

IBM Workload Automation



Overview

Version 9 Release 4

IBM Workload Automation



Overview

Version 9 Release 4

Note

Before using this information and the product it supports, read the information in "Notices" on page 83.

Contents

Figures	vii	=	New keyword for defining the latest start time in job streams	xxviii
About this publication	ix	=	New keyword for defining actions on late jobs	xxix
What is new in this release	ix	=	Higher level of control over the What-If Analysis	xxix
Who should read this publication	ix	=	Preparing and installing a Docker image for dynamic agents	xxix
Accessibility	ix	=	Upgrading from version 8.6.x	xxx
Technical training	ix	=	New options in managing workload applications	xxx
Support information	x	=	Integration with IBM UrbanCode Deploy	xxxi
How to read syntax diagrams	x	=	Satisfying Requests for Enhancements (RFEs)	xxxi
Summary of enhancements	xiii	=	IBM Workload Scheduler version 9.4	
- IBM Workload Scheduler version 9.4 Fix Pack 4 enhancements	xiii		enhancements	xxxii
- DevOps made easier with workload applications	xiii		Agent upgrade with minimal scheduling disruption	xxxiii
- Additional agent support	xiv		IBM Workload Scheduler Plug-in for Cloudant	xxxiii
- Automatically register agents to pools	xiv		Keeping track of changes to scheduling objects	xxxiv
- Option to determine when fileCreated event is sent	xv		Auditing release management	xxxv
* Satisfying Requests for Enhancements (RFEs)	xv		Version control	xxxv
* IBM Workload Scheduler version 9.4 Fix Pack 3 enhancements	xv		Backup copy of tws_env script	xxxvi
* Extra opportunities for modifying job definitions already in the plan	xvi		New event-driven workload automation action to open a ServiceNow incident	xxxvi
* Troubleshooting stalled jobs	xvi		IBM i job definition enhancements	xxxvi
* Auto refresh Plan View	xvii		Passing variables between jobs	xxxvii
* Easy installation for patches on agents	xvii		Graphical view enhancements	xxxvii
* Enhancements to IBM i job monitoring and control	xvii		IBM Workload Scheduler REST API	xxxvii
* New Plug-ins for Cloud Automation	xviii		Satisfying Requests for Enhancements (RFEs)	xxxviii
* Apache Spark Plug-in	xix		Scheduling with the Agent for z/OS 9.4	
* Restarting JSR 352 Java Batch jobs from the point of failure	xix		enhancements	xxxix
* More flexibility in managing successors for job actions	xix		EDWA for Scheduling with the Agent for z/OS	xxxix
* Satisfying Requests for Enhancements (RFEs)	xx		New opportunities for modifying job definitions already in the plan or in the database	xxxix
/ IBM Workload Scheduler version 9.4 Fix Pack 2 enhancements	xx		IBM Workload Scheduler version 9.3 Fix Pack 2	
/ Mixed pricing models	xxi		enhancements	xxxix
/ Rerun multiple jobs	xxi		New method for updating DB2 and Oracle database schema	xl
/ Rollback procedure	xxi		Hybrid Workload Automation	xli
/ Actual workstation in pool and dynamic pool	xxii		Checking prerequisites before running a silent installation	xli
/ Satisfying Requests for Enhancements (RFEs)	xxii		Automatically initializing IBM Workload Scheduler instances	xlii
= IBM Workload Scheduler version 9.4 Fix Pack 1 enhancements	xxiii		Defining and managing IBM Workload Scheduler role-based security	xlii
= Revamped graphical views	xxiii		Reply to a message for an IBM i job from Dynamic Workload Console	xlii
= What-if enhancements	xxiv		Automatic reply to waiting messages for an IBM i job	xliii
= IBM Workload Scheduler Job Management Plug-in	xxv		IBM Workload Scheduler Plug-in for Apache Oozie	xliii
= Automation of Iterative Workflows	xxv		Running a script when a job completes	xliii
= Advanced rerun flexibility	xxvi		New and improved web interfaces	xliii
= IBM Workload Scheduler Job Stream Submission Plug-in	xxvii			
= Condition-based workload automation	xxvii			
= Monitor file changes using the filemonitor utility	xxviii			

Creating a URL to launch the Plan View in context	xliv
RFE 84003 - Modifying styles when exporting the Graphical View	xliv
RFE 75635 - Graphical View enhanced for wide monitors	xliv
Auto-complete for workstation names and user ID	xliv
Managing Workload Security with Dynamic Workload Console	xliv
Restarting Informatica PowerCenter jobs from the point of failure.	xlv
Single Entry Point	xlv
Tivoli Common Reporting reports extended to MS SQL databases	xlv
IBM Workload Scheduler version 9.3 Fix Pack 1 enhancements	xlvi
Conditional dependencies	xlvi
IBM Workload Scheduler Plug-in for Liberty	xlvi
IBM Workload Scheduler Plug-in for MQTT	xlvii
WebSphere Application Server tools enhancements	xlvii
Scheduling the centralized agent update	xlvii
IBM Tivoli Monitoring integration enhancement: situation refactoring and new view associations	xlviii
Import Cron and Windows Task Scheduler jobs	xlviii
Implementing What-if Analysis changes to the plan	xlviii
IBM i enhancements	xlviii
Other Dynamic Workload Console version 9.3 Fix Pack 1 enhancements	xlx
IBM Workload Scheduler version 9.3 enhancements	xlx
IBM Workload Scheduler Plug-in for IBM WebSphere MQ	1
IBM Workload Scheduler Plug-in for RESTful Web services	1
IBM Workload Scheduler Plug-in for SAP BusinessObjects	1
IBM Workload Scheduler support for Netezza Performance Server	li
IBM Workload Scheduler Plug-in for IBM Sterling Connect:Direct	li
IBM Workload Scheduler Plug-in for Salesforce.	li
IBM Workload Scheduler Plug-in for Hadoop	lii
IBM Workload Scheduler Plug-in for BigInsights for Hadoop	lii
IBM Workload Scheduler plug-in for Oracle E-Business Suite	lii
File Transfer plug-in enhancements	liii
Enhancing your job stream definitions with the EVERY option	liii
Using IBM Workload Scheduler to extend the job capabilities on SAP.	liii
Intercept business impact of Workload Automation failures.	lv
File dependencies support for dynamic agents	lv
Centralized agent update	lv
IBM Tivoli Workload Scheduler for Applications now installed with base product	lvi

Installation of agents by using domain users	lvi
Other enhancements	lvi
New Welcome page for the Dynamic Workload Console	lvii
IBM Workload Scheduler Analytics	lvii
Monitor Workload.	lvii
Dashboards to monitor important information, at a glance	lviii
Import Cron and Windows Task Scheduler jobs	lviii
Application Lab	lviii
Tivoli Common Reporting Reports	lix
Fast and easy location of variables in a variable table.	lix
Job end time is tracked in the database and is viewable from the Dynamic Workload Console	lix
Defining a job as non-operational (NOP)	lix
Managing SAP events on dynamic workstations	lx
Other Dynamic Workload Console enhancements	lx

Chapter 1. Overview of IBM Workload

Automation	1
The state-of-the-art solution	1
Comprehensive workload planning	2
Centralized systems management	2
Systems management integration	2
Automation.	5
Workload monitoring	6
Automatic workload recovery	6
Productivity	6
Business solutions	6
User productivity.	7
Growth incentive	7
How IBM Workload Automation benefits your staff	7
Role of the scheduling manager as the focal point	8
Role of the operations manager	8
A powerful tool for the shift supervisor	8
Role of the application programmer	9
Console operators	9
Workstation operators	9
End users and the service desk	9
Summary	10

Chapter 2. IBM Workload Automation and ITUP 11

The ITUP processes.	11
Service execution and workload management	12
Managing workload with IBM Workload Automation	12

Chapter 3. Who performs workload management 15

Chapter 4. A business scenario 17

The company.	17
The challenge.	19
The solution	21
Typical everyday scenarios	24
The benefits	29

Chapter 5. IBM Workload Scheduler . . . 31

Overview	31
What is IBM Workload Scheduler	31
The IBM Workload Scheduler network	31
Manager and agent types.	33
Topology	35
Networking	35
IBM Workload Scheduler components	36
IBM Workload Scheduler scheduling objects	37
The production process	40
Scheduling	41
Defining scheduling objects	41
Creating job streams	41
Setting job recovery	42
Defining and managing mission-critical jobs	42
Scheduling workload dynamically.	43
Running production	44
Running the plan	44
Running job streams	45
Monitoring	46
Reporting	47
Auditing	48
Using event-driven workload automation	48
Options and security	48
Setting the IBM Workload Scheduler options	49
Setting security	49
Using time zones	51
Using the access methods of IBM Workload Scheduler	51
Supported agents	52

Chapter 6. IBM Workload Scheduler for z/OS 53

How your production workload is managed	53
Structure	53
Concepts	54

Using Plans in IBM Workload Scheduler for z/OS	57
Automatically controlling the production workload	58
Integration with IBM Tivoli Monitoring	63
Interfacing with other programs	64
Manual control and intervention	64
Management of critical jobs	66
Security	66
Configurations of IBM Workload Scheduler for z/OS	68
The controlling system	69
Controlled z/OS systems.	70
Remote panels and program interface applications	70
Scheduling jobs that are in IBM Workload Scheduler	70

Chapter 7. Dynamic Workload Console 73

Chapter 8. End-to-end scheduling . . . 75

End-to-end scheduling with fault tolerance capabilities	75
End-to-end scheduling with z-centric capabilities.	77
Distributed agents	78
Benefits of end-to-end scheduling	79

Chapter 9. Hybrid scheduling environment 81

Notices 83

Trademarks	85
Terms and conditions for product documentation.	85

Index 87

Figures

1. Integration scenario for IBM Workload Scheduler for z/OS. 5
2. The Fine Cola company integrated workload solution. 20
3. How to satisfy SLA response time during peak periods using the dynamic scheduling capability of IBM Workload Scheduler. . . . 27
4. IBM Workload Scheduler network made up by two domains 32
5. Supported agents configuration 51
6. Automatic recovery and restart 61
7. Production workload restart and hot standby 62
8. Security. 67
9. IBM Workload Scheduler for z/OS configurations. 69
10. End-to-end with fault tolerance capabilities configuration 77
11. End-to-end with z-centric capabilities configuration 78

About this publication

Read this publication for information about the IBM Workload Automation product suite.

IBM Workload Automation: Overview describes the family of IBM Workload Scheduler products and its enterprise workload management functions. It gives introductory information about the following products.

- IBM Workload Scheduler
- IBM Workload Scheduler for z/OS
- Dynamic Workload Console

Note: In version 9.3 the product family name changed from IBM Tivoli Workload Automation to IBM Workload Automation and the product name changed from IBM Tivoli Workload Scheduler to IBM Workload Scheduler.

What is new in this release

Learn what is new in this release.

For information about the new or changed functions in this release, see “Summary of enhancements” on page xiii.

Who should read this publication

Learn the audience of this publication.

This publication is intended for:

- Data processing (DP) operations managers and their technical advisors who are evaluating the product or planning their scheduling service
- Individuals who require general information for evaluating, installing, or using the product.

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, see the Accessibility Appendix in the *IBM Workload Scheduler User's Guide and Reference*.

Technical training

Cloud & Smarter Infrastructure provides technical training.

For Cloud & Smarter Infrastructure technical training information, see:
<http://www.ibm.com/software/tivoli/education>

Support information

IBM provides several ways for you to obtain support when you encounter a problem.

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:

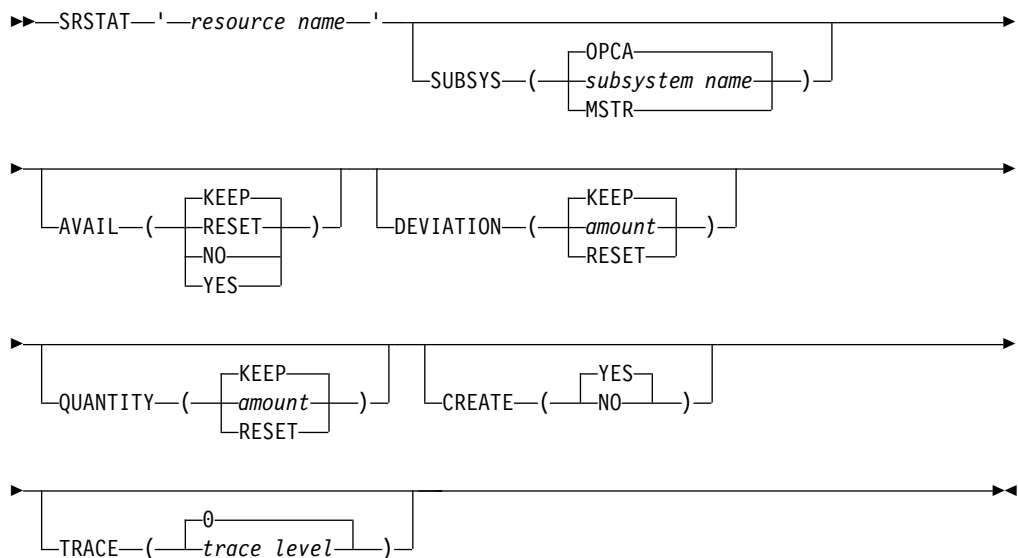
- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your 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 about support information in *IBM Workload Scheduler: Troubleshooting Guide*.

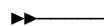
How to read syntax diagrams

Syntax diagrams help to show syntax in a graphical way.

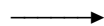
Throughout this publication, syntax is described in diagrams like the one shown here, which describes the SRSTAT TSO command:



The symbols have these meanings:



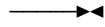
The statement begins here.



The statement is continued on the next line.



The statement is continued from a previous line.

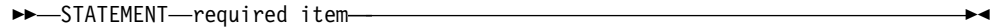


The statement ends here.

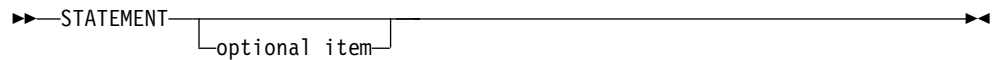
Read the syntax diagrams from left to right and from top to bottom, following the path of the line.

These are the conventions used in the diagrams:

- Required items appear on the horizontal line (main path):



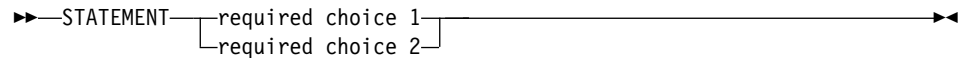
- Optional items appear below the main path:



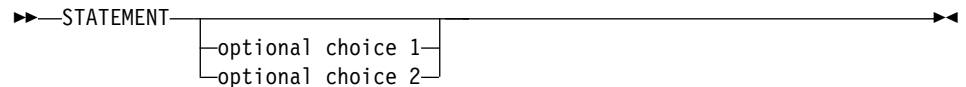
- An arrow returning to the left above the item indicates an item that you can repeat. If a separator is required between items, it is shown on the repeat arrow.



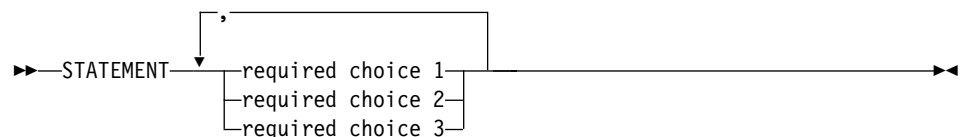
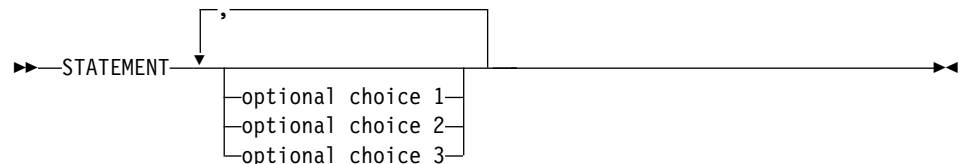
- If you can choose from two or more items, they appear vertically in a stack.
 - If you must choose one of the items, one item of the stack appears on the main path:



- If choosing one of the items is optional, the entire stack appears below the main path:



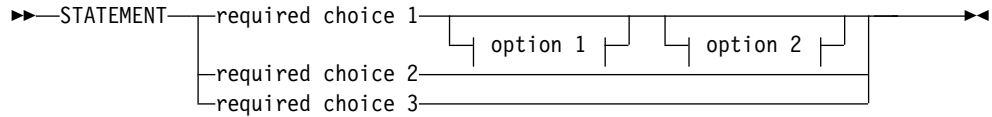
- A repeat arrow above a stack indicates that you can make more than one choice from the stacked items:



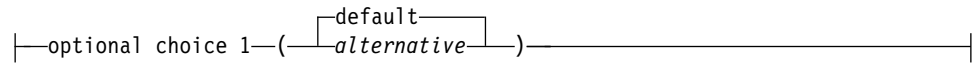
- Parameters that are above the main line are default parameters:



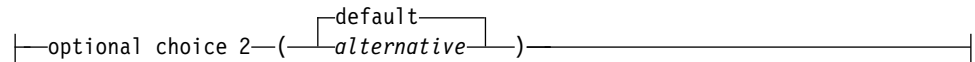
- Keywords appear in uppercase (for example, STATEMENT).
- Parentheses and commas must be entered as part of the command syntax, as shown.
- For complex commands, the item attributes might not fit on one horizontal line. If that line cannot be split, the attributes appear at the bottom of the syntax diagram:



option 1



option 2



Summary of enhancements

IBM Workload Automation provides the following enhancements:

Version 9.4 Fix Pack 4

“IBM Workload Scheduler version 9.4 Fix Pack 4 enhancements”

Version 9.4 Fix Pack 3

“IBM Workload Scheduler version 9.4 Fix Pack 3 enhancements” on page xv

Version 9.4 Fix Pack 2

“IBM Workload Scheduler version 9.4 Fix Pack 2 enhancements” on page xx

Version 9.4 Fix Pack 1

“IBM Workload Scheduler version 9.4 Fix Pack 1 enhancements” on page xxiii

Version 9.4

- “IBM Workload Scheduler version 9.4 enhancements” on page xxxii
- “Scheduling with the Agent for z/OS 9.4 enhancements” on page xxxix

Version 9.3 Fix Pack 2

“IBM Workload Scheduler version 9.3 Fix Pack 2 enhancements” on page xxxix

Version 9.3 Fix Pack 1

“IBM Workload Scheduler version 9.3 Fix Pack 1 enhancements” on page xlvi

Version 9.3

- “IBM Workload Scheduler version 9.3 enhancements” on page xlix

IBM Workload Scheduler version 9.4 Fix Pack 4 enhancements

Learn about the IBM Workload Scheduler version 9.4 Fix Pack 4 enhancements.

IBM Workload Scheduler version 9.4 Fix Pack 4 includes the following enhancements:

- “DevOps made easier with workload applications”
- “Additional agent support” on page xiv
- “Automatically register agents to pools” on page xiv
- “Option to determine when fileCreated event is sent” on page xv
- “Satisfying Requests for Enhancements (RFEs)” on page xv

DevOps made easier with workload applications

If you have a few use cases where the workload application templates replicated in your workload environment are a little too rigid, then this new enhancement will add the flexibility you are looking for.

After a workload application is exported into a workload environment, the contents such as, job streams, jobs, and other objects, are created in the

- environment and an association to the workload application is maintained. This
- allows for easy subsequent updates and synchronization between the workload
- application template exported from the source environment and the workload
- application in the target environment. The workload application can be easily
- replaced and updated. However, there might be certain circumstances where you
- prefer not to have an association to the workload application so that the imported
- objects can be freely updated or deleted.

- When importing a workload application template, you can now specify whether
- you prefer to import only the objects contained in the template, without any ties to
- the workload application, or to import the objects maintaining their association to
- the workload application. Even after you have imported the contents of a
- workload application template into a target environment, removing the association
- to the original workload application template, you still have some options through
- the **wappman** command line that enable you to manage the objects as a whole, such
- as deleting all of the objects or replacing all of the objects.

- These workload application template enhancements are supported when connected
- to an engine with version 9.4.0.4 or later

- For details about importing a workload application template, see the topic about
- importing in the *User's Guide and Reference*, as well as the topic containing the
- command syntax for the **wappman** command.

- **Additional agent support**

- New agent support.

- Support for fault-tolerant agents, dynamic agents, and z/OS agents has been
- extended on Sun Solaris SPARC 64-bit.

- **Automatically register agents to pools**

- Starting from IBM Workload Scheduler version 9.4 Fix Pack 4, you can
- automatically register dynamic agents in pools by editing a file.

- Starting from version 9.4 Fix Pack 1, you can automatically register dynamic agents
- in pools by editing the `pools.properties` file located in `<TWS_home>/ITA/cpa/
- config`.

- The file is composed by a series of lines with a list of pools to which the agent will
- be automatically registered. To make the changes in this file effective on the agent,
- you must stop and start the agent.

- Because an agent can encounter problems and is not able to register and go online,
- for example, if it does not find a pool defined in the system, there are options that
- can be used in the `pools.properties` file to allow the agent to go online even if
- some pools are not defined.

- This alternative way of registering dynamic agents to a pool can be useful when
- you need to quickly add more than one agent to a pool, or when you want to
- associate multiple pools to a dynamic agent.

- For details about how to automatically register agents to pools, see the related
- topic in the *Planning and Installation Guide*.

Option to determine when fileCreated event is sent

The **-modificationCompletedTime** option can optionally be specified with the **-event fileCreated** argument to determine when the **fileCreated** event is sent.

The **filemonitor** utility is used to check for changes in files (files that are either created or modified). You can now specify the **-modificationCompletedTime <seconds>** option with the **-event fileCreated** argument to determine when the **fileCreated** event is sent.

This option is optional and when specified, if a file is created, the event is not sent immediately, but only after the interval of time specified by **-modificationCompletedTime<seconds>** has elapsed, and during which no subsequent changes were made to the file, which includes the file being deleted and recreated with the same name.

For more information about the filemonitor utility and syntax, see the related topic in the *User's Guide and Reference*.

Satisfying Requests for Enhancements (RFEs)

IBM Workload Scheduler satisfies Requests for Enhancements (RFEs).

Requests for Enhancements (RFEs) give customers the opportunity to collaborate directly with the product development team and other users. The team prioritizes and develops new product features based on proposals made by customers.

IBM Workload Scheduler V9.4 Fix Pack 4 delivers the following RFE:

RFE 117399: Windows 2016 WSCF cluster support

Support has been extended to Microsoft Windows Server 2016 Failover Clustering (WSCF) on agents.

To view a complete list of RFEs, new, planned, and delivered, see: RFE online community .

IBM Workload Scheduler version 9.4 Fix Pack 3 enhancements

Learn about the IBM Workload Scheduler version 9.4 Fix Pack 3 enhancements.

IBM Workload Scheduler version 9.4 Fix Pack 3 includes the following enhancements:

- “Extra opportunities for modifying job definitions already in the plan” on page xvi
- “Troubleshooting stalled jobs” on page xvi
- “Auto refresh Plan View” on page xvii
- “Easy installation for patches on agents” on page xvii
- “Enhancements to IBM i job monitoring and control” on page xvii
- “New Plug-ins for Cloud Automation” on page xviii
- “Apache Spark Plug-in” on page xix
- “Restarting JSR 352 Java Batch jobs from the point of failure” on page xix
- “More flexibility in managing successors for job actions” on page xix

Extra opportunities for modifying job definitions already in the plan

Modify a job instance in the plan before it runs or modify an instance of a job in the plan that has already run and rerun the modified job.

You can modify a job definition in the database whenever and as many times as you want. However, there are times when you need to make changes to the job definition, but it has already been submitted into the plan and runs as is. This results in extra work and lost time in updating the definition in the database and then getting it into the plan to run.

This feature adds the flexibility you need so that you can now make changes to the definition even after it has already been submitted into the plan, maintaining the original definition in the database. With this additional flexibility, you can edit the job definition on-the-fly before it runs or rerun a job with a different definition. This can be done from either the Job Stream Graphical View, the job monitoring view, or from the **conman** command line.

Maybe you want to substitute the command or script executed by the job with a different one? Maybe you just need to tweak an argument? Maybe you need to rerun a job updating the logon name, priority, or connection server? Whatever the change, this enhancement allows you to quickly react and avoid possible disasters, and increase your productivity by addressing additional scenarios in your workload that were not contemplated at the time you modeled or planned your workload.

For details about how to modify the job definition in the plan see the conman commands **altjob** and **rerun** in the *User's Guide and Reference*.

For details about how to modify the job definition in the plan from the Dynamic Workload Console, see the topic about controlling job and job stream processing in the *IBM Workload Automation: Dynamic Workload Console User's Guide*.

See an introduction to this feature in the video, *New opportunities to react and recover*, available on the Workload Automation YouTube channel.

Troubleshooting stalled jobs

Quickly identify what is holding back jobs that are ready to run but, for some unknown reason, do not run.

Sometimes your jobs are all ready to go, but for some anomalous reason, they do not start. A job can sometimes encounter specific circumstances where, although everything seems to be in check, there is still something that needs to be done before the job can start. You know that any dependencies it might have on predecessors have been resolved, you also know that the start time has come and gone, but something else is holding it back.

A new action is available when monitoring jobs that helps you identify the most commonly found circumstances under which a job ready to run, does not. In addition to problem determination, it also helps you in resolving the problem for these common situations. Sometimes it is as simple as tweaking the limit, other times a workstation needs to be linked or started. Whatever the reason might be, you can save time in determining the problem and identifying the solution with a new action available from the Dynamic Workload Console, "**Why a job does not start**".

* For details about how to troubleshoot stalled jobs, see the troubleshooting topic
* about why jobs do not start in the *Troubleshooting Guide*.

* **Auto refresh Plan View**

* Continuous update of the Show Plan View graphical view at regular intervals.

* To display a graphical view of your plan, you specify filter criteria such as, a
* scheduled time range, job stream name, and workstation name to narrow the
* results in the view. You can also choose to enable an automatic refresh mechanism
* of the view to ensure you are viewing the most up-to-date results at all times.

* An auto fresh option has been added to the **Show Plan View** filter page. Enabling
* this option triggers an auto fresh of the view at regular intervals. By default, the
* view is refreshed every five minutes. From the view itself, you can control the auto
* refresh from the toolbar to pause, stop, and resume.

* You can also configure the default refresh interval by editing the
* **PlanViewAutorefresh** section in the `TdwcGlobalSettings.xml` configuration file.

* See the topic about displaying a graphical plan view in the *Dynamic Workload
* Console User's Guide*.

* For information about editing the auto refresh default value in the
* **PlanViewAutorefresh** section of the `TdwcGlobalSettings.xml` file, see the
* information about customizing the global settings in the *Administration Guide*.

* **Easy installation for patches on agents**

* You can now easily install patches on your agents

* You can now easily install patches on agents using the **twinst** command with the
* **-patch** argument.

* When you specify this argument, only the files present in the patch package are
* replaced in the installed product while all other product files remain unchanged.
* For more information, see the section about upgrading agents and master domain
* managers in Planning and Installation.

* **Enhancements to IBM i job monitoring and control**

* Monitoring and control facilities for IBM i jobs have been enhanced to simplify the
* management of IBM i inquiry messages.

* **Automated reply to inquiry messages**

* For the most frequent IBM i inquiry messages, you can define standard
* rules to automate the reply to the waiting messages. When defining an
* IBM i job, by using the Workload Designer of the Dynamic Workload
* Console or the composer command line, you can specify the list of
* messages for which you want to set up an automated reply. When
* specifying the automated replies, a new parameter has been added to the
* job definition: the **Message Max Replies** parameter. It represents the
* maximum number of automated replies accepted for a specific message.
* This new parameter optimizes the management of IBM i inquiry messages.

* For example, when you set a wrong reply to a message in the job
* definition, IBM i system keeps on sending out the same inquiry message
* repeatedly, while waiting for the correct reply. To avoid this issue, IBM

* Workload Scheduler has now the capability to intercept and disable the
* wrong reply and require, with a prompt, a direct reply from the Dynamic
* Workload Console. The job remains in SUSP (suspended) status until the
* correct reply is provided.

* **Reliable monitoring of IBM i job status changes**

* As an inquiry message receives an automated reply, the IBM i job status
* changes from SUSP (suspended) to EXEC (executing) and vice versa. All
* the job status changes are monitored and tracked. This is useful, for
* example, when you want to create an event rule definition to send an
* email every time a job status change occurs.

* **Improved trace facilities**

* Trace facilities for IBM i jobs have been improved. To specify the desired
* tracking level, customize your IBM i agent by setting the required
* parameters in the JobManager.ini file, in accordance with the settings on
* the IBM i system.

* For more information, see the section about scheduling jobs on IBM i systems in
* *User's Guide and Reference*.

* See the video: Simplify workload management for your IBM i Systems on the
* Workload Automation YouTube channel. More videos are available for the features
* released with V9.4 Fix Pack 3 on this dedicated playlist: Workload Scheduler V9.4,
* Fix Pack 3.

* **New Plug-ins for Cloud Automation**

* Amazon EC2, IBM SoftLayer, and Microsoft Azure plug-ins are available to
* manage the provisioning and de-provisioning of virtual machines in the cloud, on
* as-needed basis.

* Customers choose to move their application to the cloud to focus on business
* optimization. Once in the cloud, applications rely on systems provisioned and
* de-provisioned to run defined business workflows but unpredictable workload
* volumes. Customers face a new question: Is it better to over-provision cloud
* resources with the risk of wasting them, or under-provision with the risk of
* degrading performance and delay the business process?

* How about provisioning just the right amount of resources for only the
* time-period you need them? Exactly what you need, when you need it, and rather
* than incur extra costs and waste, de-provision when you're done, all automatically.

* To succeed in this challenge, a new approach is required, that strictly ties business
* workflow with cloud resource management. While managing a business
* application, IT organizations need to be able to orchestrate provisioning and
* de-provisioning of the infrastructure needed by the business application in the
* cloud.

* IBM Workload Scheduler provides three Cloud Automation plug-ins, for different
* cloud providers, to manage the provisioning and de-provisioning of virtual
* machines in the cloud, on as-needed basis. By orchestrating the application
* workflow and the workflow that manages the entire lifecycle of the virtual
* machines needed by the application (including the actions: start, stop, snapshot,
* etc...), IBM Workload Scheduler can increase both business and infrastructure
* agility. The plug-ins are:

- * • Plug-in for **Amazon Web Services (EC2)**

- Plug-in for **IBM SoftLayer**
- Plug-in for **Microsoft Azure**

Select your cloud provider, and add one or more jobs in the job stream that automates your business process flow to provide a flexible and dynamic allocation of cloud resources to your workload.

For more information about the Cloud Automation plug-ins, see *Scheduling Applications with IBM Workload Automation*.

See the video: Workload Scheduler for Cloud Automation on the Workload Automation YouTube channel. More videos are available for the features released with V9.4 Fix Pack 3 on this dedicated playlist: Workload Scheduler V9.4, Fix Pack 3.

Apache Spark Plug-in

With the new IBM Workload Scheduler plug-in for Apache Spark, you can schedule, monitor and control Apache Spark jobs.

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop Map Reduce and extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing. The main feature of Apache Spark is its in-memory cluster computing that increases the processing speed of an application. IBM Workload Scheduler provides a plug-in for Apache Spark that helps you manage your big data processing and analytics. With the plug-in for Apache Spark, you can define, schedule, monitor, and control Apache Spark jobs. Add one or more Apache Spark jobs in the job stream that automates your business process flow to obtain an end-to-end workload management solution.

For details, see *Scheduling Applications with IBM Workload Automation*

Restarting JSR 352 Java Batch jobs from the point of failure

You can restart JSR 352 Java Batch jobs from the point of failure.

During the execution of a JSR 352 Java Batch job, when monitoring the job from the Dynamic Workload Console, the **Workflow Details** panel displays information about the steps that already started on the JSR 352 Java Batch server.

From the Monitor jobs view, in the Job Type column, click the hyperlink Workflow Details. The **Workflow Details** panel opens. If the JSR 352 Java Batch job fails, you can restart the workflow from the first failed step, within the same job instance.

For details about defining, monitoring, and restarting JSR 352 Java Batch jobs, see *IBM Workload Scheduler: User's Guide and Reference*.

More flexibility in managing successors for job actions

Hold, rerun and release successors in the plan during the monitoring of your workload.

You can now hold, release and rerun the successors during your job monitoring.

You have a list of all successors for the selected job, view the successors in two separate tables and decide if you want to run all internal successors, all internal and external successors, or a subset of them. By default, all successors are selected.

* By selecting the mode you can also control how successors are selected: manually,
* including successors in the same job stream or including successors in other job
* streams.

* For more information about how to rerun successors from the command line, see
* the section about the **listsucc** and **rerunsucc** commands in the *User's Guide and*
* *Reference*.

* For more information about how to specify the rerun options in the job definition
* from the Dynamic Workload Console, see the section about using job recovery
* actions to control job processing in the *Dynamic Workload Console User's Guide*.

* **Satisfying Requests for Enhancements (RFEs)**

* IBM Workload Scheduler satisfies Requests for Enhancements (RFEs).

* Requests for Enhancements (RFEs) give customers the opportunity to collaborate
* directly with the product development team and other users. The team prioritizes
* and develops new product features based on proposals made by customers.

* IBM Workload Scheduler V9.4 Fix Pack 3 delivers the following RFEs:

* **RFE 89523: Modify/change job definition directly in the plan**

* Edit the definition of jobs in the plan that have not yet started or, that have
* already run and rerun them with a different definition. See "Extra
* opportunities for modifying job definitions already in the plan" on page
* xvi.

* **RFE 102777: Warning-Troubleshooting if limit is reached**

* A dedicated action to help determine why jobs that are ready to run, do
* not run, and how to solve the issue. See "Troubleshooting stalled jobs" on
* page xvi.

* **RFE 110252: Improvements for Java batch**

* Restart JSR 352 Java Batch jobs from the point of failure. See "Restarting
* JSR 352 Java Batch jobs from the point of failure" on page xix.

* **RFE 114673: Autorefresh for Plan Graphical View**

* From the Dynamic Workload Console, an option to refresh the Plan View
* at regular intervals. By default, the view is refreshed every 5 minutes. The
* refresh interval is configurable by editing the value in the
* **PlanViewAutorefresh** section in the `TdwcGlobalSettings.xml` configuration
* file. See "Auto refresh Plan View" on page xvii.

* **RFE 114590: Submit job (SBMJOB) on IBM i dynamic agents**

* A new parameter has been introduced, **MessageMax Replies**. See
* "Enhancements to IBM i job monitoring and control" on page xvii

* To view a complete list of RFEs, new, planned, and delivered, see: RFE online
* community .

/ **IBM Workload Scheduler version 9.4 Fix Pack 2 enhancements**

/ Learn about the IBM Workload Scheduler version 9.4 Fix Pack 2 enhancements.

/ IBM Workload Scheduler version 9.4 Fix Pack 2 includes the following
/ enhancements:

- / • "Rerun multiple jobs" on page xxi

- “Rollback procedure”
- “Actual workstation in pool and dynamic pool” on page xxii
- “Satisfying Requests for Enhancements (RFEs)” on page xxii

Mixed pricing models

Easily define different pricing models on workstations in your environment.

According to your IBM Workload Scheduler license, IBM® License Metric Tool helps you maintain your license compliance. By using License Metric Tool, you can generate reports that summarize your license consumption. The generated reports are maintained on the License Metric Tool server and should be periodically reviewed and signed, creating a history for audit purposes in the process. If you are contacted by a third-party software compliance auditor who plans to visit your enterprise to carry out a software audit, ensure that all reports are up-to-date and signed, and then supply copies of reports that cover the time periods that the auditor requests.

You can now take advantage of improved flexibility when defining your pricing model. For each single workstation in your environment, you can define the pricing model to be applied.

When you set the **licenseType** keyword to **byWorkstation** in **optman**, you can define for each single workstation the pricing model to be applied to it at the workstation creation time.

For more information about License Management, see the section about License Management in *Administration Guide*.

Rerun multiple jobs

Rerun multiple jobs without having to confirm the rerun action for each individual job.

You can now select multiple jobs from the Dynamic Workload Console and rerun all of them in a single action. Previously, a confirmation dialog prompted you to confirm the rerun for each job selected. When rerunning a large number of jobs this can be time consuming. With this enhancement, a confirmation dialog lists the jobs you want to rerun and prompts you a single time to confirm the rerun on all of the jobs.

A side effect of this new feature is a more responsive user interface with improved response times. The more frequently used actions such as, setting the priority or limit from the Dynamic Workload Console monitoring portlet, have been updated so that they run more smoothly.

Rollback procedure

Roll back a master domain manager to a previous fix pack level or release if the master domain manager was installed with IBM Installation Manager.

To roll back a master domain manager to a previous fix pack level or release, you first need to create a backup before installing the new fix pack or release. This allows you to then perform a rollback procedure after the fix pack or release has been installed.

/ For details about the procedure see the topic about rolling back a master domain
/ manager in the *Planning and Installation Guide*.

Actual workstation in pool and dynamic pool

/ The name of the workstation where a job, scheduled to run on a pool or dynamic
/ pool, actually ran.

/ When jobs are scheduled to run on pools or dynamic pools, you might want to
/ monitor the job or the workstation where the job ran. Previously, this information
/ was available only in the job log. With this enhancement, the name of the actual
/ workstation where the job ran is also available in a new column in the monitor job
/ query. This detail is available if the job has started or has already run. This
/ information can also be output in reports.

/ This information can also be useful when you need to determine your license
/ consumption and therefore need to know on which workstation in the pool the job
/ actually ran.

Satisfying Requests for Enhancements (RFEs)

/ IBM Workload Scheduler satisfies Requests for Enhancements (RFEs).

/ Requests for Enhancements (RFEs) give customers the opportunity to collaborate
/ directly with the product development team and other users. The team prioritizes
/ and develops new product features based on proposals made by customers.

/ IBM Workload Scheduler V9.4 Fix Pack 2 delivers the following RFEs:

RFE 179819: Database job executor handles stored procedure errors

/ Enhance the database job executor to handle stored procedure errors.

RFE 109287: Rerun multiple jobs with a single confirmation prompt

/ Rerun multiple selected jobs when monitoring jobs and confirm the action
/ for all jobs only once. See “Rerun multiple jobs” on page xxi.

RFE 102143: Lengthen text field for script path

/ Lengthen the size of the field related to a script to be run. When defining a
/ job that runs a script as the task, the text field specifying the path to the
/ script can now accommodate a much longer string. For example, if the
/ path is quite long or if it contains numerous variables, they are now
/ displayed in the text field when you view the job definition from both the
/ Workload Designer and from the List Workload Definitions portlet.

Replying to messages for IBM i child jobs

/ Support for replying to messages for an IBM i parent job was introduced
/ in a previous release. With this release, the same support is extended to
/ IBM i child jobs. When an IBM i job is in SUSP (suspended) status, waiting
/ for a reply to a message, you can reply to the message for parent and child
/ jobs from the Dynamic Workload Console when monitoring the parent job.

Add actual start date in job stream view

/ In addition to the scheduled start date, information related to the actual
/ start date has been added to the tooltip for jobs and job streams in the Job
/ Stream View on both distributed and z/OS engines.

RFE 15616: Actual workstation displayed when monitoring jobs

/ When jobs are scheduled to run on pools or dynamic pools, the name of

/ the actual workstation where the job runs can be monitored if the job has
/ started or has already run. See “Actual workstation in pool and dynamic
/ pool” on page xxii

/ **RFE 108425: File monitor support for existing files**

/ Normally the **filemonitor** utility runs an initial scan and then runs
/ subsequent scans to detect any new or changed files since the initial scan
/ that match specific criteria. This means that if there are any existing files
/ matching the criteria when the initial scan runs, they are not considered.
/ The utility has been updated with a new parameter to be able to discover
/ existing files during the initial scan that match the criteria and can
/ therefore generate an event. For more information see the
/ **-generateEventsOnFirstScan** argument for the **filemonitor** utility in the
/ *User’s Guide and Reference*

/ **RFE 106345: RUN PJP and PVU License Networks in the SAME Network**

/ Easily define different pricing models on workstations in your
/ environment. For more info see: the section about License Management in
/ IBM License Metric Tool in the *IBM Workload Scheduler: Administration*
/ *Guide*

/ To view a complete list of RFEs, new, planned, and delivered, see: RFE online
/ community .

= **IBM Workload Scheduler version 9.4 Fix Pack 1 enhancements**

= Learn about the IBM Workload Scheduler version 9.4 Fix Pack 1 enhancements.

= IBM Workload Scheduler version 9.4 Fix Pack 1 includes the following
= enhancements:

- = • “Revamped graphical views”
- = • “What-if enhancements” on page xxiv
- = • “IBM Workload Scheduler Job Management Plug-in” on page xxv
- = • “Automation of Iterative Workflows” on page xxv
- = • “Advanced rerun flexibility” on page xxvi
- = • “IBM Workload Scheduler Job Stream Submission Plug-in” on page xxvii
- = • “Condition-based workload automation” on page xxvii
- = • “Monitor file changes using the filemonitor utility” on page xxviii
- = • “New keyword for defining the latest start time in job streams” on page xxviii
- = • “New keyword for defining actions on late jobs” on page xxix
- = • “Higher level of control over the What-If Analysis” on page xxix
- = • “Preparing and installing a Docker image for dynamic agents” on page xxix
- = • “Upgrading from version 8.6.x” on page xxx
- = • “New options in managing workload applications” on page xxx
- = • “Integration with IBM UrbanCode Deploy” on page xxxi
- = • “Satisfying Requests for Enhancements (RFEs)” on page xxxi

= **Revamped graphical views**

= Enhancements made to the graphical view of your database objects has been
= extended to the plan to provide a consistent view and user experience.

= **Consistent views and revamped user experience across graphical views: Job**
= **Stream View, Preproduction Plan View, Show Plan View, Modelling Graphical**

= exclude them from the view to optimize performance when you have
= hundreds of predecessors. The default setting is to load all predecessors in
= the What-if Analysis Gantt view.

= **Setting the Earliest Start Time to the Estimated Start Time calculated by the**
= **server** Even if all predecessors are not displayed in the view, you can still
= consider the impact of predecessors on a job's Earliest Start Time. The
= server calculates an Estimated Start Time for the job and through a new
= action named "Set Earliest Start Time to server estimation", you can set the
= job's Earliest Start Time, to the time calculated by the server as the
= Estimated Start Time.

= For more information about What-if analysis capabilities, see the topic about
= analyzing the impact of changes on your environment in the *Dynamic Workload*
= *Console User's Guide*.

= **IBM Workload Scheduler Job Management Plug-in**

= The new Job Management plug-in is available to run actions on a job in a job
= stream.

= The new Job Management plug-in is available in the Automation Utilities plug-in
= category. Automation Utilities are plug-ins that facilitate specific IBM Workload
= Scheduler operations. Use the Job Management plug-in, to run one of the following
= actions on any job in the job stream where the Job Management job is running:

= *Table 1.*

Actions that you can run on a job
<ul style="list-style-type: none">• Rerun• Rerun the job and all its successor jobs• Rerun the job and its successor jobs in the same job stream• Release• Release Dependencies• Cancel• Cancel Pending• Hold• Kill• Confirm ABEND• Confirm SUCC

= The Job Management plug-in simplifies recovery scenarios and enables the
= automation of iterative workflows.

= For more information, see the topic about Job Management jobs in the *User's Guide*
= *and Reference*.

= See the video Automation of Iterative Workflows on the Workload Automation
= YouTube channel.

= **Automation of Iterative Workflows**

= The new Job Management plug-in enables iterative processing of a sequence of
= jobs within a job stream while a condition is met.

= For complex workflows, the iteration of sequence of jobs within the overall
= orchestration can be a challenging task. By selecting the option to rerun a job with
= all its successors jobs in the same job stream, the Job Management plug-in enables
= iterative processing of a sequence of jobs while a condition is met. It is like a DO
= WHILE statement in programming languages.

= It is just a matter of adding a Job Management job at the end of the sequence of
= jobs that you want to iterate. The Job Management job requests to rerun the first
= job in the sequence along with its successor jobs. The iteration is controlled by the
= output conditions of the first job in the sequence: it stops when the condition is no
= longer satisfied.

= For more information, see the topic about Job Management jobs in the *User's Guide*
= and *Reference*.

= For a complete scenario, see the video Automation of Iterative Workflows on the
= Workload Automation YouTube channel.

= **Advanced rerun flexibility**

= Ensure workload continuity with the new and improved recovery options for
= failed jobs

= The new advanced rerun options help you orchestrate your workflows seamlessly,
= building recovery logic into the job definition itself and rerunning job successors
= directly from the Monitor Workload view.

= When you create a job definition, you can now specify that you want the job to
= rerun for a specific number of times and after a specific interval, in case of failure.
= This ensures that fewer alerts are generated and the workflow continues smoothly.
= For example, if you have a job that needs to connect to a server which is
= periodically restarted, you can specify in the job definition that you want the job to
= rerun for a specific number of times and after a specified interval.

= If the parent job ran on a workstation that is part of a pool or a dynamic pool, you
= can decide whether it must rerun on the same workstation or on a different one.
= This is because the workload on pools and dynamic pools is assigned dynamically
= based on a number of criteria and the job might be rerun on a different
= workstation.

= Also, if a job fails, you can identify all its successors at a glance and decide
= whether you want to rerun the job with its successors. You can rerun either all
= successors in the same job stream, or all successors overall, both in the same job
= stream and in other job streams, if any.

= For more information about how to specify the rerun options in the job definition
= from the command line, see the section about defining job rerun and recovery
= actions in *User's Guide and Reference*. For more information about how to specify
= the rerun options in the job definition from the Dynamic Workload Console, see
= the section about using job recovery actions to control job processing in the
= *Dynamic Workload Console User's Guide*.

= For more information about how to rerun the job and its successors from the
= command line, see the section about the **listsucc** and **rerunsucc** commands in the
= *User's Guide and Reference*.

= Use the **Job Management** plug-in to further automate your workflows. With the
= plug-in, you can perform a number of actions on jobs, such as rerun the job, with
= or without its successors, release its dependencies, or cancel the job, and many
= more. For example, in a recovery scenario, you can insert a Job Management job in
= your workflow. This causes the original job to rerun automatically when the
= relevant recovery job completes successfully, reducing effort and time consumption.

= For a detailed video about this feature, see the **Advanced Rerun Flexibility** video
= available on the Workload Automation YouTube channel.

= **IBM Workload Scheduler Job Stream Submission Plug-in**

= The new Job Stream Submission plug-in is available to automate the submission of
= a job stream for processing.

= The new Job Stream Submission plug-in is available in the Automation Utilities
= plug-in category. Automation Utilities are plug-ins that facilitate specific IBM
= Workload Scheduler operations. Use the Job Stream Submission plug-in, to submit
= a job stream for processing.

= By adding the Job Stream Submission plug-in to your workflow, you can automate
= the submission of a specific job stream, minimizing code scripts and manual effort.
= Also, you can specify the earliest start time for the job stream and define the
= variable table associated to the job stream.

= For more information, see the topic about Job Stream Submission jobs in the *User's*
= *Guide and Reference*.

= **Condition-based workload automation**

= Have your workflows start at just the right time

= Condition-based workload automation provides a simple and immediate way to
= have your workflows start at just the right time. You can define in your job stream
= a condition that, when met, releases the job stream to run as scheduled.

= For example, if you have a job stream containing jobs which analyze one or more
= files, you can have the job stream start only after the file or files have been
= modified or created. Also, if the job stream contains jobs which process the data in
= a database, you might want to have the job stream start after enough rows have
= been written into the database. You can also have IBM Workload Scheduler check
= repeatedly whether the condition is met.

= You can start your workflow based on one of the following conditions:

- = • One or more files being created
- = • One or more files being modified
- = • A job completing with its output condition satisfied. You can apply this logic to
= the job stream or to specific jobs in the job stream.

= For more information, see the section about condition-based workflow automation
= in the *User's Guide and Reference*.

= You can also find a detailed explanation of the benefits provided by this
= enhancement in the Marketing-oriented video and in the How to video on the
= Workload Automation YouTube channel.

Monitor file changes using the filemonitor utility

Monitor changes in files with an easy-to-use utility command

Use the **filemonitor** utility to check for changes of files (files that were either created or modified). This could be useful when, for example, you want to make sure that a file exists before running a job that processes that file. By defining a job that runs the **filemonitor** utility, you can implement file dependency, that is, a relationship between a file and an operation in which specific activity on the file determines the starting of the operation.

You can use the **filemonitor** utility as a stand-alone command, or you can set the **filemonitor** keywords as additional parameters for the start condition of a job stream, either in the Workload Designer or from the **composer** command line. For more information about the start condition, see the section about condition-based workflow automation in the *User's Guide and Reference*.

For more information, see the section about **filemonitor** in the *User's Guide and Reference*.

For a comprehensive library of videos about the new features for this release, see the Workload Scheduler version 9.4, Fix pack 1 playlist available on the Workload Automation YouTube channel.

New keyword for defining the latest start time in job streams

Manage the latest start time for job streams more efficiently

The **jsuntil** keyword defines the latest start time of a job stream. It also determines the behavior of the jobs in the job stream when the job stream is approaching its latest start time. Use the **jsuntil** keyword to avoid that the job stream is either suppressed, canceled, or set to continue (depending on the action specified in the **onuntil** keyword) if it starts before its latest start time. For example, if you have a job stream with **jsuntil** set to 10:00 am, and one of the jobs starts running at 9:59 am, the job and its successors run as scheduled.

This keyword is mutually exclusive with the **until** keyword.

There is also a major difference with between the **until** and **jsuntil** keywords:

If you specify the **until** keyword in your job stream definition

This keyword is evaluated also after the job stream has started. As a result, if the latest start time expires before the job stream completes successfully, the action specified in the related **onuntil** keyword is performed on the job stream and on its jobs, which have not yet started.

If you specify the **jsuntil** keyword in your job stream definition

This keyword is evaluated only once, as soon as all dependencies of the job stream are satisfied and the job stream state changes to READY. If the latest start time defined using the **jsuntil** keyword has not expired at this time, it is no longer evaluated and the job stream runs independently of it. However, to prevent the job stream from remaining in READY state indefinitely, two days after the time specified in the **jsuntil** keyword has expired, the job stream is suppressed by default.

For more information, see the section about the **jsuntil** keyword in *User's Guide and Reference*.

New keyword for defining actions on late jobs

Manage the deadline for jobs in job streams more efficiently

The **onlate** keyword defines the action to be taken on a job in job stream when the job's deadline expires. If the job is running when the deadline expires, it is killed. Killed jobs end in the ABEND state. Any jobs or job streams that are dependent on a killed job are not released. If the dependency on the job is a conditional dependency on the job completing in ABEND state, that dependency is released.

For more information, see the section about the **onlate** keyword in *User's Guide and Reference*.

Higher level of control over the What-If Analysis

A new **optman** global option is available to administrators, to make the What-If Analysis feature optional in your environment.

Administrator's level of control over the What-If Analysis has increased with the introduction of a new **optman** global option. By setting the **optman** enWhatIf | wi global option to *no*, administrators can centrally disable the What-If Analysis feature, which is enabled by default in your environment to simulate and evaluate the impact of changes on the current plan. You have to run "JnextPlan" to make the change effective.

For more information, see the topic about analyzing the impact of changes on your environment in the *Dynamic Workload Console User's Guide*.

For more information, about the interaction of the enWhatIf | wi global option with the enWorkloadServiceAssurance | wa global option, which enables or disables privileged processing of mission-critical jobs and their predecessors, see the topic about disabling the What-If Analysis in the *Administration Guide*.

Preparing and installing a Docker image for dynamic agents

Quickly provision new dynamic agents with little effort.

A Docker container automates the task of installing a running dynamic agent, along with everything that is required to run it: code, runtime, system tools, system libraries, and settings.

It provides a mechanism known as a **Dockerfile**, which is used to codify the steps to install and configure your dynamic agent. This is a plain text file that uses a standard set of commands to perform the all the installation and configuration steps. The resulting Docker image is a static image of the full set of software for the dynamic agent with the exact configuration.

An image is then built to become a Docker Container, which is a unique running instance of the software with its own processes, state, and configuration. You can start as many Docker Containers from a given image as necessary. This gives you the possibility to quickly provision multiple running instances of your software, each with their own processes, state, and configuration, quickly and easily.

For more information, see the section about preparing and installing a Docker image for dynamic agents in the *User's Guide and Reference*.

= For a detailed video about this feature, see the **Docker Agent for Workload Scheduler** video available on the Workload Automation YouTube channel.

= **Upgrading from version 8.6.x**

= Support for upgrades from previous versions.

= IBM Workload Scheduler V9.4 General Availability supported upgrades from V9.x. With this fix pack, support is being extended to V8.6.x instances.

= For IBM Workload Scheduler V8.6.x instances, the upgrade is supported only for the master domain manager using the parallel upgrade method. The direct upgrade is not supported for IBM Workload Scheduler V8.6.x instances. The only supported scenario for the upgrade of a V8.6.x master domain manager is to install a new master domain manager configured as a backup.

= You can upgrade a V8.6.x single instance of the Dynamic Workload Console either on the same workstation where the back-level is installed, but in a different directory, or on a new workstation.

= When a V8.6.x instance of the Dynamic Workload Console is installed with one or more components in the same directory, then you must upgrade the Dynamic Workload Console in a new directory and then uninstall the old version. The additional components either remain at the V8.6 level or they can be upgraded to V9.4

= **Note:**

= If you are upgrading from V9.1, V9.2, or V9.3 to V9.4.x, you can upgrade directly to the latest fix pack level, with just one single step (without the need to upgrade to V9.4.0 first). For more information, see the fix pack readme file.

= **New options in managing workload applications**

= New options are available to make the management of workload applications more flexible in your environment.

= The following new options are available:

= **Export a job stream definition as a workload application template**

= From the Workload Designer, you can export a job stream definition and save it as a workload application template in a compressed file. The job stream definition can then be imported in another environment.

= For more information, see the topic about exporting a job stream definition as a workload application template in the *Dynamic Workload Console User's Guide*.

= **Rename a workload application during the import process**

= A new parameter `-workloadApplicationName <workload_application_name>` is available for the `wappman -import` command to rename the workload application during the import process. A similar option is available when importing a workload application from the Dynamic Workload Console.

= For more information, see the topic about the `wappman` command in the *User's Guide and Reference*.

= **Modify the mapping file according to rules defined using regular expressions**

= You can optionally request that the mapping file produced by the export

= process is automatically modified by the import process, according to rules
= defined using regular expressions and specified in ad-hoc files.

= For more information, see the topic about using regular expressions to
= modify the mapping file in the *User's Guide and Reference*.

= **Integration with IBM UrbanCode Deploy**

= A new integration with IBM UrbanCode Deploy is available to streamline your
= application deployment.

= Workload applications can be created and then exported so that they can be
= imported in other IBM Workload Scheduler environments.

= To export and import a workload application you can use either the Dynamic
= Workload Console or the **wappman** command line.

= As an alternative to the import process, you can automate the deployment of an
= application from one environment to another by using the Workload Automation
= plug-in of the IBM UrbanCode Deploy tool. Through the definition of a reusable
= template, the integration with IBM UrbanCode Deploy streamlines the deployment
= of your application with all of its associated scheduling objects. For details about
= this plug-in, see the IBM UrbanCode Deploy documentation.

= For a comprehensive library of videos about the new features for this release, see
= the Workload Scheduler version 9.4, Fix Pack 1 playlist available on the Workload
= Automation YouTube channel.

= **Satisfying Requests for Enhancements (RFEs)**

= IBM Workload Scheduler satisfies Requests for Enhancements (RFEs).

= Requests for Enhancements (RFEs) give customers the opportunity to collaborate
= directly with the product development team and other users. The team prioritizes
= and develops new product features based on proposals made by customers.

= IBM Workload Scheduler V9.4 Fix Pack 1 delivers the following RFEs:

= **RFE 17980**

= Restart the job stream from the given job of a job stream. (Internal ID
= 57221)

= **RFE 25295**

= Rerun a specific job and all of its successors, both in the same job stream
= and in other job streams. For more information, see the section about
= condition-based workflow automation in the *User's Guide and Reference*.
= (Internal ID 65671)

= **RFE 33200**

= Create external dependency from the graphical Plan View. (Internal ID
= 79752)

= **RFE 44226**

= Need a copy button on the Monitor Jobs panel EDIT JCL. (Internal ID
= 99136)

= **RFE 46521**

= Improve recovery options for a job. (Internal ID 103418)

=

=

=

=

=

=

=

RFE 65873

Changed behavior of selections in tables in the Dynamic Workload Console. When performing an action on a selected item in the table, either by right-clicking or from the toolbar, after the action is performed the selection is cleared so that you can perform a different action on a different selection. You can also multi-select items in a table and perform an action on all selected items. (Internal ID 125267)

=

RFE 69212

When rerunning a job in a pool, you can optionally rerun the job on the same workstation where it previously ran. For more information, see the section about condition-based workflow automation in the *User's Guide and Reference*. (Internal ID 130343)

=

RFE 78682

Additional columns are now available when monitoring jobs on multiple engines. (Internal ID 144976)

=

RFE 80759

Hyperlinked properties when updated should auto-refresh automatically in DWC. (Internal ID 146988)

=

RFE 101904

Search option improvements in the Dynamic Workload Console Workload Designer allows for object selection before inputting keyword so that the search is run against the object selected. (Internal ID 172565)

=

RFE 101905

The search field in the Dynamic Workload Console Workload Designer now provides user assistance with examples of the syntax that can be used in the field. (Internal ID 172564)

=

RFE 104082

Dynamic Workload Console - Disable IE Compatibility View. (Internal ID 175746)

=

To view a complete list of RFEs, new, planned, and delivered, see: RFE online community .

=

IBM Workload Scheduler version 9.4 enhancements

Learn about the IBM Workload Scheduler version 9.4 enhancements.

About this task

IBM Workload Scheduler version 9.4 includes the following enhancements:

- “Agent upgrade with minimal scheduling disruption” on page xxxiii
- “IBM Workload Scheduler Plug-in for Cloudant” on page xxxiii
- “**Keeping track of changes to scheduling objects**” on page xxxiv
- “**Auditing release management**” on page xxxv
- “**Version control**” on page xxxv
- “**Backup copy of tws_env script**” on page xxxvi
- “New event-driven workload automation action to open a ServiceNow incident” on page xxxvi
- “IBM i job definition enhancements” on page xxxvi
- “Passing variables between jobs” on page xxxvii

- “Graphical view enhancements” on page xxxvii
- “IBM Workload Scheduler REST API” on page xxxvii
- “Satisfying Requests for Enhancements (RFEs)” on page xxxviii

Agent upgrade with minimal scheduling disruption

Upgrade your agents with little or no impact on your job scheduling.

Planning the maintenance of your agents no longer means negotiating rare idle time windows. You want to ensure scheduling continuity while minimizing upgrade efforts. If an upgrade is not carefully planned, you risk prolonged scheduling downtime and unmet service level agreements.

With IBM Workload Scheduler agent upgrade with minimal scheduling disruption, both fault-tolerant agent and dynamic agent instances can be upgraded with minimal impact to scheduling activities. The agents are stopped for the shortest possible time, and although any jobs already running when the upgrade process begins continue to run as planned, no new jobs begin execution during this brief period. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run according to schedule, and any that have since finished running, report a successful status as usual.

The same behavior is applied to active agent command-line interfaces and processes such as **conman** or **composer**. After the agent upgrade, all processes and command-line interfaces resume and continue to run regularly.

The entire upgrade process is reliable and fail-safe with an automatic backup and restore feature in case of failure, and it all takes just a few minutes to complete.

Important: The agent upgrade with minimal scheduling disruption is supported for upgrading only version 9.x fault-tolerant and dynamic agent instances.

For information about upgrading agents with minimal disruption, see the topic about upgrading agents in the *Planning and Installation Guide*.

See the video Agent upgrade with minimal scheduling disruption on the Workload Automation YouTube channel.

IBM Workload Scheduler Plug-in for Cloudant

With the new IBM Workload Scheduler plug-in for Cloudant, you can schedule, monitor and control the execution of actions on IBM Cloudant NoSQL database, on its documents, or attachments.

With the IBM Workload Scheduler plug-in for IBM Cloudant NoSQL database, you can schedule, monitor, and control the following actions:

- Create, read, and delete a Cloudant database
- Start and monitor a Cloudant database replication
- Create, read, update, and delete Cloudant database documents
- Create, read, update, and delete Cloudant database attachments

You can gain many benefits from this integration. See the following examples:

- You can schedule and monitor a database replication on a time or event basis. You can even set a conditional dependency for the database replication, based on the value of variables passed from one job to another, such as the size of the database.
- You can monitor and control Cloudant operations from the Dynamic Workload Console or from Application Lab, from which you can also restart in case of failure.
- Daily logs generated by your Cloudant application can be automatically uploaded to your Cloudant database as documents or document attachments.
- You can balance your Cloudant workload and integrate it with the rest of your process flow.

The plug-in is available for IBM Workload Scheduler on Premises, on Cloud (SaaS), and on the Bluemix platform.

For more information, see the section about defining Cloudant jobs in *Scheduling Applications with IBM Workload Automation*.

Keeping track of changes to scheduling objects

Keep a detailed auditing track of all changes to scheduling objects in your environment

Maintaining and keeping control of a complex scheduling environment can be a challenging task: multiple schedulers, operators, and administrators have access to the job flows and can implement changes. Regardless of the complexity of this task, however, you must keep a record of every change being implemented. You can now easily track down everything with IBM Workload Scheduler and obtain detailed reports with just a few clicks using the integration with Tivoli Common Reporting.

Administrators can optionally enforce a policy by which each user making a change to an object must provide a justification for the change.

Administrators can maintain an audit trail, consisting of detailed information about the user who performed the change, the time and date when the change was performed, the reason why the change was implemented, and the details of the change for every modified item.

From the Dynamic Workload Console, IBM Workload Scheduler administrators, operators, and schedulers can review all changes to scheduling objects, both in the database and in the plan, discover which user performed a specific change, and the time and date when the change was performed.

Schedulers can review the history of all changes made on a specific object at any time and check previous versions of the object in scheduling language.

For more information, see the section about keeping track of changes in *Dynamic Workload Console User's Guide*.

You can find more information and a detailed business scenario in the Stay in control of your workload video.

This new feature causes three changes in your default optman settings:

auditStore

When you upgrade the master domain manager from a previous release, the default value for this global option is changed. The default value is now **both**. If you customized the default value in the previous release, the value is overwritten with the new value, with the exception of the **auditStore** option with the **DB** value assigned. If the **auditStore** option was set to **DB**, this value is maintained and is not overwritten.

enDbAudit

When you upgrade the master domain manager from a previous release, the default value for this global option is changed. The default value is now **1**. If you customized the default value in the previous release, the value is overwritten with the new value

enPlanAudit

When you upgrade the master domain manager from a previous release, the default value for this global option is changed. The default value is now **1**. If you customized the default value in the previous release, the value is overwritten with the new value

Auditing release management

Auditing your release management process ensures your changes are delivered consistent every time and with the same level of quality.

Automate and schedule the development, test, and promotion into production of an application or business process with the option to rollback to a previous version should something go wrong. When pushing any change into production, it usually goes through various cycles of development, testing, and fixing. It also passes through various environments to do this, and not all environments are identical. This is where versioning is key. Keeping track of different versions in the test and production environment is critical in avoiding conflicts. You need a release management process that is simple and that can push changes into production in a consistent and reliable manner.

With IBM Workload Scheduler, changes can be promoted from one environment to another by creating and exporting a workload application template. A workload application template is a compressed file containing one or more job streams and all the related jobs contained in them, including dependencies, files, resources, calendars, run cycle groups, and prompts. With this easy method for replicating job flows across environments, automating and scheduling the development, test, and promotion into production is simple and easy.

For more information, see the section about keeping track of changes in *Dynamic Workload Console User's Guide*.

For more information about defining workload application templates, see *User's Guide and Reference*.

More information and a detailed business scenario is explained in this video: Streamline your release management process.

Version control

Gain full control of changes in your scheduling environment.

Version control is most often used to track and control changes to software source code. However, to meet change management and auditing requirements, version control must also be applied to the scheduling objects, like jobs and job streams, associated to a certain application.

IBM Workload Scheduler is a modern, advanced workload automation solution supporting Version Control. IBM Workload Scheduler maintains all versions of your scheduling objects in the database, and you can easily access them from the Dynamic Workload Console. For each scheduling object, you can view the history of changes, discover which user made a specific change, the time stamp of the change, and the reason for the change. Then, you can compare two different versions in a comparison viewer and restore a previous version.

For more information, see the section about keeping track of changes in *Dynamic Workload Console User's Guide*.

More information and a detailed business scenario is explained in this video: [Version Control](#).

Backup copy of tws_env script

When upgrading an agent or master domain manager, a backup copy of the tws_env script is created, and a new one is installed.

To address new product features, the upgrade installation process for a master domain manager and for agents installs a new version of the tws_env script in the directory <TWA_HOME>/TWS, where <TWA_HOME> is the IBM Workload Scheduler installation directory. A backup copy of your original version is created in a backup directory. After the upgrade process, merge the content of the new version with the content of the original version to carry your customized content into the new version.

For more information, see the "Configuring" section that explains setting the environment variables in the *Planning and Installation Guide*.

New event-driven workload automation action to open a ServiceNow incident

About this task

IBM Workload Scheduler provides an event-driven workload automation action, *Open Incident*, available through the integration with ServiceNow to automatically trigger the opening of an incident when a job that matches a defined policy ends in error.

For more information about the *Open Incident* action, see the section "Action providers and definitions" in *User's Guide and Reference*.

IBM i job definition enhancements

The IBM i job definition has been enhanced for jobs running SBMJOB command .

The job definition for IBM i jobs has been simplified as far as the specification of the input parameters for jobs running SBMJOB command. Input parameters include the custom libraries that you can specify in addition to the system libraries.

For more information, see the section about defining IBM i jobs in *User's Guide and Reference*.

Passing variables between jobs

The IBM i job definition has been enhanced for jobs running SBMJOB command .

Passing variables between jobs has become an easy task. You can now use **variable tables** to set variables exported from a job, and pass the variables to any successor job, in the same job stream or in a different job stream.

To export variables from a job into a variable table, an ad-hoc IBM Workload Scheduler job type is available: the VariableTable job type. The VariableTable job must be added to the job stream as a successor of the job that is exporting variables. The VariableTable job sets the exported variables in a variable table and makes them available to any other successor job, in the same job stream or in a different job stream.

You can easily define a VariableTable job by using the Dynamic Workload Console or composer command line.

For more information, see the section about passing variables by using variable tables in *User's Guide and Reference*.

Graphical view enhancements

A new graphical view has been released using modern UX design principles.

A new graphical view has been released using modern UX design principles.

The graphical view has been redesigned to enhance the user experience (UX). The new design helps the user to accomplish the tasks easily and efficiently. Simple shapes to easily identify objects have been used, new icons to improve the interaction and quickly identify actions have been created, new colors and background to better visualize the objects have been applied.

For more information, see the section about Graphical View in Dynamic Workload Console User's Guide.

IBM Workload Scheduler REST API

IBM Workload Scheduler REST API to easily integrate workload scheduling capabilities with external products and solutions.

IBM Workload Scheduler provides a set of fully functional APIs that are implemented based on Representational State Transfer (REST) services. The REST APIs help you easily integrate workload scheduling capabilities with external products and solutions. The same product functionality covered by the existing J2EE APIs is available with the REST APIs. The REST APIs are programming language independent and favor easier network configuration and firewall traversal. They allow you to exploit heterogeneous environments and provide new automation opportunities with direct impact on productivity.

For information about the REST APIs available and the possibility to try them out on Swagger Docs, see the topic about driving IBM Workload Scheduler using the REST APIs in the *IBM Workload Automation: Developer's Guide: Driving IBM Workload Automation*.

Satisfying Requests for Enhancements (RFEs)

IBM Workload Scheduler satisfies Requests for Enhancements (RFEs).

Requests for Enhancements (RFEs) give customers the opportunity to collaborate directly with the product development team and other users. The team prioritizes and develops new product features based on proposals made by customers.

IBM Workload Scheduler V9.4 delivers numerous RFEs. The following are a sample of a few of them:

- **RFE 56898:** Two significant changes have been implemented in the variable tables. The first addresses character lengths for variable names and variable values. The lengths have been extended to 64 and 1024 characters respectively. The second change addresses the possibility to use a variable in place of a static file name when defining file dependencies for dynamic agents. The variable is defined in the variable table of the workstation on which the file exists, and is resolved at later time rendering the file name dynamic and more flexible for use in your plans.
- **RFE 73944:** Job log enriched with information about the start time and end time of job on a dynamic agent that ends in FAIL.
- **RFE 109970:** Job log enriched with information about user id of the user experiencing the job failure.
- **RFE 162189:** Job log for dynamic agents now shows timestamp with the same format as for FTAs.
- **RFE 81886:** **changePassword** command now provides a new parameter **-skipdbuser** to skip the update of the db user password.
- **RFE 162160:** Dynamic Workload Console V9.3 can now be installed on AIX V7.2 .
- **RFE 133854:** IBM Workload Scheduler identifies the best time window for agent upgrades. This RFE is addressed through the new features of Agent upgrade with minimal scheduling disruption.
- **RFE 125642, RFE 79392:** IBM Workload Scheduler provides more flexibility when displaying jobs and job streams in the Workload Designer graphical view.
- **RFE 79391:** The Workload Designer graphical view has been enriched with the capability to search for jobs and job streams.
- **RFE 76136:** The timestamp in the auditing information stored in the database now shows seconds and hundreds of seconds.
- **RFE 79395:** Users can now input a note for jobs and job streams that are modified in the plan.
- **RFE 108197:** IBM Workload Scheduler provides change version control for scheduling object definitions.
- **RFE 126406:** Dynamic Workload Console shows a message for every important action against the current plan.
- **RFE 141427:** A recycle bin is now available in the Workload Designer.
- **RFE 156327:** The Workload Application Template now includes also the event rules for the workstations and the job streams contained in the template.
- **RFE 156524:** A capability has been added to control resources assigned to a job in progress when the resources are no longer required by the job.

To view a complete list of RFEs, new, planned, and delivered, see: RFE online community .

Scheduling with the Agent for z/OS 9.4 enhancements

See the enhancements of the Scheduling with Agent for z/OS 9.4

IBM Workload Scheduler Distributed Agent for z/OS provides the following enhancements:

“EDWA for Scheduling with the Agent for z/OS”

“New opportunities for modifying job definitions already in the plan or in the database”

EDWA for Scheduling with the Agent for z/OS

Event-driven workload automation (EDWA) for Scheduling with the Agent for z/OS

You can now address event-driven workload automation for your z/OS network.

With this new feature, you can carry out a predefined set of actions in response to events of any kind that occur in the environment. Specifically, the scheduler can detect an activity impacting a data set and trigger any kind of action when the event of closure is verified.

New opportunities for modifying job definitions already in the plan or in the database

Modify a job instance in the plan before it runs, modify an instance of a job in the plan that has already run and rerun the modified job or modify the job definition explicitly in the production plan getting the JCL directly from a remote data set.

You can now edit the JCL on zAgent if a job is in error and modify the job definition explicitly in the database getting the JCL directly from a remote data set.

With the extra opportunities for modifying job definitions you can now modify a job instance in the plan or in the database before it runs or modify an instance of a job in the plan that has already run and rerun it.

This feature adds the flexibility you need so that you can now make changes to the definition even after it has already been submitted into the plan, maintaining the original definition in the database. With this additional flexibility you can go and get the JCL and run it again changing the data set name containing the JCL, change the JCL defined explicitly in the production plan file and you can also change the job definition explicitly in the production plan file getting the JCL from the remote data set. This can be done from either the Job Stream Graphical View, the job monitoring view, or from the **conman** command line.

For details about how to modify the job definition in the plan see the section about Edit JCL in the Chapter 7, “Dynamic Workload Console,” on page 73 and in the section JCL Editing in the *IBM Workload Scheduler: User’s Guide and Reference*.

IBM Workload Scheduler version 9.3 Fix Pack 2 enhancements

Learn about the IBM Workload Scheduler version 9.3 Fix Pack 2 enhancements.

About this task

IBM Workload Scheduler version 9.3 Fix Pack 2 includes the following enhancements:

- “New method for updating DB2 and Oracle database schema.”
- “Hybrid Workload Automation” on page xli
- “Checking prerequisites before running a silent installation” on page xli
- “Automatically initializing IBM Workload Scheduler instances” on page xlii
- “Defining and managing IBM Workload Scheduler role-based security” on page xlii
- “Reply to a message for an IBM i job from Dynamic Workload Console” on page xlii
- “Automatic reply to waiting messages for an IBM i job” on page xliii
- “IBM Workload Scheduler Plug-in for Apache Oozie” on page xliii
- “Running a script when a job completes” on page xliii

New method for updating DB2 and Oracle database schema

A new mechanism to update or upgrade the database schema for product upgrades and fix pack updates that eliminates the need for DB2® clients and, in general, provides a more dynamic, reliable and flexible method.

When creating, upgrading, or updating the master domain manager and its backup, or the dynamic domain manager and its backup, the database tables must also be created, upgraded, or updated.

If you typically update your database tables manually before updating the IBM Workload Scheduler components in your environment, then you can use this new method which includes a script that quickly generates the required SQL statements in an SQL file and optionally applies them directly to the database in a single step.

The advantages of this new update method are:

- A single script that dynamically generates the SQL statements and applies them directly to the database. Optionally, you can configure the script to generate only and then apply them at a later time but before installing the product component.
- The script can be run and rerun as many times as needed without any cleanup required between runs. For example, if for some reason a database object is corrupt, you can rerun the script to regenerate the SQL file and apply it.
- For DB2 databases, this method eliminates the requirement of installing a DB2 client to update a database located on a remote DB2 server. The script works using the JDBC drivers and connecting directly to the database server minimizing the overhead in installing and maintaining a database client.
- Use this method as a stand-alone tool to check the consistency of the database schema. The database administrator can verify if the database schema has changed and repair any inconsistencies.

Installing, updating, or upgrading the product components using either the Installation Manager wizard or the silent installation methods automatically upgrades the database tables using this same method.

For more information, see the section about creating and updating the database tables in the *IBM Workload Scheduler: Planning and Installation*.

Hybrid Workload Automation

Manage Workload Automation on Cloud or on Premises, or pick a mixture of both. Hybrid Workload Automation allows you to use a common User Interface to manage both on Premises and on Cloud engines.

Manage Workload Automation on Cloud or on Premises, or pick a mixture of both. Hybrid Workload Automation allows you to use a common User Interface to manage both on Premises and on Cloud engines. You can monitor and execute actions in plan on the following objects: job, critical job, workstation, resource, prompt, job stream, file, and domain.

The same efficient workload managed in the on premises environment can be scheduled to run in the on cloud environment. You can transfer your workload from your on premises environment to your Workload Automation on Cloud through a simple import procedure. The workload is instantly reproduced and it is ready to be employed on your agents. Handling your workload in the on Cloud environment is a more cost effective way to utilize resources in the cloud only when you need them without relying on an on premises infrastructure.

Workload Automation on Cloud reduces IT infrastructure costs, and it is a solution for utilizing resources only when you have a business need to run your workload.

For more information, see the section about Hybrid scheduling environment in the *Overview*.

Checking prerequisites before running a silent installation

Ensure a successful installation without delays or complications by running a prerequisite check script before you install either the master domain manager or the Dynamic Workload Console before initiating a silent installation.

The silent installation method is an unattended installation that does not require user intervention, however, to ensure that the installation is not halted because of missing or insufficient system requirements, you can now run a script that checks for product system requirements before initiating the installation.

The prerequisite check script is available for master domain manager and Dynamic Workload Console installations and checks requirements such as:

- Supported operating system.
- Sufficient RAM.
- Sufficient swap file space.
- Disk space for the creation of the installation and temporary directories passed in input to the script.

For more information, see the section about the silent installation in the IBM Workload Scheduler version 9.3 Fix Pack 2 readme file and the Dynamic Workload Console version 9.3 Fix Pack 2 readme file.

For details about the product system requirements see the section on disk space requirements in the product readme files.

Automatically initializing IBM Workload Scheduler instances

For Linux distributions that use **systemd** as the default initialization system, a sample service file is provided to support the automatic initialization of IBM Workload Scheduler instances at startup.

For some Linux distributions that use **systemd** as the default initialization system, such as RedHat Enterprise Linux v7.0 and SUSE Linux Enterprise Server V12, a sample service file is provided that is already configured to support the automatic initialization of IBM Workload Scheduler instances at startup.

For more information, see the section about automatically initializing IBM Workload Scheduler instances in *IBM Workload Scheduler Administration*

Defining and managing IBM Workload Scheduler role-based security

You can define and update your security file according to role-based security model, by using the **Manage Workload Security** interface from Dynamic Workload Console or the **composer** command-line program.

With the role-based security model you can update your *security file* with the security objects that you define in the master domain manager database, in a quick and easy way. Enable the role-based security model by setting the **optman** `enRoleBasedSecurityFileCreation` global option to *yes*.

According to the role-based security model, the security objects that you can define are:

Security roles

Each role represents a certain level of authorization and includes the set of actions that users or groups can do.

Security domains

Each domain represents the set of scheduling objects that users or groups can manage.

Access control lists

Each access control list is defined assigning roles to users or groups, on a certain security domain.

When the security file is updated on the master domain manager, the security settings on the master domain manager are automatically synchronized with the backup master domain manager.

For more information, see the section about getting started with security in *Administration Guide*.

Reply to a message for an IBM i job from Dynamic Workload Console

You can reply to a message for an IBM i job from Dynamic Workload Console.

When an IBM i job is in SUSP (suspended) status, waiting for a reply to a message, you can reply to the message directly from the **Monitor Workload** of the Dynamic Workload Console.

For more information, see the section about scheduling on IBM i systems in *User's Guide and Reference*.

Automatic reply to waiting messages for an IBM i job

You can define standard rules to automate the reply to waiting messages for an IBM i job.

When defining an IBM i job, by using the Dynamic Workload Console or the **composer** command line, you can specify the list of messages for which you want to set up an automated reply. In the output job log you can see the messages queued by IBM i system and the replies sent automatically by the job.

For more information, see the section about job definition for IBM i jobs in *User's Guide and Reference*.

IBM Workload Scheduler Plug-in for Apache Oozie

With the new IBM Workload Scheduler plug-in for Apache Oozie, you can schedule, monitor and control Oozie workflows and Hadoop jobs such as Hive, MapReduce, Pig, and Sqoop.

As Hadoop adoption continues to expand, Hadoop jobs are growing in volume and complexity. To optimize their execution, different Hadoop jobs can be organized into a single logical unit of work, called a workflow. Apache Oozie is a powerful tool that creates and manages complex workflows of Hadoop jobs. However, it is fundamental to integrate Oozie workflows with the rest of the business process flow. With the new plug-in for Oozie, workflows and Hadoop jobs can be managed the same way as any other IBM Workload Scheduler job, extending the central management of the company workload to the Hadoop environment.

IBM Workload Scheduler Plug-in for Apache Oozie.

Running a script when a job completes

When a job completes, you can run a script by using the information related to the job completion.

In many scenarios, when a job completes, you might want to run one or more actions, by using the information related to the job completion. For this purpose, you can write a script file and store it in a directory of the agent file system. The script is run every time that a job completes, either successfully or unsuccessfully. The script runs with the same credentials as the agent user that is running the job.

For more information, see the section about running a script when a job completes in *User's Guide and Reference*.

New and improved web interfaces

The Workload Designer and Dynamic Workload Console interfaces have been improved

The Workload Designer has been completely redesigned to improve your user experience. The new interface features a Welcome page which directs you to the main tasks to get you started with just one click. A similar function is available in the Welcome page for the Dynamic Workload Console, which provides a single entry point for accessing all your IBM Workload Scheduler web interfaces.

For more information, see the section about user interfaces in the *User's Guide and Reference* and the section about designing your workload in *Dynamic Workload Console User's Guide*.

Creating a URL to launch the Plan View in context

You can create a URL to launch the Plan View in context

You can create a URL to launch the Dynamic Workload Console and have it directly open the Plan View. You can then include this URL in an external application, for example, to quickly and easily access the Plan View.

For more information, see the section about creating a URL to launch the Plan View in context in the *IBM Workload Scheduler: Administration Guide*.

RFE 84003 - Modifying styles when exporting the Graphical View

You can define a custom style sheet for the Graphical View

You can define a custom style sheet to modify the styles used for the Graphical View, for example the background color, the color of links, and the color of objects. The changes are applied when exporting the Graphical View.

=
= **Note:** With the implementation of the newly redesigned graphical views in V9.4 FP1, this feature is not longer available.

RFE 75635 - Graphical View enhanced for wide monitors

You can display the Graphical View on wide monitors

You can display the Graphical View on wide monitors, so that also complex job streams are easily visible.

Auto-complete for workstation names and user ID

Auto-complete is now available for workstation names and user ID

An auto-complete function is now available for workstation names and user ID in job definitions in Workload Designer.

Managing Workload Security with Dynamic Workload Console

You can manage security settings in your environment quickly and easily by using Dynamic Workload Console.

From the navigation toolbar -> Administration -> Workload Environment Design, select Manage Workload Security. Here you can create and manage security objects in the database.

According to the role-based security model, the security objects that you can define are:

Security roles

Each role represents a certain level of authorization and includes the set of actions that users or groups can do.

Security domains

Each domain represents the set of scheduling objects that users or groups can manage.

Access control lists

Each access control list is defined assigning roles to users or groups, on a certain security domain.

Tooltips and online help do assist you while accomplishing security tasks.

When the security file is updated on the master domain manager, the security settings on the master domain manager are automatically synchronized with the backup master domain manager.

For more information, see the section about managing workload security in *Dynamic Workload Console User's Guide*

Restarting Informatica PowerCenter jobs from the point of failure

You can restart Informatica PowerCenter jobs from the point of failure.

When monitoring an Informatica PowerCenter job from Dynamic Workload Console, you can display detailed information about the Informatica PowerCenter workflow and run actions either on the workflow, or on any first-level tasks that are in the workflow, if these are sessions or worklets.

If the Informatica PowerCenter workflow fails, you can restart it from the failed task.

For more information, see the section about monitoring Informatica PowerCenter jobs in *Scheduling Applications with IBM Workload Automation*.

Single Entry Point

You can access all the user interfaces by using the Single Entry Point

Single Entry Point is a web-based page to access all the IBM Workload Scheduler user interfaces.

Single Entry Point is a role-based interface that you can access from any computer in your environment by using a web browser through the secure HTTPS.

For more information, see the section about Single Entry Point in the *IBM Workload Scheduler: User's Guide and Reference*.

Tivoli Common Reporting reports extended to MS SQL databases

Tivoli Common Reporting is an optional integration service of Jazz for Service Management extension for IBM WebSphere, which is installed with WebSphere Application Server.

Tivoli Common Reporting provides a web user interface functioning as a portal for IBM Cognos and allows you to administer, run, customize, and create IBM Workload Scheduler reports.

Previously the reports were available only for IBM® DB2 databases and now the support is extended to include also Microsoft SQL Server databases in a distributed environment. The reports include:

- Job Run History
- Job Run Statistics Chart
- Job Run Statistics Table

For more information, see the topic about Tivoli Common Reporting reports in the *Dynamic Workload Console User's Guide*.

IBM Workload Scheduler version 9.3 Fix Pack 1 enhancements

Learn about the IBM Workload Scheduler version 9.3 Fix Pack 1 enhancements.

About this task

IBM Workload Scheduler version 9.3 Fix Pack 1 includes the following enhancements:

- “Conditional dependencies”
- “IBM Workload Scheduler Plug-in for Liberty”
- “IBM Workload Scheduler Plug-in for MQTT” on page xlvi
- “Scheduling the centralized agent update” on page xlvii
- “WebSphere Application Server tools enhancements” on page xlvii
- “IBM Tivoli Monitoring integration enhancement: situation refactoring and new view associations” on page xlviii
- “IBM i enhancements” on page xlviii

Conditional dependencies

Using conditional dependencies, you can evaluate a particular job status or output and, based on the specified conditions, decide which jobs to run within the job stream. You might have branches of your workflow that run only when specific conditions are met.

Whenever you have conditions that determine whether or not a segment of your workflow should run, then these are conditional dependencies.

Conditional dependencies are powerful, flexible, and simple to use, and help to ensure that your workflows proceed smoothly.

For more detailed information about conditional dependencies, see the section about applying conditional branching logic in the *User's Guide and Reference*.

IBM Workload Scheduler Plug-in for Liberty

With the new IBM Workload Scheduler plug-in for JSR 352 Java Batch (plug-in for Liberty), you can schedule, run, and monitor your Java Batch applications and integrate them into more complex, composite batch workflows.

Batch processing has been a key part of information technology for many years. This type of processing might take hours or even days to complete and uses large amounts of memory or processing power while it runs. Java Batch has the same problems. For this reason, the ability to schedule and control Java Batch execution with an enterprise scheduler is fundamental. Using IBM Workload Scheduler

plug-in for JSR 352 Java Batch, you can connect to your WebSphere Liberty server and automate and control your Java Batch applications.

For more detailed information about IBM Workload Scheduler Plug-in for Liberty, see IBM Tivoli Workload Scheduler Plug-in for Liberty.

IBM Workload Scheduler Plug-in for MQTT

IBM Workload Scheduler integrates with MQTT to provide centralized management of your "Internet of Things" devices. MQTT is a publish and subscribe, simple, and lightweight messaging protocol.

MQTT allows local intelligent devices to publish data and receive commands from an application. Using the IBM Workload Scheduler plug-in for MQTT, you can connect to an MQTT message broker and:

- Subscribe to devices, publish messages, and send commands, with calendar and event-based scheduling capabilities, through an easy-to-use web interface.
- Take advantage of a gateway agent to ensure security compliance of the hybrid cloud environment.
- Integrate with analytics tools to extract real-time information and predictive insights from device-related data.

For more detailed information about IBM Workload Scheduler Plug-in for MQTT, see IBM Tivoli Workload Scheduler Plug-in for MQTT.

WebSphere Application Server tools enhancements

Credentials required to submit WebSphere Application Server tools (wastools) scripts supplied with the product are now optional parameters.

WebSphere Application Server administration scripts, such as starting and stopping the WebSphere Application Server, supplied with IBM Workload Scheduler use the `soap.client.props` file located in the properties directory of the WebSphere Application Server profile to avoid the necessity of supplying a user name and clear text password in the command line.

In addition to added security protection, the enhancement eliminates the maintenance costs associated with frequent password changes. The user ID and password are now optional and the credentials are retrieved from the `soap.client.props` file. However, unlike the master domain manager installation where the `soap.client.props` file is automatically customized with these credentials, the Dynamic Workload Console requires you to manually customize the `soap.client.props` file with the credentials to be able to use this enhancement.

For more information, see the section about application server starting and stopping in *Administration Guide*.

Scheduling the centralized agent update

You can schedule the centralized update of multiple agent instances by creating a centralized agent update job, either by using the Dynamic Workload Console or the composer command line.

When a job runs, it forwards to the master domain manager the Update agent request for all the fault-tolerant agent or dynamic agent instances that you selected, and then completes.

IBM Tivoli Monitoring integration enhancement: situation refactoring and new view associations

With this fix pack, an easy assisted creation of the default situations in a new Tivoli Workload Scheduler view, or in a view of your choice, in the Tivoli Enterprise Portal console. The default situations are associated to a view in the console with their respective severity. The default situations can be associated to a physical or logical view of your choice on the Tivoli Enterprise Portal.

For more detailed information about creating and viewing default situations, see the section about creating default situations in *IBM Workload Scheduler: Integrating with Other Products*.

Import Cron and Windows Task Scheduler jobs

You can use the Dynamic Workload Console to read Crontab files or Windows Task Scheduler jobs and convert them into IBM Workload Scheduler objects.

The jobs are added to your workload with the same specifications as the original jobs. The job streams reflect the time definition of the Cron or Windows Task Scheduler commands. You can then modify, schedule, and monitor them together with all your other objects. You can control jobs from one central place and schedule them according to complex schedules.

For detailed information about Import Cron and Windows Task Scheduler jobs by using the Dynamic Workload Console, see Import Import Cron and Windows Task Scheduler jobs.

Implementing What-if Analysis changes to the plan

The IBM Workload Scheduler What-if Analysis shows activities displayed against time and gives you a visual representation of your plan at a glance in real time.

About this task

Run the What-if Analysis before implementing the actual plan to show your predicted results at a glance. By viewing the results beforehand, you can plan for any potential problems before you start. Any changes that you make in What-if Analysis are applied only for simulation purposes until you use the option "Apply changes" to the actual plan. The "Apply changes" option is available only in a distributed environment. In a z/OS environment, changes you make are for simulation purposes only.

For more information, see the section about analyzing the impact of changes on your environment in the *Dynamic Workload Console User's Guide*.

IBM i enhancements

You can now use the Dynamic Workload Console and conman showjobs command line to check if an IBM i job is waiting for a reply to a message.

An IBM i job that is waiting for a message reply is in the SUSP (suspended) status. This status indicates that the job is running while waiting for input. When the input is received, the job status changes to EXEC (executing).

You can also view the IBM i job message that is waiting for a reply. The message text and the reply are written in the corresponding IBM Workload Scheduler job log, so that the IBM Workload Scheduler operator can see the exact message that the IBM i job is waiting for a reply to.

You can specify the name of the queue where the monitoring agent component runs by using the `MonitorQueueName` property in the native job launcher section of the `JobManager.ini` file. If you do not specify this property, the default queue (QBATCH) is used.

For more information, see the section about scheduling jobs on IBM i systems in *User's Guide and Reference*.

Other Dynamic Workload Console version 9.3 Fix Pack 1 enhancements

Other Dynamic Workload Console version 9.3 Fix Pack 1 enhancements

The following enhancements are also now available for Dynamic Workload Console version 9.3 Fix Pack 1:

- Users can set the size of columns in a permanent way: each time a resizing is done, the Dynamic Workload Console remembers the new column size, and applies the resizing wherever the same information is displayed. This means that you only need to resize the columns once.
- With the new graphical view enhancements, job dependencies are drawn nearer to the job they depend on. This greatly improves the entire layout, and complex job stream graphical views are now much more usable. Moreover, the zoom capacity has been increased, and the dimension of the boxes representing the jobs has been extended to contain even the longer jobs names; so improving the printing capability. If you use a third party tool for SVG's you can now customize the entire view, because all the information needed to identify objects is available.

IBM Workload Scheduler version 9.3 enhancements

Learn about the IBM Workload Scheduler version 9.3 enhancements.

About this task

IBM Workload Scheduler version 9.3 includes the following enhancements:

- "IBM Workload Scheduler Plug-in for IBM WebSphere MQ" on page l
- "IBM Workload Scheduler Plug-in for RESTful Web services" on page l
- "IBM Workload Scheduler Plug-in for SAP BusinessObjects" on page l
- "IBM Workload Scheduler support for Netezza Performance Server" on page li
- "IBM Workload Scheduler Plug-in for IBM Sterling Connect:Direct " on page li
- "IBM Workload Scheduler Plug-in for Salesforce" on page li
- "IBM Workload Scheduler Plug-in for Hadoop" on page lii
- "IBM Workload Scheduler Plug-in for BigInsights for Hadoop" on page lii
- "IBM Workload Scheduler plug-in for Oracle E-Business Suite" on page lii
- "File Transfer plug-in enhancements" on page liii
- "Enhancing your job stream definitions with the EVERY option" on page liii
- "Using IBM Workload Scheduler to extend the job capabilities on SAP" on page liii

- “Intercept business impact of Workload Automation failures” on page lv
- “File dependencies support for dynamic agents” on page lv
- “Centralized agent update” on page lv
- “IBM Tivoli Workload Scheduler for Applications now installed with base product” on page lvi
- “Installation of agents by using domain users” on page lvi
- “Other enhancements” on page lvi

IBM Workload Scheduler Plug-in for IBM WebSphere MQ

IBM Workload Scheduler Plug-in for IBM WebSphere MQ

You use IBM WebSphere MQ to enable your application programs to communicate with each other by using messages and queues. If, instead of individual applications you want to run job or application flows, you must link flows with applications that send and receive messages from message queues. IBM Workload Scheduler and its plug-in for IBM WebSphere MQ allows you to manage mixed processes, where multiple automated background workflows synchronize their data in real-time using message queues.

For a detailed description about creating a IBM WebSphere MQ job, see *IBM Workload Scheduler: User’s Guide and Reference*.

For more detailed information about IBM Workload Scheduler plug-in for IBM WebSphere MQ, see *IBM Workload Scheduler Plug-in for WebSphere MQ*.

IBM Workload Scheduler Plug-in for RESTful Web services

IBM Workload Scheduler Plug-in for RESTful Web services

The REST design model has gained widespread acceptance throughout the Web as a simpler, resource-based interface language to develop Web services and has lately been adopted by the most popular service providers. This IBM Workload Scheduler plug-in allows you to schedule jobs that add, download, delete, and modify resources or data on RESTful Web services via HTTP methods in any available content type such as JSON, XML, and XHTML.

For a detailed description about creating a RESTful Web services job, see *User’s Guide and Reference*.

For more detailed information about IBM Workload Scheduler Plug-in for RESTful Web services, see *IBM Workload Scheduler plug-in for RESTful Web services*.

IBM Workload Scheduler Plug-in for SAP BusinessObjects

IBM Workload Scheduler Plug-in for SAP BusinessObjects

IBM Workload Scheduler integrates with SAP BusinessObjects to automate and monitor the creation of SAP BusinessObjects Business Intelligence reports. You gain greater control of your SAP BusinessObjects report creation with both calendar-based and event-based workload automation. The integration allows you to manage and monitor SAP BusinessObjects and any other dependent operations from a central location and automatically detect problems and apply recovery actions to ensure that future strategies and decision-making are based on relevant and reliable data.

For a detailed description about creating a SAP BusinessObjects job, see *IBM Workload Scheduler: User's Guide and Reference*.

For more detailed information about IBM Workload Scheduler Plug-in for SAP BusinessObjects, see IBM Workload Scheduler plug-in for SAP BusinessObjects.

IBM Workload Scheduler support for Netezza Performance Server

IBM Workload Scheduler support for Netezza Performance Server

Most data warehouse environment data is extracted, transformed, and loaded at the end of the business day. However, with disparate data sources and high volumes of data, you need to be sure that your data includes the latest updates, so that users can create meaningful reports that serve as a reference for decision making. IBM Workload Scheduler now supports Netezza Performance Server and can condition the loading of the amounts of data upon successful execution of preliminary manipulation. It detects issues and resolves problems before they can affect users. The support for Netezza has been integrated in the Database job type.

For more information, see the section about the Database plug-in in *User's Guide and Reference*.

For more detailed information about IBM Workload Scheduler integration with Netezza Performance Server, see IBM Tivoli Workload Scheduler support for Netezza Performance Server.

IBM Workload Scheduler Plug-in for IBM Sterling Connect:Direct

About this task

IBM Workload Scheduler Plug-in for IBM Sterling Connect:Direct automates the entire file transfer process to ensure the success of any subsequent processing such as decryption, renaming, parsing, and retransmission of files. It also allows you to do real-time monitoring, reporting, event management, and auditing.

For a detailed description about creating a IBM Sterling Connect:Direct job, see *IBM Workload Scheduler: User's Guide and Reference*.

For more detailed information about IBM Workload Scheduler Plug-in for IBM Sterling Connect:Direct, see IBM Sterling file transfer.

IBM Workload Scheduler Plug-in for Salesforce

IBM Workload Scheduler Plug-in for Salesforce

IBM Workload Scheduler integrates with Salesforce to automate, monitor, and control workflows containing batch APEX jobs. You gain greater control of your Salesforce jobs with both calendar-based and event-based workload automation as well as having a single point of control to handle exceptions and automate.

For a detailed description about creating a Salesforce job, see *IBM Workload Scheduler: User's Guide and Reference*.

For more detailed information about IBM Workload Scheduler Plug-in for Salesforce, see IBM Tivoli Workload Scheduler Plug-in for Salesforce.

IBM Workload Scheduler Plug-in for Hadoop

IBM Workload Scheduler Plug-in for Hadoop

IBM Workload Scheduler provides three new plug-ins to simplify and automate your Apache Hadoop batch processing:

- A Plug-in for Hadoop Mapreduce that allows you to define, schedule, monitor, and control the execution of Hadoop MapReduce procedures.
- A Plug-in for Hadoop Distributed File System that allows you to access the Hadoop Distributed File System from any machine, and work on files and directories.
- A Plug-in for Apache Hive Database that provides you with an SQL-like interface to Hadoop.

IBM Workload Scheduler maximizes the value of big data by integrating Hadoop processes with the rest of the enterprise workflow.

For more information about creating an Hadoop Mapreduce job, see *IBM Workload Scheduler: User's Guide and Reference*.

See also the video tutorial IBM Workload Scheduler integration with Hadoop MapReduce.

For more information about creating an Hadoop Distributed File System job, see *IBM Workload Scheduler: User's Guide and Reference*.

IBM Workload Scheduler Plug-in for BigInsights for Hadoop

IBM Workload Scheduler Plug-in for BigInsights for Hadoop

IBM Workload Scheduler integrates with IBM BigInsights for Hadoop to bring the power of Apache Hadoop to the enterprise. Thanks to IBM Workload Scheduler plug-in for BigInsights for Hadoop you can:

- Monitor and control workflows containing BigInsights workbooks and applications that help enterprise find insights into new and emerging types of data.
- Fully automate BigInsights process execution with calendar and event-based scheduling, and a single point of control to handle exceptions, and automate recovery processes.

For a detailed description about creating a BigInsights for Hadoop job, see *IBM Workload Scheduler: User's Guide and Reference*.

See also the video tutorial IBM Workload Scheduler integration with BigInsights for Hadoop.

IBM Workload Scheduler plug-in for Oracle E-Business Suite

IBM Workload Scheduler plug-in for Oracle E-Business Suite

The IBM Workload Scheduler plug-in for Oracle E-Business Suite enhances your Oracle E-Business Suite scheduling capabilities and helps you to automate your entire business process chain.

For details about managing Oracle E-Business Suite jobs, see the *Scheduling Applications with IBM Workload Automation*.

For more detailed information about IBM Workload Scheduler plug-in for Oracle E-Business Suite, see IBM Workload Scheduler plug-in for Oracle E-Business Suite.

File Transfer plug-in enhancements

File Transfer plug-in enhancements

The IBM Workload Scheduler plug-in for File Transfer has been greatly enhanced with several new features.

- With wildcards support ('*' and '?') for the file name, you can easily transfer more than one file by using the same job. You can view the name, size, and transfer time for each transferred file, both in the job log and in the job properties.
- Certificate authentication is now supported for the SSH protocol: you can specify the keystore file path without having to use the user name and password to authenticate on the remote server.
- When downloading files, you can now set file permissions for the user on the local system.
- You can request that a local source file is automatically deleted after it has been transferred.

For a detailed description about creating a File Transfer job, see *IBM Workload Scheduler: User's Guide and Reference*.

Enhancing your job stream definitions with the EVERY option

About this task

IBM Workload Scheduler job stream definitions can now use the following features:

- The Repeat every option specifies a repetition rate, and the correlated end time, when instances of the same job stream are created and run in the plan. You can therefore specify that an instance of the same job stream be run every so many minutes or hours over a period of time. The option, available both in composer and in the Dynamic Workload Console, is defined in the run cycle (or in the run cycle group) used to schedule the job stream.
- The new Actions tab located in the Scheduling Options page of a job stream definition in the Dynamic Workload Console specifies if the scheduler must start anyway, enqueue, or discard a job stream instance that is about to start when another instance is still running.

In composer, this feature is implemented by the ONOVERLAP job stream definition keyword.

For more information about enhancing your job stream definitions with the EVERY option, see Every option.

Using IBM Workload Scheduler to extend the job capabilities on SAP

Using IBM Workload Scheduler to extend the job capabilities on SAP

To improve the visibility, control, and automation of your SAP environment, IBM Workload Scheduler can help you through the entire lifecycle of your workload, from design to execution and optimization. With a long-standing and certified SAP partnership, IBM Workload Automation can provide a single point of control to

manage and monitor SAP and non-SAP applications including governance of SAP environment practices, data integration with SAP Solution Manager, and other tools. R3batch process has been updated with the latest SAP NetWeaver RFC library.

- “IBM Workload Scheduler Plug-in for SAP BusinessObjects” on page l
- “Empowering SAP Solution Manager with IBM Workload Scheduler”
- “Add RFC that reports version of SAP transport installed”
- “Handling of SAP application logs” on page lv

IBM Workload Scheduler Plug-in for SAP BusinessObjects

IBM Workload Scheduler Plug-in for SAP BusinessObjects

IBM Workload Scheduler integrates with SAP BusinessObjects to automate and monitor the creation of SAP BusinessObjects Business Intelligence reports. You gain greater control of your SAP BusinessObjects report creation with both calendar-based and event-based workload automation. The integration allows you to manage and monitor SAP BusinessObjects and any other dependent operations from a central location and automatically detect problems and apply recovery actions to ensure that future strategies and decision-making are based on relevant and reliable data.

For a detailed description about creating a SAP BusinessObjects job, see *IBM Workload Scheduler: User’s Guide and Reference*.

For more detailed information about IBM Workload Scheduler Plug-in for SAP BusinessObjects, see IBM Workload Scheduler plug-in for SAP BusinessObjects.

Empowering SAP Solution Manager with IBM Workload Scheduler

About this task

IBM Workload Scheduler and SAP Solution Manager are integrated so that IBM Workload Scheduler is the engine that actually runs the job scheduling tasks available from the Solution Manager user interface. This integration is provided by the SMSE Adapter that runs in the master domain manager. The SMSE Adapter uses the SAP Solution Manager Scheduling Enabler (SMSE) interface provided by SAP to enable external schedulers to run the scheduling for Solution Manager. With this integration, when you schedule a job from the Scheduling panel of Solution Manager, IBM Workload Scheduler takes charge of the job scheduling, monitoring, and management tasks, as well as of job triggering and notification. Under these conditions IBM Workload Scheduler acts as an RFC-Server with a common interface for scheduling jobs. It is identified through an RFC-Destination registered in the SMSE.

For detailed information about scheduling jobs on IBM Workload Scheduler from SAP Solution Manager, see *Scheduling Applications with IBM Workload Automation*.

See also the video tutorial Empowering SAP Solution Manager with IBM Workload Scheduler.

Add RFC that reports version of SAP transport installed

Add RFC that reports version of SAP transport installed

A Remote Function Call (RFC) has been added to facilitate the identification of the level of SAP transports loaded for IBM Workload Scheduler applications. You can

call the RFC using a job that retrieves a string that corresponds to the release transported into SAP. This is particularly helpful when you have multiple SAP landscapes undergoing IBM Workload Scheduler upgrades in identifying the version of the IBM Workload Scheduler transports without relying on manual methods of recording the information.

Handling of SAP application logs

Handling of SAP application logs

SAP application logs can now be handled independently of the SAP application return codes so that even if the SAP application return code is not set, the SAP application log can be retrieved. R3batch can therefore handle the SAP application return code fields of the "E_T_RETUN_CODES" table independently of the information available in the SAP application log fields listed in the "E_T_LOGHANDLES" table.

Intercept business impact of Workload Automation failures

Intercept business impact of Workload Automation failures

Through real-time monitoring and analysis of business events, you can find out what the impact on your service models or lines of business would be in both a distributed and z/OS environment. You can import IBM Workload Scheduler objects like workstations, job streams, and jobs to IBM Tivoli Business Service Manager and send events related to these objects for quick analysis by IBM Tivoli Business Service Manager and priority-based response according to the business impact.

For a detailed description about integrating with IBM Tivoli Business Service Manager, see *Integrating with Other Products*.

For more detailed information about intercept business impact of Workload Automation failures, see Intercept business impact of Workload Automation failures.

File dependencies support for dynamic agents

About this task

You can now use and manage file dependencies with dynamic agents, pools, and dynamic pools.

For more information about defining file dependencies and their use in a dynamic environment, see the topic about defining file dependencies in dynamic scheduling in the *User's Guide and Reference*.

Centralized agent update

About this task

You can install fix packs or upgrade multiple fault-tolerant agent and dynamic agent instances, by downloading a package to install only on the master domain manager workstation and manage the installation or upgrade by running a single action from the Dynamic Workload Console.

For more information about updating agent instances, see the topic about the centralized agent update in *Planning and Installation*.

IBM Tivoli Workload Scheduler for Applications now installed with base product

IBM Workload Automation version 9.3 includes all of the functions of IBM Tivoli Workload Scheduler for Applications.

With this release, IBM Tivoli Workload Scheduler for Applications is no longer a separate product. The access methods and the plug-ins previously provided with IBM Tivoli Workload Scheduler for Applications are now automatically installed when you install IBM Workload Scheduler Version 9.3. However, to be entitled to use these plug-ins, you must purchase a separate chargeable component.

Note: The Oracle E-Business Suite access method previously available with IBM Tivoli Workload Scheduler for Applications has been removed from the product offering. IBM Workload Scheduler Version 9.3 does not support the extended agent for Oracle E-Business. With IBM Workload Scheduler Version 9.3, a new job plug-in has been introduced, the IBM Workload Scheduler plug-in for Oracle E-Business Suite. A conversion tool is provided to quickly and easily move the existing extended agent definition to the new job plug-in definition. See the readme file describing this solution in the Integrated Service Management Library.

For detailed information about extending IBM Workload Scheduler capabilities, see the *IBM Workload Scheduler: User's Guide and Reference*.

Use the PeopleSoft, SAP R/3, and z/OS access methods, to extend the job scheduling capabilities of IBM Workload Scheduler to external systems and applications.

You can also extend the job scheduling capabilities of IBM Workload Scheduler to external applications through the use of IBM Workload Scheduler job plug-ins.

For more information about access methods and plug-ins, see *IBM Workload Automation: Scheduling Applications with IBM Workload Automation*.

Important: Do not install IBM Tivoli Workload Scheduler for Applications, Version 8.6 on top of an IBM Workload Scheduler version 9.3 installation which can cause problems when running jobs. Version 9.3 of the product includes more updated versions of the plug-ins with respect to IBM Tivoli Workload Scheduler for Applications, Version 8.6.

Installation of agents by using domain users

Installation of agents by using domain users

It is now possible to install agents in such a way that a local user is a member of the local administrative group, and a domain user is a member of the local administrative group.

Other enhancements

Other enhancements

The following enhancements are also now available for IBM Workload Scheduler version 9.3:

- Adding support for Microsoft SQL Server as IBM Workload Scheduler database.
- Informix RDBMS support.

- FTA agent type installation and upgrade process.
- Adding installation as a non-root user.
- Improve job results analysis.

New Welcome page for the Dynamic Workload Console

New Welcome page for the Dynamic Workload Console

For a quick and rapid overview of the Dynamic Workload Console, after logging in, the Welcome page for the Dynamic Workload Console is displayed in the Dashboard Application Services Hub console window.

For a detailed description about the Welcome page, see the topic about navigating the Dynamic Workload Console in the *Dynamic Workload Console User's Guide* .

IBM Workload Scheduler Analytics

About this task

You can use IBM Workload Scheduler analytics capabilities:

- IBM Workload Scheduler What-if Analysis
- Using advanced statistics for accurate forecasts of job durations

IBM Workload Scheduler What-if Analysis

With IBM Workload Scheduler What-if Analysis you can modify the start date, add and delete dependencies, set the unavailability of the workstation, submit a job stream in the plan, and highlight the critical path to see how these changes affect the plan and to explore all the various results by using a trial and error approach.

For more information, see the topic about analyzing the impact of changes in your environment in the *Dynamic Workload Console User's Guide*.

Using advanced statistics for accurate forecasts of job durations

SPSS-based statistical analysis can help you pinpoint trouble for the timely completion of your critical jobs. Five-day forecasts, based on 90-day histories, of the duration of your jobs are provided in the form of confidence intervals. The forecasts are precise to the very second. This tool can really help you go beyond the known impacts of already accepted periodicities, and understand what hidden effects are keeping your critical jobs from making their deadlines.

For more information, see the topic about using advanced statistics to forecast the estimated duration of a job in the *User's Guide and Reference*.

See also the video tutorial *Using advanced statistics to improve your scheduling*.

Monitor Workload

About this task

IBM Workload Scheduler Monitor Workload includes the following enhancements:

- Run query in archived plan.
- View as a report on Monitor Workload
- Critical jobs on Monitor Workload

Note: With IBM Workload Scheduler version 9.3 the name changed from Dynamic Monitoring Query to Monitor Workload.

For more information about monitoring your workload, see the topic about creating a monitoring task query in the *Dynamic Workload Console User's Guide*.

See also the video tutorial [Dynamic Monitoring Query](#).

Dashboards to monitor important information, at a glance

About this task

Efficient workload monitoring allows for quick response times. By reacting quickly you avoid putting your business at risk. Monitor and drill down on a predefined dashboard, or assemble the widgets provided to monitor only those parts of the workload you need, all in a single, consolidated view.

For more information, see the section about creating a dashboard for monitoring in *Dynamic Workload Console User's Guide*.

See also the video tutorial [Control your Workload Scheduling at-a-glance with the customizable dashboards available on IBM Service Engage](#).

Import Cron and Windows Task Scheduler jobs

About this task

Use the IBM Workload Automation Application Lab to read cron definition file and Windows task scheduler files and convert them into the Application Lab processes. You can move your cron definition files and Windows task scheduler tasks defined on your servers into the Workload Automation environment to take advantage of complex scheduling scenarios.

Control all your scheduling business tasks from a single point of control and reduce the risk and time involved in using multiple schedulers.

For more information, see the section about importing Cron and Windows Task Scheduler jobs in *Application Lab User's Guide*.

For more information about Cron definition, see also the [Import Cron definition video](#) available on IBM Service Engage.

Application Lab

Model, schedule, and monitor your business processes using an easy-to-use interface and deliver business value in a shorter time.

A new graphical user interface, the Application Lab, is dedicated to application developers who require control and autonomy when driving workloads, but are not required to handle complicated workload automation concepts. Your organization might have a team of IT experts that automates complicated workloads. However, there might be individuals, or groups of individuals, in your organization who could benefit from an easy-to-use tool to automate their personal business tasks. These individuals could be considered application developers. With the assistance of an administrator, they can be set up to use the Application Lab to develop their own process or sequence of steps, and then choose to either automate the process or just run it on-demand when needed. You do not need to

install any separate components to work with the Application Lab; it is accessible through a URL when you have the Dynamic Workload Console installed.

For full details about this new interface, see the *IBM Workload Automation: Application Lab User's Guide*.

Tivoli Common Reporting Reports

Tivoli Common Reporting Reports

Tivoli Common Reporting is an optional integration service of Jazz for Service Management extension for IBM WebSphere, which is installed with WebSphere Application Server. You can administer, run, customize, and create reports with Tivoli Common Reporting.

For more information, see the information about Tivoli Common Reporting reports in the *Dynamic Workload Console User's Guide*.

Fast and easy location of variables in a variable table

You can quickly and easily locate variables in a variable table

When you have several pages of variable entries in a variable table it can be quite difficult to locate a specific variable. Now, the Dynamic Workload Console Workload Designer has been updated to allow you to sort the columns on either the variable name and variable value so you can instantly find the variable that you need to update.

Job end time is tracked in the database and is viewable from the Dynamic Workload Console

Job end time is tracked in the database and is viewable from the Dynamic Workload Console

You can now view the job end time, in addition to the job start time and duration in the job run history. Previously, to get the job end time required adding the duration to the start time. The JOB_HISTORY_V table has been updated to include this end time when defining custom reports. You can also view the job end time when you request to view the Job Run History from the Dynamic Workload Console.

Defining a job as non-operational (NOP)

Defining a job as non-operational (NOP)

You can define a job as non-operational so that when the plan runs, the job is put in Cancel Pending status and is not run. The successors of this job are however run. You can define a job as non-operational by either checking the NOP box in the job definition window in the Dynamic Workload Console, or by adding the NOP option in the job statement in composer. You can make a NOPed job operational at plan run time from either conman or the Dynamic Workload Console. This sets the job to the Ready status for that particular job stream instance only (to make the job permanently runnable, you must remove the NOP flag from its definition).

For more information, see the section about NOP in the *User's Guide and Reference*.

Managing SAP events on dynamic workstations

Managing SAP events on dynamic workstations

Event-driven workload automation for SAP events is now extended to dynamic agents. You can therefore run rules that manage basic SAP background events, IDoc records, and CCMS Monitoring Architecture alerts on dynamic agents as well as on extended agents based on the r3batch method.

For more information, see the topic about creating SAP Standard R/3 jobs from the Dynamic Workload Console in the *Scheduling Applications with IBM Workload Automation*.

Other Dynamic Workload Console enhancements

Other Dynamic Workload Console enhancements

The following enhancements are also now available for Dynamic Workload Console version 9.3:

- Importing a workload application template.
- SAP e PeopleSoft options file configurator for dynamic agents.
- Personalizing UI labels on the Dynamic Workload Console.
- Monitoring all engines from Self-Service Dashboards.

Chapter 1. Overview of IBM Workload Automation

IBM Workload Automation is the state-of-the-art production workload manager, designed to help you meet your present and future data processing challenges. Its scope encompasses your entire enterprise information system, including heterogeneous environments.

Pressures on today's data processing (DP) environment are making it increasingly difficult to maintain the same level of service to customers. Many installations find that their batch window is shrinking. More critical jobs must be finished before the morning online work begins. Conversely, requirements for the integrated availability of online services during the traditional batch window put pressure on the resources available for processing the production workload. Increasing by 7 days a week, 24 hours a day is not only a DP objective but a requirement.

Users and owners of DP services are also making more use of batch services than ever before. The batch workload tends to increase each year at a rate slightly below the increase in the online workload. Combine this with the increase in data use by batch jobs, and the end result is a significant increase in the volume of work.

Furthermore, there is a shortage of people with the required skills to operate and manage increasingly complex DP environments. The complex interrelationships between production activities, between manual and machine tasks, have become unmanageable without a workload management tool.

IBM Workload Automation simplifies systems management across heterogeneous environments by integrating systems management functions. There are three main components to the portfolio:

IBM Workload Scheduler for z/OS

The scheduler in z/OS® environments.

IBM Workload Scheduler

The scheduler in distributed environments

Dynamic Workload Console

A Web-based, graphical user interface for both IBM Workload Scheduler for z/OS and IBM Workload Scheduler.

The state-of-the-art solution

The portfolio provides leading-edge solutions to problems in production workload management. It can automate, plan, and control the processing of your enterprise's entire production workload, not just the batch subset. The portfolio works as an "automatic driver" for your production workload to maximize the throughput of work, and optimize your resources, but also allows you to intervene manually as required.

When the portfolio interfaces with other system management products, it forms part of an integrated automation and systems management platform for your DP operation.

Comprehensive workload planning

The portfolio forms operating plans based on user descriptions of the operations department and its production workload. These plans provide the basis for your service level agreements and give you a picture of the production workload at any precise time.

Good planning is the cornerstone of any successful management technique. Effective planning also helps you maximize return on your investments in information technology.

Centralized systems management

The portfolio automates, monitors, and controls the flow of work through your enterprise's entire DP operation on both local and remote systems. From a single point of control, the portfolio analyzes the status of the production work and drives the processing of the workload according to installation business policies. It supports a multiple user environment, enabling distributed processing and control across sites and departments within your enterprise.

Systems management integration

Solutions to today's systems management problems require an integration of application programs and processes. The portfolio offers you integration with the following:

- Agents for controlling the workload on non-z/OS platforms
- Other systems management applications and architecture environments.

The portfolio interfaces directly with some of the z/OS products as well as with a number of other IBM products to provide a comprehensive, automated processing facility and an integrated approach for the control of complex production workloads.

NetView®. NetView is the IBM platform for network management and automation. You can use the interface for IBM Workload Scheduler for z/OS with NetView to pass information about the work that is being processed. The portfolio lets you communicate with NetView in conjunction with the production workload processing. IBM Workload Scheduler for z/OS can also pass information to NetView for alert handling in response to situations that occur while processing the production workload. NetView can automatically trigger IBM Workload Scheduler for z/OS to perform actions in response to these situations using a variety of methods. IBM Workload Scheduler/NetView is a NetView application that gives network managers the ability to monitor and diagnose IBM Workload Scheduler networks from a NetView management node. It includes a set of submaps and symbols to view IBM Workload Scheduler networks topographically and determine the status of job scheduling activity and critical IBM Workload Scheduler processes on each workstation.

Workload Manager (WLM). WLM controls the amount of system resources available to each work unit in host environments. IBM Workload Scheduler for z/OS works in concert with WLM to detect critical jobs and move them to a higher-performance service class. In addition with WLM, critical jobs receive more system resources and complete more quickly.

Resource Object Data Manager (RODM). RODM provides a central location for storing, retrieving, and managing the operational resource information needed for

network and systems management. You can map a special resource to a RODM object. This lets you schedule the production workload considering actual resource availability, dynamically updated.

Tivoli® Decision Support for z/OS (Decision Support). Decision Support helps you effectively manage the performance of your system by collecting performance data in a DATABASE 2 (DB2) database and presenting the data in a variety of formats for use in systems management. Decision Support uses data from IBM Workload Scheduler for z/OS to produce summary and management reports about the production workload, both planned and actual results.

Output Manager for z/OS. Helps customers increase productivity and reduce the costs of printing by providing a means for storing and handling reports in a z/OS environment. When a dialog user requests to view a job log or to automatically rebuild the JCL for a step-level restart, IBM Workload Scheduler for z/OS interfaces with Output Manager. This interface removes the requirement to duplicate job log information, saving both CPU cycles and direct access storage device (DASD) space.

Tivoli Information Management for z/OS. Supports the administration of the systems management process of an enterprise's hardware, software, and related resources. An interface with Tivoli Information Management for z/OS is provided for reporting problems detected while processing the production workload.

Resource Access Control Facility (RACF®). RACF is the IBM product for data security. You can use RACF as the primary tool to protect your IBM Workload Scheduler for z/OS services and data at the level required by your enterprise. With RACF 2.1 and later, you can use a IBM Workload Scheduler for z/OS reserved resource class to protect your resources.

IBM Tivoli Monitoring (ITM). You can use it to monitor your hardware, operating systems, applications, databases. It provides proactive monitoring and automated fault management, and includes a specific module for Business Integration. You can also manage configuration and collect monitoring information for reporting, performance analysis, trend predictions and enterprise wide business impact assessment.

IBM SmartCloud Control Desk (SCCD). It is an incident management system. SCCD can function as a service desk for both IT and non-IT assets, whether internally or externally hosted. SCCD helps to improve IT performance by providing automation of processes, better visibility of service support functions, commitments, and measurements.

Tivoli System Automation for z/OS (SA z/OS). SA z/OS initiates automation procedures that perform operator functions to manage z/OS components, data sets, and subsystems. SA z/OS includes an automation feature for IBM Workload Scheduler for z/OS. You can define an automation workstation in IBM Workload Scheduler for z/OS to handle system automation operations with a specific set of options.

Data Facility Hierarchical Storage Manager (DFHSM). IBM Workload Scheduler for z/OS catalog management functions invoke DFHSM to recall migrated data sets during data set cleanup for a failed or rerun job.

CICS® and IMS™ (Computer Information Control System and Information Management System). IBM Workload Scheduler for z/OS lets you schedule the

starting and stopping of started tasks. Because IBM Workload Scheduler for z/OS tracks the status of started tasks, you can serialize work, such as backups of your transaction databases, according to the status of your CICS or IMS subsystems.

Tivoli Business Systems Manager. Tivoli Business Systems Manager provides monitoring and event management of resources, applications, and subsystems with the objective of providing continuous availability for the enterprise. Using Tivoli Business Systems Manager with the portfolio provides the ability to manage strategic applications from a unique business systems perspective. Tivoli Business Systems Manager monitors batch-related applications and operations represented by the portfolio and seamlessly integrates these objects with all other business objects monitored by Tivoli Business Systems Manager.

Tivoli Enterprise Console®. The Tivoli Enterprise Console is a powerful, rules-based event management application that integrates network, systems, database, and application management. It offers a centralized, global view of your computing enterprise while ensuring the high availability of your application and computing resources. Tivoli Enterprise Console acts as a central collection point for alarms and events from a variety of sources, including those from Tivoli applications. IBM Workload Scheduler runs a Tivoli Enterprise Console adapter that reads events from the IBM Workload Scheduler log file.

Besides these IBM products, there are many products from other software vendors that work with or process data from the portfolio.

For white papers about using IBM products, refer to the following link:
<http://www-03.ibm.com/support/techdocs/atsmasr.nsf/Web/WP-ByProduct?OpenDocument&Start=1&Count=1000&Expand=18>.

An integration scenario

About this task

This section shows how you can monitor late *critical jobs* and perform *incident management*, by integrating IBM Workload Scheduler for z/OS with the following products:

- IBM Tivoli Monitoring (ITM)
- IBM SmartCloud® Control Desk (SCCD)
- Tivoli System Automation (SA)

IBM Workload Scheduler for z/OS schedules jobs according to the defined current plan. ITM is configured with a *situation* that sends an email notification to SCCD, when a critical job is late.

The integrate components work as follows:

1. ITM detects a critical job tied to a WLM scheduling environment.
2. ITM sends an email causing the automatic opening of a service request.
3. The SCCD operator captures the service request and runs a launch in context of the Dynamic Workload Console, to perform incident analysis.
4. The analysis confirms that a critical job is waiting for a WLM scheduling environment.
5. A System Automation job is submitted through Dynamic Workload Console, to make available the WLM scheduling environment.
6. As soon as the WLM scheduling environment is available, the scheduler submits again the critical job.

7. The service request is closed.

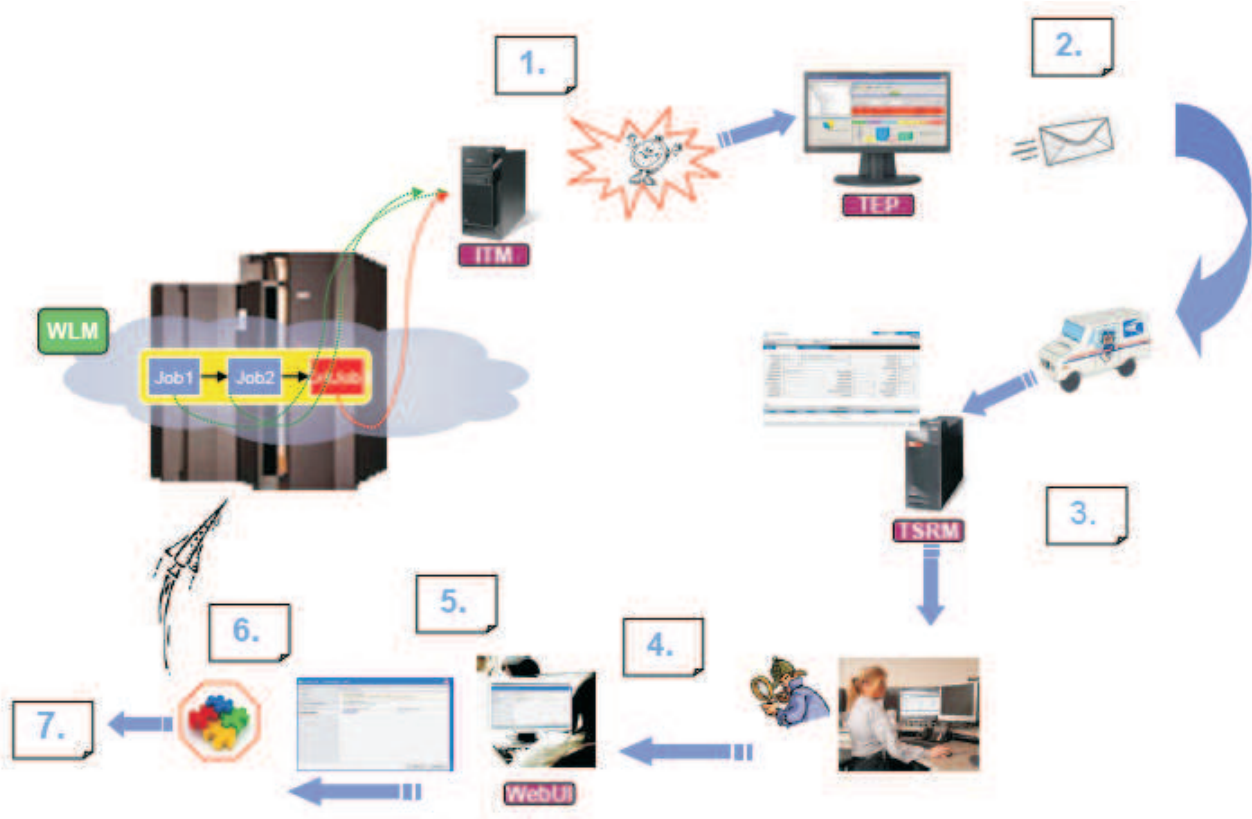


Figure 1. Integration scenario for IBM Workload Scheduler for z/OS.

Automation

About this task

By automating management of your production workload with the portfolio, you can minimize human errors in production workload processing and free your staff for more productive work. The portfolio helps you plan, drive, and control the processing of your production workload. These are important steps toward automation and unattended operations. Whether you are running one or more systems at a single site, or at several distributed sites, the portfolio helps you automate your production workload by:

- Coordinating all shifts and production work across installations of all sizes, from a single point of control
- Automating complex and repetitive operator tasks
- Dynamically modifying your production workload schedule in response to changes in the production environment (such as urgent jobs, changed priorities, or hardware failures) and then managing the workload accordingly
- Resolving workload dependencies
- Managing utilization of shared resources
- Tracking each unit of work
- Detecting unsuccessful processing
- Displaying status information and instructions to guide operations personnel in their work

- Interfacing with other key IBM products to provide an integrated automation platform

The portfolio lets you centralize and integrate control of your production workload and reduces the number of tasks that your staff need to perform.

Workload monitoring

About this task

Besides providing a single point of control for the production workload across your systems, the portfolio:

- Monitors the production workload in real time, providing operations staff with the latest information on the status of the workload so that they can react quickly when problems occur.
- Provides security interfaces that ensure the protection of your services and data.
- Enables manual intervention in the processing of work.
- Reports the current status of your production workload processing.
- Provides reports that can serve as the basis for documenting your service level agreements with users. Your customers can see when and how their work is to be processed.

Automatic workload recovery

The portfolio enables processing production workload to continue even when system or connection failures occur. If one system fails, the portfolio can restart the processing on another system. When the controlling system is running in a z/OS system complex (sysplex), a hot standby function can automatically transfer control of the production workload to another system in the sysplex. Because the portfolio continues to manage the production workload during failures, you can maintain the integrity of your processing schedules and continue to service your customers.

In IBM Workload Scheduler, a switchmgr function provides the possibility to replace a failing master domain manager or domain manager workstation with an appropriately configured backup fault-tolerant agent or domain manager .

Productivity

The portfolio represents real productivity gains by ensuring fast and accurate performance through automation. Many of today's automation solutions quote unrealistic productivity benefits. Some of the tasks automated should never be performed, or certainly not as often as they are by automation. Because of this, it is difficult to correlate real productivity benefits to your enterprise.

The tasks the portfolio performs not only have to be performed, but have to be performed correctly, every time, and as quickly as possible. Many of these tasks, traditionally performed by DP professionals, are tedious and as a result prone to error. With the portfolio, your DP staff can use their time more efficiently.

Business solutions

About this task

The portfolio provides business solutions by:

- Driving production according to your business objectives
- Automating the production workload to enhance company productivity

- Providing you with information about current and future workloads
 - Managing a high number of activities efficiently.
-

User productivity

About this task

Your DP staff and users can realize significant productivity gains through the portfolio's:

- Fast-path implementation.
 - Immediate response to dialog requests for workload status inquiries. Users are provided with detailed real-time information about production workload processing so that they can detect and promptly correct errors.
 - Automation of operator tasks such as error recovery and data set cleanup.
-

Growth incentive

About this task

As you implement automation and control you can manage greater production workload volumes. The portfolio brings growth within your DP operation by providing:

- Ways of absorbing the increasing batch workload without increasing operations personnel
 - An open interface for submitting and tracking the workload on a variety of operating systems
 - Interfaces with other systems management application programs
 - An open interface for, and communicating with, programs on other platforms
 - Management of current and future production workload volumes
 - Simulation facilities to forecast future workloads
-

How IBM Workload Automation benefits your staff

About this task

In a typical enterprise, many people contribute to the implementation and operation of IBM Workload Automation:

- Scheduling manager
- Operations manager
- Shift supervisor
- Application programmer
- Console operators
- Workstation operators, such as print operators, job setup staff, and login receptionists
- End users
- Service desk

This section describes how the portfolio can directly benefit your DP staff.

Role of the scheduling manager as the focal point

About this task

IBM Workload Automation makes it possible for the scheduling manager to maintain current and future production processing across your enterprise. The portfolio benefits the scheduling manager in the following ways:

- Automatically scheduling all production workload activities.
- Automatically resolving the complexity of production workload dependencies and driving the work in the most efficient way.
- Supporting the simulation of future workloads on the system. The scheduling manager can evaluate, in advance, the effect of changes in production workload volumes or processing resources.
- Giving a real-time view of the status of work as it flows through the system so that the scheduling manager can quickly:
 - Respond to customer queries about the status of their work
 - Identify problems in the workload processing.
- Providing facilities for manual intervention.
- Managing many workload problems automatically. The production-workload-restart facilities, hot standby, automatic recovery of jobs and started tasks, and data set cleanup provide the scheduling manager with comprehensive error-management and disaster-management facilities.
- Providing a log of changes to the production workload data through the audit-trail facility. This assists the scheduling manager in resolving problems caused by user errors.
- Managing hard-to-plan work.

Role of the operations manager

About this task

The reporting, planning, and control functions can help the operations manager do the following:

- Improve the efficiency of the operation
- Improve control of service levels and quality
- Set service level agreements for user applications and for services provided
- Improve relationships with user departments
- Increase the return on your IT investment
- Develop staff potential.

A powerful tool for the shift supervisor

About this task

The portfolio is important for the shift supervisor, especially in multisystem complexes, where local and remote systems are controlled from a central site. The portfolio can help the shift supervisor do the following:

- Monitor and control the production workload through multisystem complexes
- Control the use of mountable devices
- Separate information about work status from system and other information
- Provide users with status information directly
- Manage the workload if a system failure occurs
- Make changes to the current plan in response to unplanned events, such as equipment failures, personnel absences, and rush jobs.

Role of the application programmer

About this task

The user-authority checking functionality enables application development groups to use all the planning and control functions in parallel with, but in isolation from, production systems and services.

The portfolio can be a valuable tool for application development staff when they are doing the following:

- Packaging new applications for the production environment
- Testing new JCL in final packaged form
- Testing new applications and modifying existing ones

Console operators

About this task

The portfolio can free console operators from the following time-consuming tasks:

- Starting and stopping started tasks
- Preparing JCL before job submission
- Submitting jobs
- Verifying the sequence of work
- Reporting job status
- Performing data set cleanup in recovery and rerun situations
- Responding to workload failure
- Preparing the JCL for step-level restarts.

Workstation operators

About this task

The portfolio helps workstation operators do their work by providing the following:

- Complete and timely status information
- Up-to-date ready lists that prioritize the work flow
- Online assistance in operator instructions.

End users and the service desk

About this task

Your users often need to be informed about the status of workload processing. They can use the Dynamic Workload Console to check the status of the processing of their job streams themselves from a personal workstation. Users can make queries using the Dynamic Workload Console without having to be familiar with the portfolio, ISPF, or TSO, and without having to be logged on to a local system.

In addition, users can always be informed about the status of the processing of their job streams from their mobile device. The Self-Service Catalog and Self-Service Dashboards mobile applications connect users their environments where they can both submit job streams to run and monitor the ongoing progress.

For application developers looking to automate their business processes, the Application Lab web-based user interface can be used to create, run, and monitor a set of simple business processes without requiring any knowledge of complex workload automation concepts. It is a simplified user interface to deliver business value faster.

The help desk can use the Dynamic Workload Console in the same way to answer queries from users about the progress of their workload processing.

Summary

About this task

IBM Workload Automation communicates with other key IBM products to provide a comprehensive, automated processing facility and an integrated solution for the control of all production workloads. Here are the benefits that the portfolio offers you:

Increased automation

Increases efficiency and uses DP resources more effectively, resulting in improved service levels for your customers.

Improved systems management integration

Provides a unified solution to your systems management problems.

More effective control of DP operations

Lets you implement change and manage growth more efficiently.

Increased availability

Is made possible by automatic workload recovery.

Opportunities for growth

Are made possible by your ability to manage greater workload volumes.

Investment protection

Is made easier by building on your current investment in z/OS and allowing existing customers to build on their existing investment in workload management.

Improved customer satisfaction

Is achieved thanks to higher levels of service and availability, fewer errors, and faster response to problems.

Greater productivity

Results because repetitive, error-prone tasks are automated and operations personnel can use their time more efficiently.

Integration of multiple operating environments

Provides a single controlling point for the cooperating systems that comprise your DP operation.

The portfolio is more than just a batch scheduling tool: it is a production management system with the capability to schedule *all* the work running on any system.

Chapter 2. IBM Workload Automation and ITUP

This chapter explains where IBM Workload Automation is placed within the IBM Tivoli Unified Process (ITUP).

The IBM Tivoli Unified Process provides detailed documentation of IT Service Management processes based on industry best practices, to help users to significantly improve their organization's efficiency and effectiveness. ITUP helps users to easily understand processes, the relationships between processes, and the roles and tools involved in an efficient process implementation.

The processes described in ITUP are strongly aligned with the Information Technology Infrastructure Library (ITIL) which is based on best practices observed within the IT industry. ITIL provides high-level guidance of what to implement, but not how to implement. ITUP contains detailed process diagrams and descriptions to help users understand processes and their relationships, making ITIL recommendations easier to follow.

ITUP is based on the IBM Process Reference Model for IT™ (PRM-IT), which was jointly developed by IBM Global Services and Tivoli. PRM-IT provides detailed process guidance for all activities that fall under the office of the CIO, including, but not limited to, IT Service Management.

The ITUP processes

About this task

ITUP describes a comprehensive set of processes within an IT organization. Each process is defined by:

Tool Mentors that describe best practice use of IBM tools in a process context

Tool mentors help users identify which IBM products and solutions can be used to perform specific process activities and details their appropriate use. By following this guidance users can reduce time, effort, and errors, and get the maximum value out of their investments.

Role definitions, responsibilities, and resources

ITUP describes the roles and responsibilities of all actors in the process model. Users can identify their roles and understand the activities they need to perform and the tools available to help them.

Work products and other information

ITUP describes all work products, often referred to as artifacts, produced as output or required as input by processes and activities. Other information such as key terms and concepts are also defined.

Scenarios describing common problems and best practice solutions

Scenarios help users understand how real world problems can be addressed with process improvement and integration, proper tool use, and defined roles and responsibilities.

Service execution and workload management

About this task

Among the processes supported by IBM Workload Automation, service execution is one of the key IT processes described by ITUP. Its mission is to deliver operational services to the IT infrastructure and to the enterprise.

The main service execution activities supported by IBM Workload Automation are:

- Deliver service
- Manage delivery resources
- Manage workload

Workload management has the target to maximize the utilization of task execution resources and to minimize the total time that is required to deliver the output of task processing. This activity operates at both a macro-level and micro-level to prepare work schedules and to pre-process work items where necessary so that the delivery resources can be matched to the demands of the flow of work in an optimal fashion.

The objectives of workload scheduling focus on:

- Managing the execution of activities according to business calendars, time constraints, and resource availability.
- Managing activities that have interactions between each other and have dependencies with each other and external entities and events.
- Enabling integration with business application environments like SAP R3 and PeopleSoft, and managing the running of activities in those environments.
- Managing the lifecycle for defining the activities to run and their running.
- Enabling monitoring and control on the running of these activities and collecting results and historical running data.

Managing workload with IBM Workload Automation

About this task

IBM Workload Automation is a portfolio of products provided by IBM to automate all workload management tasks. The scheduling features of IBM Workload Automation help you plan every phase of your workload production. During the processing period, the production control programs manage the production environment and automate most operator activities. The schedulers prepare jobs for running, resolve interdependencies, and launch and track jobs. Because jobs start running as soon as their dependencies are satisfied, idle time is minimized, and throughput improves significantly. Jobs never run out of sequence, and, if a job fails, the schedulers can handle the recovery process with little or no operator intervention.

Workload management is based on a database that contains the definitions of the scheduling objects. There are two versions of the scheduling objects database depending on the placement of the main workload controller: it can be based on a mainframe computer (in this case, z/OS) or on a distributed platform. Some of the scheduling objects can exist in both of the databases, some apply only to the distributed platform, and others might apply only to the mainframe platform.

The minimum set of object definitions that are required to produce a workload consists of a workstation, a job, and a job stream. Other required scheduling objects might be predefined and exist by default.

A workstation is a definition that represents a computer system or another entity that is capable of running specific tasks, and that has the ability to report the status of task execution to the scheduler. With the IBM Workload Automation interfaces, you can identify the physical resources associated with the workstations.

A job is the representation of a task (an executable file, program, or command) that is scheduled and launched by the scheduler. The job is run by a workstation and, after running, has a status that indicates if the run was successful or not. A job definition can specify information on what to do whenever its run was not successful. Jobs not included in a job stream do not have any attribute for running, and are only the description of a task with a definition on how to perform it in a form that is known to the specified workstation.

A job stream represents a container for related jobs and organizes them, in terms of run time, sequencing, concurrency limitations, repetitions, assigning priority or resources, and so on. Job streams are the macro elements of the workload that you manage.

The scheduling plan is the to-do list that tells IBM Workload Scheduler or IBM Workload Scheduler for z/OS what jobs to run, and what dependencies must be satisfied before each job is launched. IBM Workload Scheduler or IBM Workload Scheduler for z/OS builds the plan using the elements that are stored in the scheduling database.

The running of a plan requires tracking to identify possible problems that can impact the effective delivery of the work products. It is possible to perform the tracking from a Web-based Java™ interface, the Dynamic Workload Console, on either of the platforms (z/OS and distributed). As an alternate interface to the Dynamic Workload Console on the z/OS platform you can also use the ISPF panel interface, and on the distributed platforms you can use the command-line interface.

See Chapter 4, “A business scenario,” on page 17 which describes a possible implementation of workload management based on IBM Workload Automation.

Chapter 3. Who performs workload management

About this task

The primary roles most directly responsible for workload management are:

The IT administrator

Is the general IT administrator of all the hardware and software used by the company. He is in charge of installing, supporting, and maintaining servers or other computer systems, and planning for and responding to service outages and other problems.

He installs and maintains the job scheduling tool.

The IBM Workload Scheduler IT administrator

A good deal of his time is focused on keeping job scheduling running smoothly. He rarely does any actual scheduling himself, but instead acts as the person in the background who supports those who do. The IBM Workload Scheduler IT administrator:

- Defines and maintains the security for the job scheduling tool.
- On certain occasions does a minimal amount of tuning and customization of the job scheduling tool.
- Guarantees that the job scheduling tool environments are up and running all of the time, and if something goes wrong he needs to quickly resolve the problem.
- Monitors the health status of the job scheduling tool infrastructure. Uses tools that alert him (usually via email or pager) and create alerts or automatically open a trouble-ticket to alert the responsible person when there is a problem.
- Occasionally spends his time helping to fix job scheduling problems that the job schedulers cannot understand.
- Generates and uses reports.
- Occasionally documents major problems and work-arounds on the community website.
- Interacts mainly with fellow team members, programmers, and job schedulers.
- Sometimes makes suggestions to management about capacity planning and IT software purchases.

The job scheduler

Is the primary actor in workload management and needs to easily create and maintain a plan containing the company workload. He is responsible for modeling the company workload, and for designing, fixing, and maintaining schedules. His main responsibilities are to:

- Manage workload complexity and dependencies.
- Optimize schedule efficiency, flexibility, resiliency.
- Analyze and fix modeling issues.
- Look proactively for the schedule's integrity.

The scheduling operator

Is responsible for performing all operational processes and procedures, ensuring the business continuity of the workflow. His main responsibilities are to:

- Monitor critical events and perform first analysis of problems.
- Manage and coordinate the resolution of issues.
- Ensure that operations continue.

He is usually not dedicated to monitoring job scheduling alone.

The Scheduling and Operations manager

He:

- Does not use job scheduling tools himself; but is interested in the operational data from the tools, such as reports on long and late running jobs and service level agreement status.
- Makes sure his team has the knowledge and tools they need to schedule and manage jobs efficiently.
- Is always looking for ways to reduce cost in his organization by making his team more efficient.
- Believes that process is the key to IT management and also thinks that his team's job scheduling process can be improved. He is familiar with ITUP of which his IT organization has implemented the basic aspects of change management. Consequently, his team follows this process.

Chapter 4. A business scenario

The purpose of the following scenario is to show how the choice of the correct workload scheduling product, together with process improvement and integration, and well-defined roles and responsibilities, can improve the business of a manufacturing enterprise.

The company

Fine Cola is a medium-sized enterprise that produces and distributes soft drinks to retailers across the country. It owns a production plant and several strategically located distribution centers. The primary customers of Fine Cola are foodstore chains and the quantity and size of their orders is usually regular and stable. Order quantities, however, peak in the warmer season and during holidays. Moreover, in the mid term, Fine Cola wants to increase its business by gaining market in other countries. Fine Cola's sales people are always keen to place new orders and increase the customer portfolio. These characteristics determine Fine Cola's production and distribution processes. Production and distribution can be broken down into ongoing subprocesses or phases which are constantly interlocked with each other. They are:

Inventory

Underlays the entire production process. The raw materials database is sized on the production levels supplemented by minimum safety levels. The production levels are in turn based on the order quantity for the specific period.

Ordering

Raw material quantity levels must be available to production according to the preset production levels. Orders must be planned and issued in advance to take into account delivery times by third-party suppliers.

Production

General production levels are planned for well in advance based on customer orders. Production is regularly increased by an additional five percent to provide the capability to honor unplanned-for orders.

Supply

From the production plant the soft drinks are transported to the distribution centers according to the customer delivery schedules.

Delivery

The last phase of the process. Fine Cola sodas are delivered from the distribution centers to the customer shelves.

Inventory, ordering, and production take place in the production plant. Supply takes place from the production plant to the distribution centers. Delivery takes place from the distribution centers to the end destinations.

These phases are tightly bound to each other. While each soda placed on the shelf might be regarded as the outcome of a specific sequence that starts with inventory and terminates with delivery, all phases are actually constantly interwoven. In fact, the same data is shared in one way or another by all or most phases, and applications are designed to carry on the daily operations and set up future ones.

Fine Cola uses the following databases for running the above-mentioned subprocesses:

Customer Orders

Contains all orders for the upcoming period from Fine Cola's customer base. Provides input to:

- Inventory

Raw Materials

Contains the quantities in stock of the raw materials required to produce Fine Cola's sodas. From here, orders are dispatched to suppliers when stock levels reach a pre-set minimum. Receives input from:

- Production Volumes

Production Volumes

Contains the quantities of sodas that are to be produced daily according to order volumes. Provides input to:

- Inventory
- Raw Materials

Receives input from:

- Inventory

Inventory

Contains the quantities in stock of the finished product. Is monitored to verify that the quantities in stock are sufficient to honor the orders of a specific time interval. Provides input to and receives input from:

- Production Volumes
- To Supply

To Supply

Contains the quantities of sodas that must be sent periodically from the manufacturing plant to the distribution centers to satisfy foodstore orders for the upcoming period. Provides input to:

- Inventory
- To Deliver

To Deliver

Contains the quantities that are to be delivered from each distribution center to the foodstores in its area. Provides input to:

- Customer Orders

Receives input from:

- To Supply

The company workload is both application oriented, such as accounting, payroll, supplier and utility payments, purchasing, ordering, fulfillment, and system-oriented, such as data backup, migration, export, transfer or load operations. Typically, the workload processes multiple data items such as accounts, orders, transactions, database records, at the same time.

These core applications are highly relevant for the profitability of the company and also directly influence customer satisfaction.

To create added value and exceed customers expectations, the company must strengthen integration with business applications and provide complete scheduling capabilities and tighter integration with enterprise applications.

The challenge

Currently the databases are not automatically integrated with each other and need continual human intervention to be updated. This affects Fine Cola's operations because:

- The process as a whole is onerous and prone to error.
- The interfaces between phases are slow and not very efficient.

The company realizes it needs to better integrate with the distribution centers because processing is extremely low during the regular office hours in the warmer seasons and during holidays. Users experience applications freezing, often taking considerable time before being available for them to use again. This lack of integration is causing problems for the organization in terms of lost productivity, while applications come back online. This is a problem because the interruption of important processing is not acceptable when the company wants to expand the business. The response time for service level agreements (SLAs) must continue to be met if a resource goes down, a workstation breaks, or there is urgency for maintenance, and even more during peak periods even if the resources are geographically distributed. On the other hand the company does not want to buy new IT resources (hardware, software, applications) because this would not be used during the other periods of the year.

Fine Cola realizes that their main weakness lies in their processing. They need to implement a solution that:

- Integrates the data behind their processing workflow from inventory to distribution. This makes it possible to automatically trigger the daily operations without much need for human intervention. It also gives Fine Cola complete control over the entire business process, reducing human intervention only to exception handling.
- Integrates external data coming from third parties, such as selected customers and raw material suppliers, into their process flow. Such data is provided to Fine Cola in several formats and from different applications and should be integrated into Fine Cola's databases in a seamless manner.
- Enables daily backups of their data as well as subsequent reorganization of the DB2 database with as little impact as possible on their processes. Processing of data collected online during the previous day is the next step.
- Optimizes capacity across the IT infrastructure and runs a high workload, much more than before, using shared resources, even if the resources are geographically distributed.
- Ensures 24x7x365 availability of critical business services. Disaster recovery plans are no longer sufficient because the business requires recovery within a couple of hours, not days. Recovering from last night tapes and recapturing lost transactions after a system or application failure is no longer a viable option for the company in a highly competitive market.
- Has very low probability of failure leading to maximum system reliability.

The main company goal at this time is to obtain an integrated workload solution that can entirely choreograph its business application tasks. This means solutions that optimize capacity across the IT infrastructure and run a tremendous workload, much more than before, using less resources. For example, if the company has a problem and a primary server does not process the workload, the company wants to automate the quick redistribution of system resources to process workloads and scale up or down for flawless execution. In this way the company reduces costs

because it speeds recovery time, no matter what the source. The goal is to have a view of the best available resources across this dynamically shifting cross-enterprise pool.

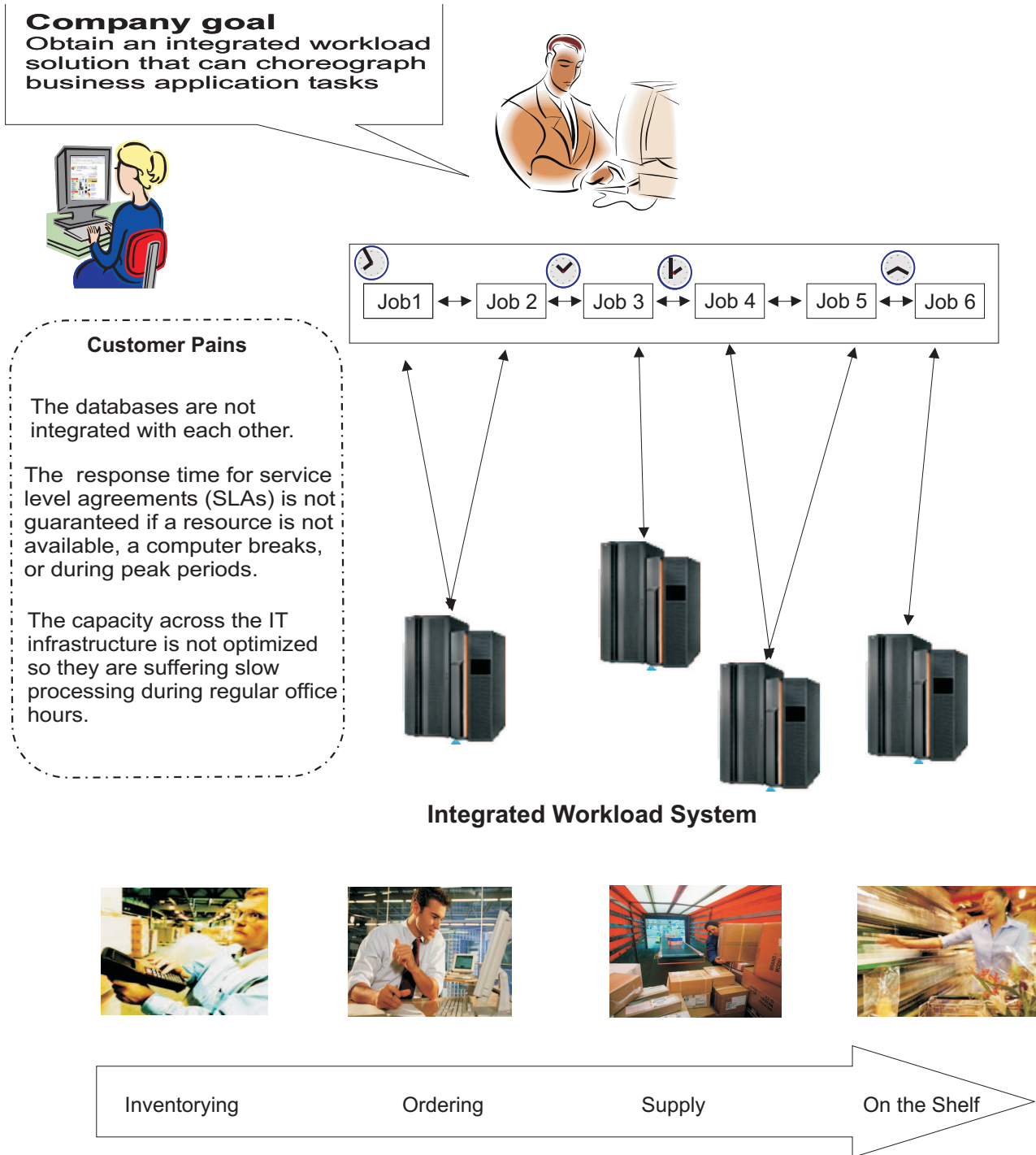


Figure 2. The Fine Cola company integrated workload solution

The solution

About this task

Fine Cola decides that one important step toward improving their process execution is to adopt a solution based on automatic and dynamic workload scheduling. The solution is based on a choice that strengthens integration with business applications to run the following tasks:

- Read data from one database to update other databases.
- Read data from external applications, process it, and add it to the appropriate databases.
- Provide the information necessary for the operation of every phase.
- Trigger some of the phases when predetermined thresholds are reached.
- Back up their data without interrupting production.
- From a capacity management perspective, understands the size of an application and what resources it requires, models that against the existing resources and is able to predict and forecast the capacity that the new application needs as it is defined in the enterprise.
- From an availability management perspective, use the resources available in the environment to support the application and understand out how to work to effectively schedule, monitor, and manage that application as it is submitted. Then if the resources are not available, interact with the change management and provisioning processes to dynamically allocate the necessary resources.
- Have a business management process monitoring all the various policies and driving a consistent view of the policies for the application.

After analyzing the workload management products available on the market, Fine Cola has chosen to use IBM Workload Scheduler and specifically the dynamic domain manager to:

- Optimize and automate the tasks to process their applications and dynamically adapt their processing in response to changes in the environment.
- Plan, choreograph, and schedule required changes to applications to minimize the impact of changes on critical production workloads, and ensure that workload processes are updated to reflect changes throughout asset life cycles.
- Minimize the total amount of time that is required to deliver the output of the task resolution processes.
- Handle dependencies between tasks, data, and external applications so that the entire workload can be managed homogeneously in the same process flow.
- Create a policy-based view of workflow automation, not just workload automation, but cross-enterprise workflow, and direct that workflow across the enterprise while planning, scheduling, managing, and monitoring all these things. Dynamically tuning the cross-enterprise capacity to support this dynamic view of workloads.
- Automatically transfer entire workloads across multiple platforms, and update policies across multiple platforms.
- Balance between the ability to provide sophisticated planning, testing, choreographing, monitoring, and adaptation of workload processes with fault tolerance and redundancy for high availability of the scheduling infrastructure, while minimizing server and network resource requirements.
- Perfectly integrate with each other.

IBM Workload Scheduler operates at both a macro-level and micro-level to prepare work schedules and to preprocess work items where necessary so that the delivery resources can be matched to the demands of the flow of work in an optimal fashion.

The dynamic domain manager dynamically routes workload to the best available resources based on application requirements and business policies. Moreover it optimizes the IT computing resource use according to SLAs.

Fine Cola's applications are mapped to what in IBM Workload Scheduler terminology are units of work called jobs. Some of these jobs are statically allocated to dedicated resources to run (static job definition), others are dynamically allocated to physical or virtual resources according to the job importance, requirements, scheduling policies, and based on the environment resource characteristics, relationships, availability, load, and performance (dynamic job definition). They drive the resource allocation to meet the job SLA and the resource optimization.

Jobs that run as a unit (such as a weekly backup application), along with times, priorities, and other dependencies that determine the exact order of the jobs are grouped into job streams.

Fine Cola's job streams are collections of jobs that are grouped for organizational purposes. The jobs of any particular job stream are related because they:

- Operate toward the completion of related tasks. For example, the jobs of Jobstream100 run tasks designed to convert incoming customer orders into operational data.
- Might be dependent on each other. Some jobs might have to wait for the completion of predecessor jobs before they can start running. The jobs are usually laid out in a sequence where the outcome of a predecessor is fed to a successor job.
- Share the same programs, applications, and databases.
- Share the same time-frames within the planning period.

Using IBM Workload Scheduler, Fine Cola's business process is structured in the following way:

1. At the start of each day, Jobstream100:
 - a. Extracts the new incoming orders from the Customer Orders database.
 - b. Checks an external application where a number of selected customers can place unforeseen orders. If there are orders, they are extracted and merged with the other data.
 - c. Copies the consolidated orders into a separate database view.
 - d. Sorts them by due delivery date and by quantity and makes a report.
2. As soon as the report is available, Jobstream200 extracts the numbers from the report and compares them with relevant data in the Inventory database. The goal is to determine the production volume required in the next production cycle to satisfy the orders.
3. Jobstream300 extracts the production volume data and updates the Production Volumes database with the quantities of each type of soda that is to be manufactured in the next cycle.
4. Jobstream400 reads the data in the Production Volumes database and:
 - a. Calculates the quantities of raw materials required to run the upcoming production cycle.

- b. Flags these quantities as allocated to next cycle in the Raw Materials database.
 - c. Checks the quantities to see if they have reached the minimum stock levels and triggers orders to Fine Cola's raw material suppliers if necessary.
5. Jobstream500 reads the report with upcoming due orders from the Customer Orders database and:
 - a. Produces the transportation schedules and destinations.
 - b. Updates the To Supply database.
 - c. Sends the delivery schedules to the distribution centers.
6. Jobstream600 reads the distribution center databases and:
 - a. Extracts the orders that have been filled.
 - b. Updates the Customer Orders database so that invoices can be prepared and sent.
7. Jobstream700 makes a backup of every database.

Fine Cola sets up a long term plan that encompasses the entire workload, spanning job streams that run on a daily basis and job streams that have other reoccurrences. From the long term plan, a current plan is extracted at the beginning of every time unit. The time period of the current plan can be chosen to vary from some hours to several days. Fine Cola has chosen to set their current plan on a daily basis. At the start of every day a new daily plan is built by their workload scheduling software: data is taken from the long term plan and from the daily plan of the previous day to include any jobs that might not have completed.

The company must also ensure that during peak periods the jobs in the critical path are run in the required time frame. To ensure this they converted some jobs from static definition to dynamic definition to manage the extra orders using the dynamic domain manager. With the dynamic domain manager, the company can:

- Manage the automatic discovery of available resources in the scheduling environment with their characteristics and relationships.
- Assign to the job the appropriate resources for running based on the job requirements and on the administration policies.
- Optimize the use of the resources by assigning to the job the required resources based on the SLA.
- Manage and control the resource consumption and load.
- Dispatch jobs to target resources that meet the requirements to run the job.

The IBM Workload Scheduler relational database contains the information related to the jobs, the job streams, the workstations where they run, and the time specifications that rule their operation. It also contains data used by the dynamic domain manager, such as information about the current IT environment, the resource real time performance, and load data. It also stores the job definitions and keeps track of resources assigned to each job.

In this way, Fine Cola's scheduling analyst can create and change any of these objects at any time and Fine Cola's IT administrator can dynamically assign the best set of resources to match allocation requests based on the defined policies, without any impact on the business.

The IT administrator can also ensure the correct concurrent or exclusive use of the resources across the different jobs according to resource characteristics. If the resource request cannot be immediately carried out, he can use dynamic

scheduling to automatically queue the resource until changes in the resource utilization or in the environment lead to its fulfillment.

The workload scheduling plan can be changed as quickly and dynamically as the business and operational needs require. The scheduling analyst makes full use of the trial and forecast planning options available in the scheduler to adjust and optimize workload scheduling and, as a consequence, Fine Cola's line of operations.

To respond to any unexpected and unplanned-for demands, individual jobs can be added ad hoc to the scheduling plan at any time.

Moreover, the company can use dynamic scheduling to rapidly adapt to the increase of workload during peak periods driving the requirement for workload virtualization, that is the ability to manage and control the workload so that it can be split, routed to appropriate resources and capacity, and dynamically moved around in logical resource pools.

If a resource is not available, the SLA defined continues to be met because the job processing is restarted from the point where the failure happens.

Typical everyday scenarios

About this task

This section describes roles and responsibilities of Fine Cola's IT staff and everyday scenarios they might face on any typical day. Fine Cola's IT staff, involved in workload scheduling are:

- The scheduling analyst. He is in charge of modeling the company workload, and for designing, fixing, and maintaining schedules. His main responsibilities are:
 - Managing Fine Cola's workload complexity and dependencies.
 - Optimizing the schedule's efficiency, flexibility, and resilience.
 - Analyzing and fixing modeling issues; look pro-actively for the schedule's integrity.
- The operations analyst. His main responsibilities are:
 - Monitoring critical events and performing first analysis of problems.
 - Managing and coordinating the resolution of issues.
 - Ensuring that operations continue.
- The IT infrastructure administrator. His main responsibilities are:
 - Fulfilling the need to assign physical or virtual resources to jobs according to the job importance, requirements, scheduling policies, and based on the environment resources characteristics, relationships, availability, load, and performance.
 - Managing the advanced reservation or provisioning of required resources.
 - Drive the resource allocation to meet the job SLA and the resource optimization data without service disruption, and possibly transparently for the users.
 - Backing up the schedule daily with no impact on operations.
 - Ensuring high availability of the infrastructure. If a resource goes down or a workstation is not available, the SLA-defined availability must continue to be met.
 - Defining and maintaining the environment topology.

Managing the workload

About this task

Together with the IT infrastructure administrator and other staff, the scheduling analyst agrees on a change in the application workflow that should go into production in a month. The change impacts Jobstream100 and includes:

- Defining a new job and replacing some job dependencies in the job stream.
- Defining two Jobstream100 instances to run twice a day for a week and during the summer season. He must therefore:
 - Define a run cycle for each of the two instances. The first run cycle has the expected start time of 9 a.m. The second is scheduled to start at 5 p.m.
 - Agree with the IT infrastructure administrator the pool of resources that satisfy the job SLA in terms of RAM and microprocessors.

He then proceeds in the following way:

1. He reviews the new stream logic and sets a plan. He wants to design the changes, test them over three days, and have a first automatic test run within a week.
2. He proceeds to apply the changes to Jobstream100. While he does this, he realizes that the application specialist must modify the tasks (scripts) contained in some of the jobs.
3. For this reason he leaves the job stream in draft state while the work is still in progress, so that it is not included in the plan generated every day.
4. To apply the changes he operates directly using the job stream editor available with Fine Cola's scheduling product: defining a new job by renaming an existing job definition and adding a new dependency.
5. After he has finished drafting the changes, he saves the job stream with a validity date set to tomorrow and active status on the test system.
6. Before launching the plan containing the modified job stream, he generates a trial plan to verify that the dependencies are correctly resolved.
7. When the trial plan ends, he analyzes job statistics and finds that a different design of dependencies could improve total elapsed time.
8. He applies the changes, sets the new dependencies, and creates a plan extension. The job stream is rescheduled and run in a test environment correctly and timely.
9. He meets the IT infrastructure administrator to verify that all the resources involved in the running of the new plan are available on the following days.
10. The IT infrastructure administrator, after analyzing the plan of availability of the IT resources, warns him that one of the required resources will not be available the next week.
11. The IT infrastructure administrator asks the scheduling analyst to run a forecast plan.
12. He then runs a forecast plan, which contains the scheduled activities for next week, to verify that the unavailability of the resources will not cause any major problem.
13. He finds that the unavailability of the resources will cause a decrease in performance because the other resources become overloaded.
14. He notifies the IT infrastructure administrator of the potential problem.
15. The IT infrastructure administrator analyzes the availability of resources between departments and realizes that the resources belonging to another department meet the requirements to run the job definition.

16. The scheduling analyst moves the workload from static to dynamic resource allocation. He uses the dynamic domain manager to route workloads to the best available systems by matching load requirements and business policies to available resource capacities.
17. He identifies the jobs in the critical path and modifies their definitions so that they can be run dynamically.
18. He finally sets a date to run the new plan in the production environment and notifies the IT infrastructure administrator.

Figure 3 on page 27 shows how the Fine Cola company can dynamically manage its workload using the added dynamic scheduling capability of IBM Workload Scheduler and satisfying the SLA response time.

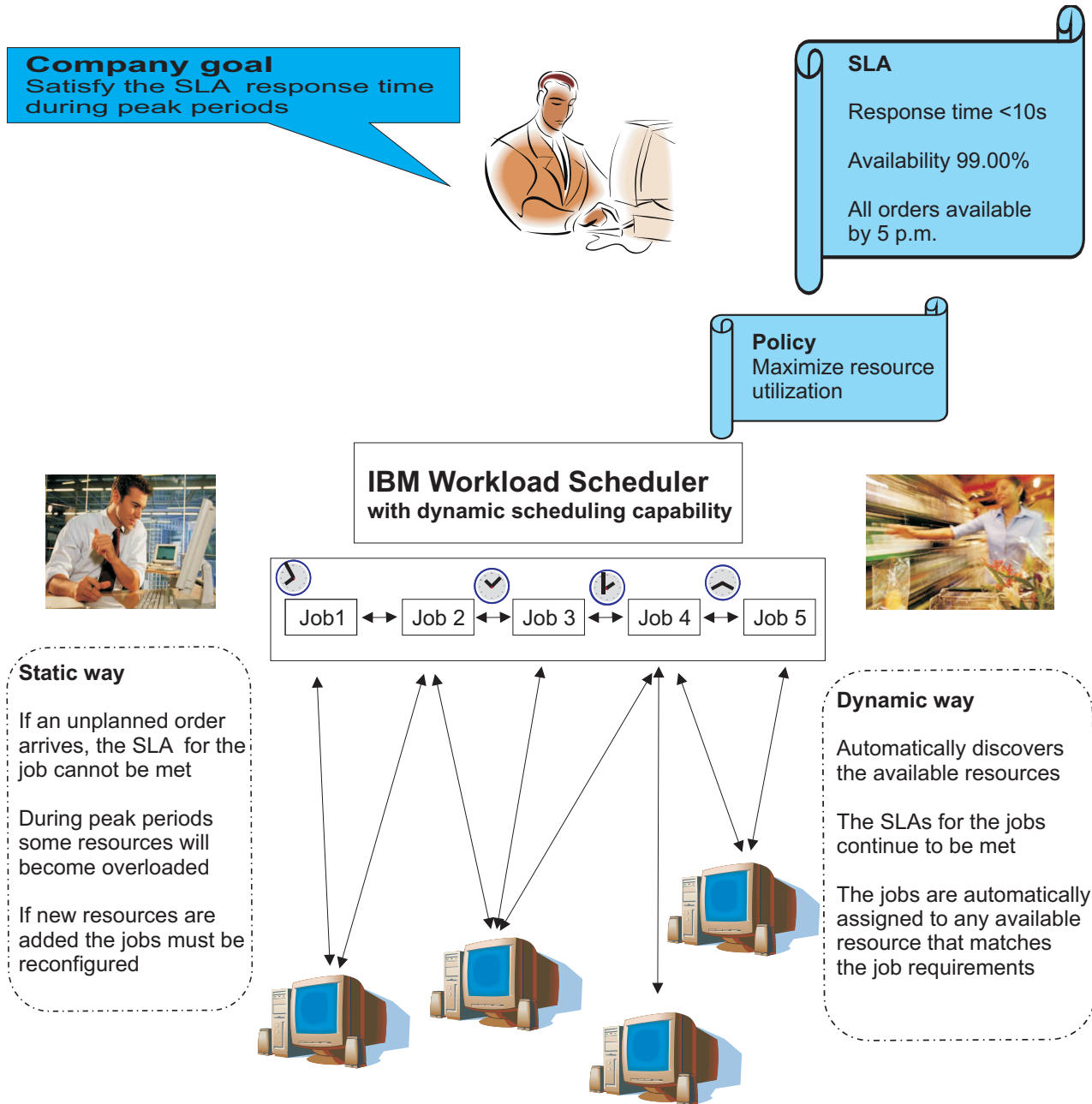


Figure 3. How to satisfy SLA response time during peak periods using the dynamic scheduling capability of IBM Workload Scheduler.

Monitoring the workload

While the operations analyst monitors the automated workload on a typical work day, he realizes that Job306, which is in the critical path of the schedule, is in the abend status. Because of this, Jobstream300 does not complete in the necessary time frame, causing a negative effect on the rest of the schedule. The consequent delay in running the plan might cause problems to Fine Cola's daily operations. With the help of the Dynamic Workload Console he then:

1. Analyzes the job and error logs in the current plan and finds that the error occurred for an unmanaged exception. The error might occur again and he cannot simply restart Job306.
2. Identifies the application specialist who is responsible for Job306 and opens a problem ticket containing all the information concerning the job.
3. Queries the status of depending jobs, exports the list in Comma Separated Variables (CSV) format, and attaches it to the ticket so that it can be viewed with a spreadsheet. Requests that the ticket be answered with high priority.

After an internal analysis, the application specialist finds that there is a broken execution path that must be fixed. The expected time for resolution is three hours, including a hot fix and a regression test.

One hour later, however, the operations analyst realizes that even if the application support team works overtime, the fix will not be completed before the end of the day and it will be impossible to close the daily processing today. He checks the status of the depending jobs and sets a target time to have the hot fix loaded into production during the night.

Then, sometime during the night:

1. The application team releases the hot fix and notifies the scheduling analyst who loads the new job into the production system, and notifies the operations analyst.
2. The operations analyst connects to the scheduling system from home to restart the job stream.
3. The operations analyst restarts Job306. The fix works and the job completes, as expected, one hour too late to complete the depending jobs before the next daily plan extension.
4. Early next morning the plan for the day is created. Because of the functionality of the latest version of IBM Workload Scheduler, the jobs depending on Job306, that could not complete in time, are now simply moved to run today, keeping their name and all their active file dependencies.
5. The operations analyst monitors the process remotely. When he arrives at work in the morning, he checks the actual completion of the daily workload. Everything completed successfully and he closes the ticket.

Managing the organization of the IT infrastructure

About this task

Two weeks before Christmas, the IT infrastructure administrator receives a notification from the scheduling analyst that an unplanned order adds so many tasks to a job stream in the critical path that its completion is delayed by a day. This causes a delay also in the completion of the plan scheduled to run the week before Christmas. The scheduling analyst advises him that he has already run the forecast plan and verified that with this addition the SLA for the job stream cannot be met and also the resources will become overloaded. To avoid this, concurrent jobs that need to use the same resource will need to wait until the requested quantity is available causing delay in the delivery of the order.

To find a solution to the potential problem and achieve the goals set for workload processing, without buying additional resources, using the dynamic domain manager, he proceeds in the following way:

1. He performs an automatic discovery of the resources available in the scheduling domain with their characteristics and relationships.

2. He finds a pool of resources in the Inventory department that meet the SLA to run the jobs. These resources have the required RAM, microprocessor, operating system, and application environments to run the new job stream and will be used at half their capacity during Christmas.

Without the use of dynamic scheduling he could not adapt the new workload processing to match load requirements with business policies and priorities, and resource availability and capacity. The only way to solve the problem would be to buy new hardware to run the added job streams increasing the cost of IT management infrastructure without optimizing the use of the existing resources.

3. He determines, based on the policies and jobs dispatching, how many new resources are required to run the new job stream.
4. He manages the definition of business-oriented performance goals for the entire domain of servers, provides an end-to-end view of actual performance relative to those goals, and manages the server resource allocation and load to meet the performance goals.
5. He identifies the required resources and finds an agreement with the Inventory department manager, to share the required resource between the two departments.
6. He defines a new logical resource in which he outlines the machines that are shared between the departments.
7. He communicates to the Ordering department the new agreement with the resource optimization.
8. Now he can guarantee the running of jobs within the time frame according to policies, rules, and resources planned availability. In this way he can also satisfy the optimization policy to maximize resource utilization.
9. The scheduling analyst now builds a feasible production plan.

Using dynamic scheduling he met the constraints imposed by rules and policies and achieved SLA goals, optimizing execution time, throughput, cost, and reliability.

The benefits

About this task

By adopting a workload scheduling strategy, and in particular by using IBM Workload Scheduler and its dynamic scheduling capabilities, Fine Cola is experiencing significant and immediate benefits, such as:

- The successful integration of all its manufacturing and distribution processes.
Because of how Fine Cola implemented their new processing flow, every customer order is active from the time a customer service representative receives it until the loading dock ships the merchandise and finance sends an invoice. Now orders can be tracked more easily, and manufacturing, inventory, and shipping among many different locations can be coordinated simultaneously. If an unplanned order arrives, it can be easily managed in the new dynamic IT infrastructure.
- The standardization and speeding up of the manufacturing process.
IBM Workload Scheduler has helped to automate many of the steps of Fine Cola's manufacturing process. This results in savings in time and increase in productivity.
- Reduce inventory

The manufacturing process flows more smoothly, and this improves visibility of the order fulfillment process inside the company. This can lead to reduced inventory of the raw materials used, and can help better plan deliveries to customers, reducing the finished goods inventory at the warehouses and shipping docks.

- Optimize IT infrastructures

The dynamic allocation of the IT resources maximizes the workload throughput across the enterprise reducing costs, improving performance, and aligning IT with business needs and service demands.

- Guarantees Fault Tolerance and High Availability

IBM Workload Scheduler can recover from server, agent, and communication failures and it can restart from the point where the failure happened. No status information will be lost due to failure events. Moreover if a computer breaks, its workload is automatically routed to another computer that can guarantee the SLAs.

In conclusion, this solution provides business value because it:

- Delivers service response times according to service level objectives.
- Understands dependencies on services for each line of business.
- Accommodates unpredictable use patterns with predictive logic.
- Understands service relationships to each other and to the IT infrastructure and business process layers.
- Provides network fault tolerance and high availability of the scheduling infrastructure.
- Reduces system and operational complexity and leverages IT staff skills and knowledge.
- Integrates systems quickly and easily, with minimal disruption to existing business processes.

Chapter 5. IBM Workload Scheduler

The IBM Workload Scheduler scheduling features help you plan every phase of production. During the plan processing period, the IBM Workload Scheduler production control programs manage the production environment and automate most operator activities. IBM Workload Scheduler prepares jobs for execution, resolves interdependencies, and launches and tracks each job. Because jobs start running as soon as their dependencies are satisfied, idle time is minimized, and throughput improves significantly. Jobs always run in sequence and, if a job fails, IBM Workload Scheduler handles the recovery process with little or no operator intervention.

Overview

The next sections provide an outline of IBM Workload Scheduler.

What is IBM Workload Scheduler

IBM Workload Scheduler is composed of the following parts:

IBM Workload Scheduler engine

The scheduling engine. It runs on every computer of a IBM Workload Scheduler network. During installation, the engine is configured for the role that the workstation will play within the scheduling network, such as master domain manager, domain manager, or agent.

The Dynamic Workload Console

Is Web-based, light, powerful, and user friendly. It can be used on any computer that has a web browser and provides access to all the current IBM Workload Scheduler functions. It is the strategic graphical user interface for the entire IBM Workload Automation portfolio.

The IBM Workload Scheduler network

An IBM Workload Scheduler network is made up of the workstations, or CPUs, on which jobs and job streams are run.

An IBM Workload Scheduler network contains at least one IBM Workload Scheduler domain, the master domain, in which the master domain manager is the management hub. Additional domains can be used to divide a widely distributed network into smaller, locally managed groups.

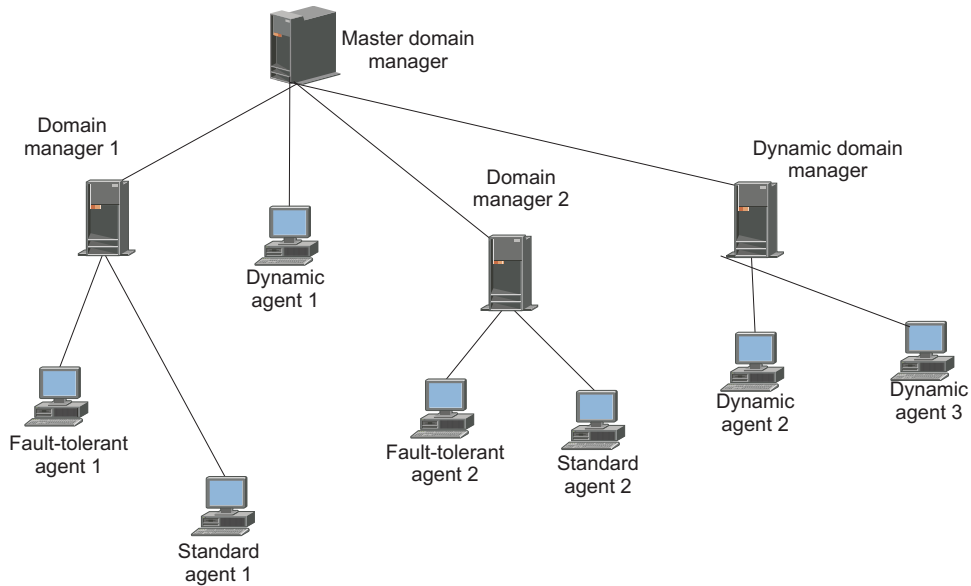


Figure 4. IBM Workload Scheduler network made up by two domains

Using multiple domains reduces the amount of network traffic by reducing the amount of communication required between the master domain manager and other computers.

In a single domain configuration, the master domain manager maintains communication with all of the workstations in the scheduling network.

In a multi-domain configuration, the master domain manager communicates with the workstations in its domain and with the subordinate domain managers. The subordinate domain managers, in turn, communicate with the workstations in their domains and with their subordinate domain managers. Multiple domains also provide fault-tolerance by limiting the problems caused by losing a domain manager to a single domain. To limit the effects further, you can designate backup domain managers to take over if their domain managers fail.

Every time the production plan is created or extended the master domain manager creates a production control file, named Symphony®. IBM Workload Scheduler is then restarted in the network, and the master domain manager sends a copy of the new production control file to each of its automatically linked agents and subordinate domain managers. The domain managers, in turn, send copies to their automatically linked agents and subordinate domain managers.

Once the network is started, scheduling messages like job starts and completions are passed from the agents to their domain managers, through the parent domain managers to the master domain manager. The master domain manager then broadcasts the messages throughout the hierarchical tree to update the production control files of domain managers and fault tolerant agents running in Full Status mode.

Manager and agent types

About this task

Primarily, workstation definitions refer to physical workstations. However, in the case of extended and network agents, the workstations are logical definitions that must be hosted by a physical IBM Workload Scheduler workstation.

IBM Workload Scheduler workstations can be of the following types:

Master domain manager (MDM)

The domain manager in the topmost domain of a IBM Workload Scheduler network. It either contains or connects to the relational database that stores the scheduling object definitions. It creates or updates the production file when the plan is created or extended and distributes it in the network. It performs all logging and reporting for the network.

Backup master

A fault-tolerant agent or domain manager capable of assuming the responsibilities of the master domain manager for automatic workload recovery.

Domain manager

The management hub in a domain. All communications to and from the agents in a domain are routed through the domain manager.

Backup domain manager

A fault-tolerant agent capable of assuming the responsibilities of its domain manager.

Dynamic domain manager

An installed component in a distributed IBM Workload Scheduler network that is the management hub in a domain. All communication to and from the dynamic agents in the domain is routed through the dynamic domain manager.

Backup dynamic domain manager

A workstation which can act as a backup for the dynamic domain manager, when problems occur. It is effectively a dynamic domain manager, waiting to be activated. Its use is optional.

Fault-tolerant agent (FTA)

A workstation capable of resolving local dependencies and launching its jobs in the absence of a domain manager.

Dynamic agent

Dynamic agents run the same types of jobs as fault-tolerant agents. From classic job types such as docommand and scripts, to more specific job types such as Java, database, file transfers, InfoSphere DataStage, Cognos, and OSLC to name a few. Dynamic agents can be defined in pools and dynamic pools.

Run workload dynamically

It communicates with the server the status of its resources. In this way the product is able to dynamically run your workload to the best available resources by:

- Automatically discovering scheduling environment resources.
- Automatically following resource changes
- Requesting additional resources when needed
- Matching job requirements to available resources
- Controlling and optimizing use of resources

The characteristics listed above provides high availability and load balancing potentialities to your environment and well suite virtualized environments.

When a job is submitted, either as part of a job stream in the plan or through ad hoc submission, IBM Workload Scheduler checks the job requirements, the available resources and the related characteristics and submits the job to the resource that best meets the requirements to run it.

Run both existing job types and job types with advanced options

It can run:

- Existing job types. For example docommand and scripts.
- Job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins. For example, those supplied with the product are DB2, file transfer, and web services. Those implemented through the custom plug-ins are the ones you developed using the Integration Workbench of the Software Development Kit (SDK). To run these job types you must also install the Java run time.

Manage dynamic workload broker logical resource

It can remotely run, from the agent, the dynamic workload broker **resource** command on the server. To manage the **resource** command you must also install the Java run time.

Pool Pools are groups in which you add specific dynamic agent workstations according to the requirements of the job. Jobs are assigned dynamically to the best available dynamic agent applying a load balancing policy by choosing the agent with the lesser amount of jobs to run.

Dynamic pool

Dynamic pools are groups in which you specify the requirements and IBM Workload Scheduler selects the dynamic agents that meet the requirements. Jobs are assigned dynamically to the best available dynamic agent applying an optimization policy to identify the best agent.

Standard agent

A workstation that launches jobs only under the direction of its domain manager. It is not fault-tolerant.

Extended agent

A logical workstation definition that helps you launch and control jobs on other systems and applications, such as PeopleSoft, SAP, and z/OS.

z-centric agent

Runs jobs scheduled from IBM Workload Scheduler for z/OS but is installed in the IBM Workload Scheduler environment. It has no fault-tolerance and communicates directly with the IBM Workload Scheduler for z/OS controller through the RESTful HTTP interface. In IBM Workload Scheduler for z/OS it has the same functionality as a computer automatic workstation even though it runs in the distributed environment.

Network Agent

A logical workstation definition for creating dependencies between jobs and job streams in separate IBM Workload Scheduler networks.

Topology

About this task

A key to choosing how to set up IBM Workload Scheduler domains for an enterprise is the concept of localized processing. The idea is to separate or localize the enterprises's scheduling needs based on a common set of characteristics.

Common characteristics are things such as geographical locations, business functions, and application groupings. Grouping related processing can limit the amount of interdependency information that needs to be communicated between domains. The benefits of localizing processing in domains are:

- Decreased network traffic. Keeping processing localized to domains eliminates the need for frequent interdomain communications.
- Provides a convenient way to tighten security and simplify administration. Security and administration can be defined at, and limited to, the domain level. Instead of network-wide or workstation-specific administration, you can have domain administration.
- Network and workstation fault tolerance can be optimized. In a multiple domain IBM Workload Scheduler network, you can define backups for each domain manager, so that problems in one domain do not disrupt operations in other domains.

Networking

About this task

The following questions will help in making decisions about how to set up your enterprise's IBM Workload Scheduler network. Some questions involve aspects of your network, and others involve the applications controlled by IBM Workload Scheduler. You may need to consult with other people in your organization to resolve some issues.

- How large is your IBM Workload Scheduler network? How many computers does it hold? How many applications and jobs does it run?

The size of your network will help you decide whether to use a single domain or the multiple domain architecture. If you have a small number of computers, or a small number of applications to control with IBM Workload Scheduler, there may not be a need for multiple domains.

- How many geographic locations will be covered in your IBM Workload Scheduler network? How reliable and efficient is the communication between locations?

This is one of the primary reasons for choosing a multiple domain architecture. One domain for each geographical location is a common configuration. If you choose single domain architecture, you will be more reliant on the network to maintain continuous processing.

- Do you need centralized or decentralized management of IBM Workload Scheduler?

A IBM Workload Scheduler network, with either a single domain or multiple domains, gives you the ability to manage IBM Workload Scheduler from a single node, the master domain manager. If you want to manage multiple locations separately, you can consider the installation of a separate IBM Workload Scheduler network at each location. Note that some degree of decentralized management is possible in a stand-alone IBM Workload Scheduler network by mounting or sharing file systems.

- Do you have multiple physical or logical entities at a single site? Are there different buildings, and several floors in each building? Are there different departments or business functions? Are there different applications?
These may be reasons for choosing a multi-domain configuration. For example, a domain for each building, department, business function, or each application (manufacturing, financial, engineering, etc.).
- Do you run applications, like SAP R/3, that will operate with IBM Workload Scheduler?
If they are discrete and separate from other applications, you may choose to put them in a separate IBM Workload Scheduler domain.
- Would you like your IBM Workload Scheduler domains to mirror your Windows domains?
This is not required, but may be useful.
- Do you want to isolate or differentiate a set of systems based on performance or other criteria?
This may provide another reason to define multiple IBM Workload Scheduler domains to localize systems based on performance or platform type.
- How much network traffic do you have now?
If your network traffic is manageable, the need for multiple domains is less important.
- Do your job dependencies cross system boundaries, geographical boundaries, or application boundaries? For example, does the start of Job1 on CPU3 depend on the completion of Job2 running on CPU4?
The degree of interdependence between jobs is an important consideration when laying out your IBM Workload Scheduler network. If you use multiple domains, you should try to keep interdependent objects in the same domain. This will decrease network traffic and take better advantage of the domain architecture.
- What level of fault-tolerance do you require?
An obvious disadvantage of the single domain configuration is the reliance on a single domain manager. In a multi-domain network, the loss of a single domain manager affects only the agents in its domain.

IBM Workload Scheduler components

About this task

IBM Workload Scheduler uses several manager processes to efficiently segregate and manage networking, dependency resolution, and job launching. These processes communicate among themselves through the use of message queues. Message queues are also used by the Console Manager (conman) to integrate operator commands into the batch process.

On any computer running IBM Workload Scheduler there are a series of active management processes. They are started as a system service, or by the StartUp command. The following are the main processes of a fault-tolerant agent:

Netman

The network management process that establishes network connections between remote mailman processes and local Writer processes.

Mailman

The mail management process that sends and receives inter-CPU messages.

Batchman

The production control process. Working from Symphony, the production control file, it runs jobs streams, resolves dependencies, and directs jobman to launch jobs.

Writer The network writer process that passes incoming messages to the local mailman process.

Jobman

The job management process that launches and tracks jobs under the direction of batchman.

The following are the main processes of a dynamic agent:

agent The network management process that establishes network connections to the master domain manager or the dynamic domain manager.

JobManager

The process that manages all of the dynamic agent operations, such as, sending events, and directing TaskLauncher to launch jobs.

TaskLauncher

The job management process that launches and tracks jobs under the direction of JobManager.

JobManagerGW

Manages the communication between dynamic agents and the master or dynamic domain manager through a firewall.

In addition, IBM Workload Scheduler uses two command line interfaces:

Composer

The command-line program used to define, manage, and store scheduling objects in the IBM Workload Scheduler database. The composer command-line program can be installed and used on any computer connected through TCP/IP to the system where the master domain manager is installed. It does not require the installation of a IBM Workload Scheduler workstation as a prerequisite. It communicates through HTTP/HTTPS with the master domain manager where the relational database management system (RDBMS) is installed. The HTTP/HTTPS communication setup and the authentication check are managed by the WebSphere® Application Server - Express® infrastructure. The composer uses edit files to update the scheduling database.

Conman

The console manager. It is the user interface for plan running activities by means of the command line interface. Conman writes information that is received by either the local netman or mailman processes.

IBM Workload Scheduler scheduling objects

About this task

Scheduling with IBM Workload Scheduler includes the capability to do the following:

- Schedule jobs across a network.
- Group jobs into job streams according, for example, to function or application.
- Set limits on the number of jobs that can run concurrently.

- Create job streams based on day of the week, on specified dates and times, or by customized calendars.
- Ensure correct processing order by identifying dependencies such as successful completion of previous jobs, availability of resources, or existence of required files.
- Set automatic recovery procedures for unsuccessful jobs.
- Forward incomplete jobs to the next production day.

Starting from version 8.3, the IBM Workload Scheduler scheduling objects are stored in a relational database. This results in a significant improvement, in comparison with previous versions, of how objects are defined and managed in the database. Each object can now be managed independently without having to use lists of scheduling objects like calendars, parameters, prompts and resources. The command syntax used to define and manage these objects has also become direct and powerful.

IBM Workload Scheduler administrators and operators work with these objects for their scheduling activity:

Workstation

Also referred to as *CPU*. Usually an individual computer on which jobs and job streams are run. Workstations are defined in the IBM Workload Scheduler database as a unique object. A workstation definition is required for every computer that executes jobs or job streams in the IBM Workload Scheduler network.

Workstation class

A group of workstations. Any number of workstations can be placed in a class. Job streams and jobs can be assigned to execute on a workstation class. This makes replication of a job or job stream across many workstations easy.

Domain

All workstations in a distributed IBM Workload Scheduler network are organized in one or more domains, each of which consists of one or more agents and a domain manager acting as the management hub. Most communication to and from the agents in the domain is routed through the domain manager. You can organize all agents in your network in a single domain, or in multiple domains. A single domain network consists of a master domain manager and any number of agents. A multiple domain network consists of a master domain manager, any number of lower tier domain managers, and any number of agents in each domain.

Job A script or command, run on the user's behalf, run and controlled by IBM Workload Scheduler.

Job stream

A list of jobs that run as a unit (such as a weekly backup application), along with run cycles, times, priorities, and other dependencies that determine the exact order in which the jobs run.

Calendar

A list of scheduling dates. Each calendar can be assigned to multiple job streams. Assigning a calendar to a job stream causes that job stream to run on the dates specified in the calendar. A calendar can be used as an inclusive or as an exclusive run cycle.

Run cycle

A cycle that specifies the days that a job stream is scheduled to run. Run cycles are defined as part of job streams and may include calendars that were previously defined. There are three types of run cycles: a Simple run cycle, a Weekly run cycle, or a Calendar run cycle (commonly called a

calendar). Each type of run cycle can be inclusive or exclusive. That is, each run cycle can define the days when a job stream is included in the production cycle, or when the job stream is excluded from the production cycle.

Run cycle group

A run cycle group is a list of run cycles that are combined together to produce a set of run dates. You can optionally define a run cycle group for your job stream instead of, or in addition to, a number of single run cycles. Unlike run cycles, run cycle groups are not tied to a specific job stream and therefore can be reused in several different job streams to apply the same scheduling rules each time.

Run cycles are organized into subsets within a run cycle groups. The subsets are in a logical OR relationship with each other. They enable you to apply exclusive run cycles to a subset of the inclusive run cycles. The result of the run cycle group is a run date or a set of run dates.

Prompt

An object that can be used as a dependency for jobs and job streams. A prompt must be answered affirmatively for the dependent job or job stream to launch. There are two types of prompts: predefined and ad hoc. An ad hoc prompt is defined within the properties of a job or job stream and is unique to that job or job stream. A predefined prompt is defined in the IBM Workload Scheduler database and can be used by any job or job stream.

Resource

An object representing either physical or logical resources on your system. Once defined in the IBM Workload Scheduler database, resources can be used as dependencies for jobs and job streams. For example, you can define a resource named *tapes* with a unit value of two. Then, define jobs that require two available tape drives as a dependency. Jobs with this dependency cannot run concurrently because each time a job is run the *tapes* resource is in use.

Variable and variable table

A variable can be used to substitute values in scheduling objects contained in jobs and job streams; that is, in JCL, log on, prompts dependencies, file dependencies, and recovery prompts. The values are replaced in the job scripts at run time. Variables are global (that is, they can be used on any agent of the domain) and are defined in the database in groups called variable tables.

Parameter

A parameter can be used to substitute values in jobs and job streams just like global variables. The difference is that a parameter is defined on the specific workstation where the related job is to run and has no global effect, but only on that specific workstation. Parameters cannot be used when scripting extended agent jobs.

User

On Windows workstations, the user name specified in the Logon field of a job definition must have a matching user definition. The definitions provide the user passwords required by IBM Workload Scheduler to launch jobs.

Event rule

A scheduling event rule defines a set of actions that are to run upon the occurrence of specific event conditions. The definition of an event rule correlates events and triggers actions. When you define an event rule, you specify one or more events, a correlation rule, and the one or more actions

that are triggered by those events. Moreover, you can specify validity dates, a daily time interval of activity, and a common time zone for all the time restrictions that are set.

Workload application

A workload application is an IBM Workload Scheduler database object that acts as a container for one or more job streams. You can use workload applications to standardize a workload automation solution so that the solution can be reused in one or more IBM Workload Scheduler environments thereby automating business processes.

You prepare a workload application template in a source IBM Workload Scheduler environment and then export it so that it can be deployed in a target environment. The export process extracts from the source environment all of the elements necessary to reproduce the solution in another environment. It produces a compressed file containing a number of files required to import the workload application into the target environment.

You can control how jobs and job streams are processed with the following attributes:

Dependencies

Conditions that must be satisfied before a job or job stream can run. You can set the following types of dependencies:

- A predecessor job or job stream must have completed successfully.
- One or more specific resources must be available.
- Access to specific files must be granted.
- An affirmative response to a prompt.

Time constraints

Conditions based on time, such as:

- The time at which a job or job stream should start.
- The time after which a job or job stream cannot start.
- The repetition rate at which a job or job stream is to be run within a specified time slot.

Job priority

A priority system according to which jobs and job streams are queued for execution.

Job fence

A filter defined for workstations. Only jobs and job streams whose priority exceeds the job fence value can run on a workstation.

Limit Sets a limit to the number of jobs that can be launched concurrently on a workstation.

The production process

IBM Workload Scheduler production is based on a plan that runs in a *production period*. The production period is defined by the user when creating or extending the production plan. It can span from a few hours to some days (by default it is a 24 hours period). Before the start of each production period, IBM Workload Scheduler executes a program that creates the production plan starting from the modeling data stored in the database and from an intermediate plan called *preproduction plan*. Then another program includes the uncompleted schedules from the previous production period into the current plan and another different program logs all the statistics of the previous production into an archive.

All of the required information for that production period is placed into a production control file named Symphony. During the production period, the production control file is continually being updated to reflect the work that needs to be done, the work in progress, and the work that has been completed. A copy of the Symphony file is sent to all subordinate domain managers and to all the fault-tolerant agents in the same domain. The subordinate domain managers distribute their copy to all the fault-tolerant agents in their domain and to all the domain managers that are subordinate to them, and so on down the line. This causes fault-tolerant agents throughout the network to continue processing even if the network connection to their domain manager is down. From the graphical interfaces or the command line interface, the operator can view and make changes in the current production by making changes in the Symphony file.

IBM Workload Scheduler processes monitor the production control file and make calls to the operating system to launch jobs as required. The operating system runs the job, and in return informs IBM Workload Scheduler if the job completed successfully or not. This information is entered into the production control file to indicate the status of the job.

Scheduling

About this task

Scheduling can be accomplished either through the IBM Workload Scheduler command line interface or one of the two graphical interfaces.

Scheduling includes the following tasks:

- Defining and maintaining workstations
- Defining scheduling objects
- Defining job streams
- Starting and stopping production processing
- Viewing and modifying jobs and job streams.

Defining scheduling objects

About this task

Scheduling objects are workstations, workstation classes, domains, jobs, job streams, resources, prompts, calendars, variables and variable tables, parameters, event rules, and run cycle groups. Scheduling objects are managed with the Composer program and are stored in the IBM Workload Scheduler database. To create or modify an object, you can use either the IBM Workload Scheduler command line interface or one of the graphical interfaces.

Creating job streams

The primary processing task of IBM Workload Scheduler is to run job streams. A job stream is an outline of batch processing consisting of a list of jobs. Job streams can be defined using either the command line interface or one of the graphical interfaces. Using either graphical interface you can easily create and modify job streams. You can use their job stream editors to work with the jobs and the follows dependencies between the jobs, as well as the job stream run cycles. You can also easily specify time restrictions, resource dependencies, file dependencies, and prompt dependencies at the job stream level.

Job streams can be defined as *draft*. A draft job stream is not considered when resolving dependencies and is not added to the production plan. It becomes actual only after the *draft* keyword is removed from its definition, and the JnextPlan command is run to add it to the preproduction plan and so to the production plan.

Setting job recovery

About this task

When defining a job, consider that in some instances it might not complete successfully. The administrator can define a recovery option and recovery actions when defining the job. The following recovery options are available:

- Do not continue with the next job. This stops the execution of the job stream and puts it in the *stuck* state. This is the default action.
- Continue with the next job.
- Run the job again.

Optionally, a recovery prompt can be associated to the job. A recovery prompt is a local prompt to display when the job completes unsuccessfully. Processing does not continue until the prompt is answered affirmatively.

Another option is to define a recovery job that can be run in the place of the original job if it completes unsuccessfully. The recovery job must have been defined previously. Processing stops if the recovery job does not complete successfully.

Defining and managing mission-critical jobs

About this task

Job schedulers can use the IBM Workload Scheduler command line or the Dynamic Workload Console to flag jobs as mission-critical and specify their deadlines. A critical job and all its predecessors make up what is called a *critical network*. At planning time, IBM Workload Scheduler calculates the start time of the critical job and of each of its predecessors starting from the critical job deadline and estimated duration. While the plan runs, this information is dynamically kept up-to-date based on how the plan is progressing. If a predecessor, or the critical job itself, is becoming late, IBM Workload Scheduler automatically prioritizes its submission and promotes it to get more system resources and thus meet its deadline.

Within a critical network, IBM Workload Scheduler dynamically identifies the path of predecessors that is potentially most at risk; this is called the *critical path*. IBM Workload Scheduler calculates the level of risk that each critical job has of missing its deadline; a high risk indicates that the estimated end of the critical job is after its deadline while a potential risk indicates that some predecessors of the critical job have a warning condition, for example are late or in error.

The Dynamic Workload Console provides specialized views for tracking the progress of critical jobs and their predecessors. Job schedulers and operators can access the views from the Dashboard or by creating Monitor Critical Jobs tasks.

The initial view lists all critical jobs for the engine, showing the status: normal, potential risk, or high risk. From this view, an operator can navigate to see:

- The hot list of jobs that put the critical deadline at risk.
- The critical path.

- Details of all critical predecessors.
- Details of completed critical predecessors.
- Job logs of jobs that have already run.

Using the views, operators can monitor the progress of the critical network, find out about current and potential problems, release dependencies, and rerun jobs.

For example:

1. To flag a critical job and follow it up, the Job scheduler opens the Workload Designer on the Dynamic Workload Console, marks the specific job as critical, and sets the deadline for 5 a.m.
When JnextPlan is run, the critical start dates for this job, and all the jobs that are identified as its predecessors, are calculated.
2. To track a specific critical job, the operator proceeds as follows:
 - a. The operator checks the dashboards and sees that there are critical jobs scheduled on one of the engines.
 - b. He clicks the link to get a list of the critical jobs.
The specific job shows a Potential Risk status.
 - c. He selects the job and clicks **Hot List** to see the predecessor job or jobs that are putting the critical job at risk.
One of the predecessor jobs is listed as being in error.
 - d. He selects the job and clicks **Job log**.
The log shows that the job failed because of incorrect credentials for a related database.
 - e. After discovering that the database password was changed that day, he changes the job definition in the symphony file and reruns the job.
 - f. When he comes back to the dashboard, he notices that there are no longer any jobs in potential risk. Also, the critical jobs list that was opened when clicking on the potential risk link no longer shows the critical job after the job is rerun.
 - g. The job is now running after being automatically promoted, getting higher priority for submission and system resources.
 - h. No further problems need fixing and the critical job finally completes at 4.45 a.m.

Scheduling workload dynamically

About this task

You can choose to set IBM Workload Scheduler to dynamically associate your submitted workload (or part of it) to the best available resources at run time.

The IBM Workload Scheduler installation process includes the option to install the dynamic scheduling capability. If you select this option, you get the following functionality:

- Automatically discover scheduling environment resources
- Match job requirements to available resources
- Control and optimize use of resources
- Automatically follow resource changes
- Request additional resources when needed

You can submit IBM Workload Scheduler jobs, including jobs defined to run on extended agents, as well as J2EE applications (if you selected the option to schedule J2EE at installation time). To schedule workload dynamically, you:

1. Use the Dynamic Workload Console to define the agents you want to use for running workload as logical resources or groups of resources.
2. Update your IBM Workload Scheduler job definitions to make as destination CPU the dynamic workload broker workstation (this workstation works as a bridge between the scheduler engine and the pool of resources)
3. For every IBM Workload Scheduler job, add a JSDL (Job Submission Description Language) job definition where you match the job with required resources, candidate hosts, and scheduling and optimization preferences. Use the Dynamic Workload Console to do this easily.

When a job is thus submitted, either as part of a job stream in the plan or through ad hoc submission, IBM Workload Scheduler checks the job requirements, the available resources and the related characteristics and submits the job to the resource that best meets the requirements.

Running production

Production consists of taking the definitions of the scheduling objects from the database, together with their time constraints and their dependencies, and building and running the production control file.

Running the plan

About this task

The production plan contains information about which jobs to run, on which fault-tolerant agent, and what dependencies must be satisfied before each job is launched. IBM Workload Scheduler creates the production plan starting from the modeling data stored in the database and from an intermediate plan called the preproduction plan. The preproduction plan is automatically created and managed by the product. To avoid problems, the database is locked during the generation of the plan and is unlocked when the generation completes or if an error condition occurs. The preproduction plan is used to identify in advance the job stream instances and the external follows job stream dependencies involved in a specified time period.

You use the `JnextPlan` command on the master domain manager to generate the production plan and distribute it across the IBM Workload Scheduler network.

To generate and start a new production plan, IBM Workload Scheduler performs the following steps:

1. Updates the preproduction plan with the objects defined in the database that were added or updated since the last time the plan was created or extended.
2. Retrieves from the preproduction plan the information about the job streams to run in the specified time period and saves it in an intermediate production plan.
3. Includes in the new production plan the uncompleted job streams from the previous production plan.
4. Creates the new production plan and stores it in a file named `Symphony`. The production plan data is also replicated in the database.

5. Distributes a copy of the Symphony file to the workstations involved in the new product plan processing.
6. Logs all the statistics of the previous production plan into an archive.
7. Updates the job stream state in the preproduction plan.

The copy of the newly-generated Symphony file is used starting from the top domain's fault-tolerant agents and domain managers of the child domains and down the tree to all subordinate domains.

Each fault-tolerant agent that receives the production plan can continue processing even if the network connection to its domain manager goes down.

At each destination fault-tolerant agent, IBM Workload Scheduler performs the following actions to manage job processing:

1. Accesses the copy of the Symphony file and reads the instructions about which job to run.
2. Makes calls to the operating system to launch jobs as required.
3. Updates its copy of the Symphony file with the job processing results and sends notification back up the tree to the master domain manager and to all full status fault-tolerant agents. The original copy of the Symphony file is stored on the master domain manager and the copies stored on the backup master domain managers, if defined, are updated accordingly.

This means that during job processing, each fault-tolerant agent has its own copy of the Symphony file updated with the information about the jobs it is running (or that are running in its domain and child domains if the fault-tolerant agent is full-status or a domain manager), and the master domain manager (and backup master domain manager if defined) has the copy of the Symphony file that contains all updates coming from all fault-tolerant agents. In this way the Symphony file on the master domain manager is kept up-to-date with the jobs still to run, the jobs running, and the jobs already completed.

After the production plan is generated for the first time, it can be extended to the next production period with the `JnextPlan` command. The Symphony file is refreshed with the latest changes and redistributed throughout the network.

Running job streams

Depending on their run cycle definition, job streams are taken from the IBM Workload Scheduler database and automatically inserted into the current production plan.

While the job stream is in the plan, and has not completed, you can still modify any of its components. That is, you can modify the job stream properties, the properties of its jobs, their sequence, the workstation or resources they use, and so on, to satisfy last-minute requirements.

You can also hold, release, or cancel a job stream, as well as change the maximum number of jobs within the job stream that can run concurrently. You can change the priority previously assigned to the job stream and release the job stream from all its dependencies.

Last minute changes to the current production plan include the possibility to submit jobs and job streams that are already defined in the IBM Workload

Scheduler database but were not included in the plan. You can also submit jobs that are being defined ad hoc. These jobs are submitted to the current plan but are not stored in the database.

Starting from version 8.3, you can create and manage multiple instances of the same job stream over a number of days or at different times within the same day. This new feature introduced the possibility to have in the same plan more than one instance of the same job stream with the same name. Each job stream instance is identified by the job stream name, the name of the workstation where it is scheduled to run, and by the start time defined in the preproduction plan.

Monitoring

About this task

Monitoring is done by listing plan objects. Using lists, you can see the status of all or of subsets of the following objects in the current plan:

- Job stream instances
- Job instances
- Domains
- Workstations
- Resources
- File dependencies
- Prompt dependencies

You can use these lists also to manage some of these objects. For example, you can reallocate resources, link or unlink workstations, kill jobs, or switch domain manager.

Additionally, you can monitor the daily plan with Tivoli Business Service Manager, an object-oriented systems management application that provides monitoring and event management of resources, applications, and subsystems, that is integrated with IBM Workload Scheduler.

Network managers can use IBM Workload Scheduler/NetView, a NetView application, to monitor and diagnose IBM Workload Scheduler networks from a NetView management node. It includes a set of submaps and symbols to view IBM Workload Scheduler networks topographically, and determine the status of job scheduling activity and critical IBM Workload Scheduler processes on each workstation. Menu actions are provided to start and stop IBM Workload Scheduler processing and to run *conman* on any workstation in the network.

Controlling with IBM Tivoli Monitoring

About this task

IBM Tivoli Monitoring is a product that applies pre-configured best practices to the automated monitoring of essential system resources. It helps you to detect bottlenecks and other potential problems and provides you with the means for automatic recovery from critical situations. In this way it eliminates the need for system administrators to manually scan through extensive performance data.

IBM Workload Scheduler integrates with IBM Tivoli Monitoring through the installation of a customizable agent, the IBM Tivoli Monitoring agent that collects and distributes data to the Tivoli Enterprise Monitoring Server.

By using data providers that are the interfaces of the IBM Tivoli Monitoring agent , you can collect data from data sources, client programs, URLs, scripts, relation tables, or SNMP agents and transfer the collected data fo the custom IBM Tivoli Monitoring agent that monitors the collected data.

The IBM Tivoli Monitoring agent uses two different data providers to integrate with IBM Workload Scheduler:

- A script data provider that collects IBM Workload Scheduler data by running a script at regular intervals and parsing the script output to look for errors, anomalies, and threshold conditions. The following resources are monitored:
 - Host availability
 - Application status
 - IBM Workload Scheduler used space
 - IBM Workload Scheduler available disk space
 - Page-outs
 - Swap space available
 - IBM Workload Scheduler message file size
- An agent log file data source that monitors data in a sequential text file. IBM Workload Scheduler writes events for all monitored objects (for example, jobs and job streams), after each new plan production, to an event file that is accessed by the Tivoli Monitoring agent for IBM Workload Scheduler. The collected data is retrieved and displayed by the Tivoli Enterprise Portal.

Reporting

About this task

As part of the preproduction and post-production processes, reports are generated which show summary or detail information about the previous or next production day. These reports can also be generated ad-hoc. The following reports are available:

- Job details listing
- Prompt listing
- Calendar listing
- Parameter listing
- Resource listing
- Job History listing
- Job histogram
- Planned production schedule
- Planned production summary
- Planned production detail
- Actual production summary
- Actual production detail
- Cross reference report

In addition, during production, a standard list file (STDLIST) is created for each job instance launched by IBM Workload Scheduler. Standard list files contain header and trailer banners, echoed commands, and errors and warnings. These files can be used to troubleshoot problems in job execution.

Auditing

About this task

An auditing option helps track changes to the database and the plan.

For the database, all user modifications, except for the delta of the modifications, are logged. If an object is opened and saved, the action is logged even if no modification is made.

For the plan, all user modifications to the plan are logged. Actions are logged whether or not they are successful.

Audit files are logged to a flat text file on individual machines in the IBM Workload Scheduler network. This minimizes the risk of audit failure due to network issues and allows a straightforward approach to writing the log. The log formats are basically the same for both the plan and the database. The logs consist of a header portion which is the same for all records, an "action ID", and a section of data which varies according to the action type. All data is stored in clear text and formatted to be readable and editable from a text editor such as vi or notepad.

Using event-driven workload automation

About this task

Use this optional feature to set up and run rules that perform predefined actions in response to particular events occurring on your agents. Your organization can benefit from using this feature by adding on-demand workload automation to plan-based job scheduling, gaining savings in time and resources.

Event-driven workload automation is based on the concept of event rule. In IBM Workload Scheduler an event rule is a scheduling object and is made up of events, event-correlating conditions, and actions. When you define an event rule, you specify one or more events, a correlation rule, and one or more actions that are triggered by those events. Moreover, you can specify validity dates, a daily time interval of activity, and a common time zone for all the time restrictions that are set.

You can set up event rules to:

- Trigger the execution of batch jobs and job streams based on the occurrence or combination of real time events
- Reply to prompts
- Notify users when anomalous conditions occur in the IBM Workload Scheduler scheduling environment or batch scheduling activity
- Invoke an external product when a particular event condition occurs

IBM Workload Scheduler includes a set of predefined event and action plug-ins, but also provides a software development kit with samples and templates for your application programmers to develop their own plug-ins.

Options and security

The IBM Workload Scheduler options files determine how IBM Workload Scheduler runs on your system. Several performance, tuning, security, logging, and other configuration options are available.

Setting the IBM Workload Scheduler options

About this task

You can set two types of properties to configure your IBM Workload Scheduler run time environment, properties that are set on the master domain manager and affect processing on all workstations in the IBM Workload Scheduler network, and properties that are set locally on a workstation and affect processing on that workstation only. The former are managed using the IBM Workload Scheduler command line program named `optman`, and the latter you define locally on the workstation by customizing the configuration files `useropts`, `localopts`, and `jobmanrc`.

Global options are used to:

- Define if the security files of all the workstations of the network can be created and managed only from the master domain manager or if the root user or administrator of each workstation can create and manage their own.
- Select whether to enable or disable database auditing.
- Control which objects in the plan the user is permitted to list when running a query.
- Select whether to enable plan auditing.
- Select whether to enable strong encryption.
- Select whether to enable or disable the fault tolerant switch manager.
- Select whether to enable or disable the time zone option.
- Enter the number of days for which you want to save job statistics.
- Set the minimum and maximum lengths of the preproduction plan in days.
- Determine if uncompleted job streams are carried forward from the old to the new production control file.
- Remove or clean up job stream instances from the plan that ended in error.
- Define the start time of the IBM Workload Scheduler processing day.

Local options are used to:

- Specify the name of the local workstation
- Prevent the launching of jobs run by root in UNIX
- Prevent unknown clients from connecting to the system
- Specify a number of performance options
- Specify a number of logging preferences
- Set SSL security options.

Setting security

About this task

Security is accomplished with the use of a security file that contains one or more user definitions. Each user definition identifies a set of users, the objects they are permitted to access, and the types of actions they can perform.

A template file is installed with the product. Edit the template to create the user definitions and compile and install it with a utility program to create a new operational security file. After it is installed, you make further modifications by creating an editable copy with another utility.

An individual file can be maintained on each workstation, or a single security file can be created on the master domain manager and copied to each domain manager, fault-tolerant agent, and standard agent.

Secure authentication and encryption

About this task

Security is enhanced for connections between protected and non-protected domains by applying the authentication and encryption mechanism based on the Secure Sockets Layer (SSL) protocol. SSL uses digital certificates to authenticate the identity of a workstation.

The IBM Workload Scheduler administrator must plan how authentication will be used within the network:

- Use one certificate for the entire IBM Workload Scheduler network.
- Use a separate certificate for each domain.
- Use a separate certificate for each workstation.

SSL support is automatically installed with IBM Workload Scheduler.

Work across firewalls

About this task

For previous versions of IBM Workload Scheduler, running the commands to start or stop a workstation or to get the standard list required opening a direct TCP/IP connection between the originator and the destination nodes. In a firewall environment, this forces users to break the firewall to open a direct communication path between the master and each fault-tolerant agent in the network.

IBM Workload Scheduler provides a configurable attribute, named *behindfirewall*, in the workstation's definition in the database. You can set this attribute to ON to indicate that a firewall exists between that particular workstation and its domain manager, and that the link between the domain manager and the workstation (which can be another domain manager) is the only allowed link between the domains.

Also, for all the workstations having this attribute set to ON, the commands to start or stop the workstation or to get the standard list will be transmitted through the domain hierarchy instead of opening a direct connection between the master (or domain manager) and the workstation.

Centralized security mechanism

About this task

A new global option makes it possible to change the security model in the IBM Workload Scheduler network. If you use this option, then the security files for the fault-tolerant agents in the network can be created or modified only on the master domain manager. The IBM Workload Scheduler administrator is responsible for creating, updating, and distributing the security files for all the agents where user access is required. Setting this global option triggers a security mechanism to identify and trust the IBM Workload Scheduler network corresponding to that master domain manager.

If you prefer the traditional security model, you can still use it by not activating the global variable.

Using time zones

About this task

IBM Workload Scheduler supports different time zones. Enabling time zones provides you with the ability to manage your workload across a multiple time zone environment. Both the 3-character and the variable length notations are supported with the current version of IBM Workload Scheduler. The variable length notation format is area/city, for example Europe/Paris as equivalent to ECT (European Central Time). The 3-character notation is supported for capability with earlier versions of the product.

Once configured, time zones can be specified for start and deadline times within jobs and job streams.

Using the access methods of IBM Workload Scheduler

Access methods are used to extend the job scheduling capability of IBM Workload Scheduler to other systems and applications. They run on:

- **Extended agents**
- **Dynamic agents and IBM Workload Scheduler for z/OS agents**

For more details, see “Supported agents” on page 52.

An access method needs to interact with the external system either through its command line or the Dynamic Workload Console. IBM Workload Scheduler includes the following access methods:

- PeopleSoft (psagent)
- SAP R/3 (r3batch)
- z/OS (mvsjes and mvsopc)

Figure 5 shows the typical configuration for the extended agent and the dynamic agent.

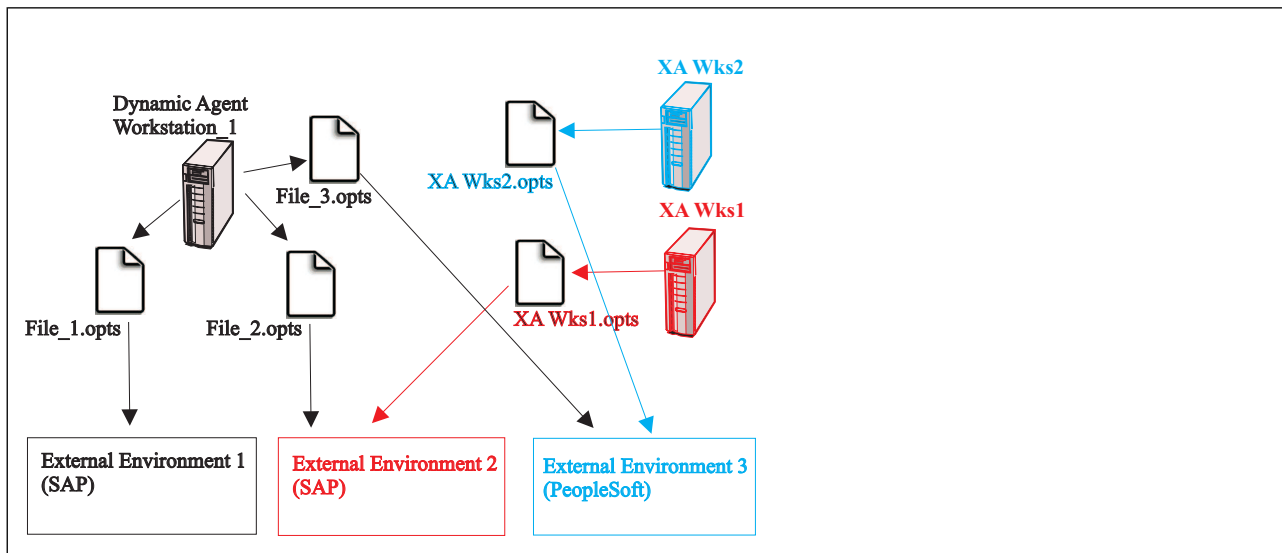


Figure 5. Supported agents configuration

Supported agents

The IBM Workload Scheduler access methods can run on extended agents and on dynamic agents (including IBM Workload Scheduler for z/OS agents).

You can extend IBM Workload Scheduler capabilities for scheduling on one or more external applications, such as PeopleSoft and SAP R/3. The agents needed to extend scheduling capabilities on these external applications using IBM Workload Scheduler are:

Extended agents

Extended agents extend the static scheduling capability. They are logical workstations related to an access method hosted by a physical IBM Workload Scheduler workstation (not another extended agent). More than one extended agent workstation can be hosted by the same IBM Workload Scheduler workstation and rely on the same access method. The extended agent is defined in a standard IBM Workload Scheduler workstation definition, which gives the extended agent a name and identifies the access method. The access method is a program that is run by the hosting workstation whenever IBM Workload Scheduler submits a job to an external system.

Jobs are defined for an extended agent in the same manner as for other IBM Workload Scheduler workstations, except for the job attributes that depend on the external system or application.

To launch and monitor a job on an extended agent, the host runs the access method, passing it job details as command line options. The access method communicates with the external system to launch the job and returns the status of the job. To launch a job in an external environment, IBM Workload Scheduler runs the extended agent access method providing it with the extended agent workstation name and information about the job. The method looks at the corresponding file named *XANAME_accessmethod.opts* (where *XANAME* is the name of the extended agent workstation) to determine which external environment instance it will connect to. The access method can then launch jobs on that instance and monitor them through completion, writing job progress and status information in the standard list file of the job.

A physical workstation can host a maximum of 255 extended agents.

Extended agents can also be used to run jobs in an end-to-end environment, where their scheduling and monitoring tasks are performed by a IBM Workload Scheduler for z/OS controller.

Dynamic agents and IBM Workload Scheduler for z/OS agents (z-centric)

To run access methods on external applications using dynamic agents, you define a job of type **Access Method**. The access method communicates with the external system to launch the job and returns the status of the job. The method looks at the corresponding file named *optionsfile_accessmethod.opts* (where *optionsfile* is the configuration file that depends on the selected access method). The dynamic agent and the IBM Workload Scheduler for z/OS agent can have more than one *.opts* associated file to determine which external environment instance to connect to. The access method can launch jobs on that instance and monitor them through completion, writing job progress and status information in the standard list file of the job.

Chapter 6. IBM Workload Scheduler for z/OS

IBM Workload Scheduler for z/OS expands the scope for automating your data processing (DP) operations. It plans and automatically schedules the production workload. From a single point of control, it drives and controls the workload processing at both local and remote sites. By using IBM Workload Scheduler for z/OS to increase automation, you use your DP resources more efficiently, have more control over your DP assets, and manage your production workload processing better.

How your production workload is managed

How does IBM Workload Scheduler for z/OS give you all this? This section describes functions that make your information systems (IS) operations run more efficiently. But first, here is a brief introduction to the structure of the product and some concepts.

Structure

About this task

IBM Workload Scheduler for z/OS consists of a base product, the *agent* and a number of features. Every z/OS system in your complex requires the base product. One z/OS system in your complex is designated the *controlling* system and runs the *engine* feature. Only one engine feature is required, even when you want to start standby engines on other z/OS systems in a sysplex.

IBM Workload Scheduler for z/OS with IBM Workload Scheduler addresses your production workload in the distributed environment. You can schedule, control, and monitor jobs in IBM Workload Scheduler from IBM Workload Scheduler for z/OS. For example, in the current plan, you can specify jobs to run on workstations in IBM Workload Scheduler.

The workload on other operating environments can also be controlled with the open interfaces provided with IBM Workload Scheduler for z/OS. Sample programs using TCP/IP or an NJE/RSCS (network job entry/remote spooling communication subsystem) combination show you how you can control the workload on environments that at present have no scheduling feature.

Additionally, national language features let you see the dialogs and messages, in the language of your choice. These languages are currently available:

- English
- German
- Japanese
- Korean
- Spanish

Panel and message text can also be modified to include enterprise-specific instructions or help.

Concepts

About this task

Terminology definition. Throughout this chapter the following synonyms are used alternately to express the same concepts:

Concept	Workload scheduling term	IBM Workload Scheduler for z/OS term
A list of jobs and of the information used to run them. A scheduling object defined in the product database.	job stream	application
The specific job stream (application) when it is entered in the IBM Workload Scheduler for z/OS plan. It is defined also by a specific execution date and time.	job stream instance	occurrence
A task, a command, or a number of tasks.	job	operation
A job in the plan.	job instance	operation

In managing production workloads, IBM Workload Scheduler for z/OS builds on several important concepts.

Plans. IBM Workload Scheduler for z/OS constructs operating *plans* based on user-supplied descriptions of the DP operations department and its production workload. These plans provide the basis for your service level agreements and give you a picture of the status of the production workload at any point in time. You can simulate the effects of changes to your production workload, calendar, and installation by generating trial plans.

Applications. Also known as *job stream*, an application is a description of a unit of production work. It can include the following:

- A list of the operations, also known as *jobs*: they are the tasks associated with that unit of work, such as:
 - Data entry
 - Job preparation
 - Job submission or started-task initiation
 - Communication with the NetView program
 - File transfer to other operating environments
 - Printing of output
 - Postprocessing activities, such as quality control or dispatch
 - Other tasks related to the unit of work that you want to schedule, control, and track
- A description of dependencies between jobs within a job stream and between jobs in other job streams
- Information about resource requirements, such as exclusive use of a data set
- Special operator instructions that are associated with a job
- How and where each job should be processed
- Run policies for that unit of work; that is, when it should be scheduled or alternatively the name of a group definition that records the run policy

IBM Workload Scheduler for z/OS schedules work based on the information you provide in your application (job stream) descriptions.

Workstations. When scheduling and processing work, IBM Workload Scheduler for z/OS considers the processing requirements of each job. Some typical processing considerations are:

- Which human or machine resources are required for processing the work, for example, operators, processors, or printers?
- When are these resources available?
- How are these jobs to be tracked?
- Can this work be processed somewhere else if the resources become unavailable?

IBM Workload Scheduler for z/OS supports a range of work process types, called *workstations*, that map the processing requirements of any task in your production workload. Each workstation supports one type of activity. This gives you the flexibility to schedule, monitor, and control any type of DP activity, including the following:

- Job setup, both manual and automatic
- Job submission
- Started-task actions
- Communication with the NetView program
- Print jobs
- Manual preprocessing or postprocessing activity

You can plan for maintenance windows in your hardware and software environments. IBM Workload Scheduler for z/OS helps you perform a controlled and incident-free shut down of the environment, preventing last-minute cancellation of active tasks. You can choose to reroute the workload automatically during any outage, planned or unplanned.

IBM Workload Scheduler for z/OS tracks jobs as they are processed at workstations and dynamically updates the plan with real-time information on the status of jobs. You can view or modify this status information online using the workstation ready lists in the dialog.

Virtual Workstations. Using virtual workstations improves workload balancing and the monitoring of system availability. This feature automatically directs the submission of workload to different destinations removing the need to associate a workstation to a specific destination. You can define a list of destinations for the submission of workload and the scheduler distributes the workload to automatically-selected active destinations, according to a round-robin scheduling approach.

You can activate this feature by specifying the new virtual option at workstation definition level. This option is allowed for computer workstations with the automatic reporting attribute, and is supported by all the interfaces available to define, modify, and monitor workstations.

Using virtual workstations the scheduler distributes the workload across your trackers evenly, thus avoiding bottlenecks when submitting or running jobs. In fact, the scheduler splits the workload among the available destinations, so that the Job Entry System (JES) and Workload Manager (WLM) do not find overloaded input queues when selecting jobs for their action.

Dependencies. In general, every DP-related activity must occur in a specific order. Activities performed out of order might create invalid output and possibly even corrupt your corporate data. This might cause costly reruns, missed deadlines, and unsatisfied customers.

You can define *dependencies* for operations (jobs) when a specific processing order is required. When IBM Workload Scheduler for z/OS manages the dependent relationships for you, the jobs are always started in the correct order every time they are scheduled. A dependency is called *internal* when it is between two jobs in the same job stream, and *external* when it is between two jobs in different job streams. A dependency takes place between a *predecessor* operation and a *successor* operation, whereby the successor can start after its predecessor has completed.

The resolution of a dependency is resolved based on the input arrival times - theoretical start times that help define a specific application occurrence - of the applications of which the predecessors and successors are part. A dependency is resolved when a best matching predecessor is found according to the criteria defined for that dependency. The criteria can be that the best matching predecessor can be found within the closest preceding occurrence with respect to the successor, or in one that runs in the same day or within a specific interval of days or hours. Also, a dependency can be defined as mandatory to various degrees.

In addition, you can specify *conditional dependencies*, where you use the return code and status of an operation to determine the start of another operation. Standard logical operators are supported to define the check on status or return code values, to implement the definition of dependencies with a *conditional logic*. If the predecessor operation is associated to a job with different steps, you can specify a conditional *step-level dependency* on individual step return codes.

IBM Workload Scheduler for z/OS lets you serialize work based on the status of any DP resource. A typical example is a job that uses a data set as input, but must not start until the data set is successfully created and loaded with valid data. You can use *resource serialization* support to send availability information about a DP resource to IBM Workload Scheduler for z/OS.

Special resources. Special resources are typically defined to represent physical or logical objects used by jobs. A special resource can be used to serialize access to a data set or to limit the number of file transfers on a particular network link. The resource does not have to represent a physical object in your configuration, although it often does.

IBM Workload Scheduler for z/OS keeps a record of the state of each resource and its current allocation status. You can choose to hold resources if a job allocating the resources ends abnormally. You can also use the IBM Workload Scheduler for z/OS interface with the Resource Object Data Manager (RODM) to schedule jobs according to *real* resource availability. You can subscribe to RODM updates in both local and remote domains.

IBM Workload Scheduler for z/OS lets you *subscribe* to data set activity on z/OS systems. The data set triggering function of IBM Workload Scheduler for z/OS automatically updates special resource availability when a data set is closed. You can use this notification to coordinate planned activities or to add unplanned work to the schedule.

Calendars. IBM Workload Scheduler for z/OS uses information about when the job departments work, so that job streams are not scheduled to run on days when

processing resources are not available (for example, Sundays and holidays). This information is stored in a *calendar*. IBM Workload Scheduler for z/OS supports *multiple calendars* for enterprises where different departments have different work days and non-working days. Different groups within a business operate according to different calendars.

The multiple calendar function is critical if your enterprise has installations in more than one geographical location (for example, with different local or national holidays).

Business processing cycles. IBM Workload Scheduler for z/OS uses business processing cycles, or *periods*, to calculate when your job streams run, for example, weekly, or every 10th working day. Periods are based on the business cycles of your customers. IBM Workload Scheduler for z/OS supports a range of periods for processing the different job streams in your production workload.

When you define a job stream, you specify when it is planned to run, using a *run cycle*, which can be:

- A *rule* with a format such as
ONLY the SECOND TUESDAY of every MONTH
EVERY FRIDAY in the user-defined period SEMESTER1

where the words in upper case are selected from lists of ordinal numbers, names of days, and common calendar intervals or period names.

- A combination of period and *offset*. For example, an offset of 10 in a monthly period specifies the 10th day of each month.

Run cycle groups. One of the elements that make up the definition of an application is the run cycle, where you specify the temporal details of when the application should run in terms of time, days, weeks, months, or periods (and several other details). An application can have several definitions of run cycles, which become part of the definition of that application. There are several types of run cycles, such as regular, exclusive, rule-based, or period.

Going one step further, you can define run cycle groups. These are database objects in their own right, and are not part of the definition of an application, but the same run cycle group can be used by more than one application. A run cycle group is a list of run cycles that, combined together, produce a set of run dates.

You can structure a run cycle group into subsets. Within a subset you can match an exclusive run cycle against a positive one to generate negative occurrences, which identify the days when an application is normally scheduled to run but is bypassed.

You can make use of the logical AND between two run cycles in a group. This enables you to easily define rules that schedule work on complex run dates.

Using Plans in IBM Workload Scheduler for z/OS

IBM Workload Scheduler for z/OS plans your production workload schedule. It produces both high-level and detailed plans. These plans both drive the production workload and show you the status of the production workload on your system at any specified time. You can produce trial plans to forecast future workloads.

Long-term planning

About this task

The *long-term plan* is a high-level schedule of your anticipated production workload. It lists, by day, the instances of job streams to be run during the period of the plan. Each instance of a job stream is called an *occurrence*. The long-term plan shows when occurrences are to run, as well as the dependencies that exist between the job streams. You can view these dependencies graphically on your workstation as a network, to check that work has been defined correctly. The plan can help you in forecasting and planning for heavy processing days. The long-term-planning function can also produce histograms showing planned resource use for individual workstations during the plan period.

You can use the long-term plan as the basis for documenting your service level agreements. It lets you relate service level agreements directly to your production workload schedules so that your customers can see when and how their work is to be processed.

The long-term plan provides a window to the future. You can decide how far into the future, from one day to four years. You can also produce long-term plan simulation reports for *any* future date. IBM Workload Scheduler for z/OS can automatically extend the long-term plan at regular intervals. You can print the long-term plan as a report, or you can view, alter, and extend it online using the dialogs.

Detailed planning

About this task

The *current plan* is the center of IBM Workload Scheduler for z/OS processing. It drives the production workload automatically and provides a way to check its status. The current plan is produced by the run of batch jobs that extract from the long-term plan the occurrences that fall within the specified period of time from the job details. The current plan selects a window from the long-term plan and makes the jobs ready to be run. They are started depending on the decided restrictions (for example, dependencies, resources availability, or time-dependent jobs).

The current plan is a rolling plan that can cover several days. A common method is to cover 1 to 2 days with regular extensions each shift. Production workload processing activities are listed by minute.

You can either print the current plan as a report, or view, alter, and extend it online, by using the dialogs.

Automatically controlling the production workload

About this task

IBM Workload Scheduler for z/OS automatically drives the production workload by monitoring the flow of work and by directing the processing of jobs to follow the business priorities established in the plan.

Through its interface to the NetView program or its management-by-exception ISPF dialog, IBM Workload Scheduler for z/OS can alert the production control specialist to problems in the production workload processing. Furthermore, the

NetView program can automatically trigger IBM Workload Scheduler for z/OS to perform corrective actions in response to these problems.

IBM Workload Scheduler for z/OS automatically:

- Starts and stops started tasks
- Edits job statements: z/OS JCL or equivalent job statements for other operating environments before submission
- Submits jobs in the specified sequence to the target operating environment every time
- Tracks each scheduled job in the plan
- Determines the success or failure of the jobs
- Displays status information and instructions to guide workstation operators
- Provides automatic recovery of jobs when they end in error, regardless of the operating environment
- Generates processing dates for your job stream run cycles using rules, such as:
 - Every second Tuesday of the month
 - Only the last Saturday in June, July, and August
 - Every third workday in the user-defined PAYROLL period
- Starts jobs with regard to real resource availability
- Performs data set cleanup in error and rerun situations for the z/OS workload
- Tailors the JCL for step restarts of z/OS jobs and started tasks
- Dynamically schedules additional processing in response to unplannable activities
- Provides automatic notification when an updated data set is closed; this can be used to trigger subsequent processing
- Generates alerts when abnormal situations are detected in the workload

IBM Workload Scheduler for z/OS also provides manual control facilities, which are described in “Manual control and intervention” on page 64.

Automatic workload submission

About this task

IBM Workload Scheduler for z/OS automatically drives work through the system, taking into account work that requires manual or program-recorded completion. Program-recorded completion refers to situations where the status of a scheduler-controlled job is set to “complete” by a user-written program. It also promotes the optimum use of resources, improves system availability, and automates complex and repetitive operator tasks. IBM Workload Scheduler for z/OS automatically controls the submission of work according to:

- Dependencies between jobs
- Workload priorities
- Specified time for the submission of particular work
- Availability of resources

By saving a copy of the JCL for each separate run, or occurrence, of a particular job in its plans, IBM Workload Scheduler for z/OS prevents the unintentional reuse of temporary JCL changes, such as overrides.

Job tailoring. IBM Workload Scheduler for z/OS provides automatic job tailoring functions to automatically edit jobs. This can reduce your dependency on time-consuming and error-prone manual editing of jobs. IBM Workload Scheduler for z/OS job tailoring provides:

- Automatic variable substitution
- Dynamic inclusion and exclusion of inline job statements
- Dynamic inclusion of job statements from other libraries or from an exit

For jobs submitted on a z/OS system, these job statements are z/OS JCL. Scheduler JCL tailoring directives can be included in jobs that are submitted on other operating systems, such as AIX®/6000.

Variables can be substituted in specific columns, and you can define verification criteria to ensure that invalid strings are not substituted. Special directives supporting a variety of date formats used by job stream programs let you dynamically define the required format and change the multiple times for the same job. You can define arithmetic expressions to calculate values such as the current date plus four work days. And you can set a temporary variable to a specific value or to an expression composed of other temporary variables.

System Automation commands tailoring. IBM Workload Scheduler for z/OS provides a function that edits system automation commands automatically. This helps you to save time and reduce the possibility of editing errors. IBM Workload Scheduler for z/OS command tailoring provides automatic variable substitution.

Automatic recovery and restart

IBM Workload Scheduler for z/OS provides automatic restart facilities for your production work. You can specify the restart actions to take if work initiated by IBM Workload Scheduler for z/OS ends in error (see Figure 6 on page 61.) You can use these functions to predefine automatic error recovery and restart actions for jobs and started tasks. The scheduler's integration with the NetView program allows it to automatically pass alerts to the NetView program in error situations. Using the z/OS cross-system coupling facility (XCF) enables IBM Workload Scheduler for z/OS to maintain production workload processing when system failures occur.

Recovery of jobs and started tasks. Automatic recovery actions for failed jobs are specified in user-defined control statements. Parameters in these statements determine the recovery actions to be taken when a job or started task ends in error.

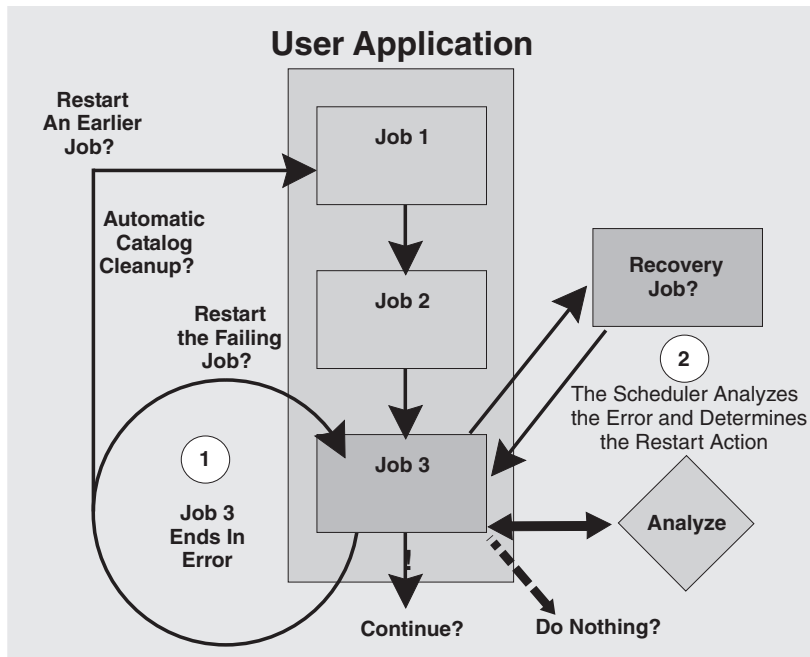


Figure 6. Automatic recovery and restart

Restart and cleanup. You can use restart and cleanup to catalog, uncatalog, or delete data sets when a job ends in error, or when you need to rerun a job. Data set cleanup handles JCL in the form of in-stream JCL, in-stream procedures, and cataloged procedures on both local and remote systems. This function can be initiated automatically by IBM Workload Scheduler for z/OS or manually by using the panels. IBM Workload Scheduler for z/OS resets the catalog to the status that it was before the job ran for both generation data set groups (GDGs) and for DD allocated data sets contained in JCL. In addition, restart and cleanup supports the use of Removable Media Manager in your environment.

Restart at both the step- and job-level is also provided in the IBM Workload Scheduler for z/OS panels. It manages resolution of generation data group (GDG) names, JCL containing nested INCLUDEs or PROC, and IF-THEN-ELSE statements. IBM Workload Scheduler for z/OS also automatically identifies problems that can prevent successful restart, providing a logic of the “best restart step.”

You can browse the job log or request a step-level restart for any z/OS job or started task even when there are no catalog modifications. The job-log browse functions are also available for the workload on other operating platforms, which is especially useful for those environments that do not support an SDSF-like facility. If you use a SYSOUT archiver, for example RMDS, you can interface with it from IBM Workload Scheduler for z/OS and so prevent duplication of job log information.

These facilities are available to you without the need to make changes to your current JCL.

IBM Workload Scheduler for z/OS gives you an enterprise-wide data set cleanup capability on remote agent systems.

Production workload restart. IBM Workload Scheduler for z/OS provides a production workload restart, which can automatically maintain the processing of your work if a system or connection fails. Scheduler-controlled production work for the unsuccessful system is rerouted to another system. Because IBM Workload Scheduler for z/OS can restart and manage the production workload, the integrity of your processing schedule is maintained, and service continues for your customers.

IBM Workload Scheduler for z/OS uses the VTAM® Model Application Program Definition feature and the z/OS defined symbols to ease the configuration and job in a sysplex environment, giving you a single system view of the sysplex.

Starting, stopping, and managing your engines and agents do not require you to know on which sysplex z/OS image they are actually running on.

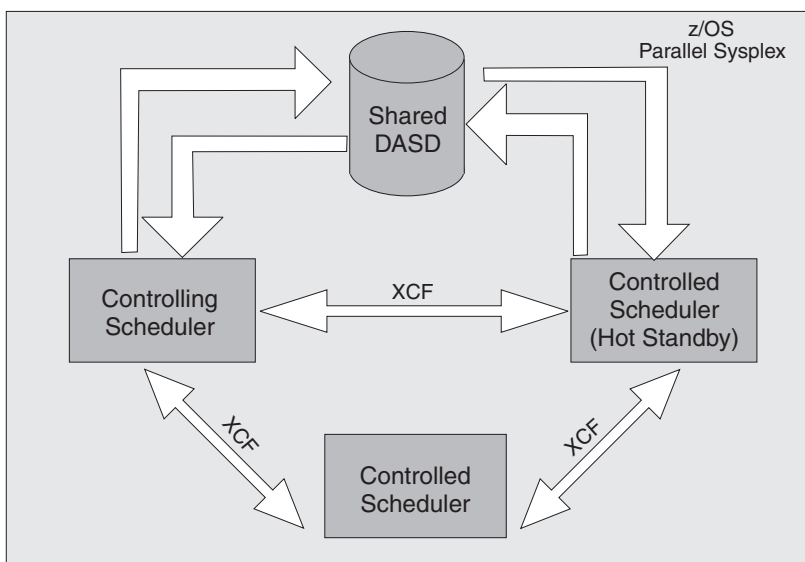


Figure 7. Production workload restart and hot standby

Hot standby. IBM Workload Scheduler for z/OS provides a single point of control for your z/OS production workload. If this controlling system fails, IBM Workload Scheduler for z/OS can automatically transfer the controlling functions to a backup system within a Parallel Sysplex®, see Figure 7. Through XCF, IBM Workload Scheduler for z/OS can automatically maintain production workload processing during system or connection failures.

z/OS automatic restart manager support

About this task

All the scheduler components are enabled to be restarted by the Automatic Restart Manager (ARM) of the z/OS operating system, in the case of program failure.

Workload Manager (WLM) support

About this task

With Workload Manager (WLM), you can make the best use of resources accessed by your scheduled jobs. In addition, your jobs maintain the highest possible throughput with WLM and IBM Workload Scheduler for z/OS. When used with WLM, the scheduler can achieve the best possible system response times.

Automatic status checking

About this task

To track the work flow, IBM Workload Scheduler for z/OS interfaces directly with the operating system, collecting and analyzing status information about the production work that is currently active in the system. IBM Workload Scheduler for z/OS can record status information from both local and remote processors. When status information is reported from remote sites in different time zones, IBM Workload Scheduler for z/OS makes allowances for the time differences.

Status reporting from heterogeneous environments

About this task

The processing on other operating environments can also be tracked by IBM Workload Scheduler for z/OS. You can use supplied programs to communicate with the engine from any environment that can establish communications with a z/OS system.

Status reporting from user programs

About this task

You can pass status information about production workload processing to IBM Workload Scheduler for z/OS from your own user programs through a standard supplied routine.

Additional job-completion checking

About this task

If required, IBM Workload Scheduler for z/OS provides further status checking by scanning SYSOUT and other print data sets from your processing when the success or failure of the processing cannot be determined by completion codes. For example, IBM Workload Scheduler for z/OS can check the text of system messages or messages originating from your user programs. Using information contained in job completion checker (JCC) tables, IBM Workload Scheduler for z/OS determines what actions to take when it finds certain text strings. These actions can include:

- Reporting errors
- Requeuing SYSOUT
- Writing incident records to an incident data set

Managing unplanned work

About this task

IBM Workload Scheduler for z/OS can be automatically triggered to update the current plan with information about work that cannot be planned in advance. This allows IBM Workload Scheduler for z/OS to control unexpected work. Because IBM Workload Scheduler for z/OS checks the processing status of this work, automatic recovery facilities are also available.

Integration with IBM Tivoli Monitoring

IBM Tivoli Monitoring monitors and manages system and network applications on a variety of platforms and can keep track of the availability and performance of your enterprise in its entirety. You can use the reports provided by IBM Tivoli Monitoring to track trends and troubleshoot problems.

The integration between IBM Workload Scheduler for z/OS and IBM Tivoli Monitoring requires the installation of a customizable agent, the IBM Tivoli

Monitoring Agent for IBM Workload Scheduler for z/OS. This agent is installed on the systems that you want to monitor, and it collects and distributes data to the Tivoli Enterprise Monitoring Server.

The IBM Tivoli Monitoring Agent for IBM Workload Scheduler for z/OS (called Tivoli Monitoring agent from now on) sends events and alerts to IBM Tivoli Monitoring (ITM). IBM Workload Scheduler for z/OS establishes an IP connection with the Socket Data Source of the Tivoli Monitoring agent which acts like a listener task. Every time one of the monitored operations changes its status, or an alert is issued, the related event is sent to the Tivoli Monitoring agent.

The data sent by IBM Workload Scheduler for z/OS is interpreted by the Tivoli Monitoring agent. The agent stores the event information in its cache and this information is later consolidated by the Tivoli Enterprise Monitoring Server. The collected data is then retrieved and displayed by a component named Tivoli Enterprise Portal.

The Tivoli Enterprise Portal is an interactive interface used for viewing and monitoring your enterprise network. The Tivoli Enterprise Portal client connects to the Tivoli Enterprise Portal server that enables retrieval, manipulation, and analysis of data collected by all existing IBM Tivoli Monitoring agents in your enterprise.

You can use the Tivoli Enterprise Portal to display and manage events related to IBM Workload Scheduler for z/OS operations such as operation start and end time. In the same way, you can configure the automatic selection of monitored operations and subtasks, providing a single point of management and control for the resources of your enterprise.

Interfacing with other programs

About this task

IBM Workload Scheduler for z/OS provides a program interface (PIF). Using this interface, you can automate most actions that you can perform online through the dialogs. This interface can be called from CLISTs, user programs, and using TSO commands.

The application programming interface (API) lets your programs communicate with IBM Workload Scheduler for z/OS from any compliant platform. You can use Common Programming Interface for Communications (CPI-C), advanced program-to-program communication (APPC), or your own logical unit (LU) 6.2 verbs to converse with IBM Workload Scheduler for z/OS through the API. You can use this interface to query and update the current plan. The programs can be running on any platform that is connected locally, or remotely through a network, with the z/OS system where the engine runs.

Manual control and intervention

About this task

IBM Workload Scheduler for z/OS lets you check the status of work and intervene manually when priorities change or when you want to run unplanned work. You can query the status of the production workload and then modify the schedule if needed.

Status inquiries

About this task

With the ISPF dialogs or the Dynamic Workload Console, you can make queries online and receive timely information on the status of the production workload.

Time information that is displayed by the dialogs is in the local time of the dialog user. Using the dialogs, you can request detailed or summary information on individual job streams, jobs, and workstations, as well as summary information concerning workload production as a whole. You can also display dependencies graphically as a network at both job stream and job level. Status inquiries:

- Provide you with overall status information that you can use when considering a change in workstation capacity or when arranging an extra shift or overtime work.
- Help you supervise the work flow through the installation; for example, by displaying the status of work at each workstation.
- Help you decide whether intervention is required to speed up the processing of specific job streams. You can find out which job streams are the most critical. You can also check the status of any job stream, as well as the plans and actual times for each job.
- Help you to check information before making modifications to the plan. For example, you can check the status of a job stream and its dependencies before deleting it or changing its input arrival time or deadline. See “Modifying the current plan” for more information.
- Provide you with information on the status of processing at a particular workstation. Perhaps work that should have arrived at the workstation has not arrived. Status inquiries can help you locate the work and find out what has happened to it.

Modifying the current plan

About this task

IBM Workload Scheduler for z/OS makes status updates to the plan automatically, using its tracking functions. However, it lets you change the plan manually to reflect unplanned changes to the workload or to the operations environment, which often occur during a shift. For example, you might want to change the priority of a job stream, add unplanned work, or reroute work from one workstation to another. You might also want to correct operational errors manually. Modifying the current plan might be the best way to handle these situations.

You can modify the current plan online. For example, you can:

- Include unexpected jobs or last-minute changes to the plan. IBM Workload Scheduler for z/OS then automatically creates the dependencies for this work.
- Manually modify the status of jobs.
- Delete occurrences of job streams.
- Graphically display job dependencies before you modify them.
- Modify the data in job streams, including the JCL.
- Respond to error situations by:
 - Rerouting jobs
 - Rerunning jobs or occurrences
 - Completing jobs or occurrences
 - Changing jobs or occurrences
- Change the status of workstations by:

- Rerouting work from one workstation to another
- Modifying workstation reporting attributes
- Updating the availability of resources
- Changing the way resources are handled
- Replan or extend the current plan

In addition to using the dialogs, you can modify the current plan from your own job streams using the program interface or the application programming interface. You can also trigger IBM Workload Scheduler for z/OS to dynamically modify the plan using TSO commands or a batch program. This adds unexpected work automatically to the plan.

Management of critical jobs

About this task

IBM Workload Scheduler for z/OS uses the capability of the Workload Manager component of z/OS to ensure that critical jobs are completed on time. If a critical job is late, IBM Workload Scheduler for z/OS favors it using the Workload Manager interface.

Management of critical path

About this task

In addition to the handling of critical jobs based on Workload Manager, IBM Workload Scheduler for z/OS provides the dynamic handling of the critical path calculated by the daily planning batch jobs process.

The critical path is the path, within a network of jobs, with the least slack time.

The slack time, in a critical job predecessor path, is the amount of time that processing of the predecessor jobs can be delayed without exceeding the deadline of a critical job. It is the spare time calculated using the deadline, input arrival, and duration settings of predecessor jobs.

The capabilities include:

- Monitoring of critical job predecessors that are late, long running, or ended with an error. This process uses the same internal logic that the scheduler applies to monitor alert conditions.
- Monitoring of the paths that are consuming their slack time, becoming more critical than the paths calculated at plan generation.
- Enhanced critical jobs monitoring, using ISPF dialog flows.
- Back-end support for new views available using the Dynamic Workload Console.

Security

Today, DP operations increasingly require a high level of data security, particularly as the scope of DP operations expands and more people within the enterprise become involved. IBM Workload Scheduler for z/OS provides complete security and data integrity within the range of its functions. It provides a shared central service to different user departments even when the users are in different companies and countries. IBM Workload Scheduler for z/OS provides a high level of security to protect scheduler data and resources from unauthorized access. With IBM Workload Scheduler for z/OS, you can easily organize, isolate, and protect user data to safeguard the integrity of your user applications (see Figure 8 on page 67

67). IBM Workload Scheduler for z/OS can plan and control the work of many user groups, and maintain complete control of access to data and services.

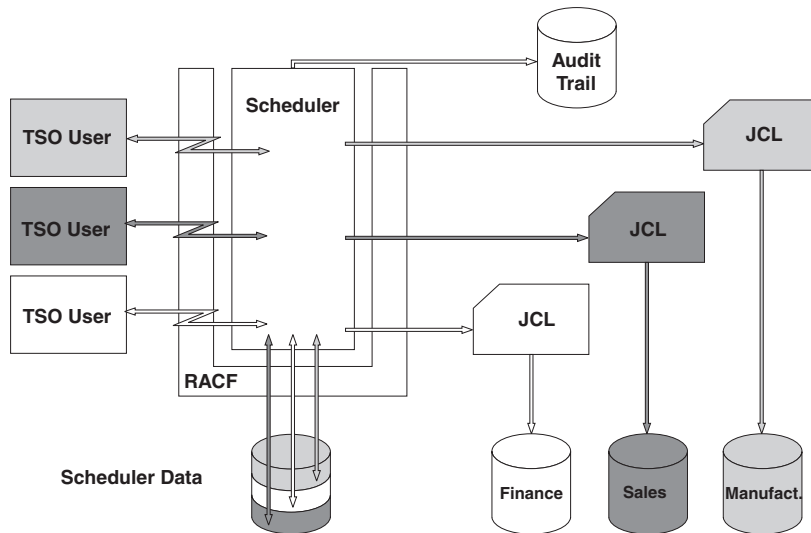


Figure 8. Security

Audit trail

With the audit trail, you can define how you want IBM Workload Scheduler for z/OS to log accesses (both reads and updates) to scheduler resources. Because it provides a history of changes to the databases, the audit trail can be extremely useful for staff that work with debugging and problem determination.

A sample program is provided for reading audit-trail records. The program reads the logs for a period that you specify and produces a report detailing changes that have been made to scheduler resources.

System authorization facility

About this task

IBM Workload Scheduler for z/OS uses the system authorization facility (SAF), a function of z/OS, to pass authorization verification requests to your security system, for example RACF. This means that you can protect your scheduler data objects with any security system that uses the SAF interface.

Protection of data and resources:

About this task

Each user request to access a function or to access data is validated by SAF. This is some of the information that can be protected:

- Calendars and periods
- Job stream names or job stream owner, by name
- Workstation, by name
- Job stream-specific data in the plan
- Operator instructions
- JCL

To support distributed, multi-user handling, IBM Workload Scheduler for z/OS lets you control the level of security you want to implement, right down to the level of individual records. You can define generic or specific RACF resource names to extend the level of security checking.

If you have RACF Version 2 Release 1 installed, you can use the IBM Workload Scheduler for z/OS reserved resource class to manage your IBM Workload Scheduler for z/OS security environment. This means that you do not have to define your own resource class by modifying RACF and restarting your system.

Data integrity during submission:
About this task

IBM Workload Scheduler for z/OS can ensure the correct security environment for each job it submits, regardless of whether the job is run on a local or a remote system. IBM Workload Scheduler for z/OS lets you create tailored security profiles for individual jobs or groups of jobs.

Compliance with the Federal Information Processing Standards (FIPS)

You can configure IBM Workload Scheduler for z/OS to comply with Federal Information Processing Standards (FIPS) standard over SSL secured connections.

You can configure IBM Workload Scheduler for z/OS to activate FIPS 140-2 compliance when scheduling end-to-end with z-centric agents or with fault-tolerant agents, and in communications between the controller and the following interfaces: tracker, server, datastore, remote ISPF dialog, program interface (PIF). To do so, you must configure the parameter named ENABLEFIPS in the TCPOPTS and/or HTTPPTS initialization statements of the IBM Workload Scheduler for z/OS controller.

Compatibility, upgrade, and coexistence

The IBM Workload Scheduler agent configured to work in a cluster environment does not impact compatibility with previous IBM Workload Scheduler versions and does not require configuration or data migration.

An IBM Workload Scheduler agent configured to work in a Windows cluster environment can be connected to both the distributed and the end-to-end network configurations.

The DLL that extends the Windows Cluster Administration program is sometimes updated in fix packs and new releases of IBM Workload Scheduler. For this reason, the program that installs the Windows Cluster Enabler has an update option that you use to update the DLL with a new version, minor (fix pack) or major (new release of IBM Workload Scheduler).

Configurations of IBM Workload Scheduler for z/OS

About this task

IBM Workload Scheduler for z/OS supports many configuration options using a variety of communication methods:

- “The controlling system” on page 69
- “Controlled z/OS systems” on page 70
- “Remote panels and program interface applications” on page 70
- “Scheduling jobs that are in IBM Workload Scheduler” on page 70

The controlling system

The controlling system requires both the agent and the engine. One controlling system can manage the production workload across all your operating environments.

The engine is the focal point of control and information. It contains the controlling functions, the dialogs, and the scheduler's own batch programs. Only one engine is required to control the entire installation, including local and remote systems (see Figure 9).

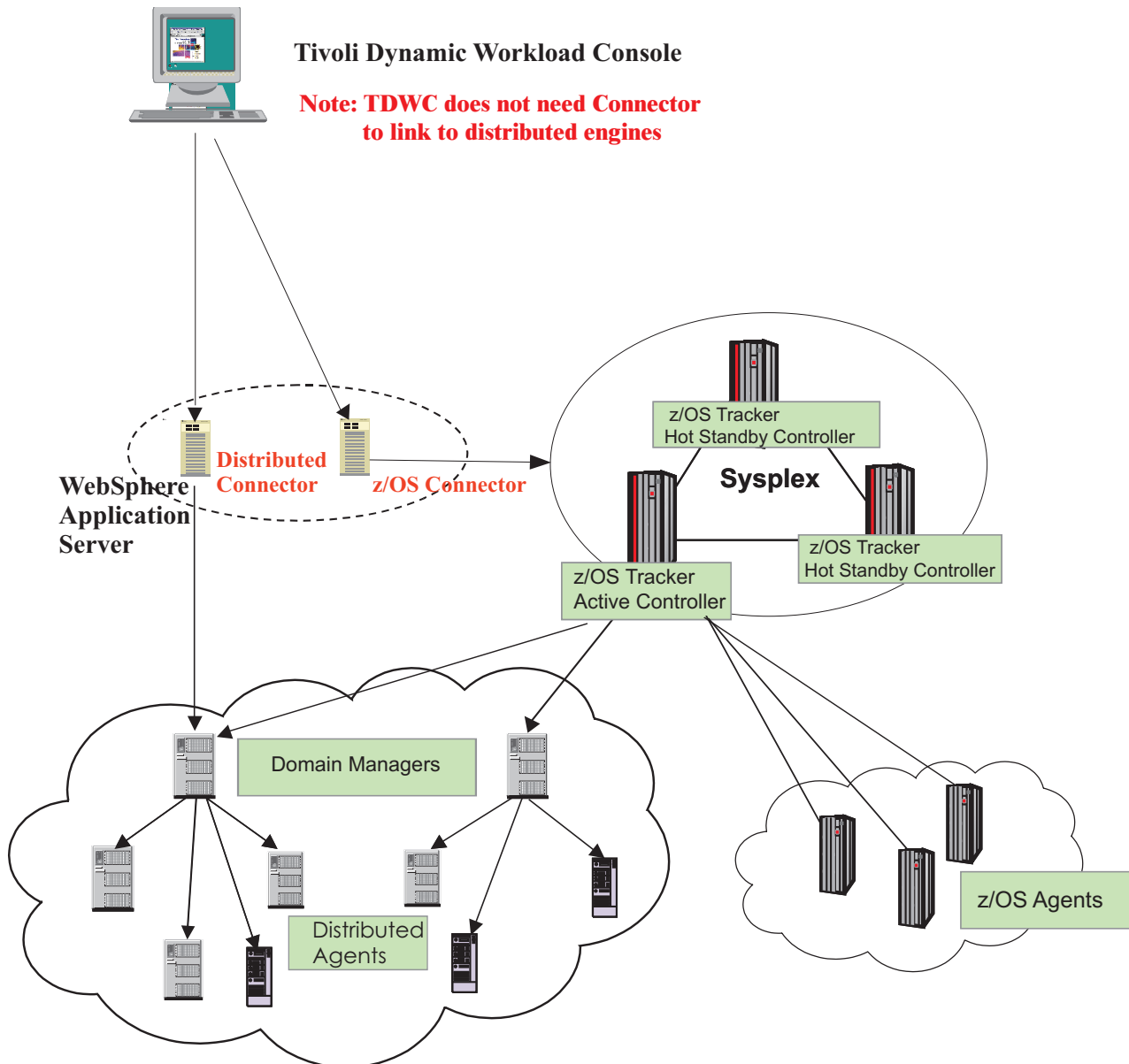


Figure 9. IBM Workload Scheduler for z/OS configurations

Controlled z/OS systems

About this task

An agent is required for every controlled z/OS system in a configuration. This includes, for example, local controlled systems within shared DASD or sysplex configurations.

The agent runs as a z/OS subsystem and interfaces with the operating system (through JES2 or JES3, and SMF), using the subsystem interface and the operating system exits. The agent monitors and logs the status of work, and passes the status information to the engine via shared DASD, XCF, or ACF/VTAM.

You can use z/OS and cross-system coupling facility (XCF) to connect your local z/OS systems. Instead of being passed to the controlling system using shared DASD, work status information is passed directly through XCF connections. XCF lets you use all the production-workload-restart facilities and its hot standby function. See “Automatic recovery and restart” on page 60.

Remote systems

About this task

The agent on a remote z/OS system passes status information about the production work in progress to the engine on the controlling system. All communication between IBM Workload Scheduler for z/OS subsystems on the controlling and remote systems is done through ACF/VTAM.

IBM Workload Scheduler for z/OS lets you link remote systems using ACF/VTAM networks. Remote systems are frequently used locally “on-premises” to reduce the complexity of the data processing (DP) installation.

Remote panels and program interface applications

About this task

ISPF panels and program interface (PIF) applications can run in a different z/OS system from the one where the engine is running. Dialogs and PIF applications send requests to and receive data from an IBM Workload Scheduler for z/OS server that is running on the same z/OS system where the target engine is running, using advanced program-to-program communications (APPC). The server communicates with the engine to perform the requested actions.

The server is a separate address space, started and stopped either automatically by the engine or by the user through the z/OS start command. There can be more than one server for an engine.

If the dialogs or the PIF applications run on the same z/OS system where the target engine is running, the server might not be involved.

Scheduling jobs that are in IBM Workload Scheduler

About this task

IBM Workload Scheduler for z/OS also allows you to access job streams (schedules in IBM Workload Scheduler) and add them to the current plan in IBM Workload Scheduler for z/OS. In addition, you can build dependencies among IBM

Workload Scheduler for z/OS job streams and IBM Workload Scheduler jobs. From IBM Workload Scheduler for z/OS, you can monitor and control the distributed agent.

Using fault-tolerant workstations

In the IBM Workload Scheduler for z/OS current plan, you can specify jobs to run on fault-tolerant agents in IBM Workload Scheduler. IBM Workload Scheduler for z/OS passes the job information to the IBM Workload Scheduler Symphony file, which in turn passes the jobs in the current plan to IBM Workload Scheduler to distribute and process. In turn, IBM Workload Scheduler reports the status of running and completed jobs back to the current plan for monitoring in IBM Workload Scheduler for z/OS.

Using z-centric workstations

z-centric workstations are agents that are installed in a IBM Workload Scheduler network and that can be connected to IBM Workload Scheduler for z/OS by HTTP or HTTPS. They provide the means to schedule from IBM Workload Scheduler for z/OS jobs that need to run on distributed platforms (UNIX, Linux, Windows). They are equivalent to computer automatic workstations in IBM Workload Scheduler for z/OS and require less configuration and a smaller supporting infrastructure than fault-tolerant workstations.

Chapter 7. Dynamic Workload Console

About this task

The Dynamic Workload Console is a Web-based user interface for:

- IBM Workload Scheduler
- IBM Workload Scheduler for z/OS

It is the strategic user interface for the IBM Workload Automation suite of products and includes support for the latest functions and enhancements available with the scheduling engines.

The Dynamic Workload Console is a light, powerful and user-friendly single point of operational control for the entire scheduling network. It allows for single sign-on and authentication to one or many schedulers, is highly scalable, and provides real-time monitoring, management and reporting of enterprise workloads. It also greatly simplifies report creation and customization.

With Dynamic Workload Console you can:

- Manage your workload to design objects in the database, handle plans, submit jobs or job streams, and monitor objects in the plan.
- Design and control the topology of your scheduling environment, that is workstations and domains.
- Define and run reports to gather historical data or details about your plans. You can also generate and run customized SQL reports.
- Define and manage logical resources or groups of logical resources for use with dynamic scheduling.

| A more simplified web-based user interface is available for application developers
| who require control and autonomy when driving workloads, but that are not
| required to handle complex workload automation concepts. The Application Lab
| connects to an existing Dynamic Workload Console distributed engine that is either
| shared or configured in single sign-on. Application developers can build and
| manage simple processes that address real business needs. For more information
| see the *Application Lab User's Guide*.

You can access the Dynamic Workload Console from any computer in your environment using a web browser through both secure HTTPS or HTTP protocol.

The first and main actions you perform when you connect to the Dynamic Workload Console are:

Creating a connection to a scheduling engine (IBM Workload Scheduler or IBM Workload Scheduler for z/OS)

You type the details (such as IP address, user name, and password) to access a scheduling engine, and, optionally, a database to operate with objects defined in plans or stored in the database. You can also define new scheduling objects in the database.

From the Dynamic Workload Console you can access the current plan, a trial plan, a forecast plan, or an archived plan for the distributed environment or the current plan for the z/OS environment.

You might want to access the database to perform actions against objects stored in it or generate reports showing historical or statistical data.

In addition, working both on the database and on plans, you can create and run event rules to define and trigger actions that you want to run in response to events occurring on IBM Workload Scheduler nodes.

Creating tasks to manage scheduling objects in the plan

You specify some filtering criteria to query a list of scheduling objects whose attributes satisfy the criteria you specified. Starting from this list, you can navigate and modify the content of the plan, switching between objects, opening more lists and accessing other plans or other IBM Workload Scheduler or IBM Workload Scheduler for z/OS environments.

The console provides also the following graphical views tools to manage your workload:

Graphical view (for modeling)

A graphical extension to the Workload Designer that shows graphical representations of job stream definitions in the database. It provides an intuitive way to create and maintain them.

Plan view (for monitoring)

A high-level representation of a plan of any type, showing a filtered set of job streams and their mutual dependencies.

Job stream view (for monitoring, troubleshooting, and impact analysis)

An expansible graphical representation of job streams and jobs in the plan. It provides a straightforward, multilevel analysis of how job and job stream completion affects plan progress and provides a direct way to work with jobs, job streams and their dependencies.

Preproduction plan view (for workload planning)

Contains job stream instances and job stream dependencies to be run during a specified time interval covered by the plan.

From each view, you can take actions on objects, view their properties, and easily switch between the views. Graphics can be exported to PNG and SVG files.

Chapter 8. End-to-end scheduling

About this task

By using end-to-end scheduling, you can schedule and control jobs on mainframe, Windows, and UNIX environments, for truly distributed scheduling. In the end-to-end configuration, IBM Workload Scheduler for z/OS is used as the planner for the job scheduling environment. IBM Workload Scheduler domain managers, standard, fault-tolerant, and z-centric agents are used to schedule on the distributed platforms. The agents replace the use of tracker agents.

IBM Workload Scheduler for z/OS also allows you to access job streams (schedules in IBM Workload Scheduler) and add them to the current plan in IBM Workload Scheduler for z/OS. In addition, you can build dependencies among IBM Workload Scheduler for z/OS job streams and IBM Workload Scheduler jobs. From IBM Workload Scheduler for z/OS, you can monitor and control the distributed agents.

You can manage distributed scheduling by activating either of the following features:

- “End-to-end scheduling with fault tolerance capabilities”
- “End-to-end scheduling with z-centric capabilities” on page 77

End-to-end scheduling with fault tolerance capabilities

Learn about End-to-end scheduling with fault tolerance capabilities.

End-to-end scheduling with fault tolerance capabilities directly connects IBM Workload Scheduler standard agents, fault-tolerant agents, and domain managers (with their underlying agents and domains) to IBM Workload Scheduler for z/OS. IBM Workload Scheduler for z/OS is seen by the distributed network as the master domain manager.

IBM Workload Scheduler for z/OS creates the production plan also for the distributed network and sends it to the domain managers and to the directly-connected agents. The domain managers send a copy of the plan to each of their agents and subordinate domain managers for execution.

The IBM Workload Scheduler domain managers function as the broker systems for the distributed network by resolving all dependencies for their subordinate managers and agents. They send their updates (in the form of events) to IBM Workload Scheduler for z/OS so that it can update the plan accordingly. IBM Workload Scheduler for z/OS handles its own jobs and notifies the domain managers of all the status changes of the IBM Workload Scheduler for z/OS jobs that involve the IBM Workload Scheduler plan. In this configuration, the domain managers and all the distributed agents recognize IBM Workload Scheduler for z/OS as the master domain manager and notify it of all the changes occurring in their own plans. At the same time, the agents are not permitted to interfere with the IBM Workload Scheduler for z/OS jobs, because they are viewed as running on the master that is the only node that is in charge of them.

In the IBM Workload Scheduler for z/OS current plan, you can specify jobs to run on workstations in the IBM Workload Scheduler network. IBM Workload Scheduler for z/OS passes the job information to the Symphony file in the IBM Workload Scheduler for z/OS server, which in turn passes the Symphony file to the IBM Workload Scheduler domain managers (DMZ) to distribute and process. In turn, IBM Workload Scheduler reports the status of running and completed jobs back to the current plan for monitoring in the IBM Workload Scheduler for z/OS engine.

Figure 10 on page 77 shows an IBM Workload Scheduler network managed by an IBM Workload Scheduler for z/OS and the flow of data.

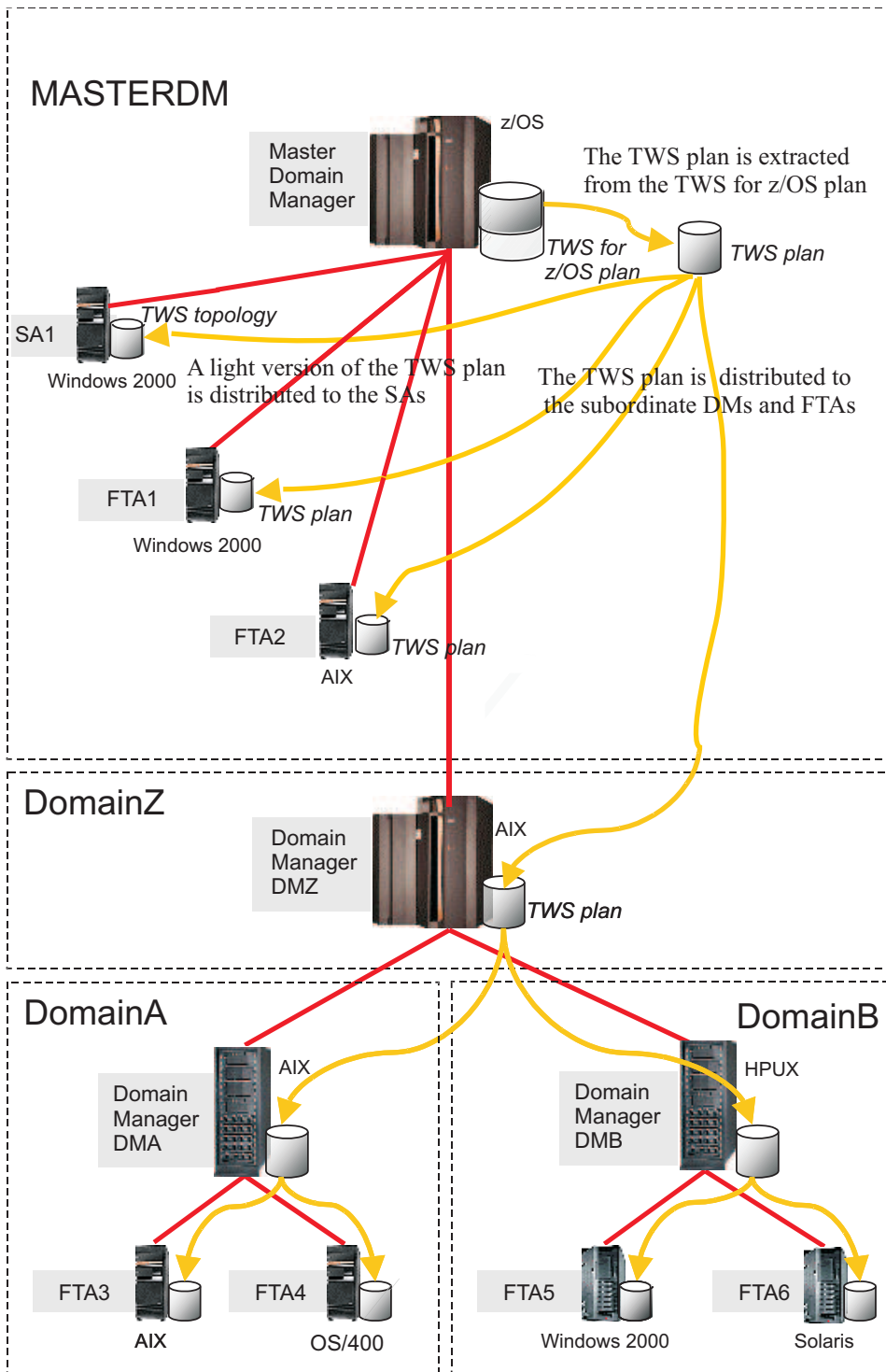


Figure 10. End-to-end with fault tolerance capabilities configuration

End-to-end scheduling with z-centric capabilities

Learn about End-to-end scheduling with z-centric capabilities.

End-to-end scheduling with z-centric capabilities directly connects IBM Workload Scheduler z-centric agents to IBM Workload Scheduler for z/OS, that is the master domain manager for the distributed network.

Powerful mainframe capabilities, such as standard variable substitution, automatic recovery statements and alternate workstation, are supported to manage distributed workload.

Communication between the z-centric agents and IBM Workload Scheduler for z/OS controller is direct, through the HTTP or HTTPS protocol.

Figure 11 shows a network with this configuration.

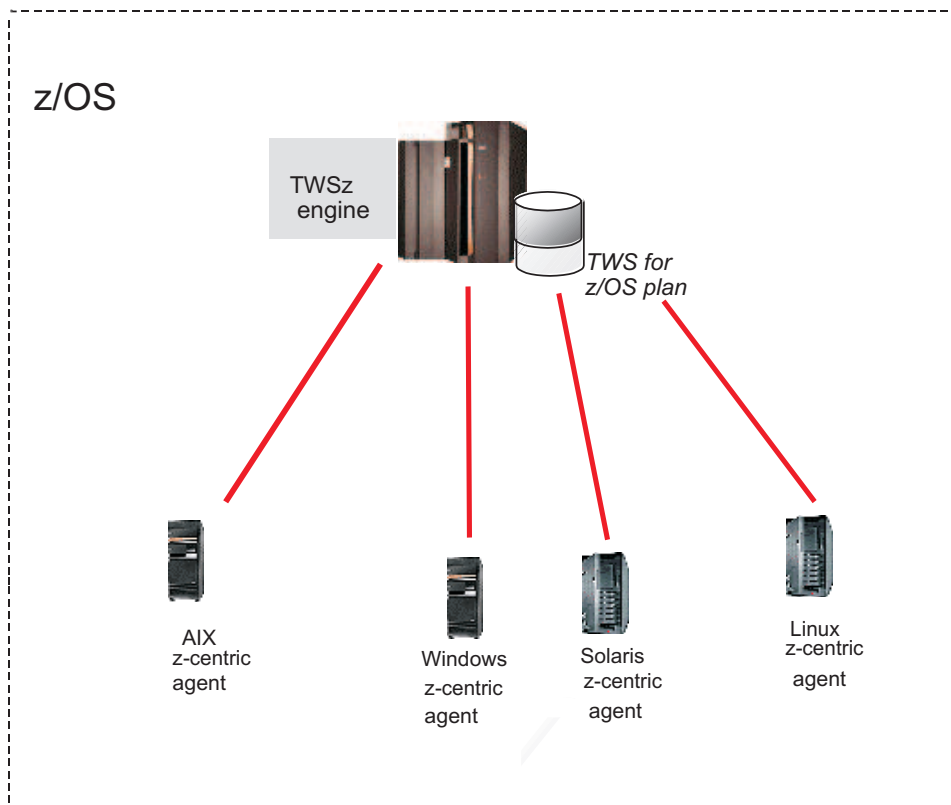


Figure 11. End-to-end with z-centric capabilities configuration

Distributed agents

About this task

A distributed agent is a computer running IBM Workload Scheduler on which you can schedule jobs from IBM Workload Scheduler for z/OS. Examples of distributed agents are the following: standard agents, extended agents, fault-tolerant agents, and domain managers.

The following is a description of the types of distributed agents:

Domain Manager

The management hub in a domain. All communications to and from the agents in a domain are routed through the domain manager.

Backup Domain Manager

A fault-tolerant agent or domain manager capable of assuming the responsibilities of its domain manager for automatic workload recovery.

Fault-tolerant Agent (FTA)

A workstation capable of resolving local dependencies and launching its jobs in the absence of a domain manager.

Standard Agent

A workstation that launches jobs only under the direction of its domain manager.

Extended Agent

A logical workstation definition that helps you launch and control jobs on other systems and applications, such as PeopleSoft, SAP, and z/OS JES2 and JES3.

z-centric Agent

A workstation that runs jobs scheduled from IBM Workload Scheduler for z/OS. The controller directly handles the communication with this type of agent.

Pool

Pools are groups in which you add specific dynamic agent workstations according to the requirements of the job. Jobs are assigned dynamically to the best available dynamic agent applying a load balancing policy by choosing the agent with the lesser amount of jobs to run.

Dynamic pool

Dynamic pools are groups in which you specify the requirements and jobs are assigned dynamically to the best available dynamic agent meeting the requirements and applying an optimization policy to identify the best agent.

Distributed agents replace tracker agents in IBM Workload Scheduler for z/OS. The distributed agents help you schedule on non-z/OS systems with a more reliable and scalable agent.

In the IBM Workload Scheduler for z/OS plan, the logical representation of a distributed agent is called a fault-tolerant workstation or a z-centric workstation.

Benefits of end-to-end scheduling

About this task

The benefits that can be gained from using end-to-end scheduling are the following:

- Connecting either fault-tolerant or z-centric IBM Workload Scheduler agents to IBM Workload Scheduler for z/OS.
- Scheduling on additional operating systems.
- Synchronization of work in mainframe and distributed environments.
- The ability for IBM Workload Scheduler for z/OS to use multi-tier architecture with domain managers.

Chapter 9. Hybrid scheduling environment

Hybrid Workload Automation allows you to use a common User Interface to manage both on premises and on Cloud engines. Manage Workload Automation on Cloud or On premises, or choose a mixture of both. You can monitor and run actions in plan on the following objects: job, critical job, workstation, resource, prompt, job stream, file, and domain.

A solution for you

The same workload efficiently managed in an on premises environment can be scheduled to run in a cloud environment. Workload Automation on Cloud reduces IT infrastructure costs and is a solution for using resources only when you have a business need to run your workload. You can transfer your workload from your on premises environment to your Workload Automation on Cloud by using a simple import procedure. The workload is instantly reproduced and is ready to be employed on your agents. Handling your workload in a cloud environment is a more cost effective way of using resources in the cloud only when you need them without relying on an on premises infrastructure.

How to work with hybrid scheduling environment

To work with the on Cloud engine from the on premises environment, define the engine connection in the Dynamic Workload Console by using port 443. This setting allows you to work with the REST APIs needed for managing Hybrid Workload Automation.

To transfer your on premises workload to the Workload Automation on Cloud environment, complete the following steps:

- Create a workload application template.
- Export the workload application template.
- Import the workload application template into the Workload Automation on Cloud environment.

Create a workload application template containing the job streams that you want to transfer and define some general information about the template, such as a name and description. Select the job streams, each containing one or more jobs, that you want to add to your template. Select the export action and save the template in a compressed file format. The compressed file contains all the files and information required to enable the workload to run in the new cloud environment with some minimal customization to adapt the workload to the cloud environment.

From the Workload Automation on Cloud dashboard, add the template to your on Cloud environment by selecting Import a custom application access point and sign in with your IBM ID. Upload the compressed workload application template file and select the agent on which to import it. You need only adjust two objects to adapt the workload to cloud environment: the prompt definition and the variable table. You must rename both of these objects by prefixing their names with your Workload Automation subscription environment ID. When the import process completes, the workload application is ready to be used in the on Cloud environment.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (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*

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may 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 provide in any way it believes appropriate without incurring any obligation to you.

Licensees 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at 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.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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



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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

ITIL is a Registered Trade Mark of AXELOS Limited.

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

Special characters

(IBM SmartCloud Control Desk (SCCD)) 3

A

access methods 51
accessibility ix
accurate forecasts lvii
advanced program-to-program communication (APPC) 64
advanced recovery options xxvi
advanced rerun flexibility xxvi
advanced rerun options xxvi
advanced statistics lvii
agents
 update xxxiii, xxxiv, xxxv, lv
agents supported in IBM Workload Scheduler 52
alerts, passing to NetView 60
Apache Spark xix
API
 rest xxxvii
API (application programming interface) 64
APPC (advanced program-to-program communication) 64
application
 definition of 54
application deployment
 deployment xxxi
application programmer 9
application programming interface (API) 64
audit-trail facility 67
authority checking 67
auto refresh
 Plan View xvii
Auto-complete
 workstation names and user ID xliv
automatic
 job submission 59
 status checking 63
 status reporting 63
automatic job and started-task recovery 60, 63
Automatic reply to waiting messages for an xliii
automation 5
Automation of Iterative Workflows, Iterative Workflows xxvi
availability 6

B

backup domain manager 33
backup dynamic domain manager 33
backup master 33
backup system 62
backup, tws_env xxxvi

batchman 37
benefits 1, 10
Big data processing
 Apache Spark xix
business processing cycle 57

C

calendar 38
 definition 56
changes
 control xxxvi
check prerequisites xli
CICS 3
Cloud & Smarter Infrastructure technical training ix
Cloudant xxxiii
 database xxxiii
Common Programming Interface for Communications (CPI-C) 64
compatibility
 with previous versions 68
Composer 37
condition automation xxvii
condition scheduling xxvii
Condition-based workload automation xxvii
conditional dependencies xlvi
configurations 69
Conman 37
console operator 9
controlled systems 70
controlling system
 description 69
 recovery of 62
CPI-C (Common Programming Interface for Communications) 64
creating a URL xliv
cross-system coupling facility (XCF) 60, 63, 70
current plan 58
customers, queries from 9

D

Data Facility Hierarchical Storage Manager (DFSHM) 3
database schema
 update xl
 upgrade xl
deadline keyword xxix
Decision Support 3
defining and managing xlii
dependencies
 conditional xlvi
 defining 56
DFSHM (Data Facility Hierarchical Storage Manager) 3
Docker container,
 installing dynamic agents xxix

domain manager 33
dynamic agents
 File dependencies lv
 install using Docker xxix
dynamic domain manager 33
Dynamic Workload Console
 accessibility ix
 check prerequisites xli

E

education ix
ENABLEFIPS
 FIPS data encryption 68
end users, queries from 9
extended agent 34
extended agents 51

F

failure
 restarting Informatica PowerCenter jobs xlv
fault-tolerant agent 33
File dependencies, dynamic agents lv
file dependency 46
filemonitor
 file change monitoring xxviii

G

global options 49
graphical view xxxvii
 enhanced xxiii
Graphical View
 expanding xliv
Graphical View styles
 exporting xliv
 modifying xliv
 on wide monitors xliv

H

Hadoop
 Oozie xliii
helpdesk 9
hybrid xli
hybrid workload automation xli

I

IBM i job xliii
 automatic replying to IBM i messages xliii
 replying to IBM i message xlii
IBM i job from Dynamic Workload Console xlii
IBM Sterling Connect:Direct, Job li

- IBM Tivoli Monitoring
 - default situations xlvi
- IBM Tivoli Monitoring (ITM) 3
- IBM UrbanCode Deploy
 - integration xxxi
- IBM Workload Scheduler 51, 53, 70
 - compatibility
 - with previous versions 68
 - supported agents 52
 - Windows Cluster Enabler
 - prerequisites 68
- IBM Workload Scheduler agents 52
- IBM Workload Scheduler instances xlii
- IBM Workload Scheduler security xlii
- IBM Workload Scheduler/NetView 46
- IBMi xxxvi
 - jobs xxxvi
- IBMi job definition xxxvi
- IMS 3
- Informatica PowerCenter jobs from the
 - point of failure xlv
- initialize xlii
- integration 2
- ISPF (Interactive System Productivity
 - Facility)
 - dialog 58
- ITM (IBM Tivoli Monitoring) 3

J

- Job liii, liv, lvii, lviii
 - IBM Sterling Connect:Direct li
- job completion checker (JCC) 63
- job dependencies 56
- job duration lvii
- Job Management
 - Plug-in xxv
- Job Management,
 - Plug-in xxv
- job recovery
 - automatic 60
 - manual 65
- Job Stream Submission
 - Plug-in xxvii
- Job Stream Submission,
 - Plug-in xxvii
- job streams 70
- job submission
 - automatic 59
 - manual 65
- job tailoring 59
- jobman 37
- jsuntil keyword xxviii

L

- local options 49
- long-term plan 58

M

- mailman 36
- manage security settings
 - Dynamic Workload Console xlv
- Managing Workload Security
 - Dynamic Workload Console xlv

- manual status control 66
- master domain 31
- master domain manager 33
- monitoring the workload 6
- multi-tier architecture 79

N

- national language features 53
- netman 36
- NetView
 - alerts 60
 - description of 2
 - RODM 2
- network Agent 34
- new and improved interface xliii
- new graphical view xxxvii

O

- occurrences 58
- onlate keyword xxix
- Oozie workflow xliii
- OpenIncident
 - integration with ServiceNow xxxvi
- operation dependencies 56
- operations manager 8
- operator, workstation 9
- Output Manager for z/OS 3

P

- parameter 39
- passing variables
 - jobs xxxvii
- passing variables between jobs xxxvii
 - installation on agents xvii
- patch installation
 - on agents xvii
- periods 57
- PIF (program interface) 64
- PIF applications
 - applications 70
- plan
 - current 58
 - definition of 57
 - detailed 58
 - long-term 57
 - modification of 65
 - trial 54
 - types 54
- Plan View
 - auto refresh xvii
 - launching in context xlv
- Plan View in context xlv
- plan view,
 - improved xxiii
- planning
 - trial plans 54
- plug-in
 - Job li
 - Plug-in for Apache Oozie xliii
 - Plug-in for Apache Spark xix
 - Plug-in for Cloudant xxxiii

- prerequisite check
 - master domain manager
 - check prerequisites xli
 - silent installation xli
- production control file 32
- production period 40
- production workload restart 60, 63
- program interface (PIF) 64
- prompt 39
- prompt dependency 46

R

- RACF (Resource Access Control
 - Facility) 3, 67
- recovery 60, 63
- recovery job 42
- recovery prompt 42
- remote dialogs
 - dialogs 70
- Reply to a message for xlii
- Requests
 - Enhancements xv, xx, xxii, xxxi, xxxviii
- Requests for Enhancements xv, xx, xxii, xxxi, xxxviii
- resource 39
- Resource Access Control Facility (RACF) 3, 67
- Resource Object Data Manager (RODM) 2
- REST API xxxvii
- restart 60, 63
- restart management 60, 63
- restarting xlv
- RFEs xv, xx, xxii, xxxi, xxxviii
- RODM (Resource Object Data
 - Manager) 2
- role-based
 - role-based security model IBM
 - Workload Scheduler xlii
- run cycle 38
- run cycle groups 57
- Running a script xliii
- Running a script when a job
 - completes xliii

S

- SA for z/OS Automation Feature 3
- SAF (system authorization facility) 67
- SCCD (IBM SmartCloud Control
 - Desk) 3
- schedule 57
- scheduling manager 7
- security 67
- ServiceNow incident management xxxvi
- shift supervisor 8
- Show Plan View
 - auto refresh xvii
 - silent installation xli
- simulation with trial plans 54
- Single Entry Point xlv
- situations
 - IBM Tivoli Monitoring xlvi

- special resources
 - definition of 56
- standard agent 34
- standard list file 47
- status checking, automatic 63
- status control
 - manual 64, 65
- status inquiries 64
- status reporting
 - automatic 63
 - from heterogeneous environments 63
 - from user programs 63
- step-level restart 61
- support
 - agents xxxiii, xxxiv, xxxv, lv
 - dynamic agents lv
- symphony 32, 41
- syntax diagrams, how to read x
- SYSOUT, checking of 63
- system authorization facility (SAF) 67
- system automation commands
 - tailoring 60
- System Automation for z/OS 3
- System Automation z/OS (SA/zOS) 3
- system failures 60
- systemd
 - initialize IBM Workload Scheduler xlii
- Systems Application Architecture
 - Common Programming Interface for Communications 64

T

- technical training ix
- Tivoli Business Systems Manager 46
- Tivoli Information Management for z/OS 3
- tracker agents 75
- training
 - technical ix
- trial plans 54
- tws_env, backup xxxvi

U

- unplannable work 63
- until keyword xxviii
- update database schema xl
- update, agents xxxiii, xxxiv, xxxv, lv
- upgrade database schema xl
- upgrade, agents xxxiii, xxxiv, xxxv
- upgrading
 - v8.6 instances xxx
- user 39
- user authority checking 67
- utility commands,
 - filemonitor xxviii

V

- variable 39
- variable passing xxxvii
- variable table 39
- version 8.6,
 - upgrading xxx

- version control xxxvi
- version
 - control xxxvi
- virtual workstation
 - definition of 55

W

- what-if
 - z/OS environment xxiv
- What-if Analysis xlvi
- What-If Analysis
 - enhancements xxix
- What-If Analysis,
 - enhancements xxix
- what-if,
 - show impact on critical xxiv
- work submission, automatic 59
- workload application
 - new options xxx
- workload applications
 - new options xxx
- Workload Manager (WLM) 2, 62
- workload monitoring 6
- workload restart 60, 63
- workstation
 - changing the status of 65
 - definition 55
 - operator 9
- workstation class 38
- workstation names and user ID
 - auto-completing xliv
- writer 37

X

- XCF (cross-system coupling facility) 60, 63, 70

Z

- z-centric agent 34



Product Number: 5698-WSH, 5698-T08

Printed in USA