

# **A Usable Web Interface for IBM's distributed Tivoli Workload Scheduler (TWS)**

**By Tully Krastins**



# Contents

<b>INTRODUCTION .....</b>	<b>1</b>
<b>A BRIEF HISTORY OF THE TIVOLI WORKLOAD SCHEDULER GUI.....</b>	<b>1</b>
<b>WHAT IS USABILITY? .....</b>	<b>3</b>
<b>A USABLE TWS GUI .....</b>	<b>4</b>
<b>A QUICK LOOK AT TWS/WEBADMIN.....</b>	<b>5</b>
WORKING WITH THE DATABASE .....	6
WORKING WITH THE PLAN .....	13
SUPPORTING THE ADMINISTRATOR.....	19
WORKING WITH REPORTS .....	24
<b>IN SUMMARY .....</b>	<b>26</b>
<b>ABOUT SEGUS INC. ....</b>	<b>27</b>

TWS/WebAdmin is a trademark of HORIZONT GmbH.  
IBM and Tivoli are trademarks or registered trademarks of  
International Business Machines Corporation.

Other company, product and service names are the trademarks or  
service marks of others.

© Copyright T.Krastins, SEGUS Inc, HORIZONT GmbH, 2005  
All Rights Reserved

### About the Author:

Tully Krastins, President and CEO of Avots International, has over 30 years experience in all facets of IT, including operations, software development, system engineering, and program and corporate management. Tully is an internationally respected subject matter expert in workload management solutions. He has been an independent professional services asset for both Unison and Tivoli. Several Fortune 500 companies are among his clientele. Tully can be contacted at [tullyk@avots.com](mailto:tullyk@avots.com).

## Introduction

Maintaining a high level of productivity within the Operations staff is the challenge every Operations Manager faces today. Service Level Agreements are demanding increasingly higher levels of availability at both the front and back ends of business processes.

Any downtime of application processing affects business process continuity to the extent that even batch job scheduling, long considered a stepchild of information technology, is now regarded as mission-critical.

Consequently, workload management software, such as the Tivoli Workload Scheduler (TWS), must be easy to use and have an intuitive User Interface (UI).

Every software developer will assure you that the user interface is an integral part of any application being brought to market. Unfortunately, UI design and development often lags behind the core application in terms of priority, budget and focus.

Yet in the eyes of the customer, the user interface **is** the product. It is mostly through the UI that the customer rates the application. Customers want to accomplish their tasks with the least amount of effort and the least daunting learning curve.

## A Brief History of the Tivoli Workload Scheduler GUI

When the company Unison ported Maestro from MPE to Unix in the mid 1980's, they outsourced the Graphical User Interface (GUI) development to a small company specializing in user interfaces.

That user interface was elegant in its simplicity. It was intuitive, making it easy to learn and use. It tracked directly to the command line interface (CLI) where



*The Maestro GUI, gmaestro, consisted of gcomposer for object definition, and gconman for plan management.*

definitions were accomplished through “composer”, and plan management was accomplished through “conman.”

The GUI provided Schedulers with a straightforward means of defining and managing scheduling objects using “gcomposer.” On the management side, “gconman” provided operators with meaningful information when a situation requiring action was required-usually with just a point and a click.

It did require the added expense of *Exceed* or other X-Windows software for desktop access. It also had its share of usability inconsistencies across panels and its some quirks (such as leaving orphaned processes behind). Overall, however, it received very high marks from the user community.


As Web-based portals emerged into general acceptance, there was some grumbling in the user community about Maestro’s continued reliance on interfacing through *X11*.

To this end, Tivoli went to work on redesigning the GUI and unveiled the new interface product with the release of Version 7 of what was now the Tivoli Workload Scheduler (TWS). The product of their development effort is the java-based Job Scheduling Console (JSC).

*The final sunset for the Maestro GUI arrived with the release of TWS V8.2.*

An unanticipated issue arose immediately: Customer operations staffs resisted the radical change in interfacing with TWS. Seasoned veterans rapidly discovered that the old gmaestro suite—the legacy GUI—still worked with TWS Version 7 and, while no longer supported, with TWS Version 8.1 as well. What was the root cause of the rebellion? The underlying issue, simply stated, was the usability factor.





*Ultimately, usability is accomplished by architecting a user interface that fits with what the end user is trying to accomplish.*

*Jacob Neilson  
Usability Engineering  
(Academic Press,  
1993)*


## What Is Usability?

Show the same user interface to any number of programmers and you will get that number of arguments on how good (or bad) it is. Everyone has a personal preference and an opinion of what makes a good design. Should the real issue not take into consideration what makes the given task easiest? Is that not what makes the UI usable?

While some end users may appreciate pretty graphics with sophisticated visual controls and a 3D look and feel, ultimately they all want a system that is easy to learn, easy to use, and that helps them do their work - quickly.

Before he became a self-anointed web design guru, Dr. Jacob Neilson focused on engineering user interfaces. In that role he developed ten fairly broad heuristics for designing a good user interface system. Five of those are rigorous enough to be called "rules" that can still provide a general framework and meaningful objectives for good user interface design:

- ⇒ **First Rule:** “Good systems are usable-without help or instruction-by a user having knowledge and experience in the application domain but no experience with the system.” - **Access**
- ⇒ **Second Rule:** “Good systems do not interfere with or impede efficient use by a skilled user having substantial experience with the system.” - **Efficacy**
- ⇒ **Third Rule:** “Good systems facilitate continuous advancement in knowledge, skill, and facility and accommodate progressive change in the usage as the user gains experience with the system.”  
- **Progression**
- ⇒ **Fourth Rule:** “Good systems support the real work that users are trying to accomplish, making it easier, simpler, faster, or more fun.” - **Support**



⇒ **Fifth Rule:** “Good systems are suited to the conditions and environment of the actual operational context within which they are developed.” – **Context**

Usability, then, is not derived from rows of pretty colored icons, from floating toolbars or from 3D dialog boxes. Usability is not in the quantity of widgets used but rather in how the included widgets work together.

## A Usable TWS GUI

As mentioned earlier, an outside company built the legacy Maestro GUI. Now an outside company has built a usable GUI for TWS. TWS/WebAdmin, from SEGUS Inc, has a familiar look and feel. It strongly reminds one of the legacy Maestro GUI. That factoid alone strongly supports the usability rules presented above.

TWS/WebAdmin does not require a specific version of the Java Runtime Environment. It is not limited to specific hardware platforms. It does not require installing and maintaining any extraneous software such as a framework or special connectors. Pick a browser, any browser; you have connectivity.

TWS/WebAdmin provides a feature-rich user interface to all aspects of TWS. Each feature incorporates a subset of the Usability Rules listed above. TWS/WebAdmin provides an intuitive interface for:

- ◆ Defining and maintaining all TWS scheduling objects and their relationships using a familiar, intuitive interface.
- ◆ Viewing and managing all domains, workstations, job streams, jobs, prompts, resources and file dependencies in the Current Plan (the Symphony File) using a self-explanatory navigation system.
- ◆ Forecasting future Plans utilizing a sophisticated capability that offers a range of dates and powerful reporting.

*TWS/WebAdmin is a highly portable, web-based application making TWS very accessible. If you have a device with a web browser, you have access to TWS through TWS/WebAdmin.*

*For this test drive, I am using Safari V1.3, the Macintosh OS X browser. I also tested it using Internet Explorer V6.0 (on Windows XP) and Galeon V1.2 (on Red Hat Linux).*

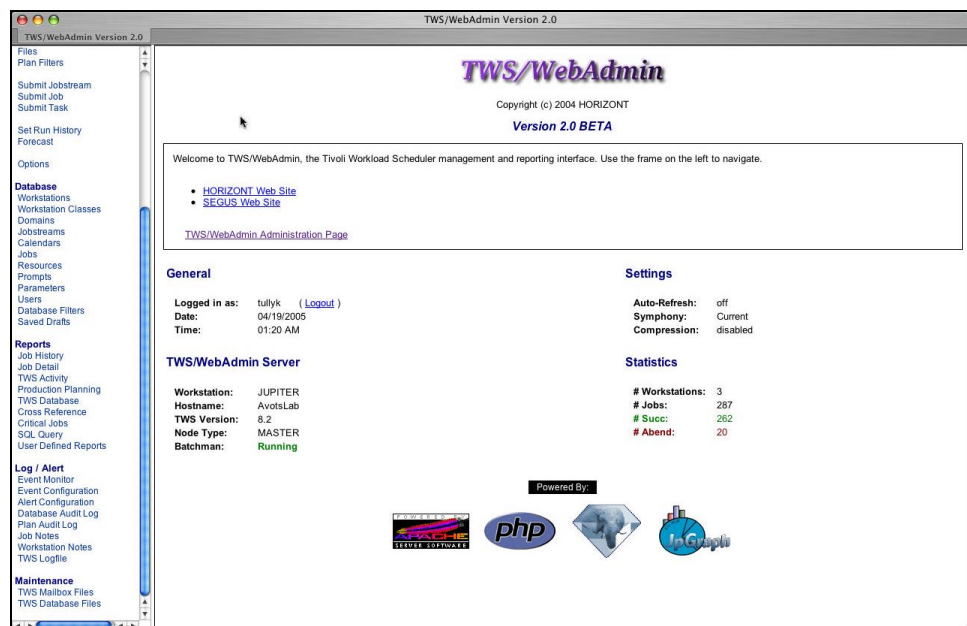
- ◆ Utilizing the Job Notes feature-accessible with one click-where instructions or on-call information may be maintained.
- ◆ Powerful reporting capability, including the option of running and saving SQL queries.
- ◆ Accessing formatted and legible audit log files for easy and quick review.

## A Quick Look at TWS/WebAdmin

This paper is not intended to be a tutorial. It is a first look to get a sense and feel for the product.

The home screen already illustrates the amount of thought put into designing TWS/WebAdmin. On the left side is a navigation bar for accessing TWS areas. The home page itself provides useful information about the server, the current settings and statistics about the TWS network.

Just from logging on, one can tell that the system is up (Batchman Running) and there are some problem jobs (# Abend).

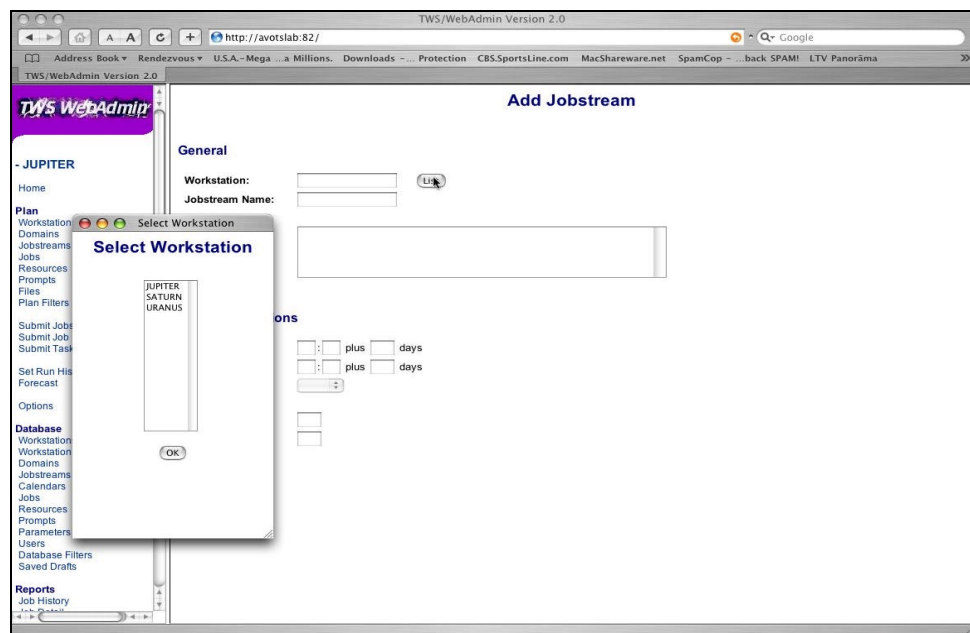


**TWS/WebAdmin Home Page Layout.**

## Working with the Database

The most complex task in job scheduling is to create a new jobstream. Creating a jobstream tests the product against the five rules discussed above. I did not read any documentation for TWS/WebAdmin (other than the install instruction) prior to exercising it for this paper.

An experienced scheduler can appreciate the orderly flow of creating a new jobstream. A novice scheduler can appreciate the guidance that flow represents. The following screen shots illustrate the steps I took for creating a new jobstream containing four jobs. To start, I selected Jobstreams from the Navigation bar on the home page.



**Where is the Jobstream Going to Run?**

**TWS/WebAdmin Version 2.0**

**Add Jobstream**

**General**

Workstation: SATURN (List)

Jobstream Name: tullytest\_stream

Description: Testing the jobstream performance on Windows server

**Jobstream Options**

At: 16:00 plus days

Until: plus days

Time Zone:

Priority:

Limit:

☐ Carryforward

Next >>

**At What Time Should the Jobstream Run?**

**TWS/WebAdmin Version 2.0**

**Run Cycle**

**On:**

☐ On Request

Weekly Cycle: ☐ Sunday ☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday ☐ Friday ☐ Saturday ☒ Everyday ☐ Weekdays ☐ Workdays ☐ Freedays

Calendar: (List) Offset: + - Days: Days

Single Date: / / (List) (MM/DD/YYYY)

**Except:**

Weekly Cycle: ☐ Sunday ☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday ☐ Friday ☐ Saturday ☐ Weekdays ☐ Workdays ☒ Freedays

Calendar: (List) Offset: + - Days: Days

Single Date: / / (List) (MM/DD/YYYY)

**Free Days:**

Calendar: ☒ Use Default ☐ Specify Calendar: (List)

Consider as Free Days: ☒ Saturday ☒ Sunday

<< Prev Next >>

**What is the Run Cycle of the Jobstream?**

Selecting the Run Cycle for this exercise gives me the opportunity to do something I strongly discourage when presenting TWS in a training session. I have selected "Everyday" as the cycle and "Except Freedays" as the

exception. (Freedays are defined as Saturday and Sunday.) For the sake of clarity and brevity, I should have just selected a Run Cycle of Weekdays with no exceptions.

That takes care of the mandatory Where and When requirements for setting up my jobstream. Now I must establish the prerequisites for the jobstream by defining “what must happen before this jobstream can run.”

The screenshot displays the TWS/WebAdmin Version 2.0 web interface. The browser address bar shows 'http://avotslab:82/'. The left sidebar contains a navigation menu with categories: JUPITER (Home, Plan, Workstations, Domains, Jobstreams, Jobs, Resources, Prompts, Files, Plan Filters, Submit Jobstream, Submit Job, Submit Task, Set Run History, Forecast, Options), Database (Workstations, Workstation Classes, Domains, Jobstreams, Calendars, Jobs, Resources, Prompts, Parameters, Users, Database Filters, Saved Drafts), and Reports (Job History). The main content area is titled 'Jobstream: SATURN#TULLYTEST\_STREAM'. It features five sections for defining dependencies: 'Jobstream Dependencies' with a single row for Workstation and Jobstream; 'Job Dependencies' with rows for Workstation, Jobstream, and Job; 'Prompt Dependencies' with a single row for Prompt; 'File Dependencies' with a single row for Workstation, Filename, and Qualifier; and 'Resource Dependencies' with a single row for Workstation, Resource, and Units. Each section includes a 'List' button and a '+-' icon. At the bottom, there are '<< Prev' and 'Next >>' navigation buttons.

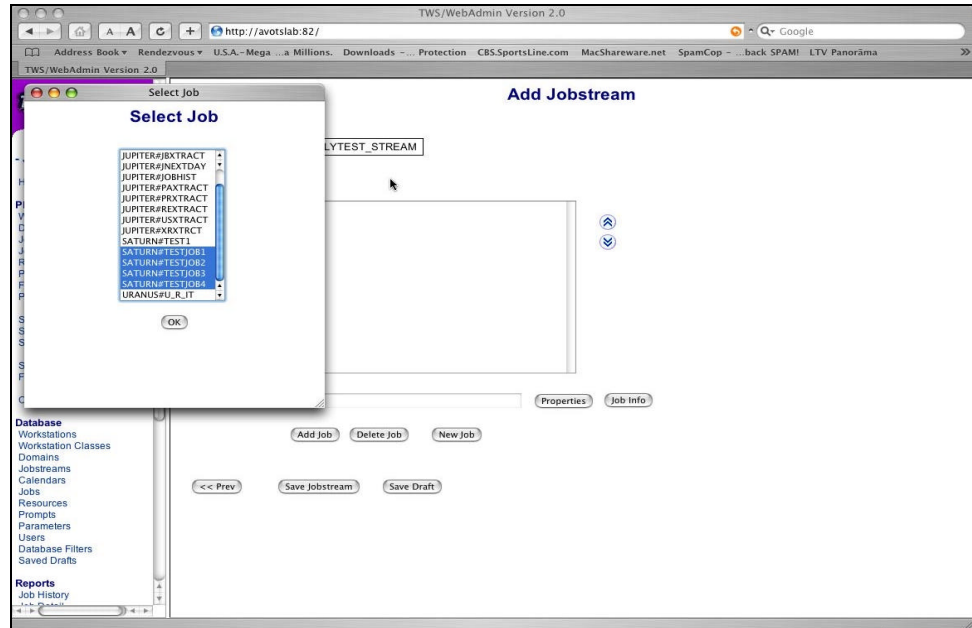
### Any Other Jobstream-Level Dependencies?

My “requirements” do not have any jobstream-level dependencies for this exercise, but having this screen presented at this point in the jobstream definition process is efficient and thoughtful. The tool is forcing me to think and fulfill my requirements in an orderly, top-down fashion. The aforementioned Rules 1 and 5 apply here.

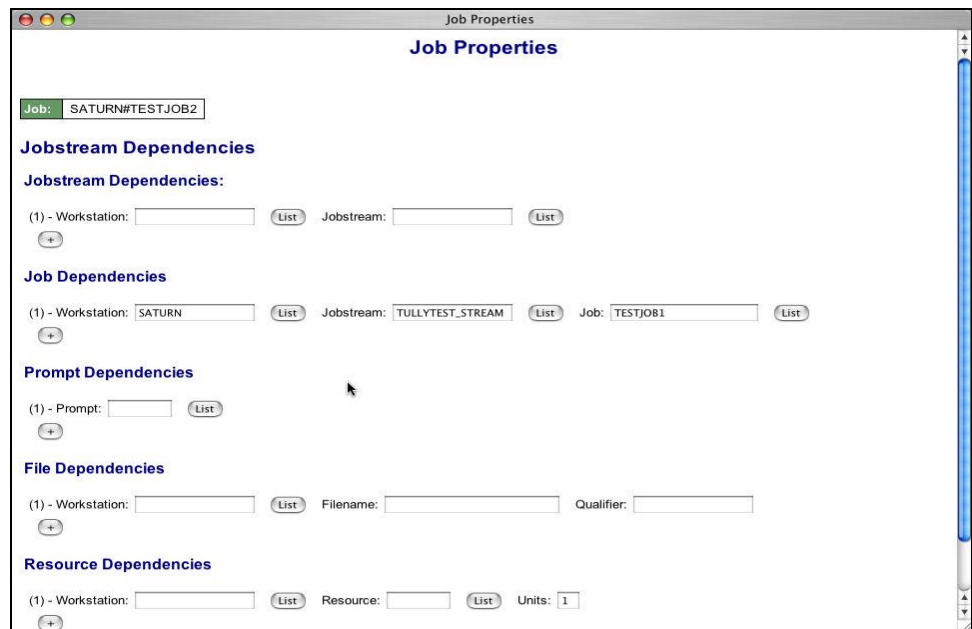
Since I have nothing to add as jobstream prerequisites, TWS/WebAdmin now takes me to the job level where I identify the actual tasks the jobstream will contain.



TWS/WebAdmin rewards good naming conventions by allowing selection of multiple job names for inclusion in the jobstream. That is a feature Schedulers will appreciate.



**TWS/WebAdmin Allows Selecting Multiple Jobs.  
(See Rule 4)**



## TestJob2 Is Made Dependent on TestJob1.

All dependencies for TestJob2 are established on this one screen. There is nothing disjointed about establishing job prerequisites. TESTJOB3 is then made dependent on TESTJOB2 in the same manner. TESTJOB4 has some additional dependencies.

The screenshot shows a window titled "Job Properties" with a sub-header "Job Properties" in blue. Below this, there is a field labeled "Job:" containing the text "SATURN#TESTJOB4". The window is divided into two main sections: "General" and "Job Options".

**General**

Description: [Empty text box]

**Job Options**

At: [19:] [00] plus [ ] days

Until: [ ] [ ] plus [ ] days

Every: [ ] [ ]

Time Zone: [Dropdown menu]

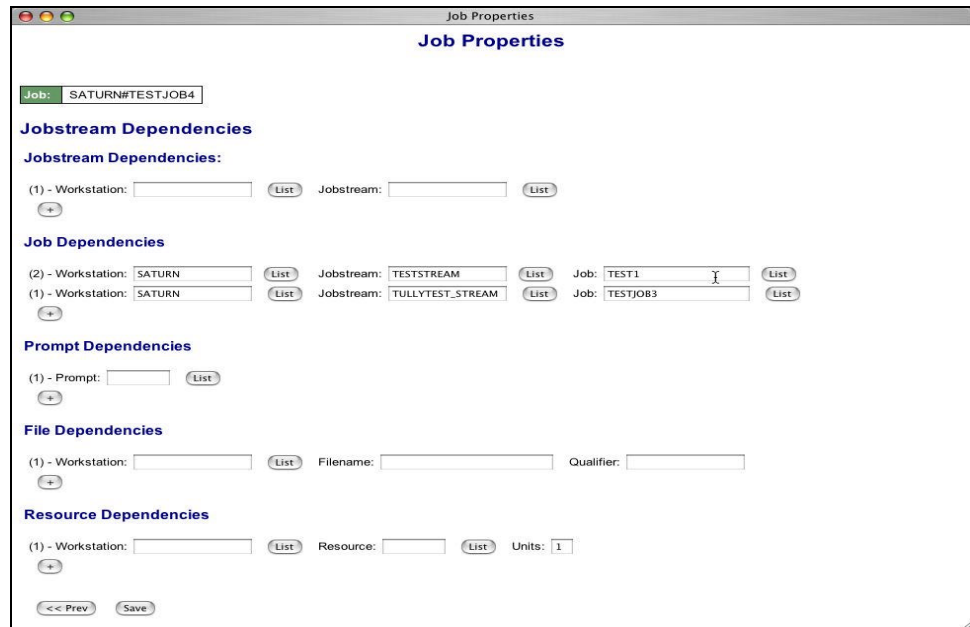
Priority: [ ]

☐ Confirmed

[Next >>]

## Job4 Has a Time Dependency and...



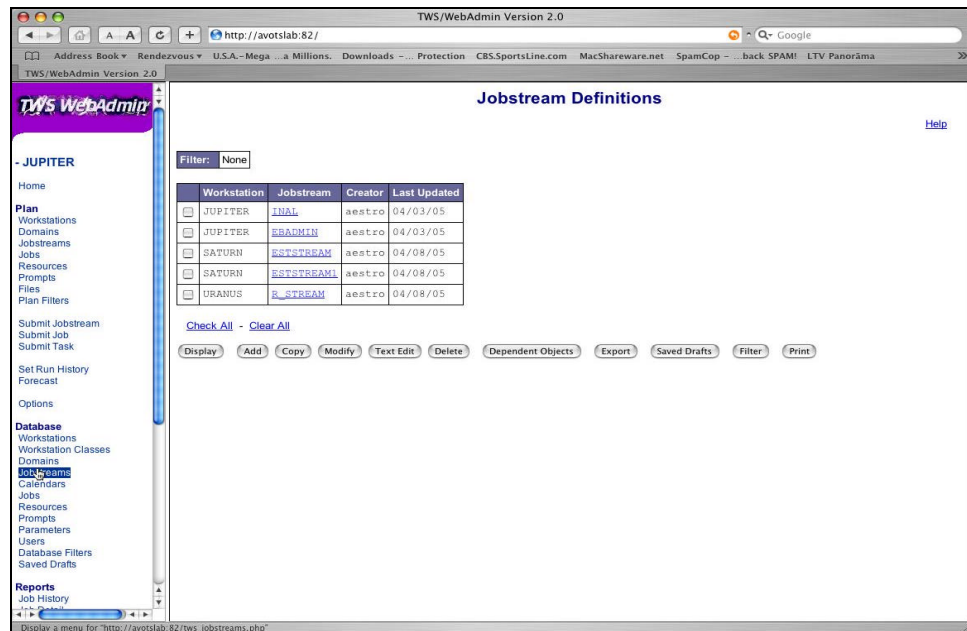


The screenshot shows a 'Job Properties' window with the following sections:

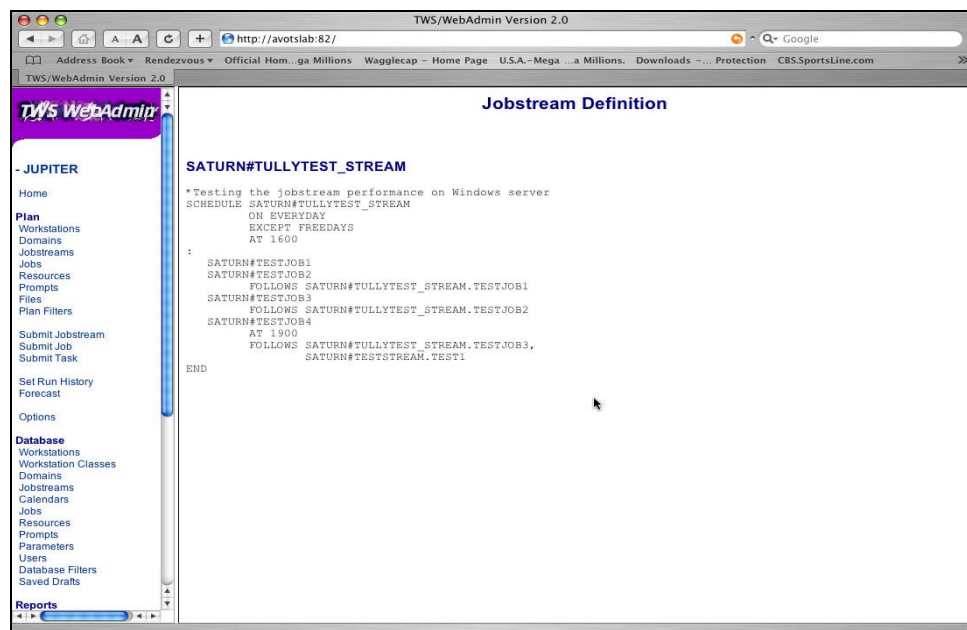
- Job:** SATURN#TESTJOB4
- Jobstream Dependencies:**
  - (1) - Workstation: [ ] List Jobstream: [ ] List
  - [+]
- Job Dependencies:**
  - (2) - Workstation: SATURN List Jobstream: TESTSTREAM List Job: TEST1 Y List
  - (1) - Workstation: SATURN List Jobstream: TULLYTEST\_STREAM List Job: TESTJOB3 List
  - [+]
- Prompt Dependencies:**
  - (1) - Prompt: [ ] List
  - [+]
- File Dependencies:**
  - (1) - Workstation: [ ] List Filename: [ ] Qualifier: [ ]
  - [+]
- Resource Dependencies:**
  - (1) - Workstation: [ ] List Resource: [ ] List Units: 1
  - [+]
- Navigation: << Prev Save

### ...Multiple Job Dependencies.

Once I have completed defining the jobstream by establishing all the prerequisites for each job and saved the new jobstream, I can perform a visual check of the jobstream to see what it looks like in the TWS scheduling language.




**Select the Jobstream and Click on DISPLAY to View...**



**...The Newly Crafted Jobstream.**

Creating a new jobstream with TWS/WebAdmin is simple and straightforward. TWS/WebAdmin keeps the jobstream developer's focus orderly and organized. Any



Production Control Analyst worth their “You Want It When?” poster will find both usability and utility in pointing their web browser to TWS/WebAdmin.

### ***Working with the Plan***

Working with the Plan through TWS/WebAdmin is no less rewarding. For the most part, working with the plan means monitoring the jobs, correcting abnormal endings (abends) when possible, resolving other scheduling problems when they occur, and escalating problem determination when necessary. (Or whatever those good folks really do 24/7!)

Again, I am only covering a few of the features - those that I feel are particularly impressive in the way they support the work Operators are trying to accomplish.

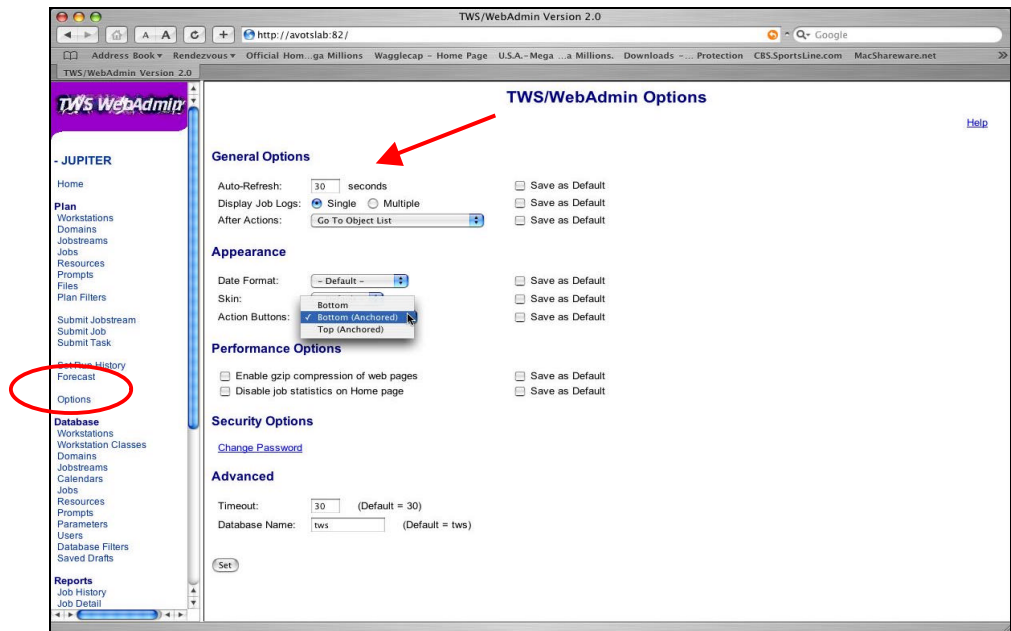
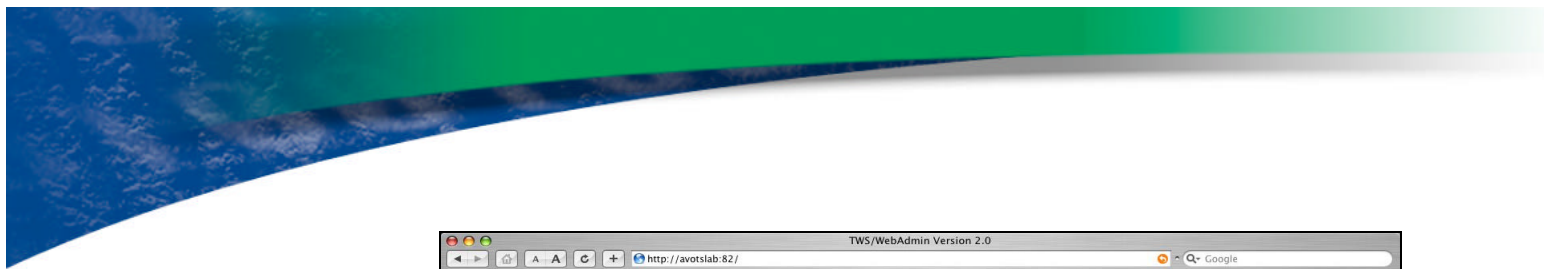
I have been to many, many shops and watched countless Operators in my TWS-related travels. There is no standard way Operators monitor TWS activity. Fact.

*TWS/WebAdmin supports the real work that Operators are trying to accomplish, making it easier, simpler and faster.*

All Operators like to filter what they watch and they all are on the lookout for abend or job failure instances. Some Operators also monitor critical jobstream execution. Operators pay very limited attention to successful job completion.

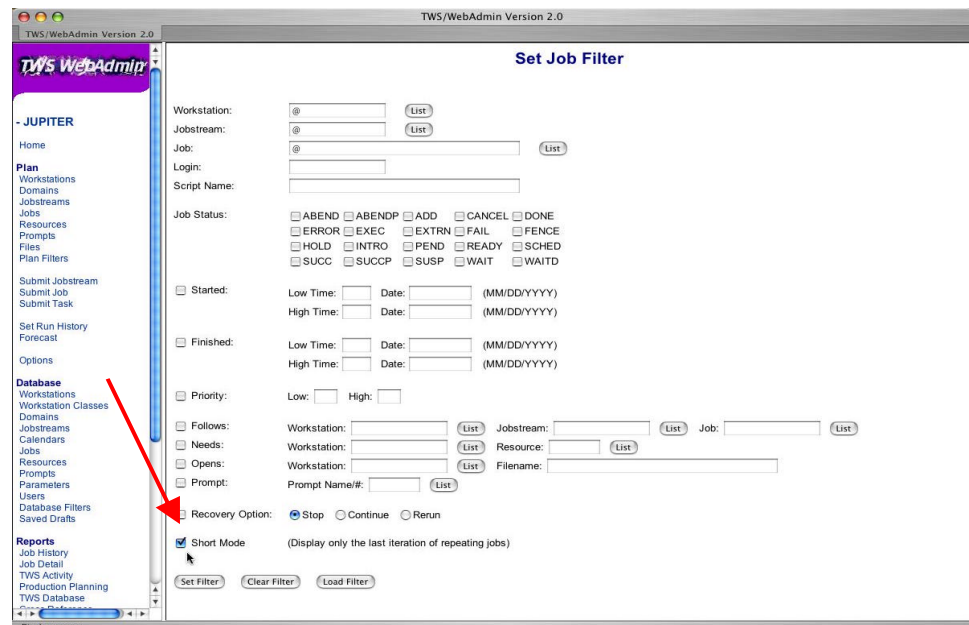
For the job monitoring exercise, I am setting two options from the Options screen. I am setting the Auto Refresh option to 30 so that the Plan screens will automatically refresh every 30 seconds for me. That mimics the way most Operations folks monitor TWS activity. (Actually, most Operators feel that real time is a bit too slow for their needs.)

I am also anchoring the Action Buttons along the bottom of the screen for quick access to operations-related activities. I’m willing to bet most Operators will also change the skin to 3-D!



### TWS/WebAdmin Options.

Filtering is an extremely powerful aid to job monitoring. TWS/WebAdmin has a full complement of filters. One really useful filtering option is to reduce the clutter by eliminating all but the last iteration of those jobs that run every few minutes throughout the day.



### Setting the Filter to Short Mode.

TWS/WebAdmin Version 2.0

**TWS WebAdmin**

**JUPITER**

Home  
Plan  
Workstations  
Domains  
Jobstreams  
Jobs  
Resources  
Prompts  
Files  
Plan Filters  
Submit Jobstream  
Submit Job  
Submit Task  
Set Run History  
Forecast  
Options  
Database  
Workstations  
Workstation Classes  
Domains  
Jobstreams  
Calendars  
Jobs  
Resources  
Prompts  
Parameters  
Users  
Database Filters  
Saved Drafts  
Reports  
Job History  
Job Detail  
TWS Activity  
Production Planning  
TWS Database  
Help

Planet	Jobstream	Status	Start Time	Run Time	Unmet Dependencies	Job ID	Estimated Run Time	Unmet Dependencies	Job ID
SATURN	TESTSTREAM	*****	06:01	00:01					
	TEST1	OK	06:01	00:01	0	#J1038			
SATURN	TESTSTREAM1	*****	09:00	(00:04)			<21:00 (OK)		
	TESTJOB1	OK	09:00	00:01	0	#J1039			
	>>every run	TESTJOB1	16:15	00:01	OK0	#J1158	<20:00 (OK0)	\$0	
		TESTJOB2	09:00	00:01	0	#J1040			
	>>every run	TESTJOB2	16:15	00:01	OK0	#J1159	<20:15 (OK0)	\$0	
		TESTJOB3	09:00	00:01	0	#J1041			
	>>every run	TESTJOB3	16:15	00:01	OK0	#J1160	<20:30 (OK0)	\$0	
		TESTJOB4	09:00	00:01	0	#J1042			
	>>every run	TESTJOB4	16:15	00:01	OK0	#J1161	<20:45 (OK0)	\$0	
SATURN	TULLYTEST_STREAM	*****	16:00	(00:04)					
	TESTJOB1	OK	16:00	00:01	0	#J1151			
		TESTJOB2	16:00	00:01	0	#J1156			
		TESTJOB3	16:00	00:02	0	#J1157			
		TESTJOB4	HOLD	(19:00)	(00:01)				
URANUS	IN YOUR FACE	*****	06:00				<23:05 (OK)		
	RUMMER	OK	06:00	00:01	5	#J9095			
	>>every run	RUMMER	16:00	00:02	5	#J15551	61:00		
URANUS	UR_STREAM	*****	HOLD	(06:30)	(00:01)		[Detail]	[Suppressed]	
	U_R_IT	OK	06:30	00:01	0	#J9381			
	>>every run	U_R_IT	15:20	00:01	0	#J15424	\$0:10		
URANUS	UR_ANOTHER_STREAM	*****	09:00	00:01					
	U_R_IT	OK	09:00	00:01	0	#J10908			

Check All - Clear All

Priority Release Cancel (NOP) Rerun Kill Confirm Tail Add Dep Delete Dep Successors Info Detail Report Notes Filter

Save Filter Layout Print

Display a menu

## Job List from the Plan.

In the Job List we see some of the detail TWS/WebAdmin provides. The color coded Status jumps out at you – white for pending, green for success and red for problems. The start time, estimated run time and unmet dependencies are what the Operators want to see and monitor. The fourth rule is at work here!

*Ultimately, usability is accomplished by architecting a user interface that fits with what the end user is trying to accomplish.*

While filtering out the multitude of iterations of jobs, which already have run, reduces the list to some extent, it still does not provide the visibility that I think a monitor would need for successful job performance.

I can further reduce the clutter of displayed data by filtering down to just the jobs that are executing and the problem jobs. TWS/WebAdmin provides the capability to define any number of filtering combinations and to save those definitions. TWS/WebAdmin will satisfy every operator's filtering desires and job execution monitoring requirements.

**Plan Job List**

Symphony: Current  
Filter: @#@+state=abend,abenp,error,exec,fail,fence,wait+short

Workstation	Jobstream	Job	State	Start	Elapse	Exit Code	Dependencies	Dependencies
JUPITER	URANUS1323409912	*****	EXEC	13:23	(01:02)			
		LONG_VIR	EXEC	13:23	(00:31)		#J23628	
		URANUS#RUNSAMHILE	WAIT	13:23	(00:31)		#J23630	
URANUS	IN_YOUR_FACE	*****	READY	06:00			<23:05 (OK)	
		SUMMER	ABEND	06:00	00:01	5	#J18607	
>>every run		SUMMER	ABEND	13:00	00:02	5	#J23266	#J1:00

Check All Clear All

Priority Release Cancel NOP Rerun Kill Confirm Tail Add Dep Delete Dep Successors Info Detail Report Notes Filter  
Save Filter Layout Print

## Plan Jobs Filtered to Executing and Abend Job View.

One very thoughtful feature of TWS/WebAdmin is showing the output of a job by clicking on the job name in the Plan's job list. This comes in very handy when there is a problem needing immediate attention.

**Plan Job List**

Symphony: Current  
Filter: @#@+state=abend,abenp,error,exec,fail,fence

Workstation	Jobstream	Job	State	Start	Elapse	Exit Code	Dependencies	Dependencies
URANUS	IN_YOUR_FACE	*****	READY	06:00			<23:05 (OK)	
		SUMMER	ABEND	06:00	00:01	5	#J27923	
							#J28490	
							#J29125	
							#J29915	
							#J30608	
							#J31310	
							#J32022	#J1:00

View Standard List

```

=====
= JOB       : URANUS#IN_YOUR_FACE.SUMMER
= USER      : maestro          TWS User,Avots
= JCLFILE    : sleep 30; exit 5
= Job Number: 30608
= Fri 04/15/05 10:00:33 EDT
=====
TWS for UNIX/JOBMANRC 8.2
AMSBJA001I Licensed Materials Property of IBM
5698-WKB
(C) Copyright IBM Corp 1998,2003
US Government User Restricted Rights
Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
AMSBIS307I Starting /usr/TWS/maestro/jobmanrc sleep
=====
TWS for UNIX/JOBINFO 8.2 (9.5)
Installed for user 'maestro'.
Locale LANG set to "C"
=====
= Exit Status      : 5
= System Time (Seconds) : 0      Elapsed Time (Minutes) : 0
= User Time (Seconds) : 0
= Fri 04/15/05 10:01:04 EDT
=====

```

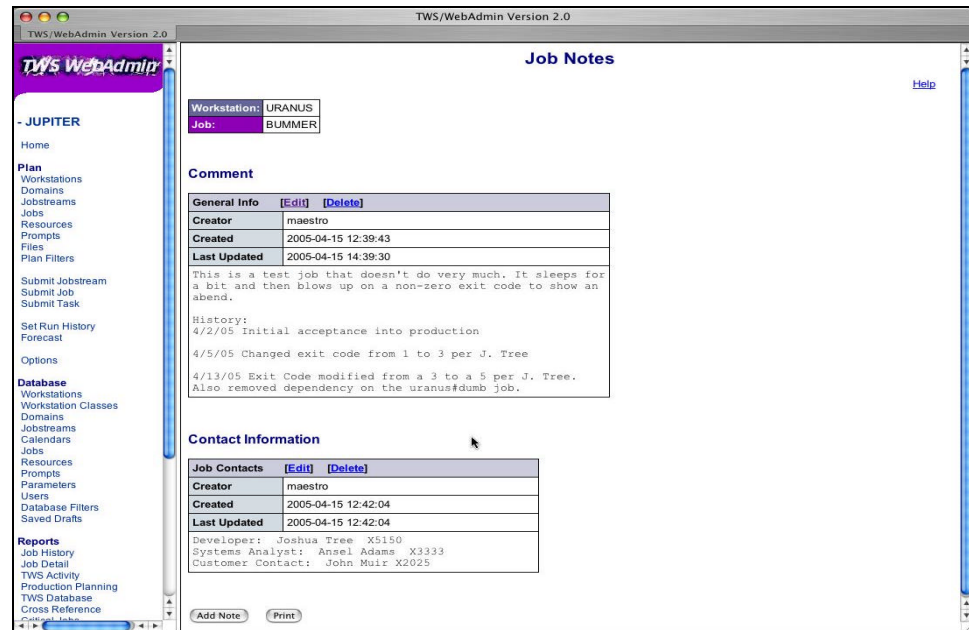
Print Close Window

Successors Info Detail Report Notes Filter

## Clicking On Job Name Immediately Pops Up Job's Output.



One TWS/WebAdmin feature that I feel has great utility is the Job Notes feature. It is a handy addition to TWS capabilities in maintaining all sorts of information for jobs. There are separate sections for Contact Information, for Recovery Instructions and for General Comments. This is a feature with unlimited utility.




### Example of Job Notes.


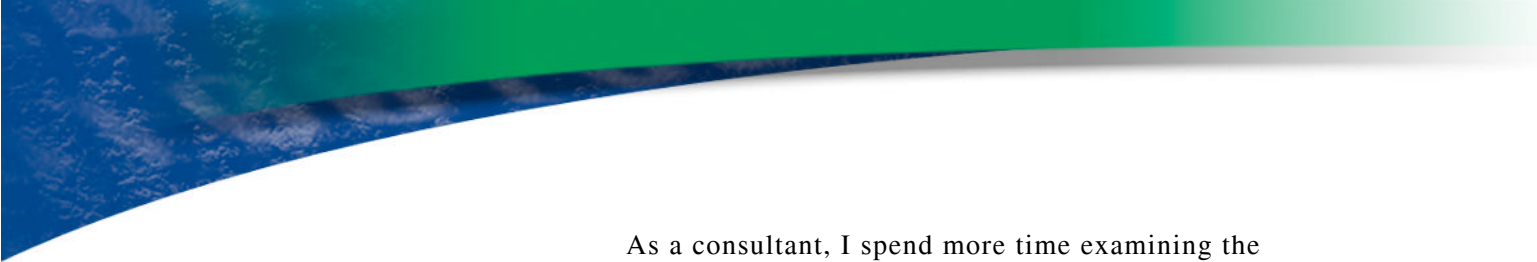
There is one TWS/WebAdmin feature that I am sure will be underappreciated and underutilized by the community in spite of being both useful and powerful. That is the ability to browse the TWS log file.

There are countless Schedulers and Operators working with TWS who are uncomfortable navigating in the command line oriented Unix environment or are not even allowed to have direct access to the Unix environment. Yet their TWS master resides on a Unix server.

The TWS log (located at *maestrohome/stdlist/ date\_ TWSMERGE.log* on a Unix master) contains all the activity regarding TWS processing. TWS/WebAdmin provides direct access to the TWS log file from the GUI.

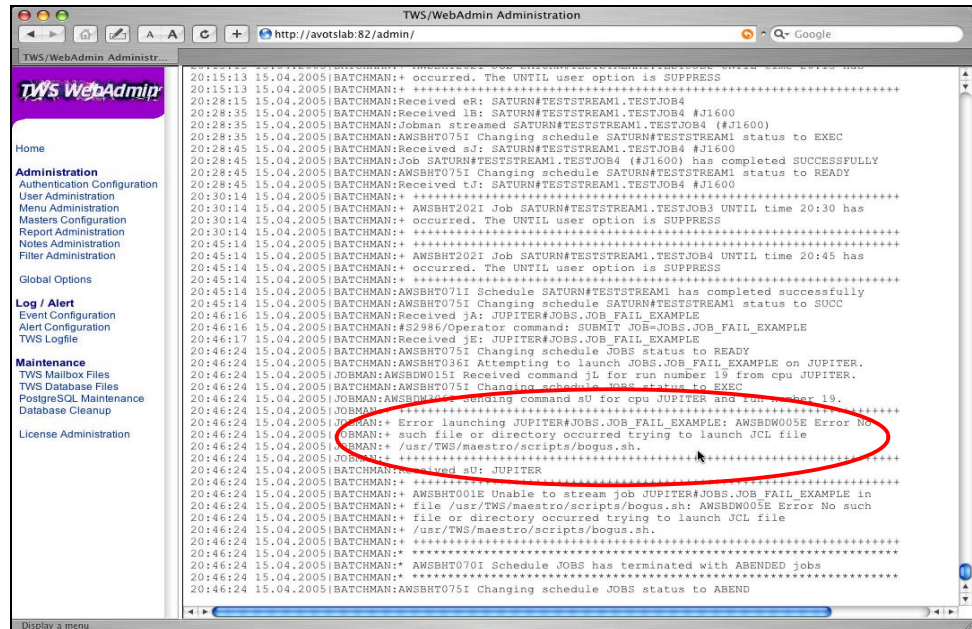


As a consultant, I spend more time examining the



As a consultant, I spend more time examining the





### ...The TWS Log Has the Answer.

I am convinced that Operators, as a class, will absolutely love using TWS/WebAdmin. TWS/WebAdmin has the utility Operators demand and the usability Operators crave. All five usability rules apply to working with the TWS Plan.

### Supporting the Administrator

TWS/WebAdmin supports the TWS Administrator with convenience and utility. I mentioned the TWS log earlier. When problems occur, it is the Administrator that has to determine their origin and cure. The TWS log allows a Scheduler or Operator to support this administration task.

Maintaining and reviewing audit files is another administration task. In their raw format audit files for both the Plan and the Database are difficult to decipher. (OK, they are really impossible to decipher.) TWS/WebAdmin eliminates counting bytes and trying to delimit fields.

As an example, assume an important jobstream was mysteriously cancelled. Selecting the Plan Audit Log provides the answer. The highlighted entry at the bottom of the list tells us user tullyk cancelled the jobstream.

TWS/WebAdmin Version 2.0

http://avotslab/82/

Q- Google

### TWS Plan Audit Log

[Help](#)

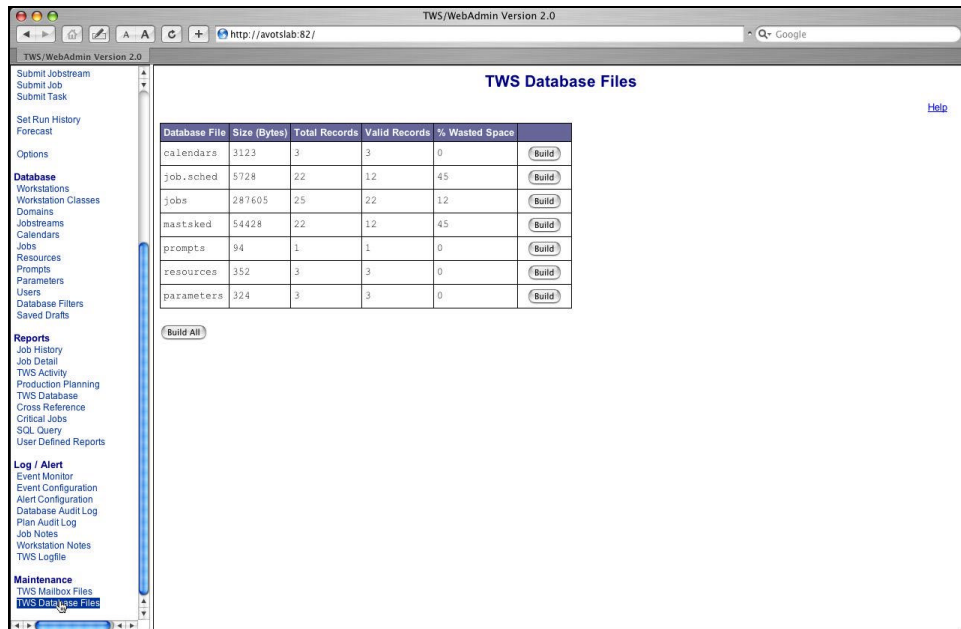
Filter: Today

Log Type	Timestamp	User ID	Framework User	Object Type	Object Name	Action Type	Workstation	Action Data
COMMAN	04/18/2005 05:59:14	maestro		PLWKSTN	SATURN	MODIFY		link @!@;noask
COMMAN	04/18/2005 05:59:15	maestro		PLWKSTN	JUPITER	MODIFY		stop @!@;wait:noask
STAGEMAN	04/18/2005 05:59:29	maestro		PLAN	/usr/TWS/maestro/schedlog/W200504180559	MODIFY	JUPITER	ANSBHV025I Old Symphony renamed /usr/TWS/ma
HEADER	04/18/2005 05:59:29	maestro		PLAN			JUPITER	Version=AL.0
STAGEMAN	04/18/2005 05:59:29	maestro		PLAN	/usr/TWS/maestro/Symphony	INSTALL	JUPITER	ANSBHV030I New Symphony file installed
STAGEMAN	04/18/2005 05:59:29	maestro		PLAN	/usr/TWS/maestro/Sinfonia	INSTALL	JUPITER	ANSBHV036I Multi CPU Symphony file copied t
COMMAN	04/18/2005 05:59:29	maestro		PLWKSTN	JUPITER	MODIFY		start
COMMAN	04/18/2005 11:20:34	maestro		PLJOB	JUPITER#TWS_MAINTENANCE	ADD		submit sched=JUPITER#TWS Maintenance
COMMAN	04/18/2005 11:39:32	maestro		PLJOB	JUPITER#TWS_MAI139325447	ADD		submit sched=JUPITER#TWS_MAINTENANCE;alias
COMMAN	04/18/2005 11:42:06	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST2	ADD		submit job=JUPITER#RMSTDLIST;into=JUPITER#T
COMMAN	04/18/2005 11:47:25	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	ADD		submit job=JUPITER#RMSTDLIST;into=JUPITER#T
COMMAN	04/18/2005 11:58:24	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	MODIFY		rerun job=JUPITER#TWS_MAINTENANCE.RMSTDLIST
COMMAN	04/18/2005 12:06:03	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	MODIFY		rerun job=JUPITER#TWS_MAINTENANCE.RMSTDLIST
COMMAN	04/18/2005 12:16:11	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	MODIFY		rerun job=JUPITER#TWS_MAINTENANCE.RMSTDLIST
COMMAN	04/18/2005 12:19:38	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	MODIFY		rerun job=JUPITER#TWS_MAINTENANCE.RMSTDLIST
COMMAN	04/18/2005 12:28:45	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	MODIFY		rerun job=JUPITER#TWS_MAINTENANCE.RMSTDLIST
COMMAN	04/18/2005 12:39:17	maestro		PLJOB	JUPITER#TWS_MAINTENANCE.RMSTDLIST3	MODIFY		rerun job=JUPITER#TWS_MAINTENANCE.RMSTDLIST
COMMAN	04/18/2005 14:08:36	maestro		PLWKSTN	JUPITER	MODIFY		stop
COMMAN	04/18/2005 14:09:07	maestro		PLWKSTN	JUPITER	MODIFY		start
COMMAN	04/18/2005 14:10:58	tullyk		PLJBSTRM	JUPITER#TWS_MAI139325447	MODIFY		cancel sched=JUPITER#TWS_MAI139325447

Filter Send CSV File Print

**TWS Plan Audit Log.**

Maintaining and compressing the TWS databases is an important part of TWS administration. Any change made to a scheduling object in the database grows that object's database file until it is rebuilt.



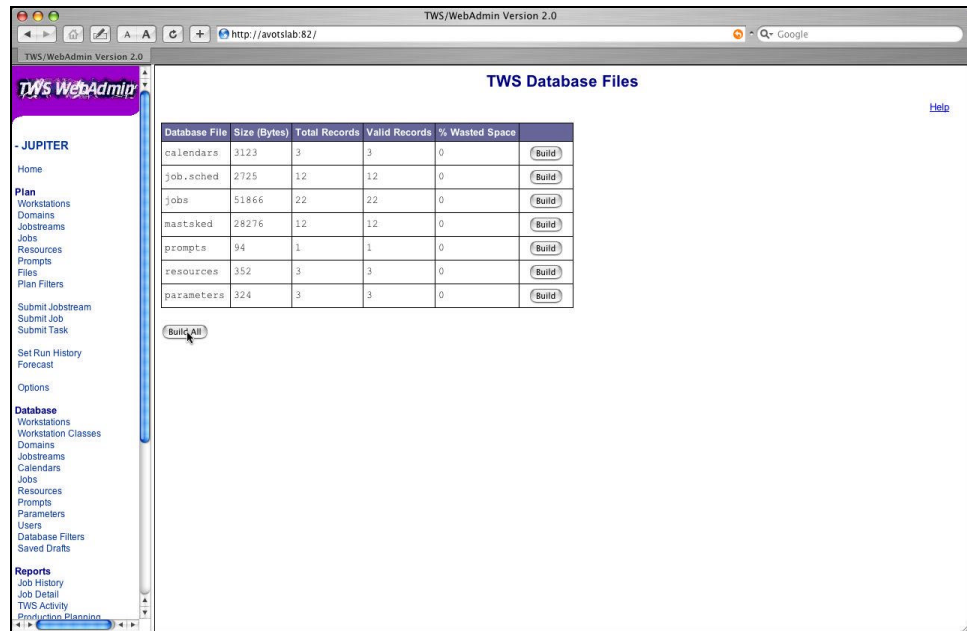
**TWS Database Files**

Database File	Size (Bytes)	Total Records	Valid Records	% Wasted Space	
calendars	3123	3	3	0	<a href="#">Build</a>
job.sched	5728	22	12	45	<a href="#">Build</a>
jobs	287605	25	22	12	<a href="#">Build</a>
mastasked	54428	22	12	45	<a href="#">Build</a>
prompts	94	1	1	0	<a href="#">Build</a>
resources	352	3	3	0	<a href="#">Build</a>
parameters	324	3	3	0	<a href="#">Build</a>

[Build All](#)

### TWS Database Before Rebuild.

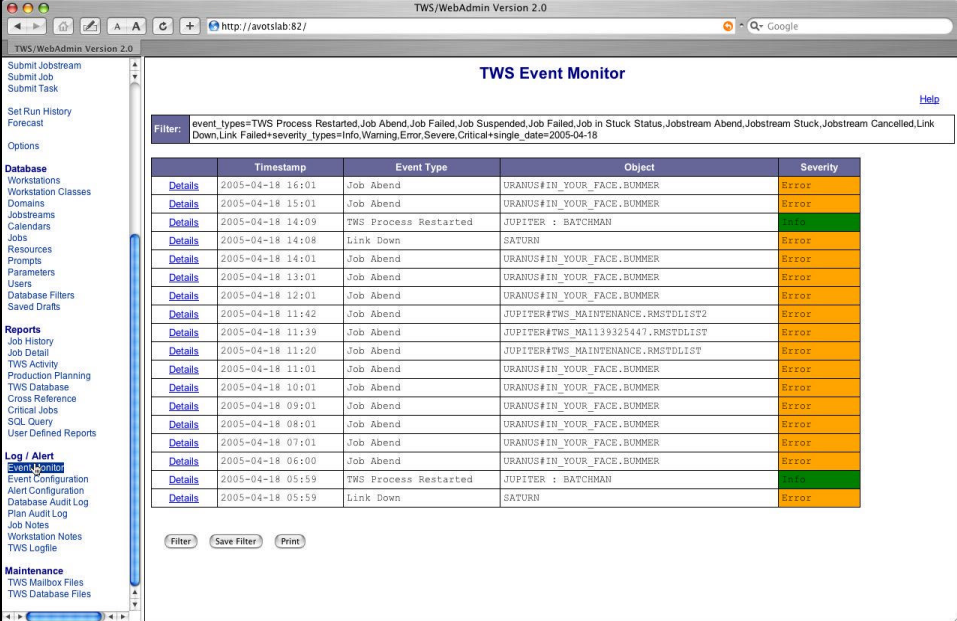
An overwhelming number of clients I visit for the first time are not aware of the importance of compressing the databases on a regular basis. The Schedulers do notice the degraded response when working with the database. TWS/WebAdmin informs the Administrator when rebuilding a database is in order.



### Clicking on "Build All" Reclaims the Wasted Space for the Files.

Not all TWS shops need/want the added expense of enterprise management software such as *Tivoli Enterprise Console* (TEC) or BMC's *Patrol*.

TWS/WebAdmin includes an Event Monitoring option. Event Monitoring is set up through a configuration screen where an Administrator selects which TWS events should be monitored. The selected events can be viewed directly or tied to an alert.



**TWS Event Monitor**

Filter: event\_types=TWS Process Restarted,Job Abend,Job Failed,Job Suspended,Job Failed,Job in Stuck Status,Jobstream Abend,Jobstream Stuck,Jobstream Cancelled,Link Down,Link Failed+severity\_types=Info,Warning,Error,Severe,Critical+single\_date=2005-04-18

	Timestamp	Event Type	Object	Severity
<a href="#">Details</a>	2005-04-18 16:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 15:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 14:09	TWS Process Restarted	JUPITER : BATCRMAN	Error
<a href="#">Details</a>	2005-04-18 14:08	Link Down	SATURN	Error
<a href="#">Details</a>	2005-04-18 14:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 13:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 12:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 11:42	Job Abend	JUPITER#TWS_MAINTENANCE.RMSTDLIST2	Error
<a href="#">Details</a>	2005-04-18 11:39	Job Abend	JUPITER#TWS_MAINTENANCE.RMSTDLIST	Error
<a href="#">Details</a>	2005-04-18 11:20	Job Abend	JUPITER#TWS_MAINTENANCE.RMSTDLIST	Error
<a href="#">Details</a>	2005-04-18 11:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 10:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 09:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 08:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 07:01	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 06:00	Job Abend	URANUS#IN_YOUR_FACE.BUMMER	Error
<a href="#">Details</a>	2005-04-18 05:59	TWS Process Restarted	JUPITER : BATCRMAN	Error
<a href="#">Details</a>	2005-04-18 05:59	Link Down	SATURN	Error

Filter Save Filter Print

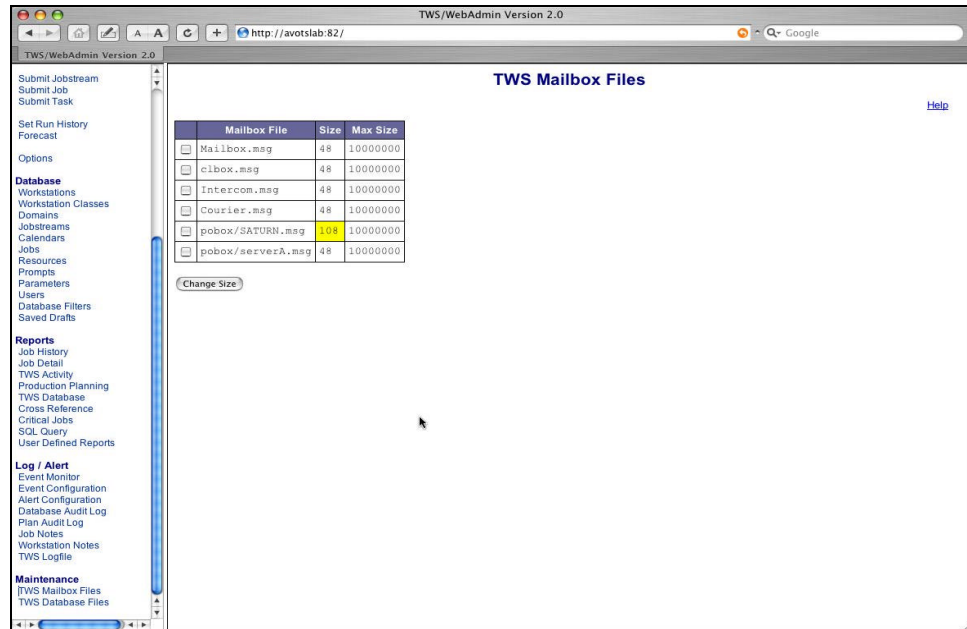
## TWS Event Monitor

If there is one thing that gets riveting attention focused on TWS, it is when TWS shuts down in the middle of the production day. The most common reason for the shutdown is a TWS message file filling to capacity. TWS uses message files to communicate between the master and agent servers.

Message files do not overflow very often—usually only when a server is taken out of service for an extended period of time and jobs are still scheduled to run there or when a server has an abnormal amount of activity.

When a message file does overflow, TWS shuts down immediately to prevent information loss. When that happens, the responsible Administrator scurries for notes to remember how to resize the files.

TWS/WebAdmin provides the capability to both monitor and resize the message files.



### TWS Message Files.

TWS Administrators will revel in the time savings TWS/WebAdmin provides. Providing a graphical interface for some of the administrative tasks is a novel and thoughtful advancement.

### Working With Reports

TWS report capability has been a sore spot with Operations Managers since forever. The standard TWS reports have been and still are poorly formatted and deemed not usable by everyone that has seen or tried to make sense of them.

*Good systems do not interfere with, or impede, efficient use by a skilled user having substantial experience with the system.*

TWS/WebAdmin provides a set of general reports that can be tailored to a shop's specific criteria. The report set includes Job History and Job Detail Reports, Scheduling Object Cross Reference Reports, and Scheduling Object Database Reports. Each of those reports can be tailored to include exactly what data a user wants to see. TWS/WebAdmin stores everything in it's own database.

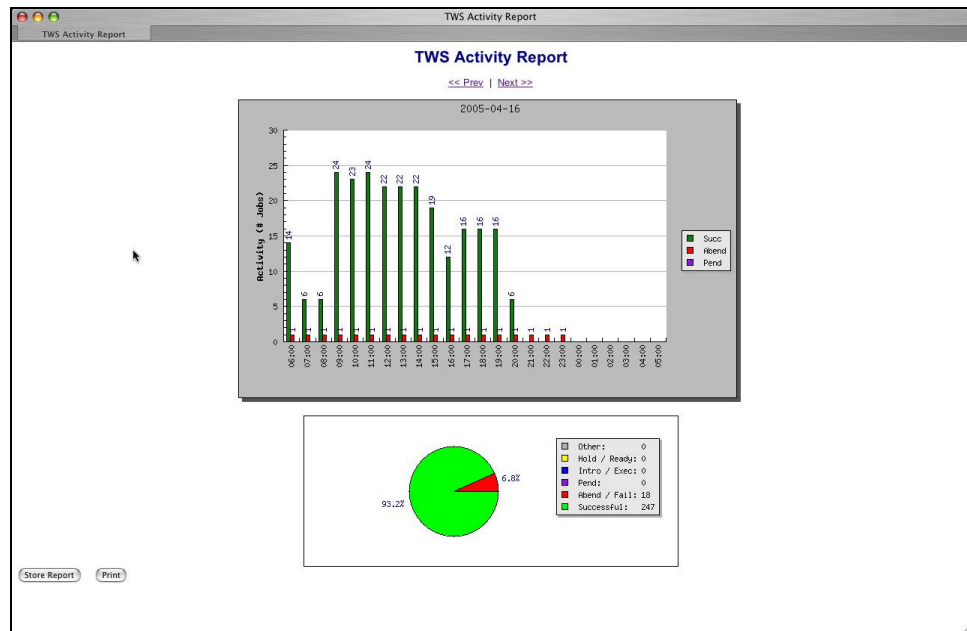
Just in case the provided reports are not sufficient, TWS/WebAdmin provides the capability to roll your



own SQL queries for reports. That seems like a useful capability so I tested it's utility. I studied the TWS/WebAdmin tables and formulated what I thought would be a useful report. I created a report of abend jobs.

It worked! I even saved the query. To my chagrin, I could have created the identical report with just a few appropriate clicks on the Job History Report screen! There are managers out there that can request something more esoteric to be reported, I am sure!

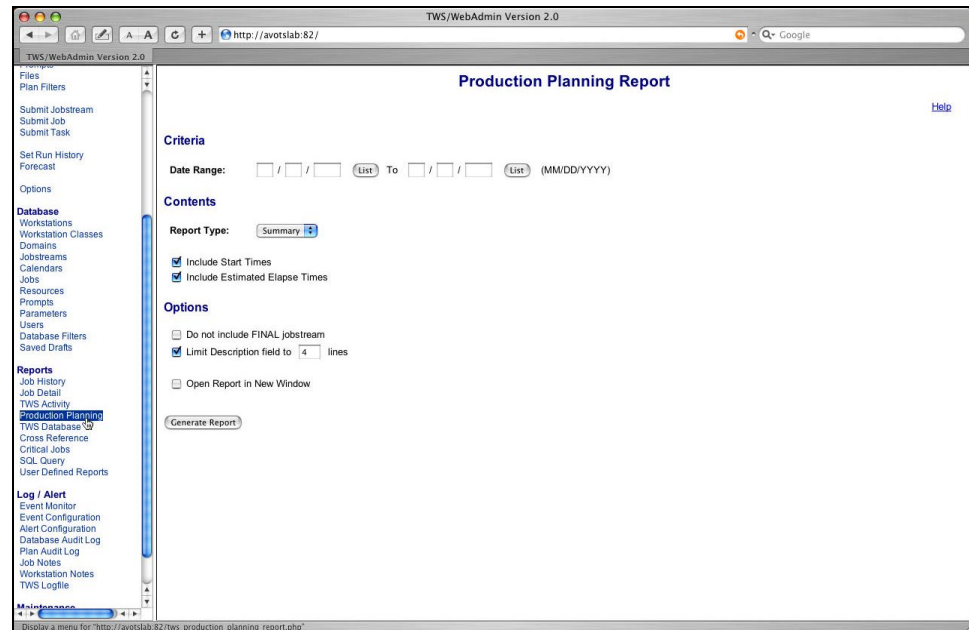
If there is one report that I have heard managers ask for regularly, it is the statistical report for TWS processing. Be it for each day, week, month, year, or all of the above - managers want that statistical data. It does not surprise to me to see that TWS/WebAdmin provides all that data with ease.



### A Day In the Life.

The clincher of the Reports usability for me personally, is the Forecasting Report. TWS/WebAdmin does forecasting beyond (my) expectations.

I install a forecasting report script at every client I visit. My script reports TWS Activity for a particular day. TWS/WebAdmin provides the capability to forecast a block of days. That block can be a weekend, a week a month, whatever. In all these years, I never considered doing that. What a fabulous idea!



### **Production Planning Report Input Page.**

Even for reporting, TWS/WebAdmin scores very high marks. The Usability Rules apply here as well.

## **In Summary**


*TWS/WebAdmin provides a user interface that fits with what the end user is trying to accomplish.*

TWS/WebAdmin is designed with the end user in mind.

TWS/WebAdmin is the solution for corporations committed to a productive environment for their TWS Schedulers, Operators and Administrators. Snappy response times for even the largest data loads and single-click information presentation provide quick problem resolution for critical path and mission critical application related problems.

The included RDBMS allows extracting every nugget of data or statistical information a manager could desire.





Intuitive and familiar interfaces allow Schedulers to confidently manage the corporation's scheduling data objects, ensuring the correct data is processed at the correct time of the correct day, with timely alerts generated before serious problems occur.

Administrators are provided with all the information to keep TWS "lean and mean" and avoid serious consequences like shut down. TWS/WebAdmin also provides the auditing information to know who did what to whom and when.

## **About SEGUS Inc.**

SEGUS Inc was founded in 1992 in Reston, Virginia and is the exclusive distributor in North America for all of HORIZONT GmbH's products - including TWS/WebAdmin.

Originally specializing only in products for and around IBM's mainframe job scheduling packages, SEGUS and HORIZONT have now entered the distributed arena and are also using their expertise to develop add-on products for other scheduling software.

For more information on SEGUS products, please go to [www.segus.com](http://www.segus.com) or call 1-800-327-9650 and tell them Tully sent you.