

IBM Tivoli Workload Scheduler



Designing Your Workload

Version 8 Release 4.0.1

IBM Tivoli Workload Scheduler



Designing Your Workload

Version 8 Release 4.0.1

Note

Before using this information and the product it supports, read the information in "Notices" on page 23.

This edition applies to version 9, release 1, modification level 0 of Tivoli Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2008, 2011.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v	Setting time restrictions	13
About this publication	vii	Setting file dependencies	15
What is new in this release	vii	Setting prompt dependencies	15
Publications	vii	Assigning resources to workstations	16
Accessibility	vii	Chapter 6. Controlling your workload processing: defining job recovery options	17
Tivoli technical training	vii	Chapter 7. Optimizing your workload processing: setting priority and limit.	19
Support information	vii	Chapter 8. Tivoli Workload Scheduler Database and plans	21
Chapter 1. Automating your business workload	1	Notices	23
Chapter 2. Organizing tasks in a logical workflow	3	Trademarks	24
Chapter 3. Deciding where to run your workload	5		
Chapter 4. Deciding when to run your workload	7		
Chapter 5. Conditioning your scheduling activities by using dependencies.	11		
Deciding the scheduling sequence	11		

Figures

1. TelBestCo logical workflow organization into job streams 4
2. TelBestCo Tivoli Workload Scheduler network 6
3. TelBestCo workflow to collect data produce monthly reports and send them to management. 12

About this publication

This publication describes how to design your workload with IBM Tivoli Workload Scheduler by providing an overview of product capabilities.

What is new in this release

For information about the new or changed functions in this release, see *Tivoli® Workload Automation: Overview*, section *Summary of enhancements*.

For information about the APARs that this release addresses, see the Tivoli Workload Scheduler Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27038323> and the Dynamic Workload Console Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27038328>.

Publications

Full details of Tivoli Workload Scheduler publications can be found in *Tivoli Workload Automation: Publications*. This document also contains information about the conventions used in the publications.

A glossary of terms used in the product can be found in *Tivoli Workload Automation: Glossary*.

Both of these are in the Information Center as separate publications.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For full information with respect to the Dynamic Workload Console, see the Accessibility Appendix in the *IBM Tivoli Workload Scheduler User's Guide and Reference*.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education website:

<http://www.ibm.com/software/tivoli/education>

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.

- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see the appendix on support information in *Tivoli Workload Scheduler: Troubleshooting Guide*.

Chapter 1. Automating your business workload

This topic describes the evolution of computer tasks into complex workload scheduling environments

Audience: Job scheduling Administrators

When data processing involved a set of tasks to be performed on a computer system, workload scheduling was primarily the responsibility of a computer operator who defined and submitted single tasks for processing. The correct order of these tasks was established by the operators who manually intervened to correct problems or to answer messages displayed on the system console. The efficiency of the tasks and workload management in general relied almost solely on the experience and expertise of the human operator. One or more operators had to check that the needed resources were available at the right time (for example tape drives or printers) so that the tasks could be performed successfully.

As data processing tasks became increasingly complex, the number of computer systems also increased and became geographically distributed over large networks. The complexity and diversity of the architectures and operating environments made it impossible to rely on human operators alone. The first workload management tools were developed inhouse by computer organizations for their specific environments. As businesses became more complex, the need for sophisticated workload management and scheduling tools became a top priority for most businesses.

The company

TelBestCo is a telephone and telecommunications infrastructure company. It provides mobile communications as well as traditional telephone services. It is divided into several business units. The Customer Relationship Management unit is in charge of the company's help desk applications. Customers submit service requests through a Web interface. TelBestCo uses Tivoli Workload Scheduler as their comprehensive workload management solution.

TelBestCo Business scenario

TelBestCo wants to automate the tasks related to the following business activities:

- Business-specific applications
- System and database backups
- Daily monitoring scripts for system resources (disk space, memory usage, and so on)
- Start, stop, and status of critical system applications and processes
- Queries on the relational databases
- Reports generated by queries on the help desk system

Each task performed in an organization is a *job*. A job is a unit of work specifying an action, such as a weekly data backup, to be performed on a computer system. A job can be an operating system command, a program, or a script.

Chapter 2. Organizing tasks in a logical workflow

This topic introduces the concept of job stream.

After you decide which tasks to automate, you must decide how to organize them in a logical workflow to best suit your business processes. Tasks are usually organized by department, by division, by function, or by processing cycle. The organization of your tasks, from now on referred to as jobs, in a logical workflow, results in the creation of a job stream. Jobs are usually grouped together in job streams because they all run on the same day, share a common function, or share common dependencies.

For example, you might define one job stream containing jobs that must run for the Accounting Department and another job stream containing jobs that must run for the Personnel Department.

Usually each job stream is scheduled to run on a specific set of dates, and consists of a list of jobs that run as a unit (such as the weekly backup application, periodical status reporting), together with priorities, times, and other properties that determine the exact order in which the jobs run.

TelBestCo Business scenario: Creating job streams for queries

The Customer Relationship management unit wants to collect the following information for its reporting activities:

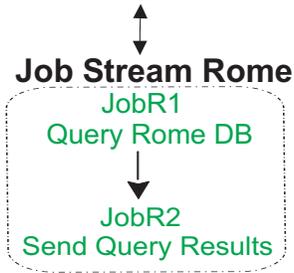
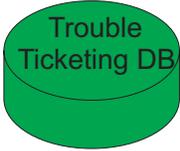
- How many trouble tickets were opened in a specific time frame
- How many tickets were opened and assigned for each area (ADSL, VoIP, telephone home)
- The ticket resolution time.
- The time taken to resolve and close the tickets.

To automate these activities, the TelBestCo divided the workflow into Customer Relationship management units, creating:

- For each Customer Relationship management unit, a job stream containing two jobs to perform the following two activities:
 1. Query the local database and collect results.
 2. Send results to the central database.
- For the Rome Central office, two job streams containing the following activities:
 - The Job StreamA Rome Central:
 1. Merges Customer Relationship management units queries
 2. Creates the worksheet of the monthly status
 3. Sends the worksheet to management
 - The Job StreamB Rome Central:
 1. Stops the Rome Central database
 2. Backs up the Rome Central database
 3. Restores the Rome Central database

Figure 1 on page 4 shows the Rome Customer Relationship management unit and the Rome Central office job stream structure.

Rome Customer Relationship management unit



Rome Central Office



Job StreamA Rome Central

Job StreamB Rome Central

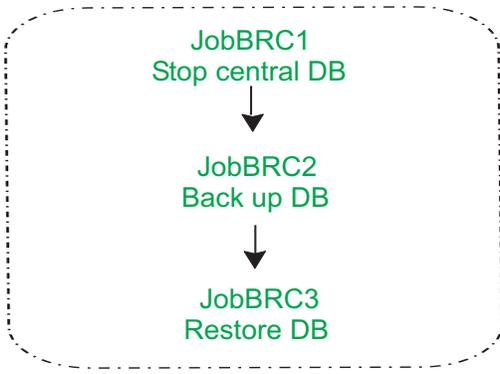
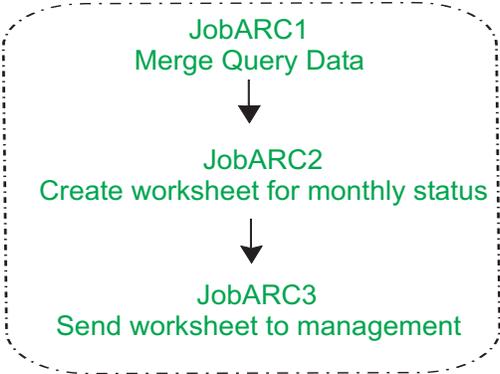


Figure 1. TelBestCo logical workflow organization into job streams

Chapter 3. Deciding where to run your workload

This topic introduces workstations, workstation classes, and domains.

The computer system where you run your jobs or job streams is called a *workstation*. When you define jobs in the Tivoli Workload Scheduler database you associate them to workstation definitions representing the physical machines or computer systems on which your jobs are scheduled to run. Workstations can also be grouped logically into *workstation classes* and organized hierarchically into *domains*. The way you define and group workstations in your Tivoli Workload Scheduler network is affected by how your business is structured.

When you create a workstation definition for a computer in your network, you associate to it a set of characteristics that uniquely identify the system and affect the way jobs run on it. You use a special type of workstation called *x-agent* to define and work with products or applications that integrate with Tivoli Workload Scheduler, such as SAP R/3. These workstations are logical definitions that are hosted by a physical workstation acting as a bridge to the application.

Designing your Tivoli Workload Scheduler Network

Tivoli Workload Scheduler network is made up of the workstations where job and job stream processing occurs. When you design your network, you assign roles to these workstations to suit your specific business requirements. You can design your network with multiple domains, to divide control of a large network into smaller manageable groups. A typical Tivoli Workload Scheduler network consists of a workstation acting as master domain manager and at least one domain. You can assign the following roles to workstations:

Master Domain Manager

A workstation acting as the management hub for the network. The master domain manager manages all the scheduling objects that you define.

Backup Master Domain Manager

Optionally, a workstation you define to act as a backup for the master domain manager. The Backup Master domain manager can assume the role of master domain manager in case of problems.

Domain Manager

A specially configured workstation controlling a subordinate domain, and sharing management responsibilities for a part of the Tivoli Workload Scheduler Network.

Fault-Tolerant Agent

A workstation that receives and processes jobs. If a network outage occurs, jobs continue to run locally.

In a single domain network, the master domain manager maintains direct communication with all of the workstations in the Tivoli Workload Scheduler network. You can define multiple domains to reduce network traffic between the master domain manager and all the other computers in the network. You assign the role of domain manager to a workstation in each domain. In a configuration with multiple domains, the master domain manager communicates with the workstations in its domain and with the domain managers of the subordinate domains.

TelBestCo Business Scenario: Deciding where to run your workload

In the Customer Relationship management units, the branch offices are divided into separate domains. Each geographical location has a domain manager and one or more *fault tolerant agents*. Each computer system in the domain has its corresponding workstation definition. All domains are connected to the Rome Central office, which is the *master domain manager*.

Figure 2 shows the TelBestCo company network mapped to a Tivoli Workload Scheduler network.

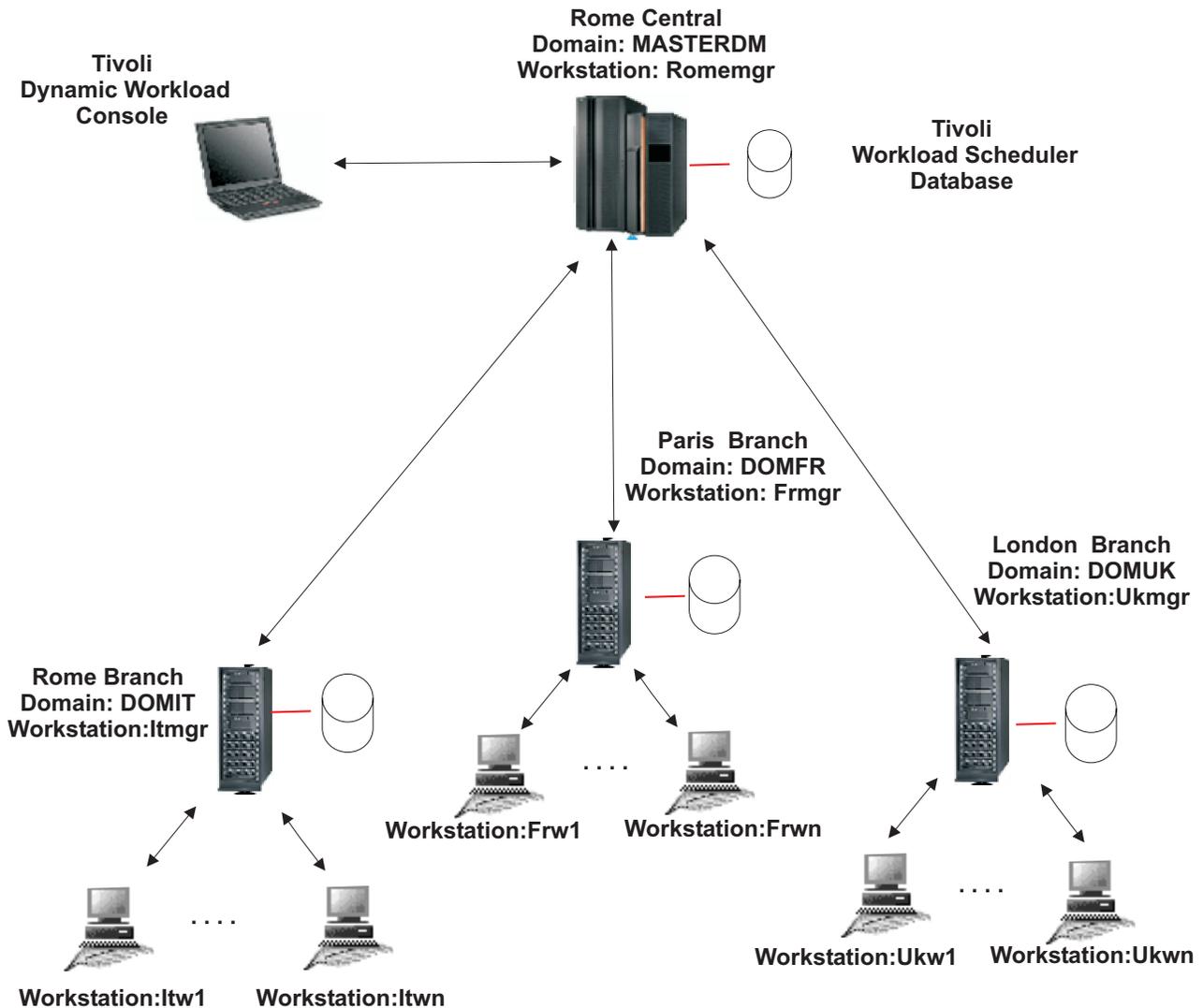


Figure 2. TelBestCo Tivoli Workload Scheduler network

Chapter 4. Deciding when to run your workload

This topic introduces the concept of run cycle.

When you have decide which workload to automate, and how to organize it into job streams, you must decide when to run your workload that is, on which days to run your job streams.

You use a run cycle to define when to run your job stream. Using run cycles you directly control job stream processing and indirectly control job processing. Table 1 shows you which run cycles you can define to run your job stream.

- Your job stream on specific dates. In this case you use a simple run cycle. For example, to run a job stream on the 20th of May 2008 and on the 28th of June 2008.
- Your job stream by day by specifying both a frequency and the type of day the job stream must run. In this case you use a daily run cycle. For example, to run a job stream once a day, every three days considering only workdays.
- Your job stream by week by specifying both a week frequency and specific days. In this case you use a weekly run cycle. For example, to run a job stream every Monday, Wednesday, and Friday every two weeks.
- Your job stream by month by specifying both a month frequency and specific days of the month. In this case you use a monthly by date run cycle. For example, to run a job stream every 1st and 2nd day of the month every two months.
- Your job stream by month by specifying both a month frequency and specific days of the week. In this case you use a monthly by day run cycle. For example, to run a job stream every 1st Monday and 2nd Monday of the month every two months.
- Your job stream by year. In this case you use a yearly run cycle. For example, to run a job stream every two years.
- Your job stream on a specific sequence of dates. In this case you use a calendar. For example, to run a job stream the days on which payment is made and if the payment day is Saturday the job stream must run the day before, if it is Sunday the job stream must run the day after.

Table 1. When to run your job stream

You can decide to run your job stream	For example	Run cycle type
On specific dates	On the 20th of May 2008 and on the 28th of June 2008	Simple run cycle
By day by specifying both a day frequency and the type of days that must be considered to establish when the job stream must run. That is every day, only during workdays, or excluding free days.	Every three days, considering only workdays.	Daily run cycle
By week by specifying both a week frequency and specific days of the week.	Every Monday, Wednesday, and Friday every two weeks.	Weekly run cycle

Table 1. When to run your job stream (continued)

You can decide to run your job stream	For example	Run cycle type
By month by specifying both a month frequency and specific days of the month.	Every 1st and 2nd day of the month, every two months.	Monthly by date run cycle
By month by specifying both a month frequency and specific days of the week.	Every 1st Monday and 2nd Monday of the month, every two months.	Monthly by day run cycle
By year.	Every two years.	Yearly run cycle
On a specific sequence of dates.	The dates you specified in a calendar.	Calendar. By assigning to the calendar run cycle type a calendar object.
	The dates you specified in a calendar plus an extension of dates. For example, if you want to run a job stream the 7th of each month and the date specified in your calendar is the 5th of each month you define an extension of +2 Days to run the job stream the 7th of each month.	Calendar plus an offset. By assigning to the calendar run cycle type a calendar object.

You can assign more than one run cycle to each job stream.

For all the run cycles you can also specify a rule that establishes when a job stream actually runs if its schedule falls on a free day. This rule is based on the definition of free days. Free days are non-workdays, such as holidays or weekends. You define them in a calendar. Free days calendars can be considered as lists of non-workdays when you do not perform any activities in your company. The rule determines which action Tivoli Workload Scheduler must perform when the running of a job stream falls on a free day. If you:

Want to run the job stream regardless of whether the schedule falls on a free day or on a workday

Do not specify any rule.

Do not want to run the job stream if the scheduled date falls on a free day.

Specify the **If freeday, do not select** rule.

Want to run the job stream on the preceding workday if the scheduled date falls on a free day

Specify the **Nearest workday before freeday** rule.

Want to run the job stream on the following workday if the scheduled date falls on a free day

Specify the **Nearest workday after freeday** rule.

For example, if you want to run the job stream that calculates the salary on the 25th of each month, or run it the day after if the 25th is a Sunday or a holiday, you specify the following run cycle information for the job stream:

- The **Nearest workday after freeday** rule
- A run cycle of type **Monthly by date** calendar
- A calendar in which you defined holidays.

Because you can assign more than one run cycle to each job stream, when you define a run cycle, you specify if it designates the days when the job stream is to run (inclusive) or when the job stream is not to run (exclusive).

TelBestCo Business Scenario: Deciding when to run your activities

Because the TelBestCo want to run the job streams that:

- Collect data to produce branch office queries daily, except on Sundays and national holidays, the company assigned to these job streams a run cycle specifying the following information:
 - The **If freeday do not select** rule
 - Inclusive run cycle
 - A run cycle of type daily
 - The **Every one day** frequency
 - A calendar in which the company defined all the holidays.
- Send data to the central office database on the 30th of each month if it is a workday and on the last available workday before the 30th if it is a holiday, the company assigned to these job streams a run cycle specifying the following information:
 - The **Nearest workday before freeday** rule
 - Inclusive run cycle
 - A run cycle of type Monthly by date
 - A calendar in which you defined all the holidays.
- Produce reports for management on the 25th of each month if it is a workday and on the last available workday before the 25th if it is holiday, the company assigned to this job stream a run cycle specifying the following information:
 - The **Nearest workday before freeday** rule
 - Inclusive run cycle
 - A run cycle of type Monthly by date
 - A calendar in which you defined all the holidays.

Chapter 5. Conditioning your scheduling activities by using dependencies

This topic introduces the concept of dependencies

In the daily operations of a business, the correct order of task processing is a critical aspect. There are activities that cannot start until others have completed successfully, or that must be performed at the beginning of a set sequence. Similarly, there are activities that need specific system resources to run correctly. Any condition or prerequisite that must be satisfied before a job or job stream can begin processing is called a *dependency*. For example, when you are running a daily backup of the production database, you must ensure that work has successfully switched to an alternate database before you begin your backup. You also must ensure that this activity starts at a set time so that it does not disrupt other critical tasks.

In a scheduling environment, you define dependencies to establish the required processing order that best suits your business needs and optimizes your workload. Dependencies can affect jobs and job streams in terms of the time a job or job stream starts, the resources it needs to complete successfully, and on whether a human operator must intervene during the scheduling process.

Deciding the scheduling sequence

The topic introduces the concept of predecessors and successors.

When you define the logical flow of your business activities, the order of processing of your daily tasks is very important. When you establish a running order for your tasks, you are defining *predecessors* and *successors* for your jobs or job streams. Tivoli Workload Scheduler automatically manages predecessor and successor relationships. You can define these and other dependencies to establish the required processing order of your workload.

TelBestCo Business Scenario: Deciding the scheduling sequence

In the Customer Relationship management units, the trouble ticketing manager sends monthly reports to upper management. These reports contain information about the number of daily incoming tickets, the average resolution time for each ticket, and the current status of active tickets. Each geographical location produces daily statistics about their customer relationship management activities. These statistics are extracted from the databases at each location and subsequently consolidated into a formatted report which is sent to management at the end of each month. The order in which these activities are carried out is very important.

Figure 3 on page 12 shows how each Customer Relationship management unit and the Rome central office organized jobs into job streams to collect data, produce monthly reports, and send them to management.

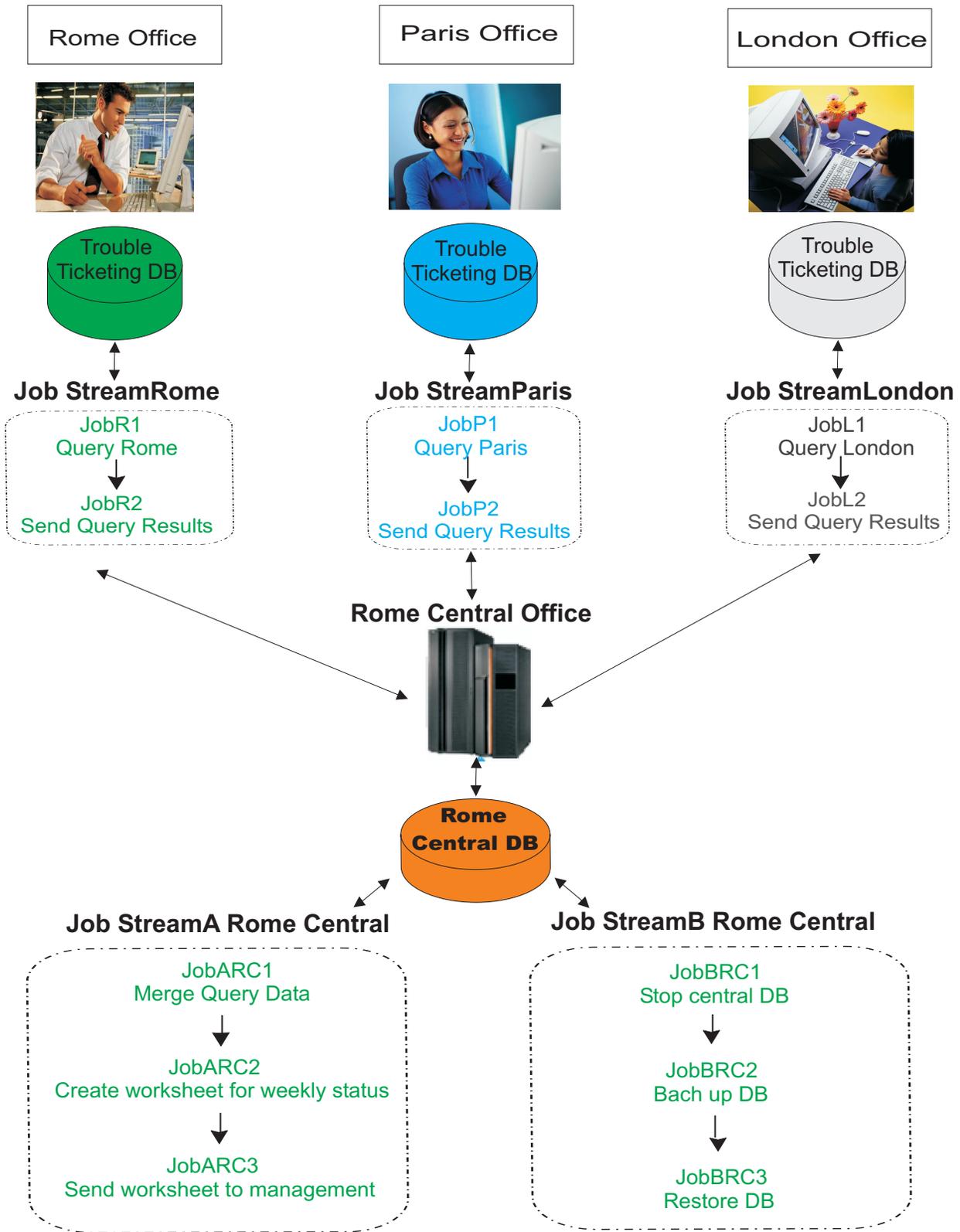


Figure 3. TelBestCo workflow to collect data produce monthly reports and send them to management.

Setting time restrictions

This topic explains how to control job and job stream processing using time restrictions

One of the key aspects of job scheduling is the capability to set time constraints or time dependencies that influence the processing activities in your organization. When you set a time restriction, you are establishing a condition that must be satisfied before processing can proceed. You can set time restrictions at the job or job stream level. You can also set repetition or frequency rates on jobs within job streams to further refine your processing. If you are using run cycles, the time restrictions you set are relative to the run cycle associated to your job or job stream.

You can set the following types of restrictions:

Earliest start time

You can define the time before which a job or job stream is not allowed to start. If the job or job stream has a run cycle defined, then the value represents the earliest start time within the specified run cycle. (AT keyword).

Latest start time

The latest time a job or job stream can start processing. (UNTIL keyword). If the job or job stream has a run cycle defined, then the value represents the latest start time within the specified run cycle. You can also decide what action to take when this time is reached. You can define to take any of the following actions: (ONUNTIL):

suppress

The job or job stream, and any other dependent job or job stream do not run

continue

The job or job stream runs when all the defined conditions are met.

cancel The job or job stream is canceled. All dependencies are released so that any other jobs or job streams that were dependent on this one (for example successors) can run.

Termination deadline

The latest time within which a job or job stream must complete. If the job or job stream has a run cycle defined, then the value represents the latest termination time within the specified run cycle. (DEADLINE keyword)

Maximum duration

Specifies the maximum amount of time that a job run can last. The value can be expressed in absolute time indicators (**Hours: Minutes**) or as a percentage of the latest estimated duration of the job. You can also specify the action that must be performed if the job run lasts longer than the specified amount of time. Possible actions are:

Continue

The workload running continues without taking any action.

Kill The job is set to **Error** status

Minimum duration

Specifies the minimum amount of time that a job run must last. The value can be expressed in absolute time indicators (**Hours: Minutes**) or as a percentage of the latest estimated duration of the job. You can also specify

the action that must be performed if the job run ends before the minimum duration is reached. Possible actions are:

Continue

The workload running continues without taking any action.

Confirm

The job is set to **Confirm** status The workload requires a user confirmation to proceed.

Abend

The job is set to **Error** status

Scheduled time (Planned Start Time)

This is not a time dependency, but it represents the time when the job stream is positioned in the plan. Specifying a scheduled time does not prevent the job stream from processing, but it uniquely identifies a given job stream instance in the plan, when multiple instances of the job stream exist for a single day (job streams that run more than once in a 24 hour period.)

You can also combine time restrictions. For example, when you define both an earliest start time and a latest start time for a job or job stream, you are creating a time frame within which processing must start. Combining a deadline adds a restriction to the time of completion of your processing window.

You can further refine your scheduling by establishing a frequency or period for jobs that run at set intervals. You do this by setting:

Job processing frequency (repetition rate or interval)

The job is launched repeatedly at the specified interval. The iteration starts when all dependencies on the job have been satisfied. (EVERY keyword)

How TelBestCo uses time restrictions

Table 2 shows some examples of how you can use time restrictions

Table 2. Time restriction types and examples

Type of time restriction	Example
Earliest start time	A job or job stream scheduled to run every Friday, cannot start before 6:45 p.m.
Latest start time	A job stream is scheduled to run the last Tuesday of each month and it cannot start later than 9 a.m.
Termination Deadline	A job is scheduled to run every day and must complete before 11 a.m.
Minimum duration	A job that runs for an unrealistically short amount of time, considering the amount of time it ran in the recent past, is automatically set to Error status.
Repetition rate (frequency)	A system disk-space check job is scheduled to run on workdays every 30 minutes.

In the Service and Support Business Unit, time dependencies are set for all the critical tasks that must be performed on a daily or regular basis:

- Database switches, database backups, and cleanup jobs are defined with an **earliest start time** of 6 p.m. This is the end of business day time, after which routine data integrity tasks are performed. These jobs must be closely monitored and use specific processing time frames and **deadlines**.
- TelBestCo must ensure a 24 x 7 service. The support department have defined jobs to check network traffic and disk space that run **every hour** so that critical resource shortages are quickly detected and managed.
- The support department also needs to know if some jobs are not running as expected. By introducing a **termination deadline** for routine database backup or database switch jobs, if the deadline time is reached and the job has not ended, a database administrator is notified. Jobs that take too long might have looped or have experienced a shortage of system resources.

Setting file dependencies

This topic explains how you control job and job stream processing using file dependencies.

You use file dependencies to control job and job stream processing based on the existence of one or more files or directories. When you specify a file dependency, Tivoli Workload Scheduler checks if the specified file or directory exists before processing begins.

You can select several conditions associated to the file that must be true for processing to begin. These are:

- The file exists
- The file exists and is a directory
- The file exists and is a regular file
- The file exists and is readable
- The file exists and its size is greater than zero
- The file exists and is writable

You can combine one or more of the above conditions.

How TelBestCo uses file dependencies

The Customer Relationship Management Business Unit collects statistical information about the trouble ticketing applications from the various branch offices. The results of the database queries collected from each branch are copied to flat files and sent to the central office for further processing. Every last day of the month, the job stream called *MonthlyRep* checks for the existence of the single flat files from each branch, and also checks that the file size of each file is greater than zero. If these conditions are satisfied, the job that collects and merges data from all these files is launched. Lastly, a job that produces a final worksheet is launched.

Setting prompt dependencies

This topic explains how you control job and job stream processing using prompt dependencies.

Some jobs or job streams are so critical that they require special attention and human intervention to ensure that each task starts when all necessary conditions are fulfilled, and that it is accurately performed and monitored. When you define *prompt* dependencies, you can momentarily suspend the processing of a job or job

stream. A prompt identifies a textual message which is presented to an operator. The prompt requires an affirmative answer to allow processing to continue. You can also use prompts to alert an operator that a specific task was performed. In this case, an operator response is not required.

You can define three different types of prompts:

A prompt that can be used by any job or job stream in your environment

It is called a *global* (or named) prompt and you define it as a scheduling object. It is identified by a unique name and can be used by any job or job stream.

A prompt that is defined and used for a specific job or job stream

It is called a *local* or *ad hoc* prompt. This type of prompt is defined within a job or job stream. It does not have a name and it cannot be used by other jobs or job streams.

A prompt that alerts an operator of an error condition

It is called a *recovery* or *abend* prompt. It is a special type of prompt that you define for when a job ends abnormally. The response to this prompt determines the outcome of the job or job stream to which the job belongs. A recovery prompt can also be associated to an action and to a special type of job called a *recovery job*.

Assigning resources to workstations

This topic explains how you control job and job stream processing by defining resources.

You can control task processing by defining and associating *resources* to workstations. These resources are used as dependencies for job and job stream processing for the workstation to which they are associated. Resources represent physical or logical scheduling resources. You define resources to introduce pauses in the logical flow of your tasks. For example, if two or more jobs require one tape unit to be available, you can define a resource called *tapedrive* and assign it a value of 1. When you associate this resource as a dependency to one or more jobs, if both jobs request access to the resource, it is assigned to the job or job stream with the highest priority. When the job using the resource ends, the resource is released and the next job can use it. Defining resources helps you control concurrent processing.

You can also define resources to avoid shortages of limited system resources. If a job on a critical system requires a fixed amount of memory and disk space to run successfully, you can associate a logical resource dependency to that job so that no other job will run at the same time causing a memory or disk space shortage.

Chapter 6. Controlling your workload processing: defining job recovery options

This topic explains how you define job recovery actions. It introduces the concept of recovery options, recovery jobs, and recovery prompts.

The interdependencies between tasks, the correct sequencing of activities, and the availability of key resources introduce complexities to workload processing. The need to monitor and correct potential problems before they become critical is a top priority for successful business process management. Tivoli Workload Scheduler addresses the need for recovery actions with a high degree of flexibility and automation. When you define jobs, you can specify the type of recovery option that is automatically initiated when the job fails:

Continue

Do nothing with the failed job and continue processing the next job in the sequence

Stop Do nothing with the failed job and do not process the next job in the sequence

Rerun Run the failed job again

In addition, you can associate a *recovery job* or *recovery prompt* to an action. For example, if Job A fails because an application was not started on a given system, the recovery Job R is launched to start the application, then the original job is rerun and a prompt is issued to the operator to alert him of the recovery condition. The recovery prompt might or might not require an operator response. You can automate problem resolution by combining recovery options to recovery jobs in the combination that better suits your business needs.

How TelBestCo sets recovery actions

The Customer Relationship Management unit records all trouble ticket data in a relational database that tracks tickets from creation to resolution. Database access is very important to store data about known problems and their solutions and to provide a real-time status to management. When the jobs performing queries on the master database fail, the information must be retrieved from a backup database, and the database administrator must be notified about the problem. A recovery job automates the query of the backup database, while a recovery prompt is issued to notify the operator. The next job will be launched only if the recovery job ended successfully (using the **continue** recovery action).

When too many concurrent accesses of the trouble ticketing database occur, sessions are killed and queries are automatically rerun.

Chapter 7. Optimizing your workload processing: setting priority and limit

This topic explains how you optimize processing by setting job priorities, and workstation fences and limits.

In a complex scheduling environment, establishing and controlling the priority of daily tasks and operations is very important. Tivoli Workload Scheduler has its own internal queuing mechanism, which consists of levels of *priority*. When you assign a specific priority to a job or job stream, you are controlling its precedence and order of processing. Another level of control over processing is provided by the *job fence*. When you set the job fence value on a workstation, this value determines the priority level above which a job is allowed to run. Any job with a priority less than or equal to the value specified for the fence will not be launched on the workstation. Additionally, you can set limits to the maximum number of jobs that can be launched simultaneously within a job stream or for a given workstation. This is done by setting the job or workstation *limit*. Combining two or more of these attributes provides a powerful means of controlling processing on given systems and to avoid resource contention or saturation. In detail you can set:

Job stream and job priority

You set a priority for a job or job stream to determine which one starts first, when all dependencies have been solved. The order of processing is determined by the following rule:

1. The job stream with the highest priority is the first one to be launched
2. Within a job stream with the highest priority, the job with the highest priority is launched first.

Job Fence on a workstation

Jobs on the specified workstation are not launched if their priorities are less than or equal to the job fence value. You set a job fence to prevent low priority jobs from being launched, regardless of the priorities of their job streams. When you assign a fence value, you can prevent all low priority jobs from running even if they belong to a high priority job stream.

Job limit within a job stream

You set a job limit to limit the number of jobs that can run simultaneously in a job stream on a given workstation.

Workstation limit

You set a workstation limit to define the maximum number of jobs that can run simultaneously on a workstation.

Chapter 8. Tivoli Workload Scheduler Database and plans

This topic introduces the concept of database and plan

Job Processing is very complex and requires close monitoring of all related attributes such as time of day , prior completion of other jobs, job streams or events, and a variety of other dependencies described in previous sections. All of these factors combined prevent job launch. We use a *database* to store the information about jobs and their scheduling dependencies within a logical job stream. The database is a repository containing all the objects needed to model your workload requirements.

Each database object has a set of properties that you define or modify. Some properties are characteristics of the object itself (such as "object name") , while others describe the conditions or dependencies that might inhibit job launch if they are not fulfilled.

The compiled and network-distributed sum of job stream requirements for a given time period of workload production we refer to as *plan*.

Notices

This information was developed for products and services offered in the U.S.A. IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and `ibm.com`[®] are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.



Java[™] and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Product Number: 5698-WSH

Printed in USA