed on your PC or laptop and connects directly to the Master. Java Client downloads and stores all the key information from the Master to build an in-memory model. This in-memory model helps Java Client to deliver a better performance on client operations like sort, filter, scroll and so on.

## Agents

The agent is any machine that runs jobs on behalf of a Tidal Workload Automation master. The master and the agent communicate with each other to execute jobs remotely. Multiple Tidal Workload Automation 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. The job definition 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 define 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 defines 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 Tidal Workload Automation, 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, Tidal Workload Automation 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 Tidal Workload Automation 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, Tidal Workload Automation recognizes the current status of a job, such as:

**Table 1-1**

Status	Description
Waiting on Dependencies	The job is waiting on Date, Time, Job, and/or File dependencies.
Waiting on Resource	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.
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.

**Note:** This is a partial list of common job statuses. See the "Job and Job Group Statuses" in the *Tidal Workload Automation User Guide* for a complete list with descriptions.

### 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 file **C:\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. Tidal Workload Automation 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 the 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 that 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 Tidal Workload Automation. One job can create many instances.

For example, if a job is defined to run every Monday, Wednesday and Friday, then Tidal Workload Automation creates one instance for Monday, one for Wednesday, one for Friday, one for the 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 that you run only on demand.

## Production Schedule

The production schedule is the timeline TWA 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 (for example, 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 [Understanding Offset Concepts, page 19](#).

## Events and Actions

Tidal Workload Automation 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 an 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 canceled 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 Tidal Workload Automation 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 Tidal Workload Automation Queue Manager decides which jobs launch based on a priority structure that includes the following in order of importance:

**Table 1-2**

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

Tidal Workload Automation 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 Tidal Workload Automation 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 Tidal Workload Automation
- 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.

Tidal Workload Automation 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 Tidal Workload Automation 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 templates:

**Table 1-3**

Default Security Policy	Available System Features
Scheduler_Administrator	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.
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

Tidal Workload Automation 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 timeframe.

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.

Tidal Workload Automation 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 TWA component (Master, Client Manager, Fault Monitor, and so on). The *.out* file is the output of the process, while the *.log* files are the logs generated.



# 2

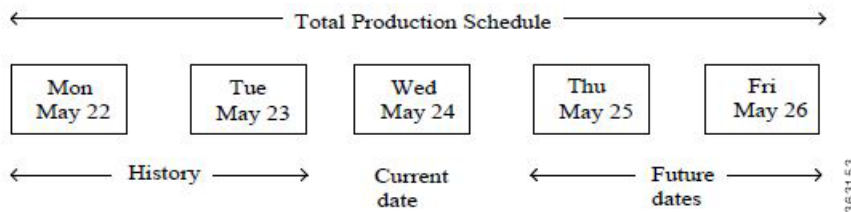
## Understanding Offset Concepts

### Overview

Tidal Workload Automation follows certain concepts when it compiles a production schedule. The concepts are basic but can result in complicated timing scenarios with date shifts that cause confusion among users when put into practice. The most complex timing issues arise from the shifting of the start of the production day through a series of time offsets as scheduled jobs run.

TWA compiles a *production schedule* from your job rules for each master. This production schedule covers at least the current day but may extend to multiple days. You determine the duration of each schedule by specifying the number of days to include. Each master in your network has its own production schedule, so schedule durations can vary. The active production schedule for a master includes history data (for dates past), the current date and any remaining days in the schedule (future).

#### Example of a Normal Schedule Length



The *production day* contains all of the job rules that are scheduled to run during the production day. A production day always contains 24 hours and by default starts at midnight and ends at 23:59:59 PM. (For simplicity's sake, we will use the 24-hour time format in this discussion so 11:59 PM is 2359.) To accommodate all of the processing needs of a business, a production day often differs from the calendar day. You can designate that the production day start at any time. For instance, you might specify that the production day starts at 0500 instead of starting at midnight to allow for the completion of the previous day's jobs. This is called a *production day offset*.

If the offset is positive, the fiscal day begins at some time after midnight and continues into the next calendar day. If the offset is negative, the fiscal day begins at some time before midnight. The maximum offset that you can define is 23 hours and 55 minutes. TWA uses the designated start of the production day to determine when the production day starts, and to select and launch jobs accordingly.

### Defining a Production Day

These topics are covered in the sections below:

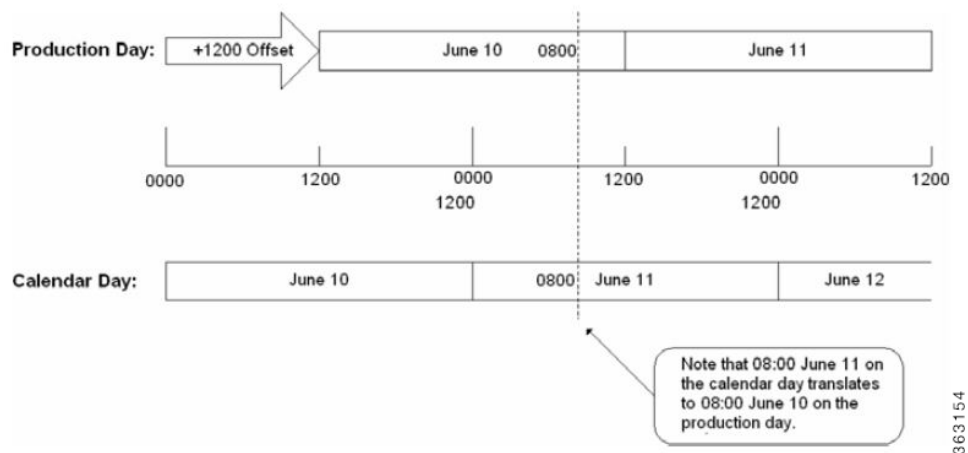
- [Positive Offset \(Late Start\), page 20](#)
- [Negative Offset \(Early Start\), page 20](#)

## Positive Offset (Late Start)

If you want your production day to begin at 1200 noon and continue until 1200 noon the following day, you define the start of the production day as 1200 (+1200). When offsetting the start time, it is important to remember that no time is lost, the hours between the start of the calendar day and the start of the production day are merely shifted from the beginning of the production day to its end. There are still 24 hours in the day.

The following figure compares the calendar day to a production day with an offset of +1200. Using this production offset, a job scheduled to launch at 0800 (8:00 A.M.) on June 10th (production date) will not actually launch until 0800 on June 11th (calendar date).

### Production Offset Defined As +1200 (Master/Agent in Same Time Zone)

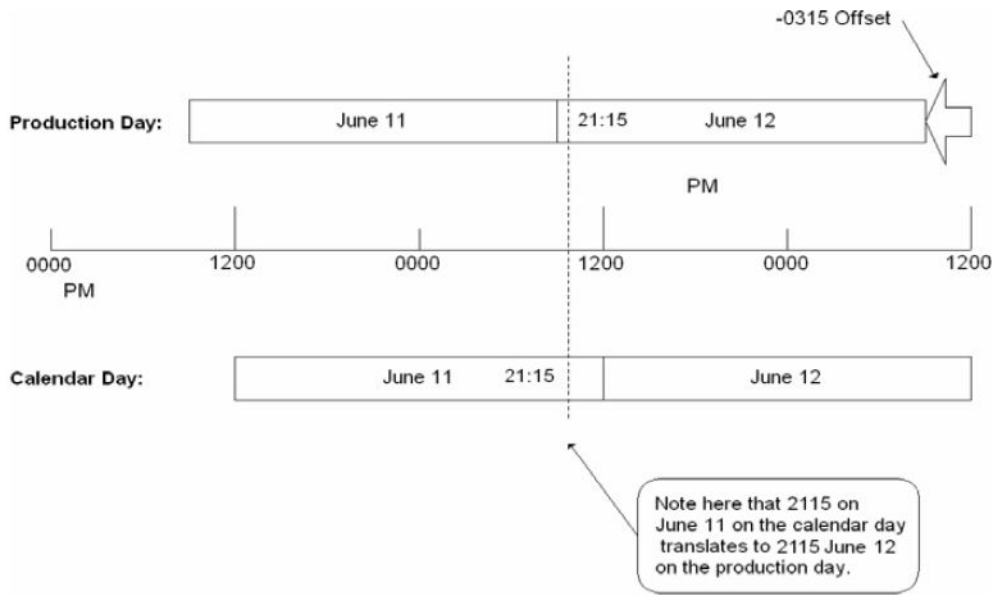


## Negative Offset (Early Start)

If you want your production day to begin at 2045 (8:45 P.M.), and continue until 2045 the following day, define the production day offset as -0315.

The following figure illustrates the calendar and production days with an offset of -0315. Using this production day offset, a job scheduled to launch at 2115 on June 12th (production date) launches at 2115 on June 11th (calendar date).

### Production Offset Defined As -0315 (Master and Agent in Same Time Zone)



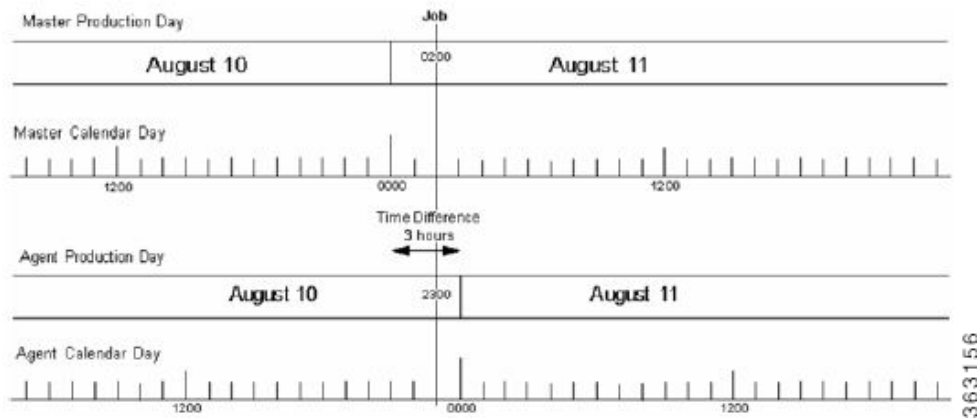
## Scheduling Based on Agent Time Zone

While jobs normally run from the master's time reference, you can launch jobs according to what time it is in the agent's time zone. Selecting the **Use Agent Time Zone** option on the **Master** tab of the System Configuration dialog will launch jobs according to the time where the agent resides. This change will take effect the next time any schedule is compiled. While the jobs will launch at the intended times in the agent's time zone, the master's viewpoint will be compiled in for the job time windows and start time. Since the master will compensate for the offset, users should not think about the difference between the different master and agent time zones when defining jobs rules. In a nutshell, using the master time zone imposes an absolute time reference while using the individual agent time zones imposes a relative time reference.

**Caution:** The master will be unable to predict shifts in times when compiling future schedules. Times will be calculated as an offset to the master time based on the timezone of the agent. If the agent shifts times, the master will not be able to predict this shift, as international daylight savings time laws constantly change, country to country. The schedule must be compiled under the influence of the new agent times.

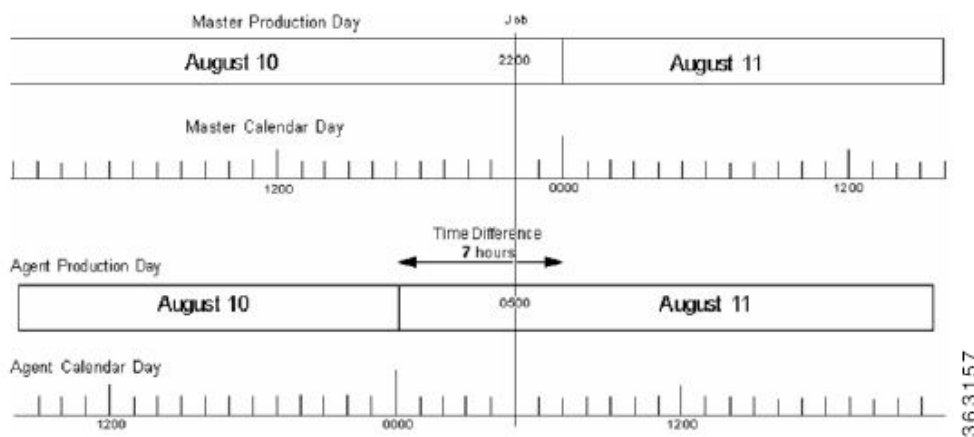
The following example illustrates the differences to be accounted for when the master and the agent reside in different time zones. The master in this example is three time zones ahead of the agent. A job defined to run at 2300 on the production day of August 10th will actually be launched by the master at 0200 on the production day of August 11th to account for the difference in time zones.

### Agent Residing Three Time Zones Behind Master (No Production Date Offset)



The following figure is another example of the master and agent in different time zones without a production offset. This example shows a master that is seven hours behind the agent. A job defined to run at 0500 on the production day of August 11th is actually launched by the master at 2200 on the production day of August 10th to account for the difference in time zones.

### Agent Residing Seven Time Zones Ahead of Master (No Production Date Offset)

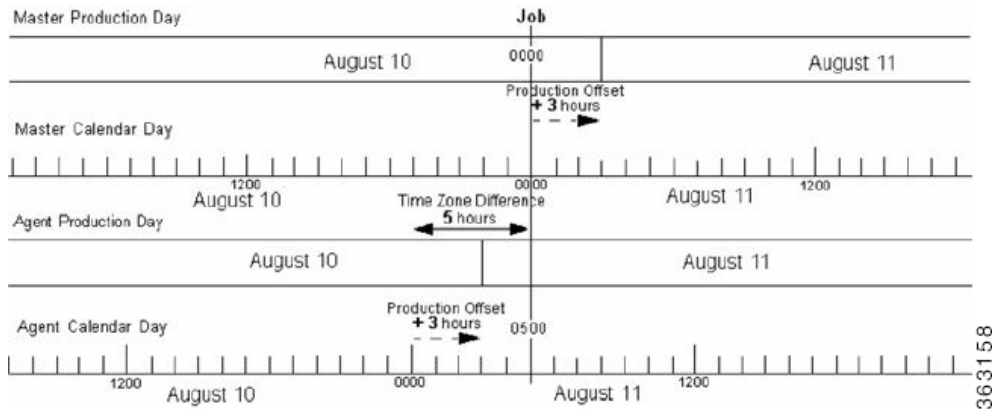


## Using a Positive Production Day Offset

### Agent Running Ahead of the Master

Setting a positive production day offset moves the start of the production day forward. In the example below, the difference between time zones is shown in a solid line and the production day offset is shown in a dotted line.

### Positive Production Day Offset With the Agent Running Ahead of the Master

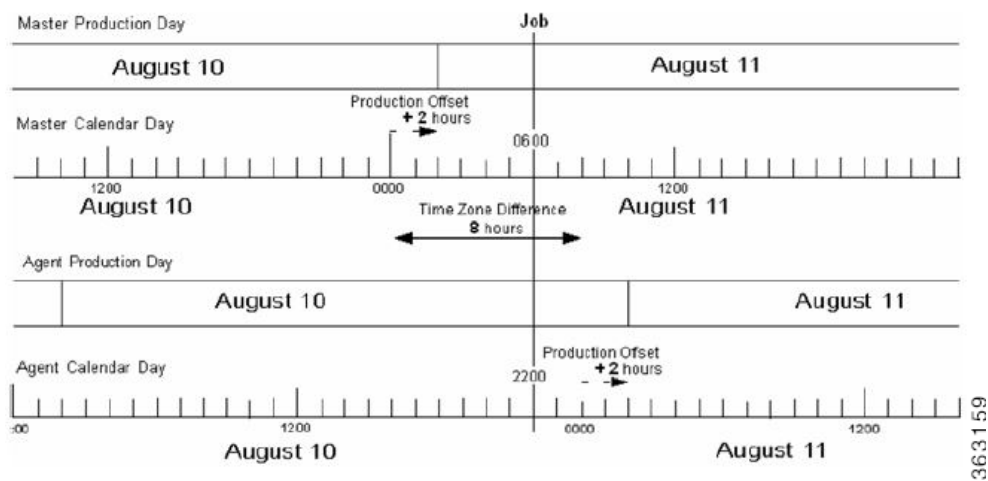


This example shows an agent that is five hours ahead of the master. A production day offset of positive three (+3) shifts the start of the production day (0000) three hours ahead in calendar time for both master and agent. A job defined to run at 0500 on the production day of August 11th launches at 0000 on August 10th on the master's production day.

### Agent Running Behind the Master

Setting a positive production day offset moves the start of the production day forward. In the example below, the difference between time zones is shown with a solid arrow and the production day offset is shown with a dotted arrow. This example has a positive production day offset of two hours ahead with the agent running eight hours behind the master. Thus a job defined to run at 2200 on the production day of August 10th is launched at 0600 on August 11th due to the difference in time zones.

### Positive Production Day Offset With Agent Running Behind Master



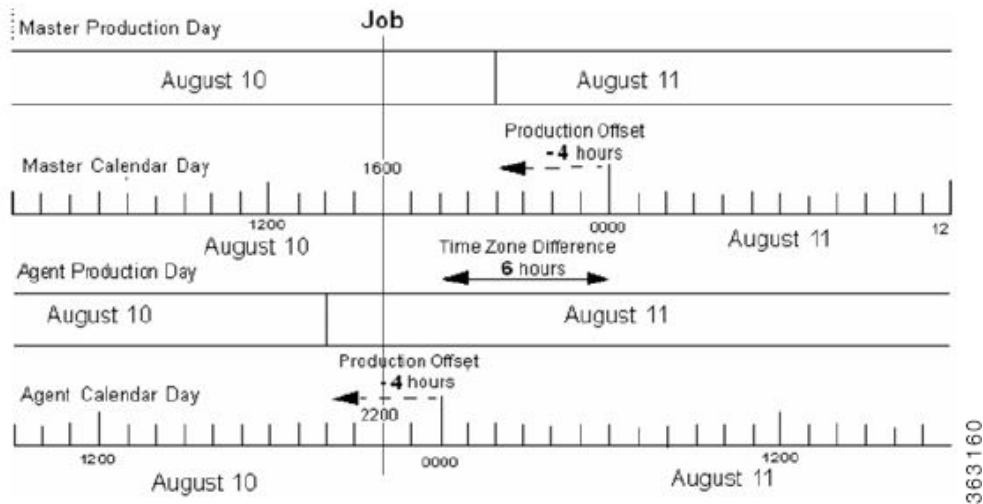
## Using a Negative Production Day Offset

### Agent Running Ahead of the Master

Setting a negative production day offset moves the start of the production day back from midnight. In the example below, the difference between time zones is shown with a solid arrow and the production day offset is shown in a dotted arrow. The example shown below has a negative 4 offset so the start of the production day is moved back four hours behind the start of the calendar day. A job defined to run at 2200 on the production day of August 11th is launched at 1600 on August 10th on the master due to the time difference.



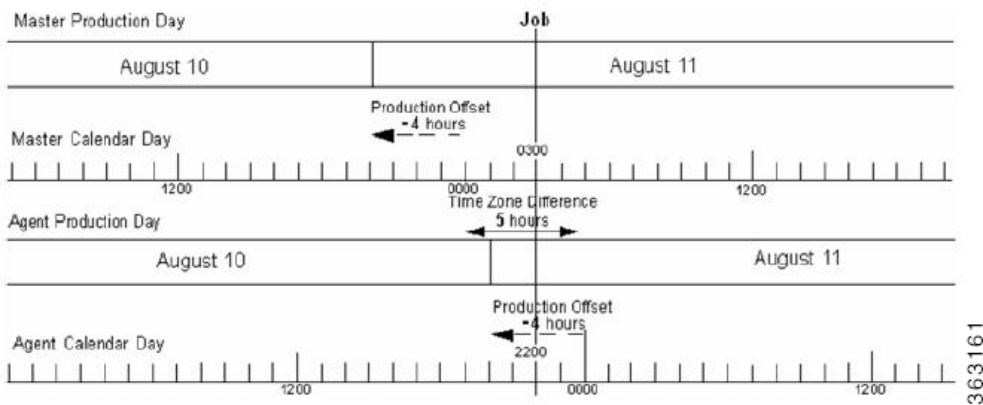
### Negative Production Day Offset With Agent Running Ahead of Master



### Agent Running Behind the Master

Setting a negative production day offset moves the start of the production day back from midnight. In the following example, there is a negative four hour production offset moving the start of the production day four hours behind the start of the calendar day. In the example below, the difference between time zones is shown with a solid arrow and the production day offset is shown with a dotted arrow.

### Negative Production Day Offset With Agent Running Behind Master

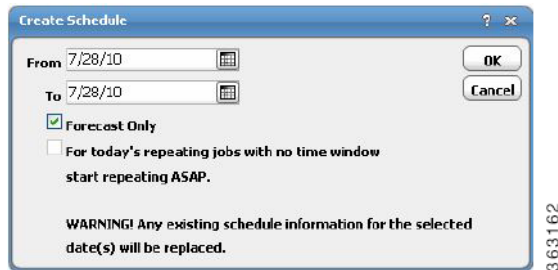


This configuration has an agent that is running five hours behind the master. A job defined to run at 2300 on the production day of August 10th is launched by the master at 0300 on August 11th.

## Defining a Compile Offset

Compiling the production schedule may consume enough CPU resources to seriously affect your system's performance and hinder other work that may be going on concurrently. It may be better to schedule such a resource-intensive operation like compiling your schedule, at a more convenient time when your system has a lighter workload. Once the schedule is compiled, it is saved until needed when the new production day starts. You can manually compile a new schedule at any time by selecting the **Create Schedule** option in the **Activities** main menu.

### Create Schedule Dialog



The compile offset is calculated from the start of the production day. The schedule will be compiled for the current day and all days that belong to the future days to include in the schedule. Any future day that was already scheduled (not forecast) will not get recompiled to reflect any job modifications or additions that were not committed to the schedule after the operation. To include any modifications that were not committed to the schedule, we must either recompile the already scheduled days or revert these schedules to a forecast type to force a compile before the day rolls into production.

There are differences between 6.x and 5.3.x as to how offsets are interpreted. If you want to preserve 5.3.x behavior, use the following.

## Sysval 150

Sysval 150 provides a way to retain 5.3.1 calendar logic for offset end date calculation in 6.x. Sysval with id 150 and value 'Y' should be created during database upgrade otherwise it can also be manually created using the below SQL query.

Sysval with id 150 and value 'Y' enforces 5.3.1 calendar logic for offset enddate calculation.

Use the following queries to update sysvals for Oracle DB:

```
delete from sysval where sysval_id in (150);

insert into sysval (sysval_id, sysval_string, sysval_integer, sysval_lstchgtm) values (150, 'Y', 0, sysdate);

commit;

/
```

Use the following queries to do so for MSSQL DB

```
delete from sysval where sysval_id in (150)

go

insert into sysval (sysval_id, sysval_string, sysval_integer, sysval_lstchgtm) values (150, 'Y', 0, getdate())

go
```

No sysval entry with id 150 or value 'N' enforces 6.x calendar logic for offset enddate calculation.

# 3

## Creating Your First Job

### Overview

This chapter describes how to use the TWA Web client to create, run and view the results of a job.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

Before you begin the tutorial, you must install at least two agent instances and define connections to them. Refer to the chapters on installing agents in the *Tidal Workload Automation Installation and Configuration Guide* for more information. These two agents must be installed and running to perform the exercises in this tutorial.

### Launching the TWA Web Client

To launch the TWA Web client, go to `http://<servername>:8080`, where `<servername>` is network name or address of Client Manager host, and log on using valid user credentials. For a list of browsers that are supported, see the *Tidal Workload Automation Compatibility Matrix*.

### Setting Default Job Parameters

Many parameters can be set from the **System Configuration** dialog. We will use the Tidal Workload Automation defaults throughout most of this tutorial.

**To check, and if necessary, to change the default agent:**

1. From the **Activities** menu, select **System Configuration** to display the **System Configuration** dialog.
2. Click the **Defaults** tab.
3. To change the default agent, select a valid agent from the **Agent Name** list.
4. Click **OK**.

### Creating the Report Job Definition

**To create the Report Job definition:**

1. Launch the TWA Web client.
2. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.

Unless other users have already created job definitions, the display does not show any data at this time.

- Right-click in the **Jobs** pane of the console and select **Add Job** from the context menu, or click the **Add Job** button on the toolbar.

The **Job Definition** dialog displays.

- In the **Job Name** field, enter **Report** as the name of your new job.
- In the **Command** field, enter the command for your operating system:

#### Windows

`C:\Program Files\TIDAL\Agent\TUTORIAL\report.bat.`

- or -

`C:\Program Files (x86)\TIDAL\Agent\TUTORIAL\report.bat`, if your agent was installed on 64-bit Windows machine, using default installation path.

**Note:** *Report.bat* is a batch file which executes the sleep.exe program for 30 seconds.

#### UNIX

`/UNIX_TEST_1.sh`

**Note:** The command `/UNIX_TEST_1.sh` lists all files in the current directory on the agent machine.

- Click the **Run** tab.
- From the **Runtime Users** list, select a runtime user with access to the Unix agent.

Note: The runtime user is the user under whose account the program or script needs to run. Since Unix Test 1 is a system command, any user with access to the Unix agent can run this command.

- Click **OK**.

A confirmation dialog displays.

This confirmation dialog reminds you that since a calendar is not associated with the job, it will not be scheduled to run automatically. We did not select a calendar because we are going to submit the job manually.

- Click **Yes**.

The **Report** job displays in the **Jobs** pane with your user name as the owner.

## Adding Your Job to the Production Schedule

You can now add the job to the production schedule.

### To add the job to the production schedule:

1. Right-click the **Reports** job and select **Insert Job Into Schedule** from the context menu.  
The **Insert Job Into Schedule** dialog displays.
2. Click **OK**.
3. When the **Information** dialog confirming that the job was added to the current production schedule displays, click **OK**.
4. In the **Navigator** pane, select **Operations>Job Activity** to monitor your job.

## Monitoring Your Job

The **Job Activity** pane is a key component in monitoring your production schedule with Tidal Workload Automation. All crucial job scheduling information is available at a glance from the **Job Activity** pane. The easy-to-read layout allows you to focus on the area of the display that is of interest to you. The columns and panes are resizable. Scrollbars facilitate access to the data you need.

From the **Job Activity** pane, you can:

- See when jobs are added or scheduled.
- Remove jobs (if you have sufficient security privileges).
- Monitor the progress of a job from a scheduled state through execution of the job.
- Control job execution.

The job statistics on the **Job Activity** pane include:

- Job owner
- Tidal Workload Automation agent assigned to run the job
- Estimated duration of the job
- Exit code of the completed job
- Actual command that the job represents
- Next date on which the job will run
- Calendar associated with the job (if any)
- Queue associated with the job (if any)
- Job's priority level
- Job group associated with the job (if any)

When jobs are scheduled, the **Jobs** pane in the center of the console shows one job per line, plus the job number, job name, job status, etc.

## Job Output Options

There are three job output options:

- **Discard** – Discards all job outputs. There will be no output log available for review.
- **Append** – Saves job outputs by appending the current output to a single log.
- **Replace** – Creates a new output file each time the job runs, replacing the previous log.

**Note:** Tidal Workload Automation's default is to discard output.

## Specifying the Disposition of Job Output

**To specify the disposition of job output:**

1. From the **Activities** main menu, select **System Configuration**.
2. Click the **Defaults** tab.
3. In the **Save Output Option** section, check **Discard**, **Append** or **Replace**. For the exercises in this tutorial, check **Replace**.
4. Click **OK**.

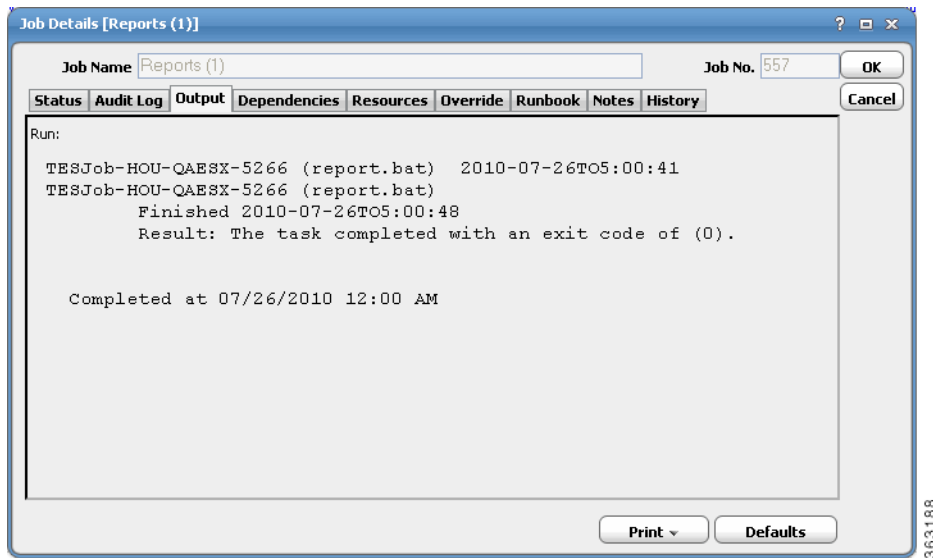
**Note:** To be able to view the output of a job, the **Append** or **Replace** option for job output must be selected prior to running the job.

## Viewing Job Output

You can view the output of a job at any time if you selected the **Append** or **Replace** option for job output prior to running the job.

**To view job output:**

1. Go to the **Job Activity** pane.
2. To display the **Job Detail** dialog, either double-click the **Report** job or right-click the **Report** job and select **Details** from the context menu.
3. Click the **Output** tab. If you do not select the **Save Output** option before running a job, you will see a Tidal Workload Automation message rather than actual job output.



4. Click **OK**.





# 4

## Using TWA Calendars

### Overview

In the previous exercise, you created an unscheduled job. In this chapter you will create a scheduled job by creating a calendar, and adding it to the job. Scheduled jobs run on a regular basis according to the dates in their calendar. Some schedules are simple with jobs running every day from Monday through Friday, or on the first day of every month. Others are more involved, with jobs running on irregular accounting periods, holidays or fiscal period offsets.

Tidal Workload Automation uses calendars to define the set of dates on which jobs run. Tidal Workload Automation contains five different methods for creating calendars, depending on the period in which you need your job to repeat. For example, you may need your job to run as needed on a, daily, weekly or monthly basis. Each method addresses a particular period.

This chapter describes how to build four calendars:

- **first wkdy** – A list calendar (as needed) representing the first workday of each week.
- **work day** – A daily calendar representing Monday through Friday.
- **Fiscal Week End** – A calendar to use when your work week ends on a Thursday.
- **Fiscal Quarter End** – A calendar for scheduling on the last business day of a quarter.

You will then schedule a job using the **work day** calendar, and confirm that it is scheduled properly.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

### Building the First Workday Calendar

In this exercise, you will build a list calendar called **first wkdy** which refers to the first workday in every week. In the **List Calendar** dialog, you choose each day that the job is to be scheduled.

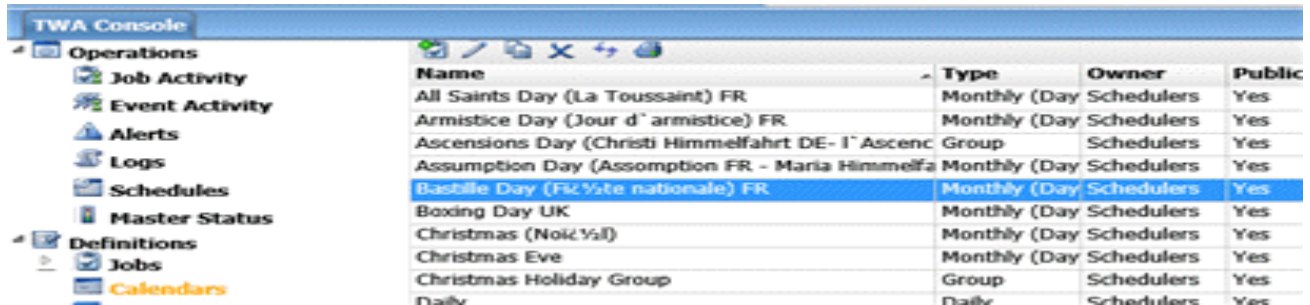
#### To build a List calendar:

1. From the **Navigator** pane, select **Definitions>Calendars** to display the **Calendars** pane.
2. Click the **Add** button on the TWA toolbar or right-click a calendar and select **Add Calendar>List** from the context menu to display the **List Calendar Definition** dialog.
3. In the **Calendar Name** field, type **first wkdy**.
4. In the Calendars view, double-click the dates that represent Monday of each week for the current month.

All selected dates appear under the **Selected Dates** section to the right of the calendar.

**Note:** To add scheduling dates to other months, use the inner right arrow buttons. To add scheduling dates to other years, use the outer right arrow buttons.

- Click **OK** after you have completed your selections.



Name	Type	Owner	Public
All Saints Day (La Toussaint) FR	Monthly (Day	Schedulers	Yes
Armistice Day (Jour d'armistice) FR	Monthly (Day	Schedulers	Yes
Ascensions Day (Christi Himmelfahrt DE- I" Ascenc	Group	Schedulers	Yes
Assumption Day (Assomption FR - Maria Himmelfa	Monthly (Day	Schedulers	Yes
<b>Bastille Day (Fête nationale) FR</b>	<b>Monthly (Day</b>	<b>Schedulers</b>	<b>Yes</b>
Boxing Day UK	Monthly (Day	Schedulers	Yes
Christmas (Noël) FR	Monthly (Day	Schedulers	Yes
Christmas Eve	Monthly (Day	Schedulers	Yes
Christmas Holiday Group	Group	Schedulers	Yes
...	...	...	...

Congratulations, you just created your first list calendar!

**Note:** You can create a calendar for any dates you specify. This calendar can be associated with a job from within the **Job Definition** dialog, but will not be used in any future tutorial exercises.

## Building the Work Day Calendar

Now you will create the **work day** daily calendar to use for scheduling a job. The **work day** daily calendar schedules your job to run on each weekday.

### To build a Work Day calendar:

- From the **Navigators** pane, select **Definitions>Calendars** to display the **Calendars** pane.
- Click the **Add** button on the TWA toolbar or right-click a calendar and select **Add Calendar>Daily** from the context menu to display the **List Calendar Definition** dialog.
- In the **Calendar Name** field, type **Work Day**.

Notice the **Public** option in the lower left corner of the dialog. Selecting this option makes the calendar available to all Tidal Workload Automation users.

We will accept the defaults in the **Details** tab. The defaults automatically set the calendar to generate dates for Monday through Friday of each week.

- Click the **Forecast** tab to generate the calendar.

This tab displays all the dates calculated from the **Details** tab. The **Forecast** tab is viewed the same way as in the **List Calendar** dialog, when the calendar **first wkday** was entered. You can add individual dates from this tab by double-clicking in the calendar view on the dates you want to change. You can delete dates by selecting the date in **Selected Dates** and either right-clicking the date to select **Delete** or clicking the **Delete** button. The forecast displays a four-year time window, but the calendar extends indefinitely.

**Note:** This step can take from a few seconds to a minute, depending on your system's configuration.

- Click **OK**.

The Calendars view displays the two calendars you created. By default, the rows of the display are sorted alphabetically according to calendar name (the **Name** column). You can also sort by other columns by clicking the column name. For example, if you click the **Owner** name at the top of the **Owner** column, calendars are grouped by their creator. If you click the **Modified** label at the top of the **Modified** column, your newest calendars are grouped together.

## Building the Fiscal Week End Calendar

We will create the **Fiscal Week End** calendar to use for scheduling a job to run when your fiscal week ends on Thursday evening. If Thursday is a holiday, the job will execute on Wednesday evening, through the use of a condition definition.

### To build a Fiscal Week End calendar:

1. From the **Activities** menu, select the **System Configuration** option.
2. In the **Week Begins** list, select **Friday**. This sets your fiscal week to start on Friday and end on Thursday.
3. Click **OK**.
4. Click **OK** when the **Warning** dialog displays. This warning displays whenever you change the calendar week starting day.

**Note:** The **Recalculate Calendars** option, available from the **Calendars** pane context menu, will change all calendars to begin on the new Week Begins day. If you do not recalculate calendars, existing calendars are not changed; only those calendars created selecting the new Week Begins day are affected.

5. From the **Navigator** pane, select **Definitions>Calendars** to display the **Calendars** pane.
6. Click the **Add** button on the TWA toolbar or right-click a calendar and select **Add Calendar>Subset** from the context menu to display the **Calendar Subset Definition** dialog.
7. In the **Calendar Name** field, enter **Fiscal Week End**.
8. In the **Occurrence** list, select **Every**.
9. In the **Calendar** list, select **Friday**.
10. In the **Time Frame** list, select **Week**.
11. (Optional) Multiple fiscal calendars can be defined for Tidal Workload Automation. You can choose to base a calendar definition on a fiscal year, rather than a standard January- through-December calendar year.

Making no selection in the **Based on Fiscal** list results in the use of a standard calendar. For this exercise, leave the field empty.

12. Now, we need to set a condition to let Tidal Workload Automation know what to do if a holiday occurs on Thursday. Click the **Conditions** tab.
13. Click **Add** to display the **Condition** dialog.
14. From the **If conflicts with calendar** list, select the **Holidays** calendar.
15. From the **Take action** list, select **Previous Calendar**.
16. The **Calendar** field defines which calendar to use and how many days in the past to go. The default is **0** days. Select a value of **1**, and in the adjacent list, select **Workdays**.
17. Click **OK**.

The condition now displays in the **Conditions** view.

Now if a holiday falls on Friday of the **Fiscal Week End**, Tidal Workload Automation will schedule the **Fiscal Week End** calendar for Wednesday.

18. Click the **Forecast** tab to verify the dates Tidal Workload Automation has set.
19. Click **OK**. The **Fiscal Week End** calendar now appears in the **Calendars** view.

## Alternate Method of Creating a Calendar

In most cases, there is more than one way to define a calendar such as our **Fiscal Week End**. The following is an example of another way to create the **Fiscal Week End** calendar. We will call this calendar **Fiscal Week Ending**, so that the results of both exercises will be visible.

### To create a calendar:

1. From the **Navigator** pane, select **Definitions>Calendars** to display the **Calendars** pane.
2. Click the **Add** button on the TWA toolbar or right-click a calendar and select **Add Calendar>Weekly** from the context menu to display the **Weekly Calendar Definition** dialog.
3. In the **Calendar Name** field, enter **Fiscal Week Ending**.
4. Click the **Details** tab. In the **Include the following days** section, de-select **Monday, Tuesday, Wednesday** and **Thursday** options.
5. Click the **Conditions** tab, and then click **Add**.
6. In the **If conflicts with calendar** list, select the **Holidays US** calendar.
7. In the **Take action** list, select **Previous Calendar**.
8. The **Calendar** field defines which calendar to use and how many days in the past to go. The default is **0** days. Select a value of **1**, and in the adjacent list, select **Workdays**.
9. Click **OK**.
10. Click **Forecast** to verify the dates that Tidal Workload Automation has chosen.
11. Click **OK** in the **Weekly Calendar Definition** dialog.

Now, both the **Fiscal Week End** and the **Fiscal Week Ending** calendars are visible in the **Calendars** view.

## Building the Quarter End Calendars

We will create the **Fiscal Quarter End** and **Calendar Quarter End** calendars to use for scheduling jobs to run when your fiscal quarter ends on the last work day of the quarter. If the quarter ends on a holiday, the calendar should execute on the previous work day. For this exercise, we will consider the fiscal year to run from November to October.

### Fiscal Quarter End

We first have to define a fiscal calendar period, to be used in documenting the **Fiscal Quarter End** calendar. Many companies run production from more than one fiscal calendar period.

#### To define a Fiscal calendar period:

1. From the **Navigator** pane, select **Definitions>Fiscal Calendars** to display the **Fiscal Calendars** pane.
2. Click the **Add** button on the TWA toolbar or right-click a calendar and select **Add Fiscal Calendar** from the context menu to display the **Fiscal Calendar Definition** dialog.

Fiscal Calendar Definition (Create Mode)

Fiscal Calendar Name: Corporate Fiscal Year

Fixed Start: ☒

Starting Month: November

Starting Day: 1

Quarter Breakdown: 4-4-5 (Weeks in each month of quarter)

OK Cancel

363191

3. In the **Fiscal Calendar Name** field, enter **Corporate Fiscal Year**.
4. In the **Starting Month** list, select **November**. Keep the default values for the remaining text fields.
5. Click **OK**.
6. From the **Navigator** pane, select **Definitions>Calendars** to display the **Calendars** pane.
7. Click the **Add** button on the TWA toolbar or right-click a calendar and select **Add Calendar>Subset** from the context menu to display the **Calendar Subset Definition** dialog.

Calendar Subset Definition (Create Mode)

Calendar Name: Fiscal Quarter End

Owner: Schedulers

Starting On: 1/1/09

Details Conditions Forecast (through) Description

Occurrence: Last

Calendar: Workdays US

Time Frame: Quarter

Based on Fiscal: Corporate Fiscal Year

Adjust by: 0 days

☒ Public

OK Cancel

363192

8. In the **Calendar Name** field, enter **Fiscal Quarter End**.
9. If not already selected, click the **Details** tab.
10. From the **Occurrence** list, select **Last**.
11. From the **Calendar** list, select **Workdays US**.
12. From the **Time Frame** list, select **Quarter**.
13. In the **Based on Fiscal** list, select **Corporate Fiscal Year**. Leave the **Adjust by ... days** field at **0**.
14. We have completed the definition for the **Fiscal Quarter End** calendar. Click the **Forecast** tab to verify the dates that will be selected for production execution and then click **OK**.

The calendar definitions are ready to be used in production.

## Calendar Quarter End

To create a quarter end calendar that will work on the last business day of the calendar quarter, where the fiscal year starts in January, just repeat Steps 6 through 12 above with the following exceptions:

- In **Step 8**, change the Calendar Name to **Calendar Quarter End**.
- In **Step 12**, leave the **Based on Fiscal** field blank.

You have created two quarterly calendars, one based on a fiscal year calendar and one based on a standard calendar year starting in January.

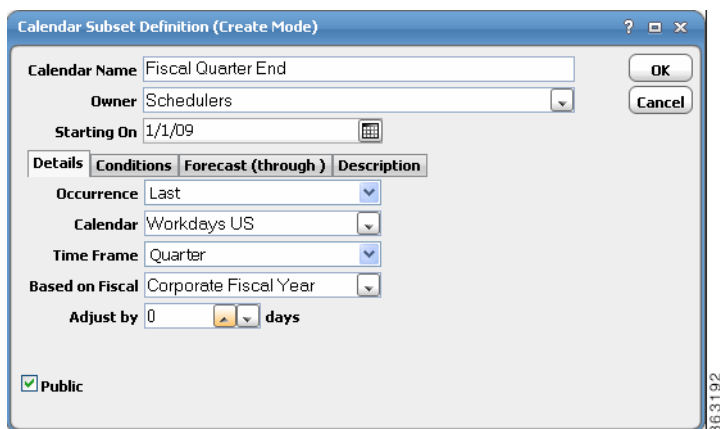
## Scheduling the Report Job with a Calendar

Now, return to the **Report** job and schedule it using the **work day** calendar.

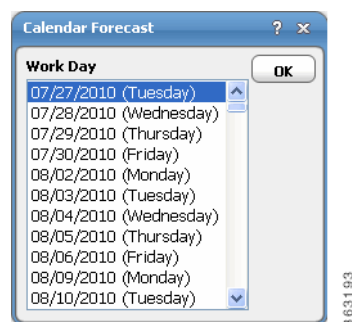
**To schedule the Report job using the word day calendar:**

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click the **Report** job and select **Edit Job/Group** from the context menu to display the **Job Definition** dialog.
3. Click the **Schedule** tab on the **Job Definition** dialog.
4. In the **Calendar Name** list, select **Work Day**.
5. To view the dates in the calendar, click the **Forecast** button and then click **OK**.

TWA displays the **Calendar Forecast** dialog.



6. Click **OK** again on the **Job Definition** dialog to display the **Effective Date** dialog.



The **Effective Date** dialog displays asking you when the job should enter the production schedule. The dates represent the current span of the production schedule.

7. Click **OK** to accept the default (today).

Another occurrence of the **Report** job enters the production schedule.

8. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.

The job runs according to the **work day** calendar. It will run today (unless you are running this tutorial on a weekend). Your window may appear slightly different.

You will also see the previous occurrence of the **Report** job that was added to the schedule manually in the previous chapter. The job occurrence has been renamed **Report(2)** to reflect the first occurrence of the same job (Report) that ran earlier.





## 5

## Setting Up a Job Dependency

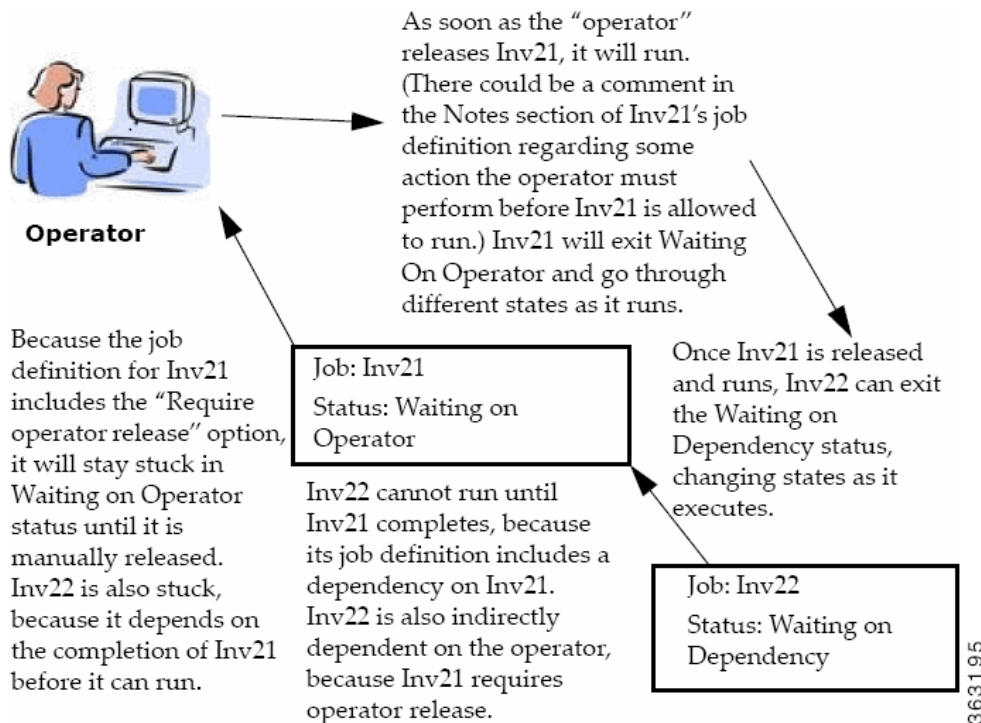
## Overview

In many cases, a job should run only after some external requirements are satisfied. For example, a job may need data generated by another job, or it can run only after another job has completed successfully.

Tidal Workload Automation has the flexibility to encompass the different scheduling needs encountered in a business environment. You use job dependencies to prevent a job from running until the preceding job completes or enters a predefined status. You can also set jobs to run only when manually released.

In this chapter, we are going to define (add) a job called **Inv21**. We will assign a calendar to this job, but we will also make it **Require operator release**. Even if according to its calendar, the job is due to run, it does not run until it is released from **Waiting on Operator** status. We will define a second job, **Inv22**, which depends on **Inv21**. **Inv22** does not run until **Inv21** completes normally. Finally, we release **Inv21**, and both jobs end with a **Completed Normally** status.

## Interdependence of Inv21, Inv22, and the Operator



This chapter describes how to:

- Define a job that waits for an operator to release it
- Define a job that depends on the completion of another job

- Monitor jobs and job dependencies
- Release a job that requires operator intervention

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

## Defining a Job for Operator Release

You can define a job requiring an operator to release it manually before running, after all its other dependencies are satisfied. Before the job is released, it enters the **Waiting on Operator** status. The job definition can include instructions for the operator.

Requiring an operator release is a good technique when testing a job or a set of sequential jobs for the first time. This is also good for jobs requiring operators to verify an event, or perform a related operation external to Tidal Workload Automation.

## Setting the Require Operator Release Option

**To set the Require Operator Release option:**

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click in the **Jobs** pane of the console and select **Add Job** from the context menu, or click the **Add Job** button on the TWA toolbar.

The **Job Definition** dialog displays.

3. Set the following options:
  - a. In the **Job Name** field, enter the name **Inv21**.
  - b. In the **Command** field, enter the full path to the Inv21 executable for your operating system:

### Windows

`C:\Program Files\TIDAL\Agent\TUTORIAL\Inv21.bat.`

- or -

`C:\Program Files (x86)\TIDAL\Agent\TUTORIAL\Inv21.bat`, if your agent was installed on 64-bit Windows machine, using default installation path.

**Note:** Although the command executable has the same name as the job in this case, this is not required. The Job Name is any logical name that is meaningful to you. However, the command name refers to the physical file and must match the actual command.

4. Click the **Schedule** tab.
5. In the **Calendar Name** list and select the **work day** calendar.
6. Click the **Run** tab, and in the **Runtime Users** list, select a runtime user with access to the Unix agent.
7. Click the **Options** tab.

8. Select the **Require operator release** option.
  9. Click **OK** to save the job definition.
  10. Click **OK** in the **Effective Date** dialog to submit the job into today's schedule.
  11. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.
- Inv21** displays a **Waiting On Operator** status.

We will leave this job for now, and release it to the Production Schedule later after creating a second job with a dependency on **Inv21** completing normally.

## Adding a Job with a Dependency

We will now create the **Inv22** job definition with a dependency on the **Inv21** job already in the production schedule.

### To create the **Inv22** job definition with a dependency on the **Inv21** job:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
  2. Right-click in the **Jobs** pane of the console and select **Add Job** from the context menu, or click the **Add Job** button on the TWA toolbar.
- The **Job Definition** dialog displays.
3. Set the following options:
    - a. In the **Job Name** field, type the name **Inv22**.
    - b. In the **Command** field, enter the full path to *Inv22.bat* (Windows).
  4. Click the **Schedule** tab.
  5. In the **Calendar Name** list, select the **work day** calendar.
  6. Click the **Run** tab, and in the **Runtime Users** list, select a runtime user with access to the Unix agent.
  7. Display the **Dependencies** tab, by clicking its tab on the **Job Definition** dialog.
  8. Click the **Add** button to display the list.
  9. Select the **Add Job Dependency** option to display the **Job Dependency Definition** dialog.
  10. Go to the **Job/Group** list, click the down arrow button and select **Inv21**. It will appear in the **Job/Group** field.
  11. Accept the defaults and click **OK**. The defaults are preset to satisfy the dependency when **Inv21** completes normally.

Now **Inv21** shows as a dependency for **Inv22**. The **Inv21** dependency is part of the job definition of **Inv22**.

12. Click **OK** in the **Job Definition** dialog to close it.
13. Click **OK** in the **Effective Date** dialog.

The **Inv22** job is added to the production schedule.

## Monitoring Your Jobs

Both jobs are now scheduled and ready to run. Use the **Job Activity** pane to monitor them.

From the **Navigator** pane, select **Operations>Job Activity** to display your scheduled jobs.



Job No.	Name	Status
120	Dhivya (1)	Agent Disabled
576	Group 1 (1)	Timed Out
588	Group 1 (1)	Waiting On Children
591	Parent Group 12 (1)	Waiting On Children
592	Copy of Parent Group 1	Waiting On Children

Notice that both jobs are in waiting mode. **Inv21** needs manual intervention before it can run (Waiting On Operator) and **Inv22** is waiting for **Inv21** to complete normally (Waiting On Dependencies).

## Viewing Dependencies from the Job Activity Pane

To view the dependence **Inv22** has on **Inv21**:

1. Double-click the **Inv22** job in the **Job Activity** pane to display its **Job Detail** dialog.
2. Click the **Dependencies** tab.

The dependency appears in the **Predecessors** field.

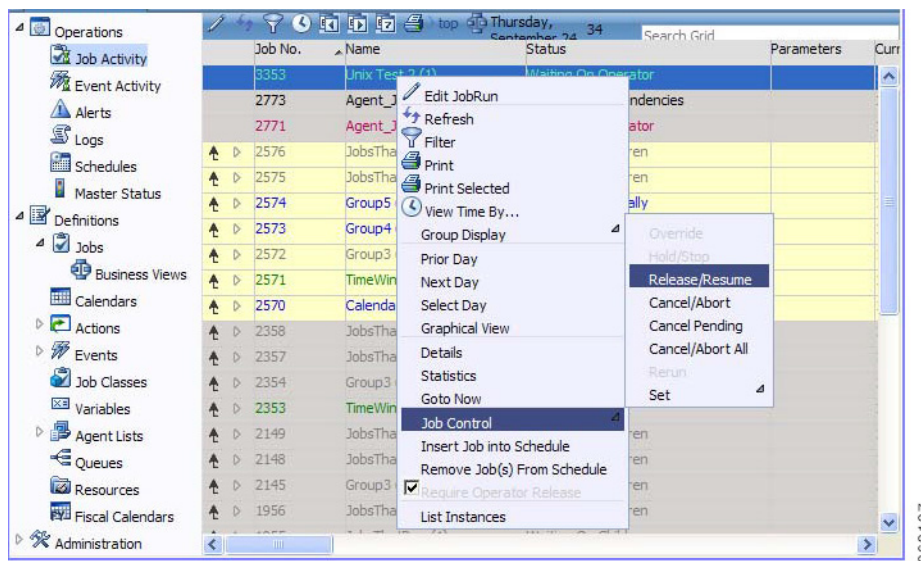
This shows that **Inv22** is waiting for **Inv21** to complete normally. Currently **Inv21** is in **Waiting on Operator** status. **Inv22** cannot run until **Inv21** goes to **Completed Normally** status.

3. Click **OK**.

## Releasing the Jobs

To release the jobs:

1. Click the **Inv21** job in the **Job Activity** pane.
2. Right-click the job to display the context menu.
3. Select the **Job Control** option and then choose the **Release/Resume** option to release the job.



The **Confirm** dialog displays.

4. Click **Yes**.

Now **Inv21** is free to run. Once complete, **Inv21** enters a **Completed Normally** status. The color of **Inv21** changes to green and then to blue (if using the default status colors), and then **Inv22** runs. **Inv22** displays similar status and color changes.

5. Click **OK** when the **Information** dialog displays.

Congratulations! In this chapter, you have defined jobs requiring manual release and a job dependency. You have scheduled and monitored your jobs through the various states until completion.



## 6

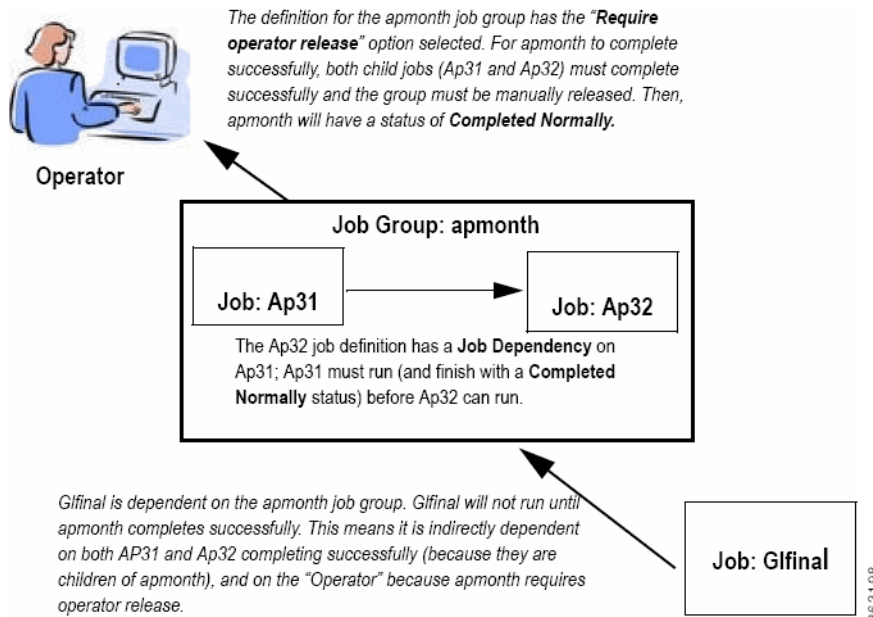
## Scheduling Jobs in Groups

## Overview

Tidal Workload Automation lets you organize your jobs into job groups to make it easier to manage and control jobs, and to make it more efficient to set job attributes. Jobs can inherit characteristics such as dependencies and calendars from their associated job group, eliminating the need to specify common job details for each job.

When a job group is added to the schedule manually, all of its member jobs (called child jobs) are added as well, saving you additional work and reducing the potential for mistakes. A job group can be configured the way a single job can be to depend on another job or job group.

In this exercise, we will create a job group called **apmonth**. **apmonth** will consist of two jobs, **Ap31** and **Ap32**. We will make the **Ap32** job dependent on the **Ap31** job completing successfully. Once the **apmonth** job group has completed normally, **Glfinal** will run.

Interdependency of **Glfinal**, **apmonth**, **Ap31** and **Ap32**

This chapter describes how to:

- Define a job group.
- Create jobs that belong to the group.
- Schedule the job group.
- Create a job that depends on a group.

- Release the job group.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

## Job Group Hierarchy

A job group is represented with a colored background. All of a group's child jobs are indented underneath the group name. Each job group can include jobs and other job groups, which in turn can include other jobs and job groups. The window helps you visualize and navigate through the hierarchy of job groups. When you click the symbol to the left of a job group name, the display expands to show all of the jobs and job groups within the selected job group.

## Job Group Scheduling

In this chapter we will develop a set of jobs under a job group, then schedule the jobs by submitting the job group only. We will do this by initially setting up the job group and its jobs as unscheduled (without an assigned calendar). This will prevent the jobs from entering the schedule before all the definitions are complete. We will then give the job group a calendar. We will set the child jobs to inherit the calendar so that all the jobs in the job group will be scheduled in one step.

## Adding a Job Group

### To add a job group:

1. From the Navigator pane, select Definitions>Jobs to display the Jobs pane.
2. Right-click in the Jobs pane of the console and select Add Job Group from the context menu.

The Job Group Definition dialog displays.



The Job Group Definition dialog is similar, but not the same, as the Job Definition dialog. The Job Group Definition dialog does not have a field for specifying the command to execute, because a job group is a container for jobs.

3. In the **Job Group Name** field, enter **apmonth**.
4. Click the **Run** tab and from the **Runtime Users** list, select a user that can access the Unix agent. This runtime user will now be inherited by default by all of the child jobs within the job group. If needed you can always change the default runtime user in individual job definitions.
5. Click the **Options** tab, and select the **Require operator release** option. This causes the group to wait for a manual release before it will launch.
6. Click **OK**.

A **Confirm** dialog displays.

7. Click **Yes**. The **apmonth** job group is added to the **Jobs** pane.

## Creating Jobs that Belong to a Job Group

Now, you will add child jobs within the job group that you just created. Because you are adding jobs within the job group, inheritable properties will be set for the jobs from the job group, such as the agent that the jobs will run on.

### Creating the Ap31 Child Job

**To create the Ap31 child job belonging to the apmonth job group:**

1. From the Navigator pane, select Definitions>Jobs to display the **Jobs** pane.
2. Select the apmonth job group.
3. Click the Add Job button on the Tidal Workload Automation toolbar or right-click the **Jobs** pane and select Add Job from the context menu to display the **Job Definition** dialog.
4. Enter Ap31 in the **Job Name** field.

Notice that the apmonth job group is displayed in the Parent Group field. You can also specify apmonth as the job group by selecting \apmonth from the Parent Group list.

5. In the Command field, enter the full path to the file for your platform:


#### Windows



Ap31.bat

6. Click **OK**.

A Confirm dialog displays.

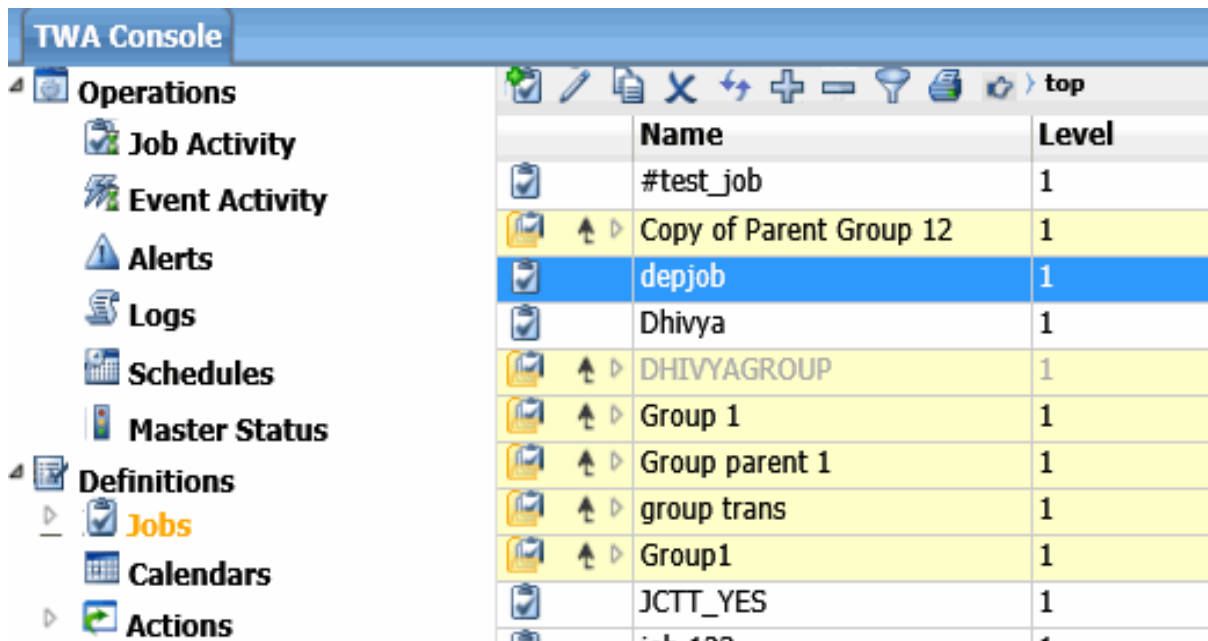
7. Click **Yes**.

Now, when the **apmonth** job group is displayed in the jobs window, it will have a  icon to the left of it. This icon indicates there are jobs and/or more job groups underneath it that are not currently visible. You can:

- Click the  icon to expand the selected job group, showing the child jobs and/or job groups that belong to it.
- Click the  icon again to collapse the job group.

Jobs do not have the  icon next to them because a job is singular and cannot have child jobs.

**Note:** Ap31 is identified as a Level 2 job, meaning it is nested on the next level below apmonth. The indentation of the job name also serves as a visual cue of the hierarchy. Apmonth is at Level 1 (or root) level.



	Name	Level
	#test_job	1
	Copy of Parent Group 12	1
	depjob	1
	Dhivya	1
	DHIVYAGROUP	1
	Group 1	1
	Group parent 1	1
	group trans	1
	Group1	1
	JCTT_YES	1

All of the jobs and job groups within **apmonth** will be listed beneath it. If they are not, expand the **apmonth** listing by clicking the icon next to it.

## Creating the Ap32 Child Job with a Dependency

Now, we will create the **Ap32** child job that depends on the **Ap31** job. This job also belongs to the **apmonth** job group.

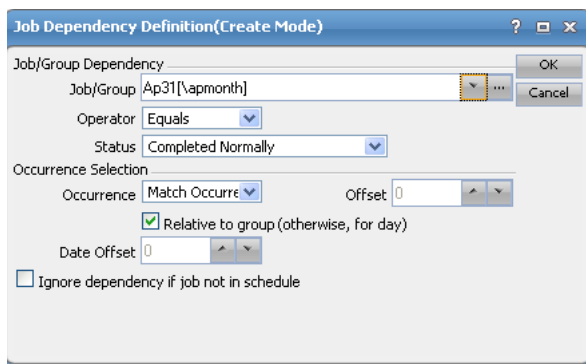
### To create the Ap32 child job with a dependency:

1. From the Navigator pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Select the **apmonth** job group.
3. Click the **Add Job** button on the Tidal Workload Automation toolbar or right-click the **Jobs** pane and select **Add Job** from the context menu to display the Job Definition dialog.
4. In the **Job Name** field, enter the job name Ap32. The apmonth job group should be displayed in the Parent Group field. If it does not appear, select \apmonth from the Parent Group list.
5. In the Command field, enter the full path to the file for your platform:

#### Windows

Ap32.bat

6. Click the **Dependencies** tab.
7. Click **Add** and select **Add Job Dependency** to display the **Job Dependency Definition** dialog.
8. Click the down-arrow button in the **Job/Group** list and select the **Ap31** job.



9. Click **OK**.

Ap31 is now displayed on the Dependencies tab of the Job Definition dialog. Notice that the Type of dependency is identified as JOB.

10. Click **OK**.

The Confirm dialog displays, reminding you that a calendar has not been specified.

11. Click **Yes** to accept the job definition without assigning it a calendar.

The **Ap32** job appears along with **Ap31** in the hierarchy under the apmonth job group.

## Scheduling the Job Group

We have created two child jobs under the job group called **apmonth**, but no jobs have been scheduled. We will now assign a calendar to the job group, and when closed, the job group and all its child jobs will enter the schedule to run based on the calendar.

### To schedule the job group:

1. Double-click the **apmonth** job group to open its definition.

2. In the **Calendar Name** list, select the **Daily** calendar.

Since child jobs default to inheriting the parent group's attributes, by setting this option for the group, you are effectively assigning a **Daily** calendar to each child job.

3. Click **OK**. The **Effective Date** dialog appears.

4. Accept the defaults and click **OK**.

The job group and jobs are submitted and displayed in the **Job Activity** pane.

## Scheduling a Job that Depends on a Group

One of the benefits of using job groups is that you can work with multiple jobs as though they were one big job. You can schedule jobs that depend on the status of the job group. You do not have to determine the order in which the jobs will complete, or set a dependency on each individual job.

The status of the job group is determined by its child jobs. For example, a job group cannot reach completed status until every job in the job group has completed.

### To schedule a job that depends on a group:

1. From the Navigator pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Verify that no job group (such as **apmonth**) is highlighted, click the **Add** button on the Tidal Workload Automation toolbar or right-click in the **Jobs** pane and select **Add Job** from the context menu.

The **Job Definition** dialog displays.

3. In the **Job Name** field, enter **Glfinal**.
4. In the Command field, enter the full path to the file for your platform:

#### Windows

Glfinal.bat

5. Click the **Schedule** tab.
6. Select the **Daily** calendar from the **Calendar Name** list.
7. Click the **Run** tab and from the **Runtime Users** list, select a user that can access the Unix agent. (Since this job is not part of a job group it does not inherit the agent it will run on.)

## Adding a Job Group Dependency

Now, you can add the dependency that causes job **Glfinal** to wait until the **apmonth** job group completes normally.

### To add a job group dependency:

1. Click the **Dependencies** tab in the **Job Definition** dialog.
2. Click **Add**, and select **Add Job Dependency** to display the **Job Dependency** dialog.
3. Go to the **Job/Group** field, and click the down-arrow button.
4. Select the **apmonth** job group.

The dependency will be met when the **apmonth** job group reaches **Completed Normally** status. Each job in the group must complete normally for **apmonth** to complete normally.

5. Click **OK** to return to the **Job Definition** dialog.

Notice that, to the right of **apmonth**, the **Type** of dependency is identified as **JOB**.

6. Click **OK**, then click **OK** again in the **Effective Date** dialog to accept the current production schedule.

The **Glfinal** job now displays in the hierarchy.

## Releasing a Job Group

### To release a job group:

1. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.

The **apmonth** job group and the **Glfinal** job are now in the production schedule. **apmonth** is **Waiting on Operator**, and the **Glfinal** job is waiting for the normal completion of all the jobs in the **apmonth** group.

2. Highlight and then right-click the **apmonth** job group to display the context menu.
3. Select the **Job Control** option, then select the **Release/Resume** option to release the job group.

The **Confirm** dialog displays.

4. Click **Yes**.

The **Information** dialog displays.

5. Click **OK** in the **Information** dialog.

**apmonth** will enter **Active** status. Then, the **Ap31** job runs, and **Ap32** follows after **Ap31** completes normally. When **Ap32** completes, the **apmonth** job group shows a **Completed Normally** status. Then the **Glfinal** job runs. The total process should take a few minutes.



## 7

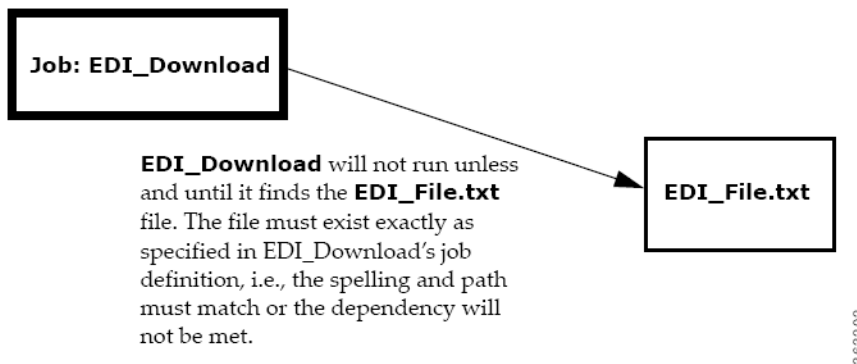
# Creating a File Dependency

## Overview

Job dependencies mean that jobs depend on the status (such as **Completed Normally**) of other jobs to run. Using file dependencies, you can have jobs depend on the availability and status of files as a condition of execution. File dependencies are set up using the **File Dependency** dialog which is accessible when you create or edit a job definition.

In this chapter, we will create a job (**EDI\_Download**) that is dependent on the existence of the *EDI\_File.txt* file.

### EDI\_Download Job Depends on the Existence of the EDI\_File File



To properly complete this exercise, you need to have a solid understanding of the following concepts:

- How to add and create a new job definition.
- How to navigate to and between the **Jobs** pane and the **Job Activity** pane.

This chapter describes how to:

- Create a job with a file dependency.
- Create a file to satisfy the dependency.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

## Creating a Job with a File Dependency

### To create a job with a File dependency:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click in the **Jobs** pane of the console and select **Add Job** from the context menu, or click the **Add Job** button on the toolbar.

The **Job Definition** dialog displays.

3. In the **Job Name** field, type **EDI\_Download**.
4. On the **Program** tab, enter the full path to the file for your platform in the **Command** field:

#### Windows

Edi.bat

5. Click the **Schedule** tab and from the **Calendar Name** list select the **work day** calendar.
6. Click the **Dependencies** tab.
7. Display the **File Dependency** dialog by clicking the **Add** button and selecting **Add File Dependency** field. Follow the steps below for your platform:

#### Windows

- a. Enter **C:\Program Files\<Installation Directory>\Agent\Tutorial\Edi\_File.txt** in the **File Name** field. The full path to this file must be typed in because the file does not yet exist. We will create *Edi\_File.txt* later, in the section [Creating a File to Satisfy the Dependency, page 57](#).
- b. The **Agent Name** field should already have a Windows agent listed—the agent you selected as your default agent at the beginning of this tutorial. If there is no agent listed or an agent other than the Windows agent is selected, then select your local Windows agent from the **Agent Name** list.

#### UNIX

- a. Enter **/Edi\_File.txt** in the **File Name** field. The full path to this file must be typed in because the file does not yet exist. We will create *Edi\_File.txt* later, in the section [Creating a File to Satisfy the Dependency, page 57](#).
- b. The **Agent Name** field should already have a Unix agent listed—the agent you selected as your default agent at the beginning of this tutorial. If there is no agent listed or an agent other than the Unix agent is selected, then select your local Unix agent from the **Agent Name** list.

**Note:** The *Edi\_File.txt* file must reside in the **/** (root level) of the machine where the Unix agent is installed so this tutorial works properly. Click **OK** to accept the default settings. Tidal Workload Automation's default is that the dependency is met after the file is created.

8. Click **OK** to accept the default settings. Tidal Workload Automation's default is that the dependency is met after the file is created.
9. Click the **Dependencies** tab.

The full path to *Edi\_File.txt* appears in the **Dependency Name** column. The **Type** of dependency is listed as **FILE** to indicate that it is a file dependency.

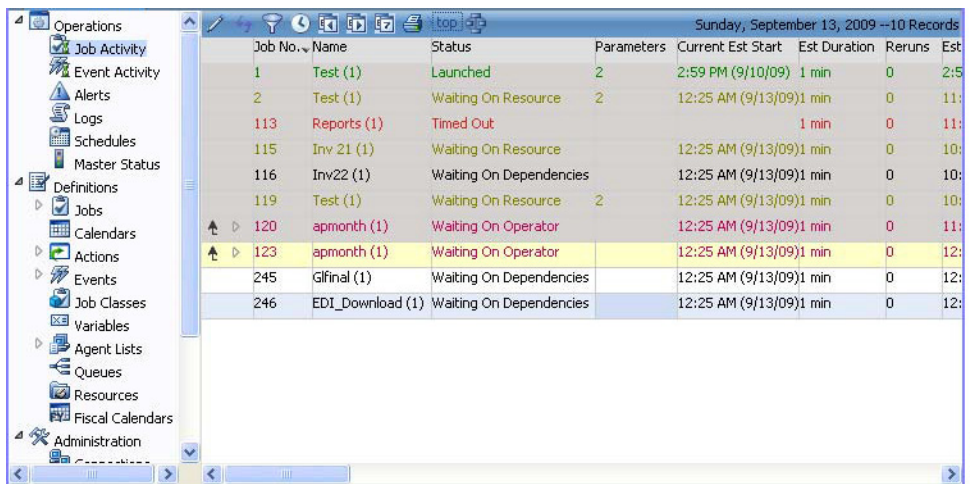
10. Click **OK** to save the job definition for **EDI\_Download**.
11. Click **OK** in the **Effective Date** dialog to add the job to the schedule.



## Viewing the Job in Production

To view the job in production:

1. Open the **Job Activity** pane.
2. The **EDI\_Download** job first appears with a **Waiting on Dependencies** status. It goes through different stages and ends in a **Scheduled** status.



Job No.	Name	Status	Parameters	Current Est Start	Est Duration	Reruns	Est
1	Test (1)	Launched	2	2:59 PM (9/10/09)	1 min	0	2:5
2	Test (1)	Waiting On Resource	2	12:25 AM (9/13/09)	1 min	0	11:
113	Reports (1)	Timed Out			1 min	0	11:
115	Inv 21 (1)	Waiting On Resource		12:25 AM (9/13/09)	1 min	0	10:
116	Inv22 (1)	Waiting On Dependencies		12:25 AM (9/13/09)	1 min	0	10:
119	Test (1)	Waiting On Resource	2	12:25 AM (9/13/09)	1 min	0	10:
120	apmonth (1)	Waiting On Operator		12:25 AM (9/13/09)	1 min	0	11:
123	apmonth (1)	Waiting On Operator		12:25 AM (9/13/09)	1 min	0	12:
245	Glfinal (1)	Waiting On Dependencies		12:25 AM (9/13/09)	1 min	0	12:
246	EDI_Download (1)	Waiting On Dependencies		12:25 AM (9/13/09)	1 min	0	12:

3. Select the **EDI\_Download** job occurrence.
4. Right-click the **EDI\_Download** job and select **Details** from the context menu to see its **Job Detail** dialog.
5. Click the **Dependencies** tab to view the file dependency status.

The **Job Detail** dialog displays the file dependency for this job.

6. Click **OK** to close the dialog.

## Creating a File to Satisfy the Dependency

In this example, the dependency only requires that the file exists and does not require any specific data. For this exercise, we just create an empty file since the job does not require any data from the file. Normally, however, this kind of dependency is set up because the job uses the data from the file's input.

To create a file to satisfy the dependency:

1. Open a new document in a text editor or word processing program for your platform, such as Notepad or Microsoft Word (Windows) or vi, joe, or pico (UNIX).
2. Save the file as **EDI\_File.txt** in the location that you specified when defining the file dependency for your platform:

### Windows

C:\Program Files\<Installation Directory>\Agent\Tutorial EDI\_File.txt

Substitute the name of the directory where TWA is installed for **<Installation Directory>**.

### UNIX

/Edi\_File.sh

File names are case-sensitive in Unix so even if a file has the same spelling, if some of the letters in the name are not the same case, the name is not recognized.

For your file to satisfy the dependency, the name must match exactly. Saving as a plain text file helps ensure the correct file name.

3. Double-check that the file exists, is correctly named and in the correct directory.

## Monitoring the Job in Production

### To monitor jobs:

1. Open the **Job Activity** pane.

You can see that the file dependency for the **EDI\_Download** job is met and the job goes from a **Waiting on Dependencies** state into an **Active** state.

Congratulations! In this chapter, you have created a job with a file dependency, and fulfilled that dependency requirement by creating the file. By this time, you should also feel comfortable with moving between the **Job Activity** pane and **Jobs** pane, and with the basic procedure for adding a job.

# 8

## Defining Actions Based on Job Conditions

### Overview

Using Tidal Workload Automation job events allows you to automatically trigger actions whenever certain event triggers (job conditions) arise. Actions include user notification via email, console alert messages, launching new jobs, job control commands, variable updates, log message posting and SNMP traps to HP OpenView or other network managers. This chapter describes how to:

- Define an operator alert action.
- Add the action to the job by using the **Job Events** dialog.
- Submit the job to production and watch the operator alert action occur.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

### Understanding Actions

An action is a predefined response to a job event that is triggered by a job in the production schedule. You define event triggers for jobs in the **Job Events** dialog and associate them to actions. Actions can be linked to multiple jobs and job groups.

There are different types of actions that can be defined within TWA . Each type of action can be triggered based on any job's status, job control by an operator and/or an event triggered by the system. Furthermore, you can personalize actions from a set of included variables such as system, job, job event, action and user-defined information.

- **Email** – An email action is a standard format email message that issues each time the associated job event is triggered. Email can be sent to any user on your network.
- **Alert** – An alert action is an alert sent to the Job Activity Console, with one of four separate levels of severity: *Critical*, *Error*, *Warning* and *Information*. This is the type of action used in this exercise
- **SNMP** – An SNMP message action can be sent to any SNMP-compliant manager.
- **SAP Event** (SAP only) – Notifies an SAP instance that a specified event occurred in TWA . Can be used to trigger an event-based job in SAP.
- **Job** – TWA can initiate automated job recovery procedures with a job action. The new job action also allows you to override certain command parameter values at submission time.

- **Variable** – This variable update action updates user-defined variables. Since jobs can be dependent on variable values, this action can be used to start a job based on a wide selection of events experienced by another job which sets the variable.
- **Log** – The log message action posts a user-defined message to the TWA logging facility and optionally to the NT Application Events log, viewable from the **Logs** window and the NT Event Viewer (respectively).

## Creating an Alert Action


In the following exercise, you define an action that sends an alert to the **Job Activity** window when the job *EDI\_Download* (created in the previous chapter) ends with a **Completed Abnormally** status.

### To create an Alert action:

1. In the **Navigator** pane, select **Definitions>Actions**.

The **Actions** window displays.

2. Double-click **Alert** to display the **Alert** window.

3. In the TWA toolbar, click the **Add**  button.

The **Action Definition:Alert** dialog displays.

4. Type the following message and select the variables shown in angle brackets from the **Variables** list. The variables appear between the brackets as <variable>.

The Job <Jobname>, owned by <JobOwner> went to <JobStatus> status on <SysDate.M/d/yy>. <JobOwner>'s pager number is <JobPager>.

When this action is triggered, an alert record is sent to the Job Activity console for the operator to take action.

**Table 1**

Variable	Path	Description
<JobName>	Job Variables\JobNAME	The name of the job that triggered the action.
<JobOwner>	Job Variables\JobOWNER	The name of the user that owns the job.
<JobStatus>	Job Variables\JobSTATUS	The status the Job is in at the time of the message.
<SysDate.M/d/yy>	System Variables\System Date\M/d/yy	The day, month, date, year action was triggered.
<Job Pager>	Job Variables Owner's Pager Number	The pager number of the owner.


5. Click **OK** to accept the action.

The action is displayed in the **Actions** window.

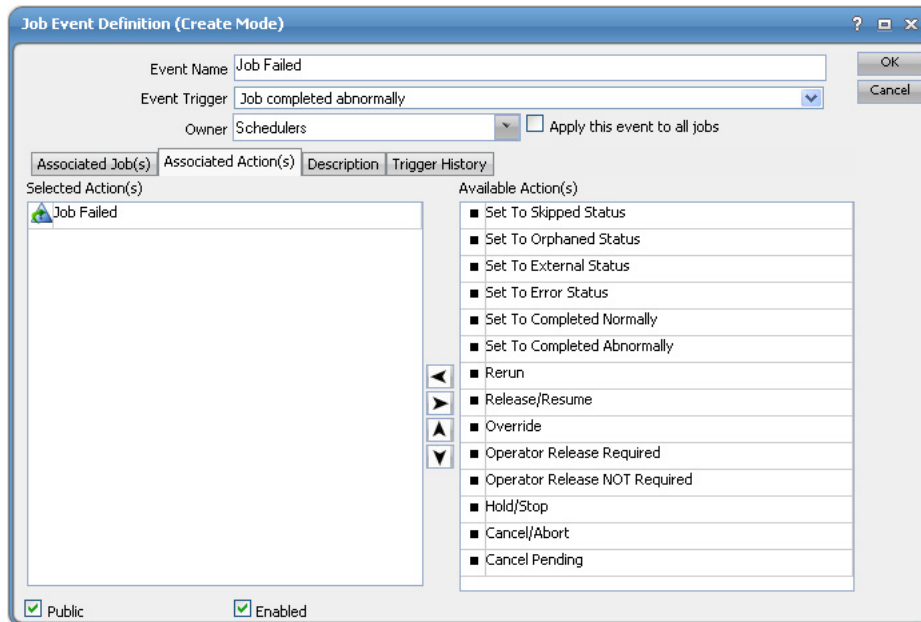
## Associating the Action to the Job

You can assign the action to the *EDI\_Download* job through the **Job Events** dialog.

### To associate the action to the job:

1. In the **Navigator** pane, select **Definitions>Events** to display the **Events** window.
2. Double-click **Job Events** to display the **Job Events** window.
3. In the TWA toolbar, click the **Add**  button.

The **Job Event Definition** dialog displays.



4. In the **Event Name** field, enter **Job Failed**.
5. From the **Event Trigger** list, select **Job Completed Abnormally**.
6. In the **Available Action(s)** list, highlight **Job Failed** and click the left arrow button or drag-and-drop the **Job Failed** action to the **Selected Action(s)** section.
7. Click the **Associated Job(s)** tab to associate the *EDI\_Download* job to the job event.
8. Select *EDI\_Download* from the list, then click **Add**.
9. Select the **Public** option, if not already selected.

This makes your event available to other TWA users.

10. Click **OK**.

The *EDI\_Download* job is now linked to the **Job Failed** alert action through the **Job Failed** job event.

## Changing the edi.bat file to Fail

### To change the edi.bat file:

1. Right-click the **Edi** batch file and select **Edit** from the context menu.

The *Edi.bat* file opens in Microsoft Notepad.

2. Change the last line of the file *ocsexit 0* to *ocsexit 1*.
3. From Notepad's **File** menu, select **Exit**.

A Notepad dialog displays.

4. Click **Yes** to save your changes.

5. Insert the job into the schedule by right-clicking the *EDI\_Download* job and selecting **Insert Job into Schedule** from the context menu.

The **Insert Job Into Schedule** dialog displays.

6. Click **OK**.
7. On the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** window.

Watch as the job goes into the *Active* state and runs. The job eventually terminates with an **Completed Abnormally** status, initiating the action you defined and changing the Alerts status light at the bottom of the console to red.

## Changing the Status of the Alert

### To change the status of the alert:

1. In the **Navigator** pane, select **Operations>Alerts** to display the **Alerts** window.
2. Double-click the alert to display the **Alert Detail** dialog.
3. Select the **Acknowledged** option.
4. You can enter a response or comment in the **Response** field.
5. Click **OK** to close the dialog.

# 9

## Creating Variable Dependencies

### Overview

In the previous chapter, we created an alert message that used system variables. In this chapter, we will create a job that depends on the value of a user-defined variable. A user-defined variable is a variable you create manually. A job can change the value of a variable you have assigned to it by using a variable update action. A job can be dependent upon the value of a variable. Using user-defined variables, you are given unlimited control over the conditional structuring of jobs.

For example, a job can run, and set a variable to a specific date. Another job can be triggered to run based on that variable's date. When the specific date arises, the second job runs.

In this chapter, you will increment a variable with the action of "Job A" and make "Job B" dependent on that variable reaching a specific value. The variable will start with a value of 0 and increment by 1 each time Job A runs. Job B will depend on the variable equaling 3. After Job A repeats 3 times, Job B will start. You could set up a job to start after any defined number of previous dependency jobs complete.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

### Creating a Variable

#### To create a variable:

1. In the **Navigators** pane, select **Definitions>Variables** to display the **Variables** pane.
2. Display the **Variable Definition** dialog by either clicking the **Add Variable** button or by right-clicking in the **Variable** pane and selecting **Add Variable** from the context menu.
3. In the **Name** field, enter **Gather Orders Var**.
4. From the **Type** list, select **Number**.
5. In the **Value** field, enter the number **0**.
6. In the **Description** field, enter **The value will be incremented by 1 for each successful run of the Gather Job**.
7. Select the **Public** field, if it is not already selected.
8. Click **OK**.

The newly-created variable now appears in the **Variables** pane.

## Creating the Variable Update Action

### To create the variable update action:

1. From the **Navigator** pane select **Definitions>Actions>Variable** to display the **Variable Actions** pane.
2. Select the **Add Action** button or right-click the **Variable Actions** pane and select **Add Action** from the context menu to display the **Action Definition: Variable** dialog.
3. In the **Action Name** field, enter **Set Gather Var**.
4. From the **Variable Name** list, select **Gather Orders Var: Number**.
5. In the **Type of Update** list, select **Increment variable value by**.
6. In the **Value** field, enter **1** if it is not already set at **1** by default.
7. Click **OK**.

The **Set Gather Var** action now displays in the **Variable Actions** pane.

## Creating a Job Event for the Gather Report Job

### To create a job event for the Gather Report job:

1. From the **Navigator** pane select **Definitions>Events>Job Events** to display the **Job Events** pane.
2. Select the **Add Event** button or right-click the **Job Events** pane and select **Add Event** from the context menu to display the **Job Event Definition** dialog.
3. In the **Job Event Name** field, type **Event Action Gather**.
4. In the **Event Trigger** list, select **Job completed normally**.
5. Highlight **Set Gather Var** and click the left arrow button or drag-and-drop the **Set Gather Var** variable to the **Selected Action(s)** field.
6. Click **OK**.

## Creating the Gather Orders Job Group

### To create the Gather Orders job group:

1. From the **Navigator** pane select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click the **Jobs** pane and select **Add Job Group** from the context menu to display the **Job Group Definition** dialog.
3. In the **Job Group Name** field, enter **Gather Orders**.
4. In the **Calendar Name** list, select the **work day** calendar.
5. Click **OK**.
6. Click **OK** at the **Effective Date** dialog.

**Note:** When you click **OK** at this dialog, the job group enters the production schedule, but nothing will run because there are no jobs associated with the job group yet. Since you can manually add the job group to the schedule later, clicking **Cancel** at this dialog is also acceptable.



## Creating a Repeating Job and Associating a Job Event

### To create a repeating job and associate a job event:

1. In the **Jobs** pane, select the job group **Gather Orders** and click the **Add Job** button or right-click in the **Jobs** pane and choose **Add Job** from the context menu to display the **Job Definition** dialog.
2. In the **Job Name** field, enter **Gather\_Report**.
3. In the **Command** field, enter the full path to the **Gather** batch file.
4. Click the **Schedule** tab.
5. In the **Repeats** section, select the **Run new occurrence every** option. Click in the **minutes** field and enter **2** (or use the up and down arrows) and click in the **up to \_\_\_ times** field and enter **5**.
6. Click **OK**.
7. Click **OK** in the **Effective Date** dialog.
8. If the **Gather Report** job is not visible in the **Jobs** pane, expand the **Gather Orders** job group by clicking the expand icon to its left.

The **Gather Report** job will run every two minutes for a total of five times, activating the **Event Action Gather** job event each time. **Set Gather Var** is tied to **Event Action Gather** and will increment by **1** each time **Gather Report** runs. When **Set Gather Var** reaches **3**, it will have fulfilled the conditions necessary to set off the next job, which we will define as **Order Reports** in [Creating a Job Dependent on the Gather Orders Variable, page 65](#).

The [Overview, page 63](#) explains in general terms the interdependence of **Gather Report** (Job A), **Set Gather Var** (a user-defined variable), and **Order Reports** (Job B).

9. Right-click **Gather Report** and from the context menu select **Edit Job/Group**.
10. Click the **Job Events** tab.
11. Click **Insert** to display the **Select Job Event** dialog.
12. Select the **Event Action Gather** event.
13. Click **OK**. The job event displays in the **Events** field.
14. Click **OK** to close the **Job Definition** dialog.
15. Click **OK** to confirm the **Effective Date** dialog.

Every time the **Gather Report** job completes normally, the variable **Gather Orders** will be incremented by **1**.

## Creating a Job Dependent on the Gather Orders Variable

### To create a job dependent on the Gather Orders variable:

1. Select the job group **Gather Orders** and click the **Add Job** button or right-click and select **Add Job** from the context menu to display the **Job Definition** dialog.
2. In the **Job Name** field, enter **Order Reports**.
3. In the **Command** field, enter the full path to the **Ordr\_rpt** batch file.
4. Click the **Dependencies** tab.

5. Click the **Add** button and select **Add Variable Dependency**.
6. Select **Gather Orders Var: Number** from the **Variable Name** list.
7. In the **Operator** list, if **Equals (=)** is not already displayed, select it.
8. In the **Variable Value** field, type the number **3**.
9. Click **OK** to accept the variable dependency.

The **Job Definition** dialog now displays the new dependency.

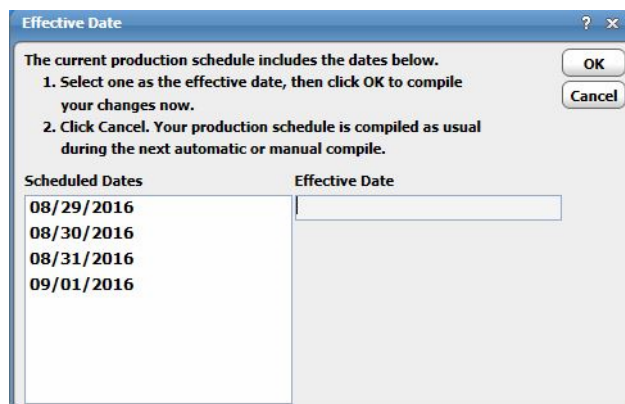
10. Click **OK**.
11. Click **OK** in the **Effective Date** dialog.

## Scheduling and Monitoring the Gather Orders Jobs

Now that all the components are defined and configured, we will run the jobs and monitor them in the **Job Activity** pane.

### To schedule and monitor the Gather Orders jobs:

1. In the **Jobs** pane, double-click the **Gather\_Report** job to display the **Job Definition** dialog.
2. Click **OK** to display the **Effective Date** dialog.
3. Select the **Start today's repeating job(s) now** option at the bottom of the **Effective Date** dialog; otherwise, Tidal Workload Automation will start repeating the job at the beginning of its time window, which is 12:00 AM for this job.



4. Click **OK**. The job enters the production schedule.
5. Right-click the **Order Reports** job and select **Insert Job into Schedule** from the context menu and click **OK**.
6. Click **OK** in the **Information** dialog.
7. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.
8. Double-click the **Order Reports** job occurrence.
9. Click the **Dependencies** tab.

The variable dependency appears showing the present value of the variable and the value it needs for the dependency to be met.

10. Click **Cancel** and look at the jobs in the **Job Activity** pane.

The three occurrences of the **Gather Report** job run before the **Order Reports** job runs.



# 10

## Managing Jobs in the Production Schedule

### Overview

In this chapter you learn to use job control, typically performed by an operator, to manage the current production schedule. Job control consists of releasing, stopping, restarting and aborting jobs. You can also rerun jobs, override their dependencies and change their completion status.

This chapter describes how to:

- Cancel a job.
- Remove a job.
- Release a job group in the **Waiting on Operator** status.
- Place a job on Hold.
- Override a job dependency.
- Rerun a job that fails.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

### Creating the Parent Job Group

**To create the parent job group:**

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click in the **Jobs** pane and select **Add Job Group** from the context menu to display the **Job Group Definition** dialog.
3. Right-click in the **Jobs** pane and select **Add Job Group** from the context menu to display the **Job Group Definition** dialog.
4. In the **Job Group Name** field, enter **Production Jobs**. The **Parent Group** field should be empty.
5. In the **Calendar Name** list, select **work day**.
6. Click **OK**.
7. Click **OK** in the **Effective Date** dialog

## Creating Three Child Jobs

### To create three child jobs:

1. To create a child job of a job group, select the **Production Jobs** job group on the **Jobs** pane.
2. Click the **Add Job** button or right-click and select **Add Job** from the context menu to display a **Job Definition** dialog.
3. In the **Job Name** field, enter **Prod Job 1**.
4. In the **Command** field, enter the full path to the file for your platform:  
**Windows**  
`C:\Program Files\TIDAL\Agent\TUTORIAL\Ap31.bat.`
5. Click the **Options** tab.
6. Select the **Require operator release** option.
7. Click **OK** and then click **OK** again in the **Effective Date** dialog.  
The first job has been created.
8. In the **Jobs** pane, select the **Prod Job 1 (UNIX=Unix Prod Job 1)** job if not selected.
9. Right-click **Prod Job 1** and select **Copy Job/Group** from the context menu.  
A job called **Copy of Prod Job 1** appears in the **Jobs** pane.
10. While the **Copy of Prod Job 1** job is selected, click the **Edit Job\Group** button or right-click and select **Edit Job\Group** from the context menu.
11. In the **Job Name** field, change the name to **Prod Job 2**.
12. Click **OK** and then click **OK** again in the **Effective Date** dialog.  
The second job has been created.
13. Create a third job, called **Prod Job 3**, the same way we created **Prod Job 2**.  
For this job, we will add an unreasonable time dependency to show how we can override this dependency in the **Job Activity** window.
14. Double-click **Prod Job 3**.
15. Click the **Schedule** tab of the **Prod Job 3** definition.
16. Clear the **Inherited** option in the **Time Window** section.
17. Enter a **Time Window** of **11:45 PM to 12:00 AM**.  
**Note:** If you set a time window that is before the present time, e.g., 12:00 AM to 12:15AM, the job will enter today's schedule with a Timed Out status.
18. Click the **Options** tab.
19. Make sure the **Require operator release** option is cleared. This job will wait based on its time window. (We are assuming that you are not running this tutorial at midnight.)
20. Click **OK**, then click **OK** again in the **Effective Date** dialog.
21. Right-click the new job group that you defined, then select **Insert Job into Schedule** and click **OK**.

22. Click **OK** in the **Information** dialog.

## Viewing the Jobs in the Job Activity Pane

From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.

You see all three jobs in the job group.

## Job Sorting

Jobs can be sorted and filtered by most columns in the **Job Activity** pane. On every refresh, sorting occurs on the column which contains a caret (^) in the column header.

**To set the default sort column to be the Estimated Start Time:**

1. Click the **Est Start Time** column header. The jobs are sorted by the estimated start time.

## Removing a Job from the Production Schedule

If a job has not launched, you can remove it from the production schedule.

**To remove a job from the production schedule:**

1. On the **Job Activity** pane, right-click the **Prod Job 1** job.
2. Select **Remove Jobs(s) from Schedule** from the context menu.
3. Click **Yes** on the **Confirmation** dialog.

**Note:** You can also remove a job by clearing the **Enabled** option in the job definition, or by deleting the job from the **Jobs** pane. You can only remove a job from the production schedule if it is not currently running or has not yet run.

## Inserting a Job into the Production Schedule

**To insert a job into the production schedule:**

1. From the **Activities** main menu, select **Insert Job into Schedule** to display the **Insert Job Into Schedule** dialog.

There are several ways to enter a job in this dialog. For this exercise, we will use Tidal Workload Automation's sophisticated search function.

2. Click the ellipses button in the **Job/Group** field to display the **Job Search** dialog.
3. In the **Name** field, type **Prod\***. Leave the other text fields blank.

The asterisk (\*) is a wildcard character, so this search will find all jobs that start with **Prod**. (The asterisk matches any character or combination of characters.)

4. Click the **Find** button.

The **Job Search Results** section should include the three **Prod Jobs** and the **Production Jobs** job group.

5. Select **Prod Job 1**.
6. Click **OK** in the **Job Search** dialog.
7. Click **OK** in the **Insert Job Into Schedule** dialog.

8. Click **OK** at the **Information** dialog.

The job is added to the schedule outside its job group. Note that jobs outside of a job group are not indented under the group.

9. Right-click the **Prod Job 1 (1)** job and from the **Job Control** menu select **Release/Resume**.
10. Click **Yes** in the **Confirmation** dialog.
11. Click **OK** in the **Information** dialog.

The job begins to run.

## Canceling a Job in the Production Schedule

You can cancel a job using the **Cancel/Abort** command when it is in the **Held** state. You can abort a job using the **Cancel/Abort** command when it is in the **Active** state. In the next procedure, we will cancel a job that is in the **Held** state.

**To cancel a job in the production schedule:**

1. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.
2. Select the **Prod Job 2** job.
3. Right-click the **Prod Job 2** and from the **Job Control** menu select **Cancel/Abort**.
4. Click **Yes** at the **Confirmation** dialog.
5. Click **OK** in the **Information** dialog.

The status of the job changes to **Canceled**.

## Placing a Job on Hold

You can put any job that is waiting on dependencies on hold. Even if the dependencies are met, the job will not run while in the **Hold** state.

**To place a job on hold:**

1. Select the **Prod Job 3** job.
2. Right-click the **Prod Job 3** job and from **Job Control** submenu select **Hold/Stop**.
3. Click **Yes** in the **Confirmation** dialog.
4. Click **OK** in the **Information** dialog.

The job enters **Held** status.

## Removing the Job from Hold Status

**To remove a job from Hold status:**

1. Right-click the **Prod Job 3** job and from the **Job Control** submenu select **Release/Resume**.
2. Click **Yes** in the **Confirmation** dialog, and then click **OK** at the **Information** dialog.

The job re-enters the **Waiting On Dependencies** status. If the job's dependencies were met, the job would launch.



## Overriding a Dependency

You can also override job, file, time and variable dependencies with the **Override** command. For this exercise, we will override the time dependency.

### To override a dependency:

1. Right-click the on **Prod Job 3** job and from the **Job Control** submenu select **Override**.
2. Click **Yes** in the **Confirmation** dialog, and then click **OK** at the **Information** dialog. The job launches even though the dependencies have not been met.

**Note:** When a dependency is overridden, the Job Detail dialog for the job whose definition includes the dependency will show which dependency was overridden.

## Estimated and Excessive Durations

Two of the most common questions asked of system operators and schedulers are “How long is this job going to run?” and “Which job is going to run next?” To help answer these questions, Tidal Workload Automation provides the Estimated Duration feature.

When a job enters the production schedule for the first time, if an estimated duration time is specified in the **Job Definition** dialog, that time will be displayed in the **Job Activity** pane. The **Estimated Duration Time** value is automatically updated after each job occurrence runs, creating a historical average that is used for subsequent submissions. Of course, you can override this value in the job definition at any time.

For unattended operations, you may want a job to:

- Terminate if it runs over its maximum time.
- Notify an operator if it completes under its minimum time.
- Perform other actions automatically.



# 11

## Managing Command Parameters

### Overview

This chapter describes how to:

- Pass command parameters to an executable.
- Determine the job completion status based on the jobs output.
- Define command parameters and enable alternative tracking methods.

In this chapter, you will create two jobs (**Report Writer 1** and **Report Writer 2**) that use the same batch file but with different command parameters. The command is coded to echo the command parameters to the output. You use the echoed value to determine the success or failure of the job. This example serves to demonstrate two important features: passing input and determining the jobs success by examining the output.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

### Passing Command Parameters

The command parameters feature allows you to pass information to a command when it is launched by Tidal Workload Automation. You can alter parameters such as dates or other data at runtime, rather than hard-coding the command file itself. You can also use parameters to add flexibility to the same command.

For example, you may have a batch file that can process several files. By specifying its file name as a command parameter, the batch file knows which file to process at any given time.

Tidal Workload Automation uses the standard MS-DOS or command prompt syntax for command parameters. The command parameters are substituted into a command file containing the variable names **%1**, **%2**, etc. Through the DOS prompt, parameters are passed as a list of values separated by spaces after the command. In Tidal Workload Automation, the command parameters are displayed in a list below the command. The first value replaces **%1**, the second value replaces **%2**, and so on, when the batch file is executed.

### Specifying Command Parameters and Viewing the Batch File (Windows)

After specifying command parameters in the **Job Definition** dialog, sometimes it is useful to view or edit the batch file to which the command parameters apply.

**To view or edit the batch file to which the command parameters apply:**

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.

2. Right-click and select **Add Job** from the context menu to display the **Job Definition** dialog.

You specify command parameters from the **Command Parameters** section in the **Program** tab.

3. From the **Tutorial** directory, right-click the *Argdemo* batch file and select **Edit** from the context menu.

The Microsoft Notepad text editor opens the batch file and you see the batch file text. The command parameters specified in the **Job Definition** dialog will be entered into the batch file in the locations **%1**, **%2**, etc.

4. Exit **Notepad** and click **Cancel** twice to return to the **Jobs** pane.

**Note:** For more information on batch processing, please refer to your Windows or DOS documentation.

## Tracking Methods

A job's completion status is determined by its exit code. The exit code can be set by using the **ocsexit** command at the end of a batch file. Using **ocsexit** to facilitate job tracking, the completion status of a job can be:

- **Completed Normally** indicated by an exit code of 0 to notify that the job succeeded
- **Completed Abnormally** indicated by an exit code  $\neq 0$  to notify that the job ran, but failed.

**Note:** Completion statuses of **Skipped**, **Orphaned**, **Error Occurred** and **Externally Defined** are also possible but are not determined by the exit code.

The completion status is frequently used to control dependencies. For example, you can set up jobs that run if its predecessor completes with a **Completed Normally** status.

Tidal Workload Automation supports four other tracking methods:

- **External** – You can use this option if you need to inspect the job before determining how the job completed. Tidal Workload Automation sets the status to **Set Externally** when the job completes.
- **Exit Code of cmd pipe** – You can use this option to pipe the output of your job as an input to a command, such as the Windows **Find** command or other output analysis tool. Tidal Workload Automation then uses that command's exit code as the job's exit code. You must type the command that invokes the associated tool.
- **Scan Output: Normal String** – Select this option to use a particular text string in the job output to determine if the job completed normally. Enter the exact text that determines “normal” in the text box below.
- **Scan Output: Abnormal String** – Select this option to use a particular text string in the job output to determine if the job completed abnormally. Enter the exact text that determines “abnormal” in the text box below.

The tracking selections are located in the **Tracking** section of the **Run** tab in the **Job Definition** dialog.

## Using the Exit Code

In this section, you will create a **Reporting** job group with two jobs, Report Writer 1 and Report Writer 2. Each job will use the same command, but will pass a different command parameter. The command parameter is then echoed in the output of both jobs after they complete.

To determine whether the jobs completed successfully or not, both jobs will use the **Exit code of cmd pipe** tracking method using the Windows Find command to search for the saved output. The result of the **Find** command will determine the job's final status.

You will set up Report Writer 1 to succeed, and Report Writer 2 to fail. You do this by setting the command parameter for job Report Writer 1 to OK and the parameter for Report Writer 2 to FAILED.

## Creating the Reporting Job Group

### To create the reporting job group:

1. From the **Navigators** pane, select **Definitions>Jobs** to display the Jobs pane.
2. Right-click and select **Add Job Group** from the context menu.
3. In the Job Group Name field, enter **Reporting**.
4. Click the **Schedule** tab, if not already selected.
5. From the **Calendar Name** list, select **work day**.
6. Click **OK** to save the job group.
7. Click **Cancel** at the Effective Date dialog.

**Note:** If you click **OK** at this dialog, the job group enters the production schedule, but nothing will run because there are no jobs associated with the job group yet. Clicking **Cancel** means the job group will be added to the schedule manually.

## Setting the Report Writer 1 Job to Succeed

### To set the Report Writer 1 job to succeed:

1. Right-click the **Reporting** job group and select **Add>Job** from the context menu.
2. In the **Job Name** field, enter **Report Writer 1**.
3. In the **Command** field, enter the full path to the Argdemo batch file.
4. Enter **OK** in the **Command Parameters** field. This passes as Argument 1 (%1) to the batch file.
5. Click the **Run** tab.
6. In the **Tracking** section, select the **Exit code of cmd pipe** option.
7. In the text field below, type **Find "OK"**. This tells TWA to search through the output of the job for the string "OK". Since "OK" is passed as a command parameter, it will appear in the output.
8. Click **OK**.
9. Click **Cancel** in the Effective Date dialog. This job will enter the schedule when manually submit its job group.

## Setting the Report Writer 2 Job to Fail

### To set the Report Writer 2 job to fail:

1. Right-click the **Report Writer 1** job in the jobs view.
2. Select **Copy Job/Group** from the context menu.
3. Double-click the **Copy of Report Writer 1** job to open its definition for editing.
4. In the **Job Name** field, change **Copy of Report Writer 1** to **Report Writer 2**.
5. In the **Command Parameters** field, change **OK** to **FAILED**.
6. The **Run** tab should contain **Find "OK"** in the Tracking section. This was inherited when you copied the job.

7. Click **OK**.
8. Click **Cancel** at the Effective Date dialog.

## Adding the Reporting Job Group to the Production Schedule

**To add the Reporting job group to the production schedule:**

1. Right-click the **Reporting** job group.
2. Select **Insert Job** into Schedule from the context menu.
3. Click **OK** in the **Insert Job Into Schedule** dialog.
4. Click **OK** in the **Information** dialog.

The jobs start with one completing normally and the other completing abnormally. The job group results in a **Completed Abnormally** status because one of its jobs completed abnormally.

## Setting Report Writer 2 to Complete Normally

**To set Report Writer 2 to Complete Normally:**

1. In the **Job Definition** dialog, double-click the **Report Writer 2** job.
2. In the **Job Definition** dialog, change the **Command Parameter** value from FAILED to OK.
3. Click **OK**.
4. Click **OK** at the Effective Date dialog.
5. Re-submit the jobs by adding the **Reporting** job group using the **Insert into Schedule** option, from the right-click context menu or selecting the option from the **Actions** main menu.
6. Click **OK** when the **Insert Job Into Schedule** dialog displays.
7. Click **OK** in the **Information** dialog.

Both **Report Writer 1** and **Report Writer 2** complete successfully.

# 12

## Setting Up Queues

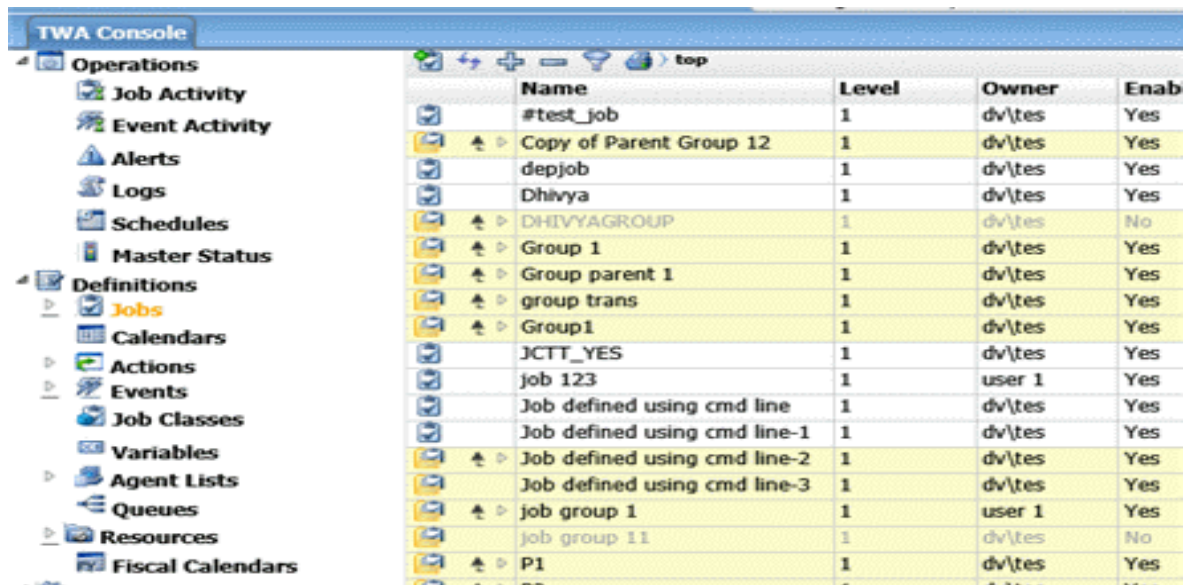
### Overview

This chapter explains how to set up queues so that your jobs run efficiently on your network. Jobs enter queues when their dependencies are met and are released from a queue by the master. In the previous chapters, the jobs you created ran from the default queues.

Queues are most effective when more jobs are scheduled than you have resources to handle them. Each queue has a job limit and a priority that helps to determine from which queue jobs are launched. When a queue's limit is reached, Tidal Workload Automation selects the next queue to run jobs on according to queue priorities. Within a queue, a job's individual priority determines which job in the queue is launched first.

Queues are defined in a hierarchy, with the **System Queue** at the top.

To view Tidal Workload Automation's predefined job queues, select **Scheduling>Queues** in the **Navigator** pane.



Name	Level	Owner	Enab
#test_job	1	dv\tes	Yes
Copy of Parent Group 12	1	dv\tes	Yes
depjob	1	dv\tes	Yes
Dhivya	1	dv\tes	Yes
DHIVYAGROUP	1	dv\tes	No
Group 1	1	dv\tes	Yes
Group parent 1	1	dv\tes	Yes
group trans	1	dv\tes	Yes
Group1	1	dv\tes	Yes
JCTT_YES	1	dv\tes	Yes
job 123	1	user 1	Yes
Job defined using cmd line	1	dv\tes	Yes
Job defined using cmd line-1	1	dv\tes	Yes
Job defined using cmd line-2	1	dv\tes	Yes
Job defined using cmd line-3	1	dv\tes	Yes
job group 1	1	user 1	Yes
job group 11	1	dv\tes	No
P1	1	dv\tes	Yes

Each queue contains a set of queue filters. Queue filters are designed to accept or reject jobs based on their properties.

In order for a job to match a queue, all categories defined in a queue must be met, and each one must be met at least once. All parent queue categories will also have to be met, and no queue with a higher priority must match the job at all.

For example, a queue defined as jobs starting with TEST\* on agent Example Agent will match a job named TEST JOB run on Example Agent, but will not match job TEST JOB 1 run on Example Agent 2. Also, all parent queues must also be matched before this queue can be considered for a match.

Besides prioritizing the execution of large quantities of jobs, queues can be used for a number of other purposes:

- Limit the number of jobs of a particular type or group running concurrently

- Control the flow of jobs executed, overall or by an individual queue, so that system resources are used more evenly and peak demands are reduced
- Single-thread jobs requiring a particular resource, such as exclusive access to a database
- Restrict the hours when jobs run to a time window defined by the queue, for example, to allow batch processing to occur overnight
- Launch jobs immediately for special cases where it is required

This chapter covers:

- Setting up the **Inventory Weekly** job group.
- Creating three jobs for the **Inventory Weekly** job group.
- Setting up a queue.
- Adding the **Inventory Weekly** job group to production.
- Monitoring the job group in production.
- Changing the queue limit.
- Re-submitting the **Inventory Weekly** job group.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

## Setting Up the Inventory Weekly Job Group

In this example, we will design a queue to accept jobs of a particular job group. Otherwise, there will be no job group to choose from when the queue is created.

### To set up the Inventory Weekly job group:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click in the **Jobs** pane and select **Add Job Group** from the context menu.
3. In the **Job Group Name** field, enter **Inventory Weekly**. Make sure the **Parent Group** field is empty.
4. Click the **Schedule** tab, if not selected.
5. In the **Calendar** list, select the **work day** calendar.
6. Click the **Options** tab.
7. Select the **Require operator release** option.
8. Click **OK**.
9. Click **Cancel** at the **Effective Date** dialog. (The job group should not enter the schedule yet.)



## Creating Jobs for the Inventory Weekly Job Group

Now, we will create three jobs (**Inventory\_10**, **Inventory\_11** and **Inventory\_12**) that belong to the **Inventory Weekly** job group, with no dependencies.

### To create jobs for the Inventory Weekly job group:

1. In the **Jobs** pane, select the **Inventory Weekly** job group.
2. Click the **Add Job** button or right-click and select **Add Job** from the context menu to display the **Job Definition** dialog.
3. In the **Job Name** field, enter **Inventory\_10**.
4. Verify that **Inventory Weekly** is displayed in the **Parent Group** list. If not, select it.
5. In the **Command** field, enter the full path to the **Inv10** batch file for your platform:

#### Windows

C:\Program Files\TIDAL\Agent\TUTORIAL\Inv10.bat

6. Click the **Notes** tab. We will use the **Notes** tab to enter a short description of the job.

The **Notes** tab is a good place to add special instructions or explanatory comments without affecting the job's execution.

7. In the **Other Notes** field, enter **Inventory Processing Program**.
8. Click **OK**.
9. Click **Cancel** at the **Effective Date** dialog.

## Duplicating a Job Definition

### To duplicate a job definition:

1. Select the **Inventory\_10** job in the **Jobs** pane if not already selected, then right-click and select **Copy Job/Group** from the context menu.

A job displays in the **Jobs** pane named **Copy of Inventory\_10**.

2. Double-click **Copy of Inventory\_10** and rename it to **Inventory\_11**.
3. In the **Command** field:

#### Windows

Change *Inv10.bat* to *Inv11.bat*.

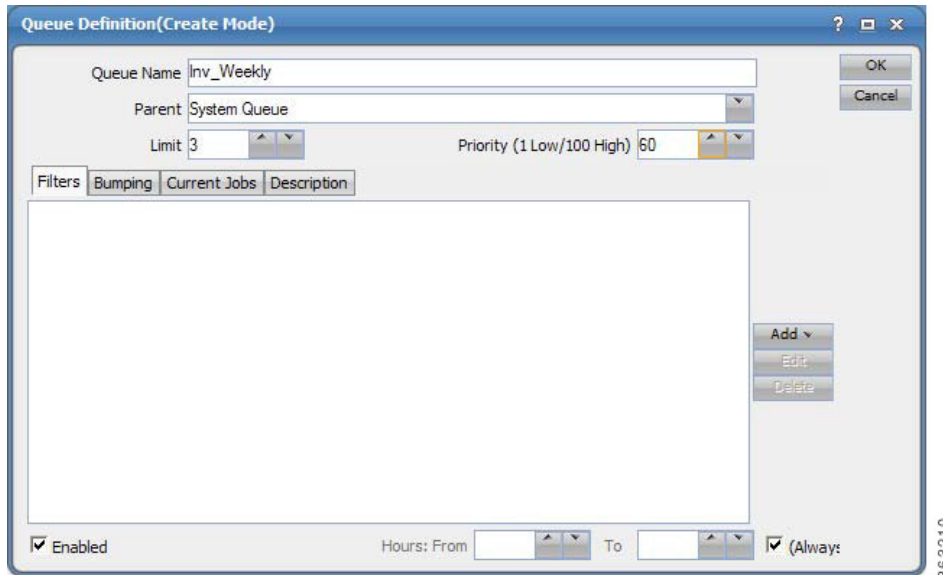
4. Click **OK**.
5. Click **Cancel** to prevent the job from being scheduled.
6. Repeat the steps above to create a third job called **Inventory 12** and enter the full path to the *Inv12.bat* (Windows) file in the **Command** field.

## Setting Up a Queue

We will set up a queue named **Inv\_Weekly** and give it a queue limit of three. The jobs in the **Inventory Weekly** group will run in this queue. The queue configuration parameters will single-thread (run one at a time) the jobs in the group.

**To set up a queue:**

1. From the **Navigator** pane, select **Scheduling>Queues** to display the **Queues** pane.
2. Right-click the **System Queue** and select **Add Queue** from the context menu to display the **Queue Definition** dialog.



3. In the **Queue Name** field, enter the name **Inv\_Weekly**.

4. Leave **System Queue** in the **Parent** field.

We defined **Inv\_Weekly** as a direct sub-queue of the system queue in the queue hierarchy. This means there are no intermediate queues to impose their own restrictions and affect job execution.

5. Change the **Limit** value to **3**. This allows a maximum of three jobs from this queue to run concurrently.

6. Change the **Priority** value to **60**.

Default queues have a priority setting of **50**. Since default queues *can* accept the jobs that you created, this higher priority value will ensure that the jobs enter this queue. When the queue manager sees two or more queues that can accept a job, it always assigns the job to the queue with the highest priority.

7. On the **Filters** tab, click the **Add** button to display a list and select **Job Group**.

The **Queue Filter Condition** dialog displays.

8. Select the **Inventory Weekly** job group.

9. Click **OK**. The filter is entered into the **Filters** tab.

The **Always** option in the **Hours Available** section should be selected by default. This means that the **Inv\_Weekly** queue can run at any time. If you de-select the **Always** option, you can restrict the queue to operate within a specific time window. Jobs assigned to the queue would then run only if the current time were within that time window.

10. Click **OK**. **Inv\_Weekly** is added to the list of queues on your system.

## Adding the Inventory Weekly Job Group to Production

Now, submit the group **Inventory Weekly** into the production schedule.

### To submit the group into the production schedule:

1. In the **Jobs** pane, right-click the **Inventory Weekly** job group and select **Insert into Schedule** from the context menu.
2. Click **OK** in the **Insert Job Into Schedule** dialog.
3. Click **OK** in the **Information** dialog.

## Monitoring the Job Group in Production

### To monitor the job group in production:

1. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.  
Your job group is listed in bold showing the **Waiting On Operator** status.
2. Right-click the **Inventory Weekly** job group and from the **Job Control** submenu and select **Release/Resume** to start the job group.
3. Click **Yes** in the **Confirm** dialog.
4. Click **OK** in the **Information** dialog.

The **Inventory Weekly** job group goes active, along with the three jobs: **Inventory\_10**, **Inventory\_11**, and **Inventory\_12**. Because the limit on the **Inv\_Weekly** queue is set to **3**, all three jobs in the group **Inventory Weekly** are launched at once, and now run to completion.

5. Scroll to the right in the **Jobs** pane to view the **Queue** column.

Note that the **Inventory\_10**, **Inventory\_11** and **Inventory\_12** jobs are running from the **Inv\_Weekly** queue.

## Changing the Queue Limit

Changing the queue limit from any number to **1** changes the queue from multiple to single-threaded processing.

### To change the queue limit:

1. From the **Navigator** pane, select **Definitions>Queues** to display the **Queues** pane.
2. Double-click the **Inv\_Weekly** queue to display its definition dialog.
3. Lower the **Limit** from **3** to **1**.
4. Click **OK** to save the changes.

The limit change is displayed in the **Queues** pane.

## Resubmitting the Inventory Weekly Job Group

Now we re-submit the **Inventory Weekly** job group into the production schedule.

### To re-submit the job group:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.

2. Right-click the **Inventory Weekly** job and from the context menu select **Insert Job into Schedule**.

3. Click **OK** in the **Insert Job Into Schedule** dialog.

4. Click **OK** in the **Information** dialog.

5. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.

You can now see another occurrence of your job group and its jobs in the **Job Activity** pane.

6. Click the new occurrence of the **Inventory Weekly** job group.

7. Right-click the occurrence of the **Inventory Weekly** and select **Job Control** from the context menu.

8. Select the **Release/Resume** option.

9. Click **Yes** in the **Confirm** dialog.

10. Click **OK** in the **Information** dialog.

The **Inventory Weekly** job group goes active, and the jobs within the group run again in no particular order. This time, however, they run one at a time because their assigned **Inv\_Weekly** queue has a limit of **1**. While they are running, if you insert or rerun any other job on the schedule that belongs to a different queue, it runs immediately since it does not belong to the **Inv\_Weekly** queue and is not restricted by its limit.

# 13

## Using TWA Agents

### Overview

If you have installed two or more TWA Agents for Windows or UNIX on separate machines, you can perform the procedures in this chapter. If you have only one agent, you may still want to read this chapter to understand how other agents are integrated in Tidal Workload Automation.

In previous exercises, you scheduled jobs to run on the Tidal Workload Automation master system which included an agent of its own. You can also run jobs on multiple external agents. Agents let you expand your computing power and flexibility, helping you get jobs done more quickly and efficiently.

For a Tidal Workload Automation master to recognize an agent, the Tidal Workload Automation agent software must be installed on that agent machine, and a valid license file must exist for it on the master. Refer to the *Tidal Workload Automation Installation and Configuration Guide* for more information on licensing and registering agents.

The TWA master provides centralized control over scheduling in a distributed computing environment. There is no limit on how many agents you can attach to a master, if your license contains the unlimited (floating) provision for agents. You can also schedule a job to run on any agent in your network.

Tidal Workload Automation lets you group multiple agents into sets called **agent lists**. When you associate one of these lists with a job, Tidal Workload Automation launches that job depending on the type of list:

- **Ordered List** - Runs on the first agent available in the list, chosen sequentially from the top of the list
- **Balanced List** - Runs on the agent with the lightest load at the time the job is ready to run.
- **Random List** - Runs on an agent picked at random from the list.
- **Rotation List** - Runs on the agent following the last agent used for execution.
- **Broadcast List** - Runs on all agents in the list.

In this chapter, you learn how to run a job on an external agent. You will also learn how to create an inter-agent dependency, that is, a dependency on a job that runs on a different machine. We will add an agent and run a job that is located on it.

To complete the exercises in this tutorial, you need to:

- Install Tidal Workload Automation in the TWA default directory (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.

## Defining An Agent Connection

### To define an agent connection:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Double-click the agent connection to display the **Connection Definition** dialog.
3. Click the **Connection** tab.
4. In the **Machine Name** field, enter the machine with the Tidal Workload Automation agent installed on it.
5. In the **Master-to-Agent Communication Port** field, enter the agent's listener port number used during agent installation. The default port number for the first agent is **5912** but subsequent agent installations use an incremented port number (for example, 5913 for the second agent, etc.).
6. Select the **Enabled** option, if not already selected.
7. Click **OK**.

The agent will begin the connection process. The agent status light is yellow until the agent is connected. The status light then turns green. This normally takes about 45 seconds or less.

## Adding a Job that Requires Operator Release

### To add a job that requires operator release:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Click the **Add Job** button to display a **Job Definition** dialog.
3. In the **Job Name** field, enter **Agent\_Job1**.
4. In the **Command** field, enter the full path to the **Inv21** batch file for your platform:

#### Windows

C:\Program Files\TIDAL\Agent\TUTORIAL\Inv21.bat

5. Click the **Schedule** tab and select **work day** from the **Calendar Name** list.
6. Click the **Run** tab and from the **Agent Name** list, select the remote agent you are going to use.
7. Click the **Options** tab and select the **Require operator release** option.
8. Click **OK** to close the dialog.
9. Click **OK** to display the **Effective Date** dialog.

The job is placed in the production schedule with a **Waiting on Operator** status. We will release this job, but first let's create a job with a dependency on this job to show the agent dependency feature.

## Adding an Agent Job Dependency

In this section, you define a remote agent job dependency. You will use **Agent\_Job1**, created in the previous exercise, in the dependency condition for the new job. Defining a job that depends on a job run by a remote agent is the same as defining any other job dependency.

**To add an agent job dependency:**

1. Click the **Add Job** button to display a **Job Definition** dialog.
2. In the **Job Name** field, enter **Agent\_Job2**.
3. From the **Command** field, enter the full path to the **Inv22** batch file for your platform:

**Windows**

C:\Program Files\TIDAL\Agent\TUTORIAL\Inv22.bat

4. Click the **Schedule** tab and in the **Calendar Name** list, select the **work day** calendar.
5. Click the **Run** tab and select the agent that exists on the master.
6. On the **Dependencies** tab, click the **Add** button and select **Add Job Dependency** from the list of options to display the **Job Dependency Definition** dialog.
7. In the **Job/Group** list, select the **Agent\_Job1** job.

**Note:** Only jobs that have already been defined will appear in the menu. If you have not defined any jobs, none will appear.

8. Click **OK** to close the **Job Dependency Definition** dialog. This accepts the default dependency condition.

The **Agent\_Job1** job dependency appears in the **Dependency** section.

9. Click **OK** to close the **Job Definition** dialog.
10. Click **OK** again in the **Effective Date** dialog.

When the dialog closes, the **Agent\_Job2** job appears in the **Jobs** pane.

## Releasing the Agent Jobs

**To release the Agent jobs:**

1. From the **Navigator** pane, select **Operations>Job Activity** to display the **Job Activity** pane.
2. Select the **Agent\_Job1** job.
3. Right-click the job, and then select **Job Control>Release/Resume** from the context menu to release **Agent\_Job1** from the **Waiting on Operator** status.
4. Click **Yes** in the **Confirm** dialog.
5. Click **OK** in the **Information** dialog.

The **Agent\_Job1** runs on the external agent. When it completes successfully, the **Agent\_Job2** runs on the local agent.

## Summary

Congratulations! With the completion of this chapter, you have built a good foundation of knowledge about Tidal Workload Automation. You are ready for Tidal Workload Automation to increase and complement your productivity.

If you need any further information or training, please contact the Technical Support team.

