

Tidal Enterprise Scheduler™ Playbook

Monitoring Files on Remote FTP Servers

Date published: 06/20/07

Tidal Enterprise Scheduler™ v5.3

Radically | Simplify™ IT Operations

Monitoring Files on Remote FTP Servers

Tidal Enterprise Scheduler™ Playbook

Abstract

Processing a file when it becomes available on an external FTP server is a common data processing requirement that does not have an obvious scheduling solution. The problem is that without a Tidal agent to monitor the availability of files on an external FTP server, an alternative mechanism is required to detect these files to satisfy Scheduler job dependencies and/or trigger file events.

For example, an accounting department needs to download bank transactions as soon as they are made available from a secure FTP server to expedite reconciliations. This process is critical to generate exception reports as soon as possible to avoid severe financial penalties caused by delays in postings. For obvious reasons, as a customer, the company requiring the financial data cannot install monitoring software on the remote server owned by the bank.

This playbook illustrates a method for building a job sequence that determines when to download a file from an external FTP server when that file becomes available. In addition, this playbook illustrates how to build a notification mechanism to alert if the file has not arrived by an expected time (but allows the process to continue trying to find/process the file). This is all accomplished by building a flow of sequential jobs where the first job determines when the file is available, the second job gets the file, and the third job processes the file.

Features Employed

- Business Views
- SFTP job
- Job Tracking – Scan Output for Normal String
- Rerun a job via Job Event and Action
- “Do not timeout” option for trigger
- Send notification via Job Event and Action

Highlights

- Determine file availability
 - ✓ Define SFTP directory listing job
 - ✓ Use alternate “job tracking” method
 - ✓ Dynamic scheduling – rerun the job if necessary via Job Event and Action
- Retrieve file
 - ✓ Define SFTP “get” job
- Process file
 - ✓ Define file processing job

- Provide notification if job has not run by specified time
 - ✓ “Do not timeout” at end of time window (trigger only)
 - ✓ Job Event and Action

Requirements

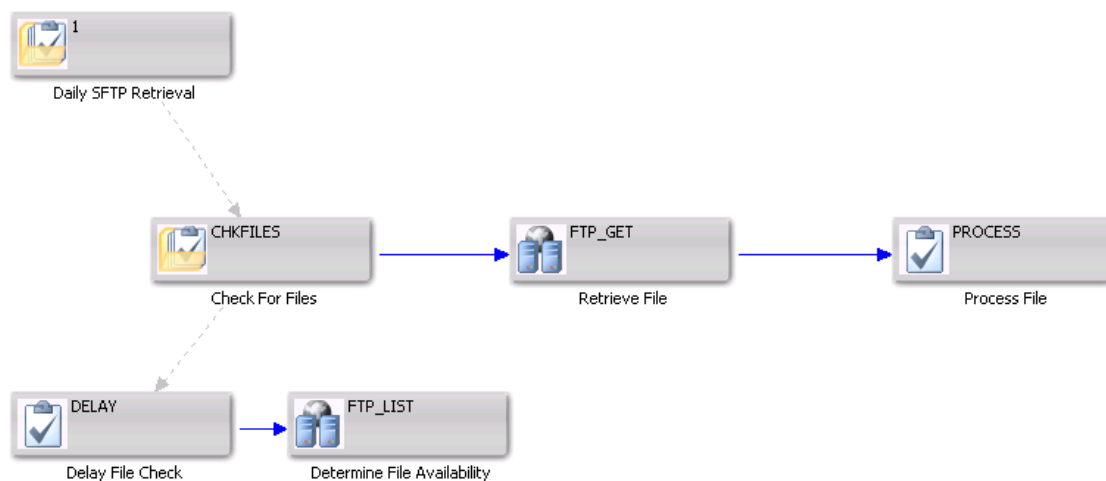
This playbook was created using Tidal Enterprise Scheduler v5.3 (Windows 2003 SP1/MSSQL 2000).

Play by Play

Define Business View for SFTP Download Process

1. Create a job group for a simple daily SFTP Retrieval process and create a business view from the group.
 - a. In the group view, add a subgroup containing one OS job and one FTP job.
 - b. To the main group, add an FTP job and an OS job, and create dependencies for the normal run as shown below. In this example, *Process File* depends on *Retrieve File*, which depends on the *Check For Files* (job group). *Check For Files* (job group) contains *Determine File Availability*, which depends on *Delay File Check*.

Daily SFTP Retrieval



Fill In Job Details (Normal Run)

Next, fill in the job details of the above steps for a “normal” run assuming for the moment that the desired file is present.

2. Define a job group, *Check For Files*, as a subgroup of *Daily SFTP Retrieval*.

The screenshot shows the 'Job Group Definition (Edit Mode)' dialog box for the job group 'Check For Files'. The dialog is titled 'Job Group Definition (Edit Mode) - \Daily SFTP Retrieval [Schedulers]\Check For Files'. It has a 'Schedule' tab selected, with other tabs including 'Run', 'Dependencies', 'Variables', 'Job Events', 'Group's Events', 'Options', 'Runbook', 'Notes', 'History', and 'Images'. The 'Job Group Name' is 'Check For Files', 'Job Class' is empty, 'Owner' is 'Schedulers', and 'Parent Group' is '\Daily SFTP Retrieval'. The 'Calendar' section shows 'Calendar Name' as 'Daily', 'Offset' as '0', and 'Inherited' checked. The 'Time Window' section shows 'Only start the job between the hours of:' with empty fields, 'and', and 'Inherited' checked. Below this is 'If not ready by end of day:' with a checkbox for 'Run again tomorrow' which is unchecked. The 'Repeats' section shows 'After job execution:' with three radio buttons: 'Do not repeat' (selected), 'Run new occurrence', and 'Rerun same occurrence'. Below this is 'Repeat every:' with empty fields for 'minutes' and 'times'. At the bottom left is a checkbox for 'Enabled' which is checked. At the bottom right is the text 'Last Modified : 6/21/2007 11:33 AM'. There are 'OK' and 'Cancel' buttons in the top right corner.

- Define job details for the first job, *Delay File Check*. This is a simple OS job that runs a sleep command for a specified interval to avoid performance issues. In this example, the delay interval is stored as a TIDAL variable that can be throttled globally.

The screenshot shows the 'Job Definition' dialog box for a job named 'Delay File Check'. The dialog has a title bar with the path: 'Job Definition - \Daily SFTP Retrieval [Schedulers]\Check For Files [Schedulers:\Daily SFTP Retrieval]\Delay File Ch...'. The 'Job Name' field is 'Delay File Check'. The 'Job Class' is set to 'Schedulers'. The 'Parent Group' is '\Daily SFTP Retrieval\Check For Files'. Below these fields are tabs for 'Program', 'Schedule', 'Run', 'Dependencies', 'Resources', 'Job Events', 'Options', 'Runbook', 'Notes', 'History', and 'Images'. The 'Command' field contains 'C:\Program Files\TIDAL\Agent\Bin\SLEEP.EXE'. The 'Command Parameters' field contains '<Delay Interval.1>'. There are 'OK' and 'Cancel' buttons on the right.

- Define job details for the next job, *Determine File Availability*, which is an SFTP job. Unlike the typical FTP job, the purpose is not to transfer a file, but to use FTP to do a remote directory listing to determine file availability.

The screenshot shows the 'FTP Job Definition' dialog box for a job named 'Determine File Availability'. The dialog has a title bar with the path: 'FTP Job Definition - \Daily SFTP Retrieval [Schedulers]\Check For Files [Schedulers:\Daily SFTP Retrieval]\Determi...'. The 'FTP Job Name' is 'Determine File Availability'. The 'Job Class' is 'Schedulers'. The 'Parent Group' is '\Daily SFTP Retrieval\Check For Files'. Below these are tabs for 'FTP', 'Schedule', 'Run', 'Dependencies', 'Resources', 'Job Events', 'Options', 'Runbook', 'Notes', 'History', and 'Images'. The 'FTP' tab is active, showing 'Protocol' as 'SFTP' and 'FTP Operation' as 'LIST Directory'. The 'Format' field is empty. The 'FTP File/Directory Details' section has a 'Remote Path' field containing '/incoming'. The 'SFTP Options' section has 'SSH2 only' selected. The 'Authentication Type' section has 'User/Password' selected. There is an 'Advanced...' button and a 'Variables' dropdown. At the bottom, there is a checkbox for 'Enabled' (checked) and a timestamp 'Last Modified : 6/21/2007 11:34 AM'. 'OK' and 'Cancel' buttons are on the right.

5. The directory listing in the job's output will be scanned for the file name of interest.

FTP Job Definition - \Daily SFTP Retrieval [Schedulers]\Check For Files [Schedulers:\Daily SFTP Retrieval]\Determi...

FTP Job Name: Determine File Availability

Job Class: [] Owner: Schedulers

Parent Group: \Daily SFTP Retrieval\Check For Files

FTP | Schedule | Run | Dependencies | Resources | Job Events | Options | Runbook | Notes | History | Images

Agent/Adapter Information

Agent (Local Host): REM-KLARSON-CON (Windows) Inherited ☐

Local User: [] ☐ Agent List

FTP Host: ftp.tidalsoft.com

FTP User: sftpuser ☐ Anonymous

Tracking

Use: ☐ Exit Code ☐ External ☒ Scan Output: Normal String(s) ☐ Scan Output: Abnormal String(s)

Normal String: banktrans.dat

Duration (in minutes)

Estimated: 1:00 Minimum: 1:00 Maximum: 1:00

☒ Exclude Completed Abnormally

☒ Enabled Last Modified : 6/21/2007 11:34 AM

- Define the FTP job details for *Retrieve File* that does the actual file transfer; the **Run** tab will be the same as the previous job that did the directory listing.

The screenshot shows the 'FTP Job Definition' dialog box for a job named 'Retrieve File'. The 'Run' tab is selected. The 'Protocol' is set to 'SFTP' and the 'FTP Operation' is 'PUT File'. The 'Format' is set to 'Binary'. The 'FTP File/Directory Details' section shows 'Filename' as 'banktrans.dat', 'New Filename' as 'banktrans.dat', 'Local Path' as 'D:\data', and 'Remote Path' as '/incoming'. The 'SFTP Options' are set to 'SSH2 only'. The 'Authentication Type' is 'User/Password'. The 'Variables' dropdown is empty. The 'Enabled' checkbox is checked. The 'Last Modified' date is 6/21/2007 11:26 AM.

- Finally, define the details for *Process File*, a standard OS job (in this case, Windows) that runs a script to process the incoming file.

The screenshot shows the 'Job Definition' dialog box for a job named 'Process File'. The 'Run' tab is selected. The 'Command' is 'D:\scripts\processTransactions.bat' and the 'Command Parameters' are 'banktrans.dat'. The 'Enabled' checkbox is checked. The 'Last Modified' date is 6/21/2007 11:26 AM.

Define a Job Event to Rerun *Check For Files* until File is Present

If the file is not available, the job group *Check For Files* will complete abnormally and the rest of the processing will not occur. In order to configure the jobs to continue polling for the file until it is present, and then proceed processing, we need to use a job event.

8. Open the job group definition for *Check For Files* to define a job event. This event should be triggered if the job group completes abnormally, with an associated action to rerun the job an unlimited number of times.
 - a. Go to the **Job Events** tab, select **Insert...**, then right-click and choose **Add Job Event**.

The screenshot shows the 'Job Group Definition (Edit Mode)' window for the job group 'Check For Files'. The window has a blue title bar and a standard Windows interface. At the top, there are fields for 'Job Group Name' (set to 'Check For Files'), 'Job Class' (a dropdown menu), 'Owner' (set to 'Schedulers'), and 'Parent Group' (set to '\Daily SFTP Retrieval'). To the right of these fields are 'OK' and 'Cancel' buttons. Below the fields is a tabbed interface with the following tabs: 'Schedule', 'Run', 'Dependencies', 'Variables', 'Job Events' (which is currently selected), 'Group's Events', 'Options', 'Runbook', 'Notes', 'History', and 'Images'. The 'Job Events' tab contains two empty tables, each with columns 'Event Name' and 'Event Trigger'. To the right of these tables are buttons for 'Insert...', 'Edit...', 'Delete', 'Clear', 'Move Up', and 'Move Down'. At the bottom left, there is a checkbox labeled 'Enabled' which is checked. At the bottom right, it says 'Last Modified : 6/21/2007 11:33 AM'.

- b. Choose the *Job completed abnormally* event trigger and choose the *Rerun* action.

Job Event Definition (Insert Mode)

Event Name: Rerun File Existence Check

Event Trigger: Job completed abnormally

Owner: Schedulers

Apply this event to all jobs: ☐

Associated Job(s) | Associated Action(s) | Description | Trigger History

Selected Action(s):

- Rerun

Available Action(s):

- Cancel Pending
- Cancel/Abort
- Hold/Stop
- Override
- Release/Resume
- Set To Completed Abnormally
- Set To Completed Normally
- Set To Error Status
- Set To External Status
- Set To Orphaned Status
- Set To Skipped Status

☒ Public ☒ Enabled

- c. When you choose the *Rerun* action, you are prompted for **Maximum Reruns**. Click **OK** to accept the unlimited rerun value of 0.

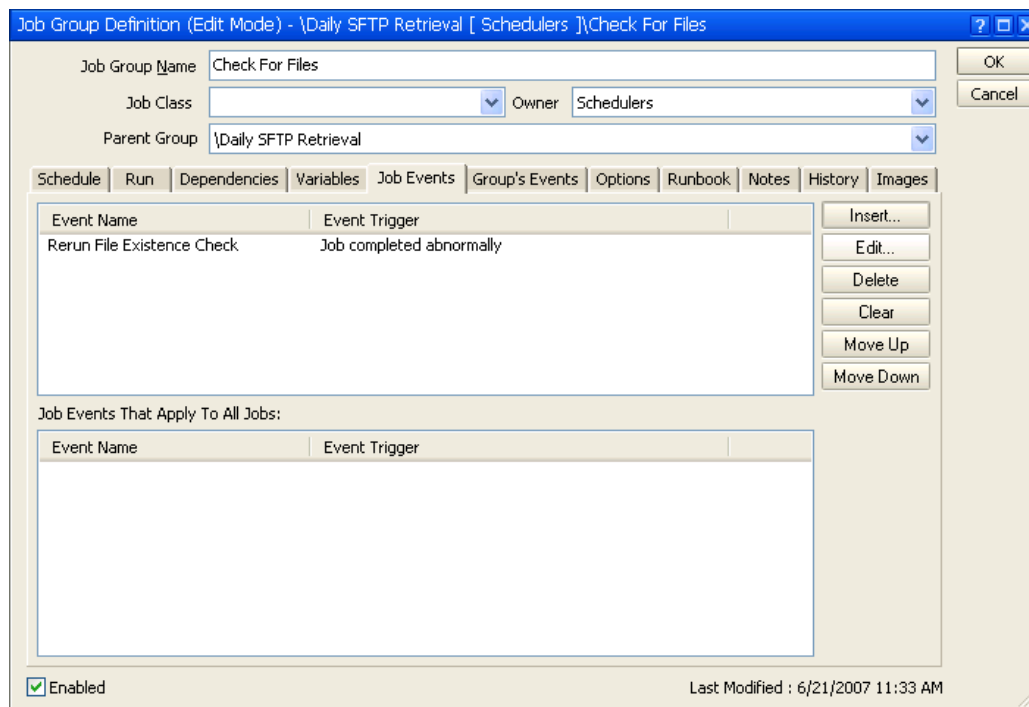
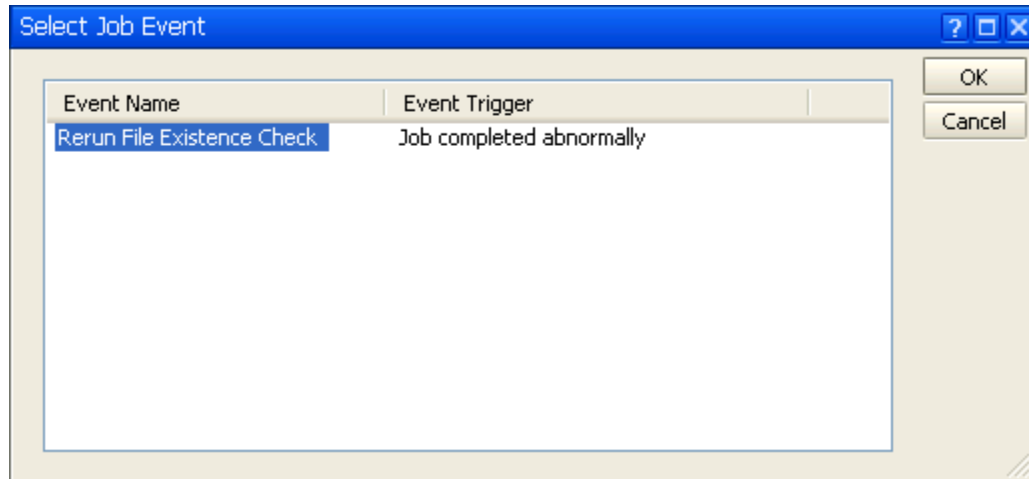
Event Parameter

Maximum Reruns: 0

(0 indicates unlimited)

OK Cancel

Next, select the new job event to associate it with the FTP directory job.



Set up Notification for File not Arriving by Specified Time

Under the current setup, our job sequence will continue polling for the file all day, and if the file still has not arrived by the end of the day, the process will end without any notification. Using the end of time window trigger only feature (“do not timeout”), you can set a “checkpoint” time to notify you or others if the file has not arrived by the specified time. The process will not timeout, but continue to check for file availability.

9. Open the job definition for *Retrieve File* and set as the end time for the job the time you want to be notified. Also select the “Do not time out” option. This causes our end time to be a “soft” end time – an event that we can capture, but will not cause the job group to stop running.

FTP Job Definition - \Daily SFTP Retrieval [Schedulers]\Retrieve File

FTP Job Name: Retrieve File

Job Class: Owner Schedulers

Parent Group: \Daily SFTP Retrieval

Calendar: Inherited ☒

Calendar Name: Daily

Offset: 0

From: to Forecast

Time Window: Inherited ☐

Only start the job between the hours of: and 5:00 PM

If not ready by end of time window: ☒ Do not time out

☐ Run again tomorrow

Repeats:

After job execution: ☒ Do not repeat

☐ Run new occurrence

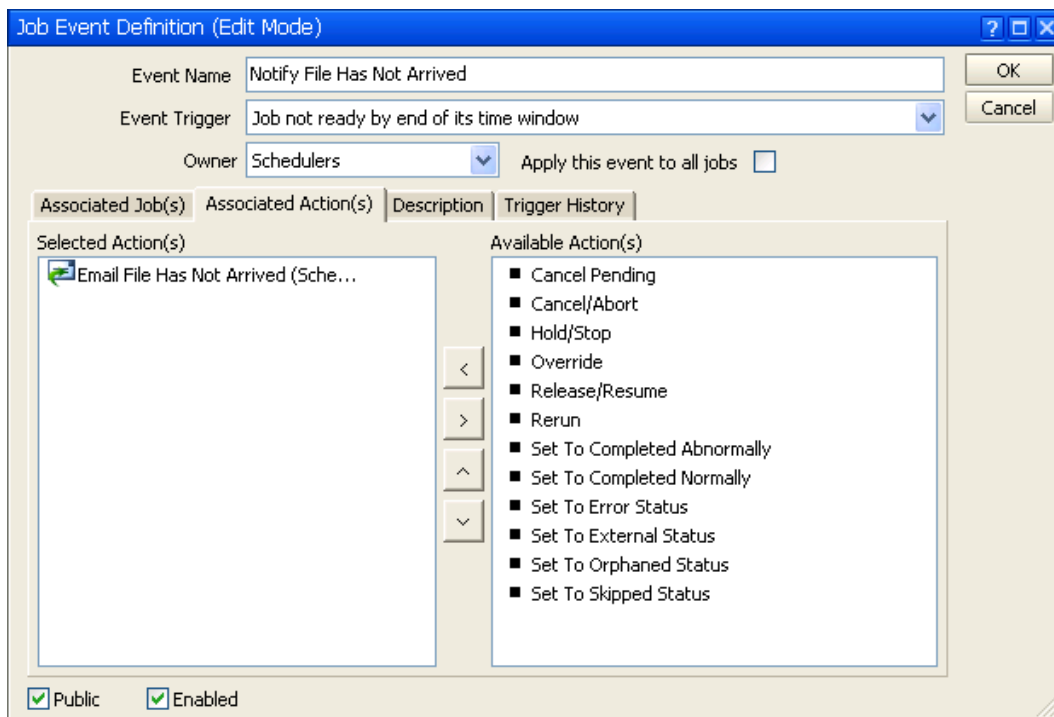
☐ Rerun same occurrence

Repeat every: minutes, up to times

☒ Enabled

Last Modified : 6/21/2007 11:29 AM

10. Define a job event to send an email if the job has not yet run by the end of its time window. Note that because we have selected “Do not timeout”, we are able to get our notification at the desired time, and our process is allowed to continue. The process will continue checking for file existence, and if the file is found, download and process the file.



Job Event Definition (Edit Mode)

Event Name: Notify File Has Not Arrived

Event Trigger: Job not ready by end of its time window

Owner: Schedulers

Apply this event to all jobs: ☐

Associated Job(s) | Associated Action(s) | Description | Trigger History

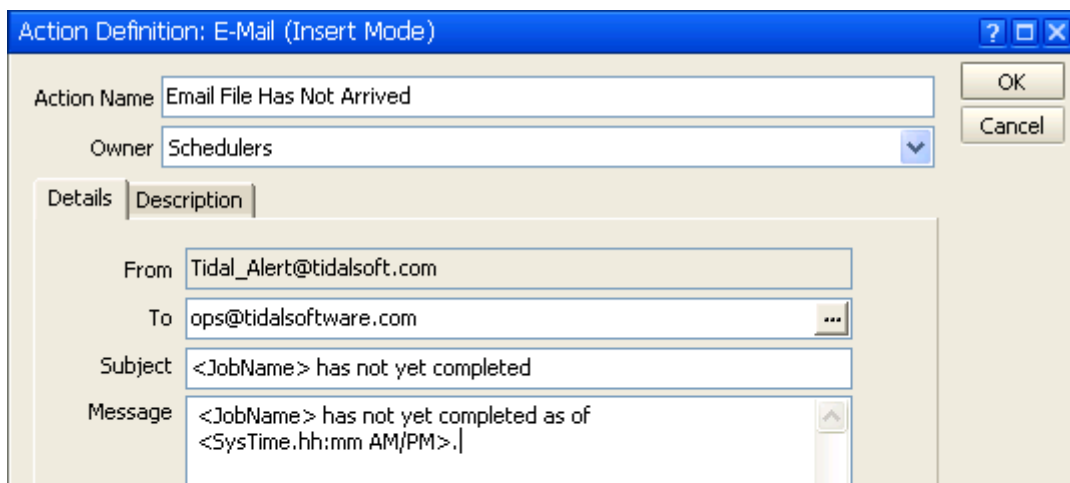
Selected Action(s):

- Email File Has Not Arrived (Sche...

Available Action(s):

- Cancel Pending
- Cancel/Abort
- Hold/Stop
- Override
- Release/Resume
- Rerun
- Set To Completed Abnormally
- Set To Completed Normally
- Set To Error Status
- Set To External Status
- Set To Orphaned Status
- Set To Skipped Status

☒ Public ☒ Enabled



Action Definition: E-Mail (Insert Mode)

Action Name: Email File Has Not Arrived

Owner: Schedulers

Details | Description

From: Tidal_Alert@tidalsoft.com

To: ops@tidalsoftware.com

Subject: <JobName> has not yet completed

Message: <JobName> has not yet completed as of <SysTime.hh:mm AM/PM>.

FTP Job Definition - \Daily SFTP Retrieval [Schedulers]\Retrieve File

FTP Job Name: Retrieve File

Job Class: [] Owner: Schedulers

Parent Group: \Daily SFTP Retrieval

FTP | Schedule | Run | Dependencies | Resources | Job Events | Options | Runbook | Notes | History | Images

Event Name	Event Trigger
Notify File Has Not Arrived	Job not ready by end of its time window

Insert...
Edit...
Delete
Clear
Move Up
Move Down

Job Events That Apply To All Jobs:

Event Name	Event Trigger
------------	---------------

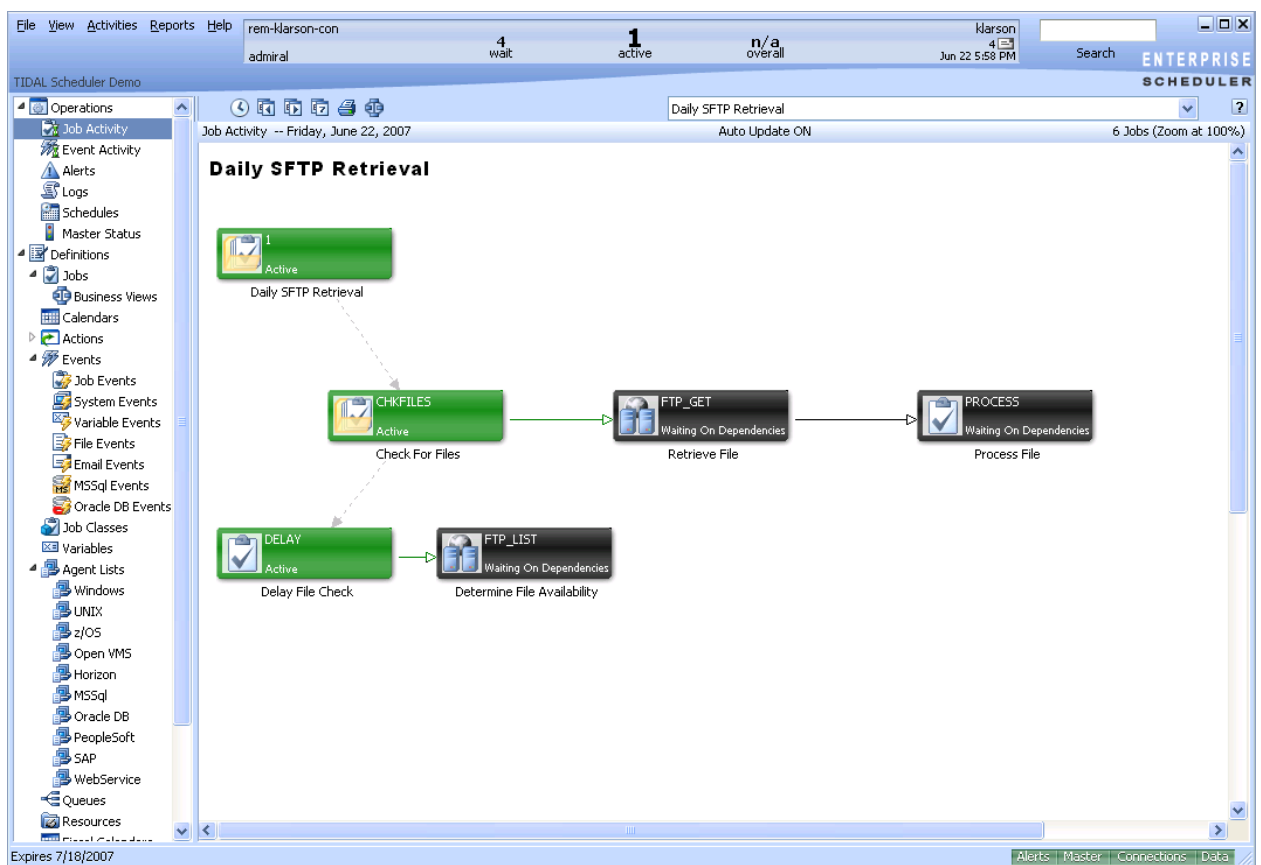
☒ Enabled

Last Modified : 6/21/2007 11:29 AM

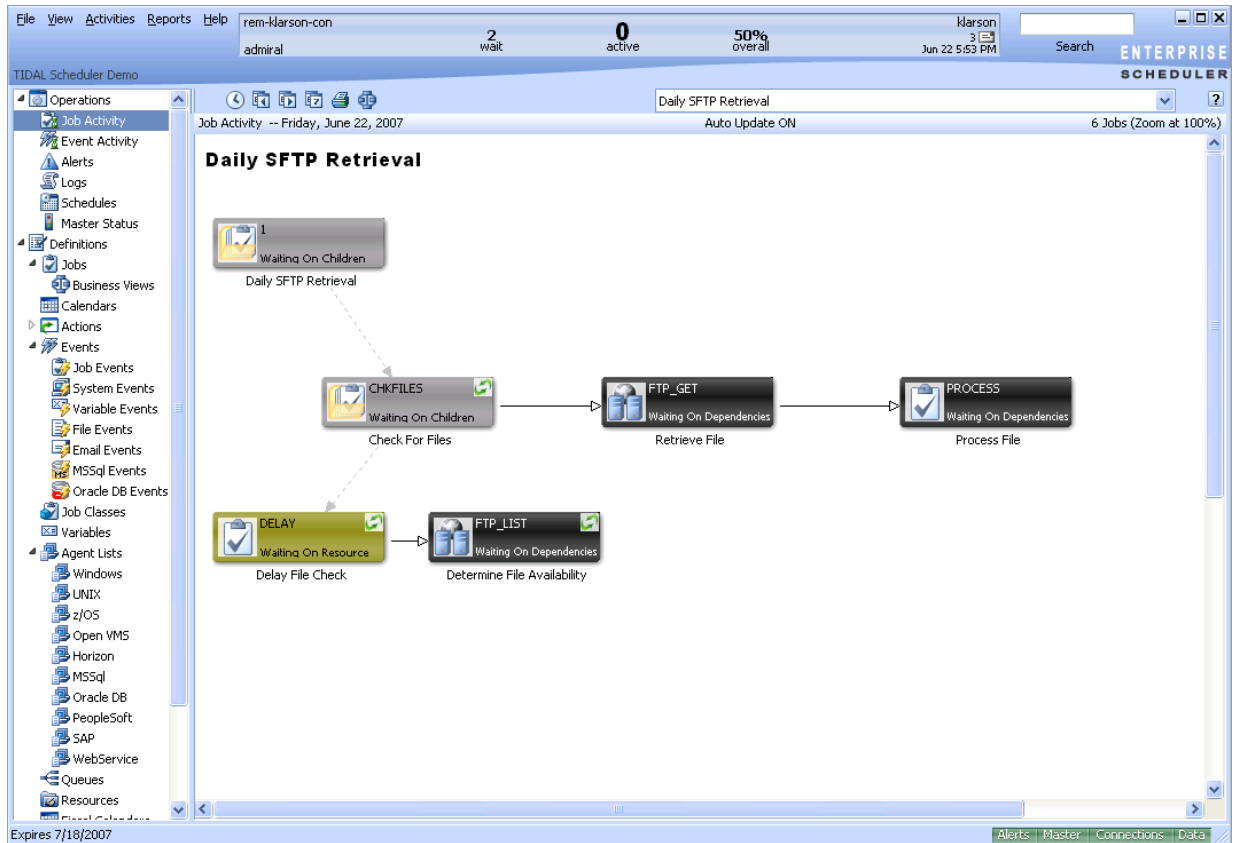
Monitoring Process Execution

During process execution, you can use the real-time business view and job details to monitor the progress of this process. The following is an example of the above process at runtime.

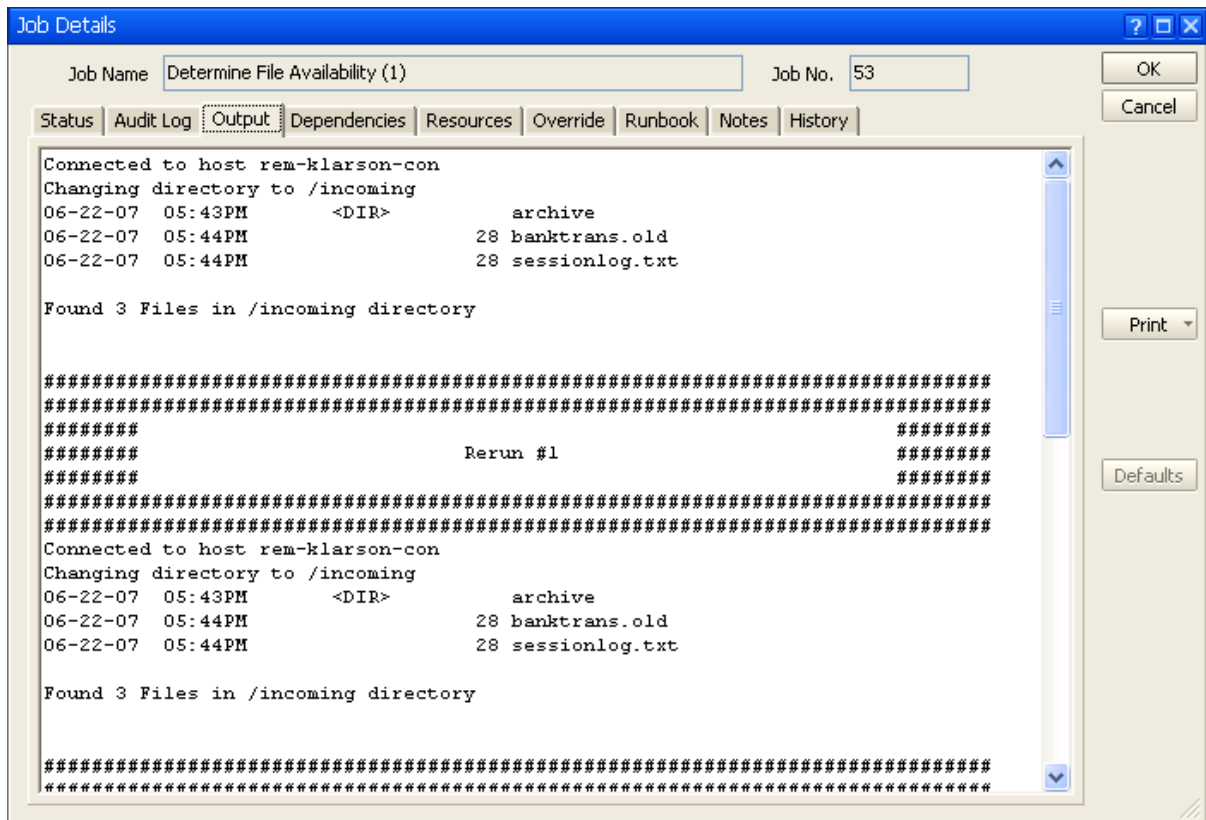
1. Observing the process during execution provides a graphical representation of current progress.



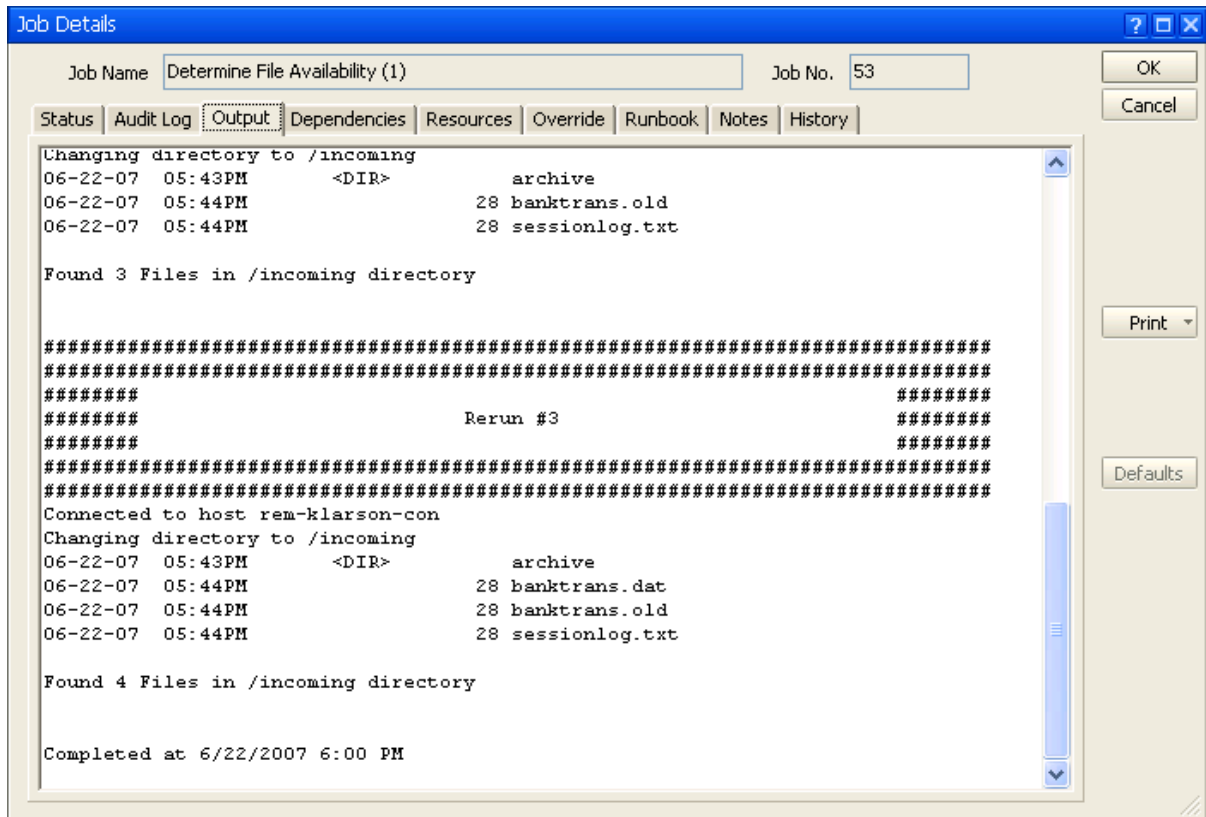
- Note that when *Determine File Availability* ends with a *Completed Abnormally* status, the job group *Check For Files* immediately is set to rerun.



- Looking at the Output tab of *Determine File Availability*, you see the appended output of each directory listing command. (If you don't want a running history, use the job option to replace rather than append output).

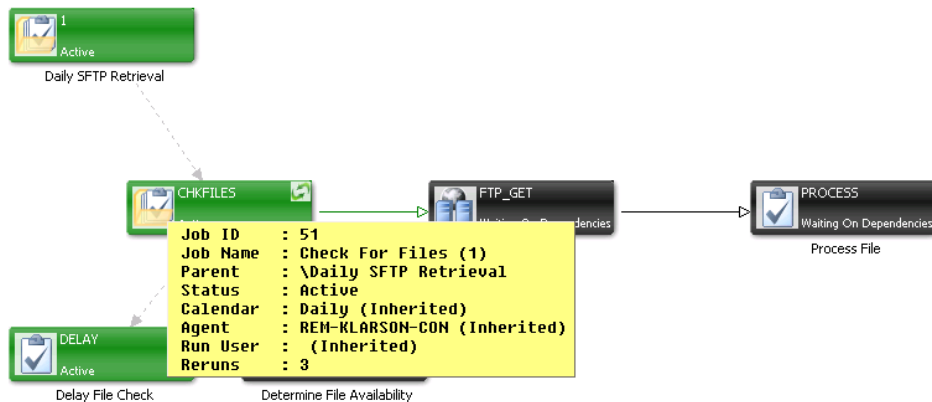


4. The output of a successful run might look as follows.

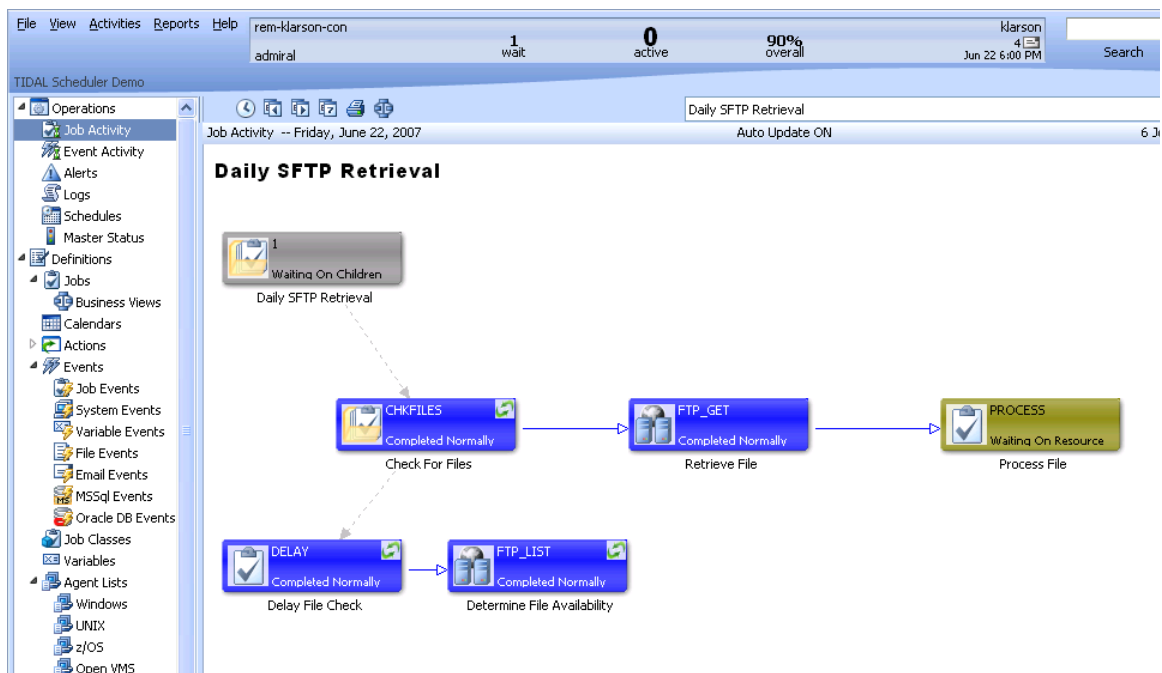


- Note that during execution you can move the mouse over the *Check For Files* job group to quickly view how many reruns have occurred.

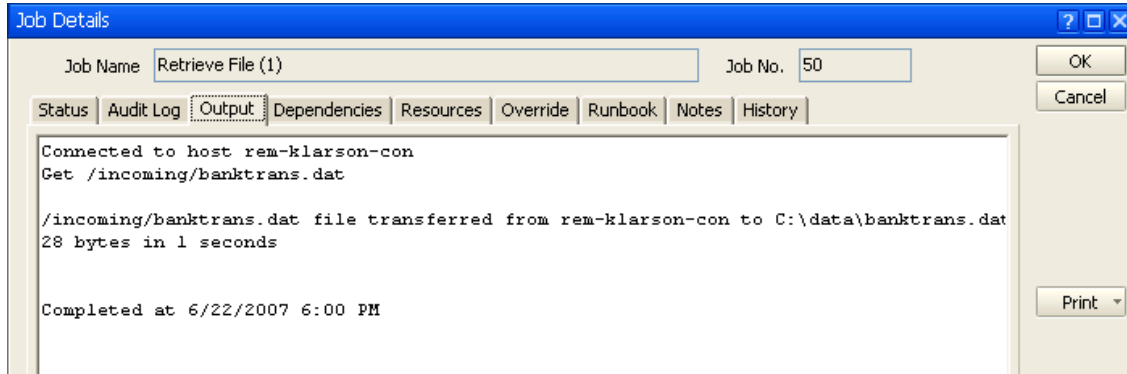
Daily SFTP Retrieval



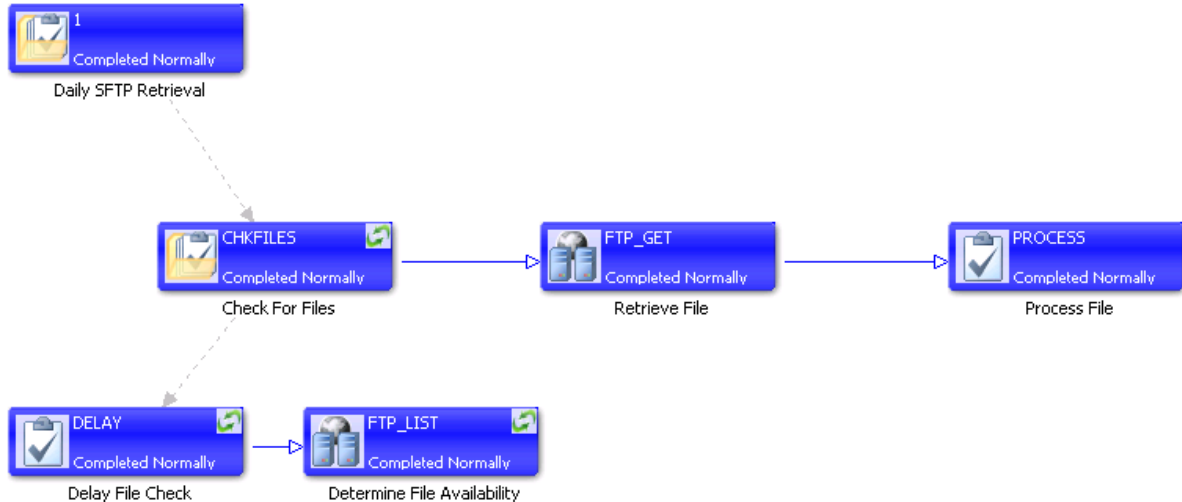
- Now that the file has arrived, *Retrieve File* runs, followed by *Process File*.



7. Inspecting the output of *Retrieve File*, you see the successful execution of the “get” command.



8. Once the file is retrieved, *Process File* runs and succeeds.



Tidal Enterprise Scheduler™ Playbook

Monitoring Files on Remote FTP Servers

Copyright

Copyright ©2004-2007 by Tidal Software®, Inc.

Tidal Software, Inc. 2100 Geng Road, Ste 210, Palo Alto, California 94303, U.S.A.

Unpublished rights reserved under the Copyright Laws of the United States. Printed in the U.S.A.

This document may not be stored in a retrieval system, reproduced, or transmitted in any form or by any means, either in whole or in part, without the express prior written permission of an officer of Tidal Software, Inc ("Tidal"). Copying includes translating into another language or format. This document contains proprietary information of Tidal. Tidal has prepared this document for use only by Tidal's employees and customers under Tidal's current commercial license and maintenance agreements. The only undertakings of Tidal respecting the information in this document are as contained in such contracts, and nothing contained in this document shall be construed as changing said contracts.

Information in this document is subject to change without notice and does not represent a commitment on the part of Tidal. Any use of this information except as defined by a written contract with Tidal, is not authorized and, with respect to any such unauthorized use, neither Tidal nor any of the contributors to this document makes any representation or warranty, nor shall any warranty be implied, as to the completeness, accuracy, or usefulness of the information contained in this document or that such use of such information may not infringe privately owned rights, nor do they assume any responsibility for liability or damage of any kind which may result from such use of such information.

Tidal™ Horizon™, Tidal™ Enterprise Scheduler, Tidal Software logo, and Correlex™ are trademarks of Tidal Software, Inc. Patent pending on Correlex engine technology.

SAP® Solutions and SAP® Solution Manager are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

MVS®, OS/390® and z/OS® are registered trademarks of IBM Corporation. UNIX, is a registered trademark in the US and other countries, licensed exclusively through X/Open company, LTD.

Microsoft Operations Manager®, MOM®, .NET™ and all variations of the Windows® product family are trademarks or registered trademarks of Microsoft Corporation in the United States of America and in several other countries. All other trademarks are property of their respective owners and are used purely for informational purposes.

Revision History

June 2007: P1021-A06/2007

TIDAL
SOFTWARE

2100 Geng Road Suite 210 Palo Alto, CA 94303

1 (650) 475-4600
playbook@tidalsoftware.com

www.tidalsoftware.com