

Course Guide

IBM Workload Scheduler 9.4 Administration

Course code TX318 ERC 1.0



September 2017 edition

NOTICES

This information was developed for products and services offered in the USA.

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
United States of America*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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 supply in any way it believes appropriate without incurring any obligation to you.

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

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines 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.

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.

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.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

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

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.

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.

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

© Copyright International Business Machines Corporation 2017.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

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

Contents

About this course	ix
Course objectives	xi
Audience	xii
Prerequisites	xii
Agenda	xiii
Unit 1 Introducing IBM Workload Scheduler 9.4	1-1
Objectives	1-2
Lesson 1 Using dynamic batch computing	1-3
Workload Automation concepts	1-4
Eliminating batch silos	1-5
Lesson 2 IBM Workload Automation configurations	1-7
IBM Workload Automation configurations	1-8
IBM Workload Automation solutions	1-9
Mainframe-based workload automation	1-10
Distributed-based workload automation	1-12
Workload Automation on Cloud	1-14
Peer-to-peer workload automation	1-15
Lesson 3 Workload Scheduler architecture	1-17
Workload Scheduler topologies	1-18
Using the Dynamic Workload Broker and dynamic scheduling	1-20
Lesson 4 Workload Scheduler agents and workstations	1-21
Types of Workload Scheduler agents and workstations	1-22
Lesson 5 The Workload Scheduler production process	1-26
The production planning process	1-27
Creating the production plan	1-27
Distributing the plan	1-27
Monitoring the plan	1-27
Lesson 6 Workload Scheduler user interfaces	1-29
Workload Scheduler graphical interfaces	1-30
Workload Scheduler command-line interfaces	1-31
Application programming interfaces	1-33
Lesson 7 Using the Dynamic Workload Console	1-35
Tasks that use the Dynamic Workload Console	1-37
Opening the Dynamic Workload Console	1-38
Using the Welcome page	1-39
Lesson 8 Setting up the Dynamic Workload Console	1-41
The engine connection	1-42
Creating and managing engine connections	1-44
Lesson 9 Getting started with the command-line interfaces	1-48
Preparing to use the command-line interfaces	1-49

Using the command-line interfaces	1-52
Getting help with command or composer	1-53
Instructor demonstration	1-55
Exercises	1-56
Review questions	1-57
Review answers	1-58
Unit summary	1-60
Unit 2 Planning the Workload Scheduler implementation.....	2-1
Objectives	2-2
Lesson 1 Planning the Workload Scheduler topology	2-3
Domain hierarchies for fault-tolerant agents	2-4
Dynamic Workload Broker	2-5
Hierarchies for dynamic agent domains	2-7
Using gateway agents	2-8
Lesson 2 Creating workstation and domain definitions	2-9
Workstation definitions attributes	2-10
Workstations attributes requirements	2-14
Creating workstation classes	2-17
Lesson 3 Workload Scheduler networks	2-19
Workload Scheduler component ports	2-20
Lesson 4 Checking hardware and software prerequisites	2-23
Software prerequisites	2-25
Lesson 5 Preparing a database	2-27
Preparing the databases and tables	2-29
Lesson 6 IBM Workload Automation users	2-30
Workload Automation user types	2-31
Users' entry points	2-32
Exercises	2-35
Review questions	2-36
Review answers	2-37
Unit summary	2-38
Unit 3 Installing and configuring Workload Scheduler	3-1
Objectives	3-2
Lesson 1 Preparing for the installation	3-3
Workload Automation related services	3-4
Workload Automation collocation	3-6
Product installation paths	3-7
Installation registries	3-11
Lesson 2 Choosing an installation method	3-13
Using the launchpad	3-15
Using the installation wizard	3-16
Using silent mode	3-17
Using twsinst to install agents	3-19
Installation scenario	3-21
Lesson 3 Configuring Workload Scheduler	3-23
Setting global configuration options	3-24
Auditing and security settings	3-24

Planning and scheduling	3-26
Reporting	3-27
Sending electronic mail	3-27
Lesson 4 Customizing the Dynamic Workload Console	3-29
Setting Dynamic Workload Console configuration options	3-30
Settings that you can customize	3-31
Linking to online job and job stream documentation	3-34
Managing settings per role	3-37
Exercises	3-39
Review questions	3-40
Review answers	3-41
Unit summary	3-42
Unit 4 Upgrading Workload Scheduler	4-1
Objectives	4-2
Lesson 1 Planning and preparing the upgrade	4-3
Choosing top-down or bottom-up network upgrades	4-4
Choosing parallel or direct upgrades	4-6
Lesson 2 Upgrading Workload Scheduler	4-9
Upgrading Workload Scheduler	4-10
Upgrading the master domain manager	4-11
Upgrading agents	4-12
Lesson 3 Maintaining Workload Scheduler	4-13
Applying product fixes	4-14
Maintaining the data	4-16
Maintaining the file system	4-17
Exercises	4-19
Review questions	4-20
Review answers	4-21
Unit summary	4-22
Unit 5 Managing the production cycle	5-1
Objectives	5-2
Lesson 1 Production and preproduction plans	5-3
Production and preproduction plans	5-4
Changing the start of day setting	5-5
Generating the preproduction plan	5-6
Checking preproduction plan status	5-9
Managing the production plan generation process	5-10
Resetting the plans	5-12
Recovering a corrupted plan	5-14
Lesson 2 Automating the production planning cycle	5-15
Scheduling the production planning cycle	5-16
Customizing the FINAL job stream	5-18
Exercises	5-21
Review questions	5-22
Review answers	5-23
Unit summary	5-24

Unit 6 Running agent processes	6-1
Objectives	6-2
Lesson 1 Agent processes and process communication	6-3
Agent processes	6-4
Interprocess communication	6-7
Network communication store and forward	6-9
Lesson 2 Using the console manager	6-11
Using the console manager	6-12
Using conman commands	6-13
Starting, stopping, linking, and unlinking workstations	6-14
Starting and stopping other processes	6-18
Lesson 3 Configuring the job environment	6-23
Configuring the job environment	6-24
Job environment variables	6-26
Exercises	6-29
Review questions	6-30
Review answers	6-31
Unit summary	6-32
Unit 7 Securing Workload Scheduler.....	7-1
Objectives	7-2
Lesson 1 Providing specific access	7-1
User access types	7-2
Access entry points	7-3
Lesson 2 Configuring authentication	7-5
Authentication configuration points	7-6
Configuring authentication for the Dynamic Workload Console and engine connection users	7-7
Configuring the use of Single Sign-On (SSO)	7-11
Lesson 3 Configuring Dynamic Workload Console roles	7-13
Adding users and groups to roles	7-14
Dynamic Workload Console roles: TWSWEBUI	7-15
Dynamic Workload Console roles: Dynamic Workload Broker	7-16
Lesson 4 Configuring user authorization	7-17
Using the Security file	7-19
Configuring the Security file	7-21
Lesson 5 Managing the Security.conf file	7-23
Using the dumpsec and makesec commands	7-24
Configuring the Security file	7-25
Defining in the correct order	7-28
Lesson 6 Using role-based user authorization	7-43
Security objects in the database	7-44
Configuring security roles	7-46
Configuring security domains	7-48
Configuring access control lists	7-49
Lesson 7 Managing role-based user authorization by using the Dynamic Workload Console	7-51
Managing workload security	7-52
Managing security roles	7-52
Managing security domains	7-53

Managing access control lists	7-53
Lesson 8 Managing role-based user authorization by using the composer command	7-55
Managing workload security objects	7-56
Lesson 9 Providing auditing and version controls	7-57
Tracking changes to scheduling objects	7-58
Enabling auditing justification and reporting	7-59
Enabling auditing	7-60
Auditing dynamic workload scheduling	7-61
Exercises	7-63
Review questions	7-64
Review answers	7-65
Unit summary	7-66
Unit 8 Finding and repairing problems	8-1
Unit objectives	8-2
Lesson 1 Built-in troubleshooting features	8-3
Available features	8-4
Lesson 2 Checking workstation status	8-7
Viewing workstation status by using comman	8-8
Displaying workstation status	8-8
Checking the health of fault-tolerant agents	8-10
Alternative workstation views	8-11
Resetting a fault-tolerant agent	8-13
Lesson 3 Managing domains	8-15
Switching domain managers	8-16
Lesson 4 Gathering troubleshooting data	8-19
Finding log files	8-20
Gathering data for IBM support	8-22
Event rules for systems administrators	8-23
Exercises	8-25
Review questions	8-26
Review answers	8-27
Unit summary	8-28

About this course

IBM Training



IBM Workload Scheduler 9.4
Administration

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This 2-day course relates the administrative functions necessary to plan, install, configure, maintain, and troubleshoot an IBM Workload Scheduler 9.4 network.

Training occurs in a classroom environment or online instructor-led facility. Hands-on lab exercises include planning a network, defining workstations and domains, and installing and configuring Workload Scheduler components. The lab environment for this course uses the Red Hat Enterprise Linux operating system.

Details	
Delivery method	Classroom or instructor-led online (ILO)
Course level	ERC 1.0
	This course is an update of TX315: IBM Workload Scheduler Administration ERC1.0
Product and version	IBM Workload Scheduler 9.4 with Fix Pack 1

Details	
Recommended duration	2 days
Skill level	Intermediate / Advanced

Course objectives

In this course, you learn how to perform the following tasks:

- Describe the components and their interactions
- Create and describe agents and workstations
- Plan a Workload Scheduler network topology for your organization
- Install and configure the components
- Upgrade agents to the latest version
- Customize how jobs run
- Manage how users are authenticated and authorized to use Workload Scheduler
- Demonstrate the daily planning functions
- Diagnose and troubleshoot problems with Workload Scheduler

© Copyright IBM Corporation 2017

Course objectives

This course includes an in-depth look at designing an IBM Workload Scheduler network architecture, background processes in Workload Scheduler, workstation and network configuration options, workload production cycle and processing, effective use of Workload Scheduler security, and techniques for identifying and resolving common problems.

Audience

This course is designed for the following audiences:

- New IBM Workload Scheduler administrators who seek proficiency in Workload Scheduler 9.4.
- Existing administrators, installers, and support personnel who are converting software from previous versions to Workload Scheduler 9.4.
- Anyone needing detailed guidance on designing Workload Scheduler networks, managing background process, security, configuration, and backups.

Prerequisites

Before you take this course, make sure that you can use a web browser and the UNIX or Linux shell to run basic commands and browse the file system, or that you complete TX319, IBM Workload Scheduler Operations and Scheduling.

Agenda

This course contains the following units:

- Introducing IBM Workload Scheduler 9.4
- Planning the Workload Scheduler implementation
- Installing and configuring Workload Scheduler
- Upgrading Workload Scheduler
- Managing the production cycle
- Running agent processes
- Securing Workload Scheduler
- Finding and repairing problems

© Copyright IBM Corporation 2017

Agenda

The course contains the following units:

1. [Introducing IBM Workload Scheduler 9.4](#)

In this unit, you learn about the basic functions of IBM Workload Scheduler. You learn about IBM's offerings for workload automation and terminology that is used in the class. You also learn about the basic components that comprise a workload automation offering.

2. [Planning the Workload Scheduler implementation](#)

In this unit, you learn how to gather and apply information about your infrastructure. You can then use this information to build an effective Workload Scheduler network environment. This unit includes lessons about finding hardware and software prerequisites and planning Workload Scheduler networks, including networks with firewalls.

3. [Installing and configuring Workload Scheduler](#)

In this unit, you learn how to plan, prepare, and install Workload Scheduler.

4. [Upgrading Workload Scheduler](#)

Workload Scheduler developers continually work to improve the software in each version. In this unit, you learn how to upgrade and maintain the IBM Workload Scheduler server, agent, and Dynamic Workload Console.

5. [Managing the production cycle](#)

In this unit, you learn how the production planning cycle in Workload Scheduler functions. You learn how to create and extend production plans, how to create trial and forecast plans, and how to recover plans that are not correctly functioning.

6. Running agent processes

In this unit, you learn about the processes on each workstation and how the agents that run on a workstation manage jobs that run there.

7. Securing Workload Scheduler

In this unit, you learn how to secure access to Workload Scheduler. You learn about how to grant or deny access to the objects and how users are authenticated to use Workload Scheduler.

8. Finding and repairing problems

In this unit, you learn how to troubleshoot problems in the Workload Scheduler environment. You learn how to use the console manager interface, where to find log files, and how to resolve common issues.

Unit 1 Introducing IBM Workload Scheduler 9.4

IBM Training



Introducing IBM Workload Scheduler 9.4

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn about the basic functions of IBM Workload Scheduler. You learn about IBM's offerings for workload automation and terminology that is used in the class. You also learn about the basic components that comprise a workload automation offering.

Objectives

In this unit, you learn how to perform the following tasks:

- Describe the concepts, needs, and terminologies of dynamic batch computing, and how batch computing silos are reduced through common infrastructures
- Plan to use a Workload Scheduler solution that fits your requirements:
 - Mainframe-based
 - Distributed-based
 - Peer-to-peer
- Describe typical Workload Scheduler architectures
- Outline Workload Scheduler agent and workstation types
- Set up and log in to the Dynamic Workload Console
- Start using the command line interfaces

© Copyright IBM Corporation 2017

Objectives

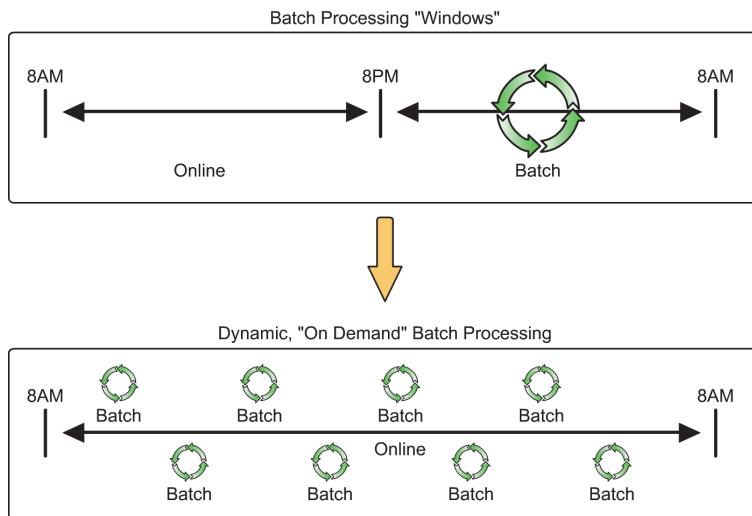
Lesson 1 Using dynamic batch computing

Using dynamic batch computing

© Copyright IBM Corporation 2017

In this lesson, you learn about workload automation and batch computing. You learn about how IBM Workload Automation solutions address the needs of dynamic batch processing.

Workload Automation concepts



1. Unattended work, run repeatedly, such as daily or weekly
2. Integrated with online processing
3. Crosses system and network boundaries
4. Consistently defined and ready to occur
5. Human intervention and error reduced or eliminated

Workload Automation concepts

© Copyright IBM Corporation 2017

Scheduling does not happen in batch windows in current IT enterprises. Data or user-triggered batches of work are more commonplace and occur simultaneously with online transactions.

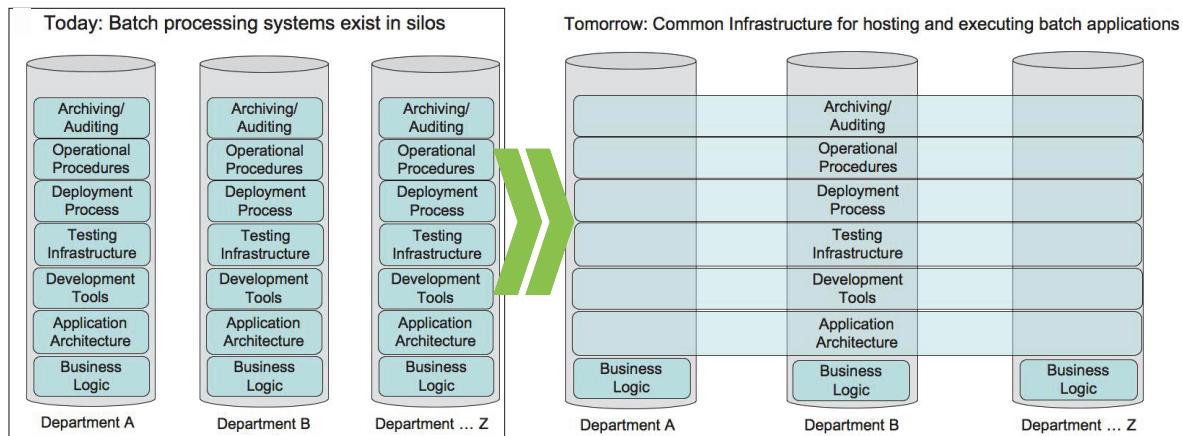
Batch scheduling of workloads is a fundamental asset in large enterprise computing; it is an IT metaphor for bulk business processes. Batch processing volume increases year after year, even with the effects of global technology trends, such as outsourcing and cloud computing.

The pictures on the slide depict traditional batch processing windows, and on-demand batch processing. These windows occurred when online processing ended for the evening, and batch processing took place through the night. On-demand batch processing occurs continuously, often in parallel with online processing.

Batch scheduling addresses economies of scale and efficiencies that are necessary to process large quantities of data. It provides the infrastructure to transform IT projects by reusing and integrating assets. The cost and complexity of maintaining and operating existing batch applications continues to grow, creating demand for improved run times and tools.

Variants of batch processing are emerging that run on new operating systems, infrastructure, and middleware. Batch scheduling must become more elastic. With this flexibility, you can implement newer or modified business processes, consolidate IT systems, and adopt a model for a specific purpose.

Eliminating batch silos



© Copyright IBM Corporation 2017

Eliminating batch silos

As new applications are deployed, independent application architectures and scheduling code cause batch silos. With a unified batch architecture, such as IBM Workload Automation, you can transcend applications and operating systems to minimize the effect of development, operations, and maintenance costs.

An adequate Workload Automation end-to-end configuration can accommodate any batch movement. As an example, clients can move batch from mainframe to distributed systems and keep the control on the mainframe.

In any configuration, you use a single point of monitoring, management, and reporting by using a web-based user interface, called the Dynamic Workload Console. IBM Workload Automation can absorb and integrate existing workloads with any new type of workload that provides APIs or commands.

Lesson 2 IBM Workload Automation configurations

IBM Training

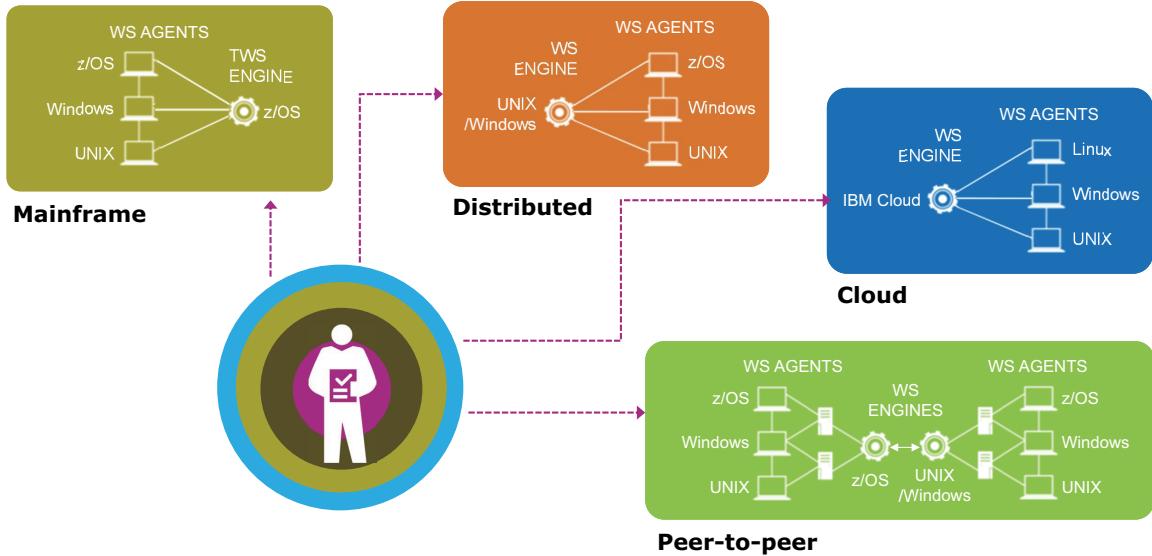


IBM Workload Automation configurations

© Copyright IBM Corporation 2017

In this lesson, you learn about IBM Workload Automation configurations.

IBM Workload Automation configurations

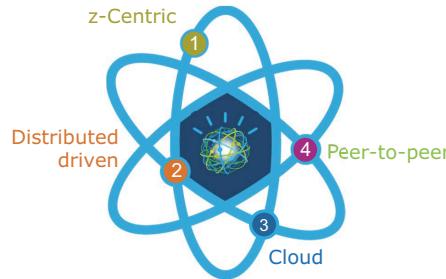


© Copyright IBM Corporation 2017

IBM Workload Automation configurations

IBM Workload Scheduler is part of a suite of workload automation products. You can use this suite for product visibility, integrated management, and maximized productivity; you can manage the workloads that span your enterprise. IBM Workload Scheduler, IBM Workload Automation on Cloud, and Workload Scheduler for z/OS can be used in different configurations, as shown on the slide.

IBM Workload Automation solutions



Mainframe

- z/OS workload environment with graphical user interface
- End-to-end workload environments, with static, dynamic, or both types of scheduling

Distributed

- Distributed workload environment with static, dynamic, or both types of scheduling
- End-to-end workload environments with static, dynamic, or both types of scheduling

Cloud

- Distributed workload environment hosted on the IBM cloud
- Local agents with dynamic scheduling

Peer-to-peer

- Mainframe and distributed workload environments collaborating for end-to-end workloads

© Copyright IBM Corporation 2017

IBM Workload Automation solutions

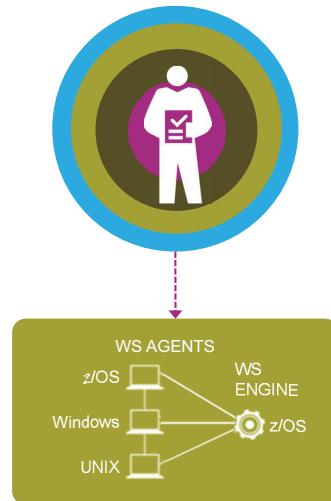
IBM Workload Automation solutions provide configurations in four categories, as described in the next sections.

- Mainframe-based workload automation
- Distributed-based workload automation
- Workload automation on cloud
- Peer-to-peer based workload automation

Mainframe-based workload automation

Controlled by IBM Workload Scheduler for z/OS

- z/OS workload environment with graphical user interface
 - Workloads running on the mainframe
 - ISPF and web-based user interfaces
- End-to-end workload environments, with static, dynamic, or both types of scheduling
 - Connected agents run workloads on distributed platforms
 - Fault-tolerant capabilities
 - Lightweight agents
 - Dynamic scheduling with dynamic domain managers



© Copyright IBM Corporation 2017

Mainframe-based workload automation

IBM Workload Scheduler for z/OS is the workload automation engine for scheduling in mainframe-based environments. Workload Scheduler for z/OS is used in several configurations.

z/OS workload environment with graphical user interface

Use this configuration on mainframe systems where the workloads run on the mainframe and are typically isolated from workloads that run elsewhere. IBM Workload Scheduler for z/OS controls scheduled and on-demand workloads that run on the mainframe in the z/OS environment. The Dynamic Workload Console provides a web-based graphical interface to the scheduling environment, in addition to the traditional ISPF panel-based interface.

End-to-end workload environments

End-to-end configurations use the IBM Workload Scheduler distributed fault-tolerant agents, dynamic agents, and domain managers, which are connected to IBM Workload Scheduler for z/OS controllers. In these scenarios, the mainframe computer runs the scheduling configuration and distributes workload plans. An end-to-end configuration is useful when most batch workloads are scheduled on the mainframe. This configuration is also useful where most of the scheduling skills and knowledge belong to mainframe-oriented staff. The following end-to-end workload environments are typical:

- End-to-end static workload environments with fault-tolerant scheduling capabilities

You use fault-tolerant agents that connect to the Workload Scheduler for z/OS controller to manage workloads that are running on distributed systems. Static workloads that are always run on the same computer.

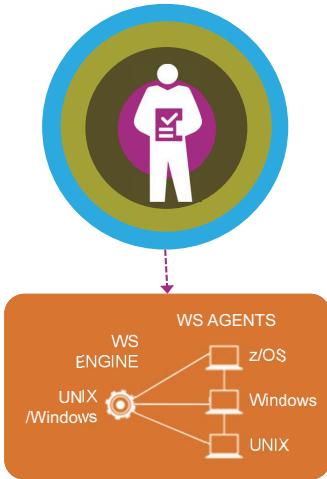
- End-to-end static workload environments with the IBM Workload Scheduler for z/OS agent

You use the dynamic agents that connect to the Workload Scheduler for z/OS controller for managing distributed workloads.

- End-to-end static and dynamic workload environment with the IBM Workload Scheduler for z/OS agent

You use the dynamic agent for managing distributed workloads and add dynamic scheduling capabilities for managing pools of dynamic agents. Dynamic workloads can run on different computers each time they are scheduled.

Distributed-based workload automation



Controlled by IBM Workload Scheduler

- Distributed workload environment with static, dynamic, or both types of scheduling
 - UNIX, Linux, and Windows-based workloads
 - Command-line and web-based user interfaces
 - Static or dynamic scheduling
- End-to-end workload environments with static, dynamic, or both types of scheduling
 - UNIX, Linux, and Windows-based workloads
 - Use the distributed agent for z/OS to create JCL-based jobs in Workload Scheduler that JES2 processes

© Copyright IBM Corporation 2017

Distributed-based workload automation

IBM Workload Scheduler is the workload automation engine for scheduling in UNIX and Windows environments. Workload Scheduler is used in several configurations.

Distributed workload environment with static scheduling

This configuration is for running workload statically across your distributed network. Static workloads run on the same computer, as defined in the workstation configuration.

Distributed workload environment with dynamic scheduling

Use this configuration to run workload dynamically across your distributed network. Dynamic workloads do not necessarily always run on the same computer, even though they are scheduled to run on a single IBM Workload Scheduler workstation.

You can choose to also add the runtime environment for Java jobs to the agent. With the runtime environment, you can run advanced job types that are supplied with IBM Workload Scheduler. These job types include file transfer, database (SQL), or web services jobs. Extra job types can be implemented through provided or custom job type plug-ins.

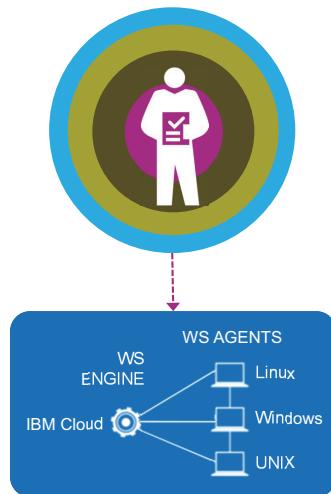
Distributed workload environment with static and dynamic scheduling

This configuration combines the static and dynamic scheduling capabilities into a single environment. With this configuration, you can schedule both static and dynamic workloads, and share most of the installed components on each computer.

Distributed agent for z/OS

You can schedule workloads to run on z/OS by using the IBM Workload Scheduler distributed agent for z/OS. Use this configuration to submit workloads that JES2 processes, without defining the workload directly on the z/OS system. This scenario can be combined with other distributed scenarios.

Workload Automation on Cloud



Controlled by IBM Workload Automation on Cloud

- Distributed workload environment with dynamic scheduling
 - UNIX, Linux, and Windows-based workloads
 - Use the agent for z/OS to create JCL-based jobs in Workload Automation that JES2 processes
- Virtually no cost of ownership for the central server and its backup

© Copyright IBM Corporation 2017

Workload Automation on Cloud

IBM Workload Automation on Cloud is the cloud-based workload automation engine for scheduling in Linux, UNIX, mainframe, and Windows environments. In Workload Automation on Cloud, the Workload Scheduler engine is hosted in the cloud, with virtually no cost of ownership for your central server and its backup.

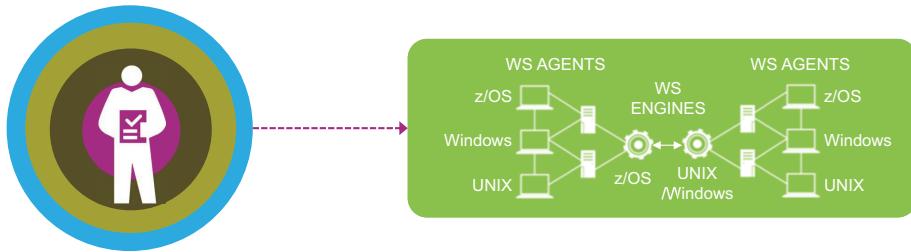
Cloud-based workload environment with dynamic scheduling

With Workload Automation on Cloud, you use a central server that is hosted on the IBM cloud to drive workloads on local agents that you install and configure.

Use this configuration to run workload dynamically across your distributed network. Dynamic workloads do not necessarily always run on the same computer, even though they are scheduled to run on a single IBM Workload Scheduler workstation.

You can choose to also add the runtime environment for Java jobs to the agent. With the runtime environment, you can run advanced job types that are supplied with IBM Workload Scheduler. These job types include file transfer, database (SQL), or web services jobs. Extra job types can be implemented through provided or custom job type plug-ins.

Peer-to-peer workload automation



Use Workload Scheduler and Workload Scheduler for z/OS together

- Mainframe and distributed workload environments collaborating for end-to-end workloads
- Use a single web-based user interface to manage combined environments
- Use shadow jobs to schedule workloads that cross environments, platforms, and time zones

© Copyright IBM Corporation 2017

Peer-to-peer workload automation

You can use Workload Scheduler and Workload Scheduler for z/OS in a peer-to-peer configuration. You can schedule workloads that cross environments, operating systems, and time zones by using two or more Workload Scheduler networks in combinations that fit your requirements. By using a shadow job, you can design workload in a Workload Scheduler environment that waits for the completion of successful work in separate Workload Scheduler environment, even if it is managed on a different operating system.

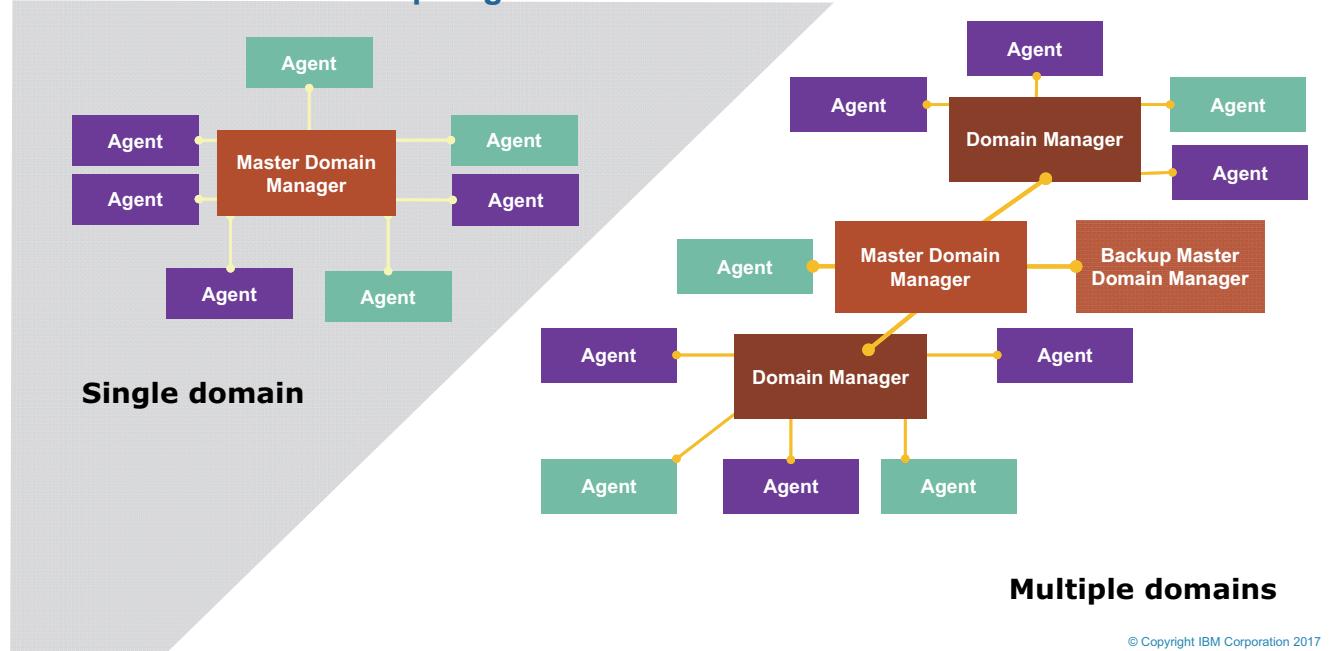
Lesson 3 Workload Scheduler architecture

Workload Scheduler architecture

© Copyright IBM Corporation 2017

A *Workload Scheduler network* has workstations, where tasks run. A Workload Scheduler network has at least one domain: the *master domain*. The *master domain manager* is the management hub and connects to the database. You define new database objects from the master domain manager. Workload Scheduler domains concentrate processing based on a common set of characteristics, such as location.

Workload Scheduler topologies



© Copyright IBM Corporation 2017

Workload Scheduler topologies

You can group the Workload Scheduler network into domains that share common characteristics. Common characteristics are things such as geographical locations, business functions, and application groupings.

In a single-domain configuration, the master domain manager communicates with the computers (running agents) in the Workload Scheduler network. You can use more domains to divide a widely distributed network into smaller, locally managed groups.

In a multiple-domain configuration, the master domain manager communicates with the workstations in its domain and the subordinate domain managers. These subordinate domain managers then communicate with workstations in their domains and their subordinate domain managers.

Multiple domains reduce network traffic by reducing communications between the master domain manager and other computers in the network. Multiple domains also provide fault tolerance by limiting the problems that are caused by losing a domain manager to a single domain. To limit the effects further, you can designate a backup domain manager to communicate with the computers and subordinate domain managers if a domain manager fails.

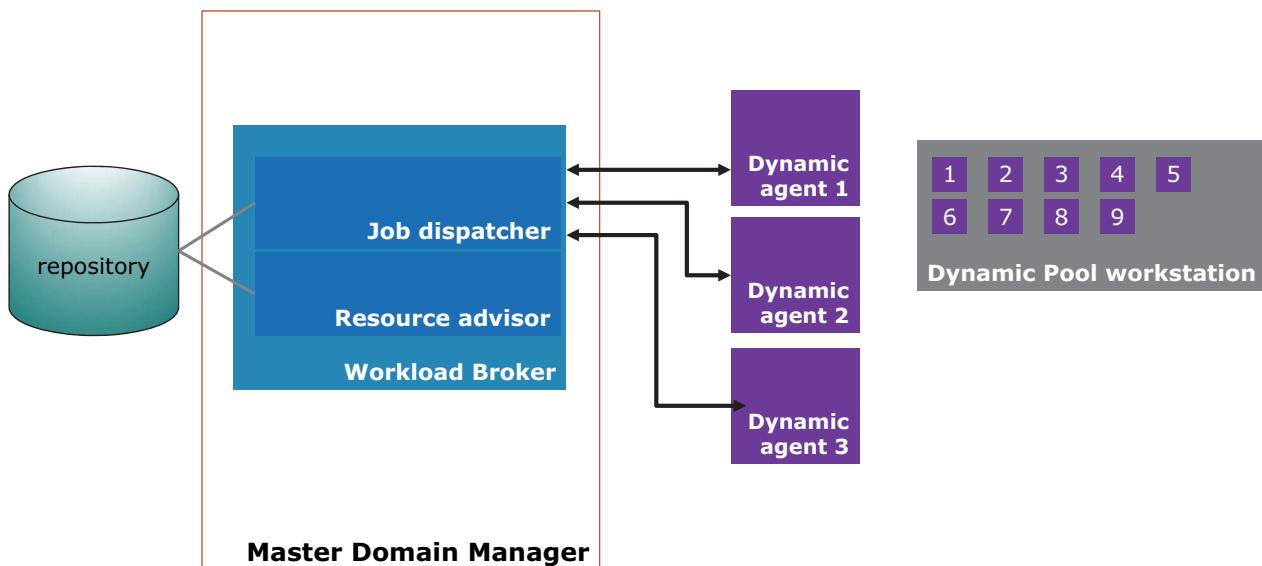
Grouping related processing can limit the amount of interdependency information that must be communicated between domains. The benefits of concentrating processing in domains are listed.

- **Decreased network traffic.** Keeping processing concentrated to domains eliminates the need for frequent interdomain communications.
- **Better security.** Domains provide 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.

- **Optimized fault tolerance.** Network and workstation fault tolerance can be optimized. In a multiple domain Workload Scheduler network, you can define backups for each domain manager so that problems in one domain do not disrupt operations in other domains.

Using the Dynamic Workload Broker and dynamic scheduling



© Copyright IBM Corporation 2017

Using the Dynamic Workload Broker and dynamic scheduling

You can choose to set Workload Scheduler to dynamically associate your submitted workload or part of it to the best available resources at run time. These characteristics provide high availability and load balancing potential to your environment, and are suited to virtualized environments. To schedule workload dynamically, you perform these tasks:

- Use the *Dynamic Workload Console* to define pool or dynamic pool workstations you want to use for running workload as logical resources or groups of resources. For example, you can create a dynamic pool workstation to represent all of the Linux agents in a load-balanced mode. When a job runs on this workstation, the dynamic workload broker chooses the least currently used agent that runs Linux.
- Create Workload Scheduler job definitions that use the pool or dynamic pool workstations that are defined before as their destinations.

When a job is thus submitted, either as part of a job stream in the plan or through ad hoc submission, the *Dynamic Workload Broker* checks the job requirements. The broker compares the available resources and the related characteristics and submits the job to the resource that belongs to the agent that best matches the requirements.

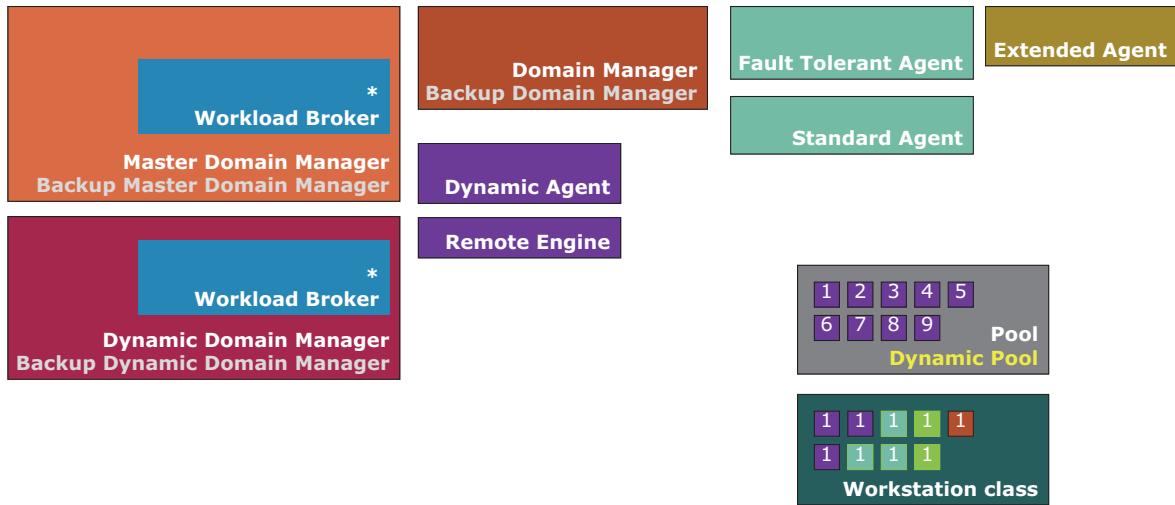
Lesson 4 Workload Scheduler agents and workstations

Workload Scheduler agents and workstations

© Copyright IBM Corporation 2017

You install IBM Workload Scheduler *agents* on the computer systems where you run jobs. You then create *workstation* definitions in the IBM Workload Scheduler database. Workstation definitions control where and how jobs run on the computer systems where agents are running. Workstations can be grouped logically into workstation classes and organized hierarchically into domains, managed by domain managers. In this lesson, you learn about the types of workstations and how they are used in your IBM Workload Scheduler network.

Types of Workload Scheduler agents and workstations



© Copyright IBM Corporation 2017

Types of Workload Scheduler agents and workstations

Agents

You install software agents that run jobs and automate work flow administration tasks. The primary categories of agents are as follows:

- A **fault-tolerant agent** resolves local dependencies and starts jobs in the absence of a domain manager. Fault-tolerant agents receive a copy of the production plan and run jobs locally. If communication problems with its domain manager exist, a fault-tolerant agent can run jobs without a remote connection.
- A **dynamic agent** manages a wide variety of job types, for example, specific database or FTP jobs, in addition to other job types. Additionally, dynamic agents periodically update the workload broker with updated monitored data. You can group dynamic agents in pools and dynamic pools. The Dynamic Workload Broker manages dependencies for dynamic agents; they do not receive copies of the production plan.
- An **extended agent** is a logical definition that you can use to start and control jobs on other systems and applications. Extended agents run jobs under the control of a hosting fault-tolerant agent or dynamic agent by using an access method script or program.

Workstations

A workstation definition specifies the characteristics that uniquely identify the system and how it controls how jobs that are run on it. The following types of workstations are available:

- The **master domain manager** (MASTER or MDM) is the management hub for the IBM Workload Scheduler network. It manages all of the scheduling objects, connects to the database, and produces the daily production plan for all the workstations in the network. One master domain manager is required in each Workload Scheduler network.
- A **backup master domain manager** (BKM) acts as a backup for the master domain manager when problems occur. It receives a copy of workload status so that it can be promoted to master domain manager if needed. A backup master domain manager is optional. The backup domain manager is installed like the master domain manager.
- A **domain manager** (DM) controls a part of the Workload Scheduler network, which is known as a domain, and shares management responsibilities for part of the network. Domain managers are installed as fault-tolerant agents and then configured as domain managers in the IBM Workload Scheduler database.
- A **backup domain manager** (BDM) is a backup for the domain manager when problems occur. Backup domain managers are optional.
- A **dynamic domain manager** (DDM) adds dynamic workload brokering capabilities to a domain manager. When a dynamic domain manager is installed, all the communication to and from dynamic agents in the domain are routed through the dynamic domain manager.
- The **backup dynamic domain manager** acts as a backup for the dynamic domain manager when problems occur. Its use is optional.
- The **fault-tolerant agent** (FTA) is a workstation that receives and runs jobs. If communication problems with the domain manager exist, the fault-tolerant agent can continue to run jobs locally.
- An **extended agent** (X-AGENT) is used to extend the job scheduling functions of IBM Workload Scheduler to other systems and applications.
- The **workload broker** workstation (BROKER) manages jobs that are defined within the Dynamic Workload Broker environment. It is installed as part of the Dynamic Workload Broker feature within IBM Workload Scheduler.
- The **agent** (AGENT), also called dynamic agent, manages various job types. The agent manages database, FTP, web services, and Java Platform, Enterprise Edition jobs, in addition to scripts and commands that the operating system shell runs. IBM Workload Scheduler agent for z/OS runs on mainframe computers that run z/OS. When you create jobs for the agent workstation for z/OS, you include the JCL that the job runs, or name the data set and member that includes the JCL to reference. IBM Workload Scheduler agent for IBM i runs on Power Systems (including AS/400, iSeries, and System i) computers that run IBM i.
- A **pool** workstation (POOL) groups sets of dynamic agents into one workstation. Workload Scheduler balances the number of jobs that run on each agent within the pool. When an agent

in the pool becomes disabled, IBM Workload Scheduler automatically runs jobs on other agents in the pool.

- A **dynamic pool** (D-POOL) workstation is similar to a pool workstation. The workstation grouping for dynamic pools is defined dynamically based on resource requirements that you select in the workstation definition.
- A **workstation class** is a workstation definition that defines replication. When you define a workstation class, you add other workstations as members of the class. Then, you define jobs and job streams for the workstation class. IBM Workload Scheduler replicates the jobs and job streams that you created for the workstation class to each member of the class.
- **Remote engine** (REM-ENG) workstations represent a remote IBM Workload Scheduler engine, like the one you create in the Dynamic Workload Console. You use the remote engine workstation to run shadow jobs. A shadow job runs locally and maps to another job that runs on a remote engine.

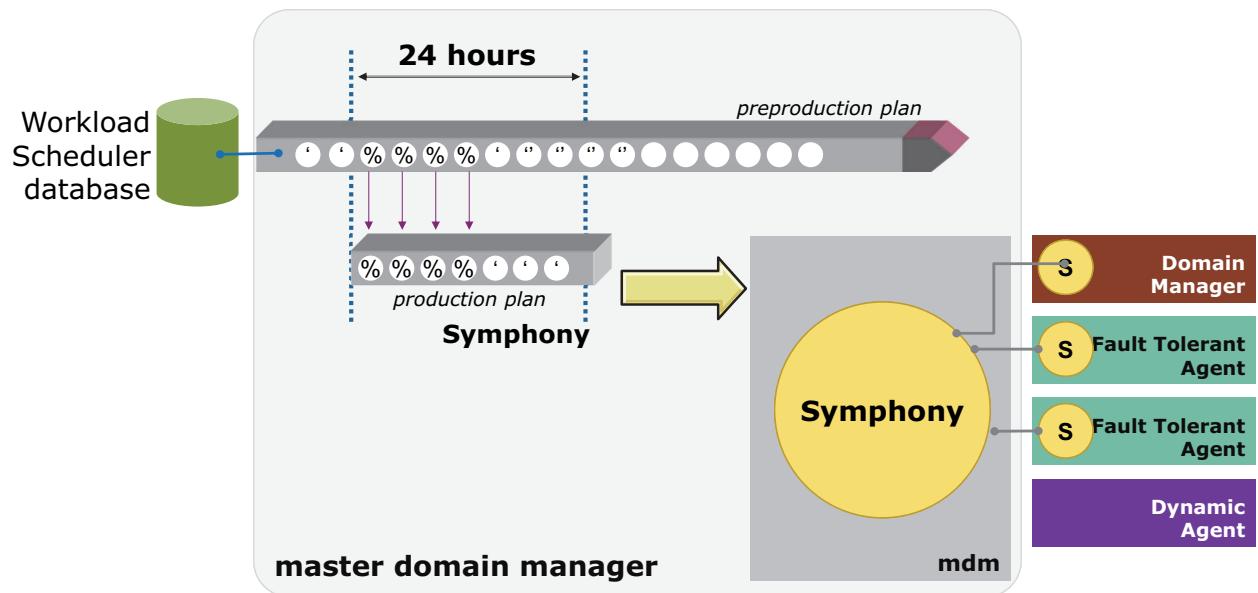
Lesson 5 The Workload Scheduler production process

The Workload Scheduler production process

© Copyright IBM Corporation 2017

IBM Workload Scheduler production is based on a plan that runs in a production period. The production period can span from a few hours to several days; by default it is 24 hours. Before the start of each production period, Workload Scheduler runs a program that creates the production plan from the modeling data that is stored in the database and from an intermediate plan that is called the preproduction plan. Then, another program includes the job streams that did not complete from the previous production period into the current production plan, and records all the statistics of the previous plan into an archive.

The production planning process



© Copyright IBM Corporation 2017

The production planning process

Creating the production plan

All of the information for that production period is placed into a *production control file* that is 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 completed work.

Distributing the plan

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. The distribution allows fault-tolerant agents throughout the network to continue processing even if the network connection to their domain manager is down. Dynamic agents do not receive copies of the Symphony file. From the graphical interfaces or the command-line interface, the operator can view and change the current production by changing the plan.

Monitoring the plan

Workload Scheduler processes monitor the Symphony file and make calls to the operating system to start jobs as needed. The operating system runs the job, and in return informs Workload Scheduler if the job completed successfully or not. This information is entered into the Symphony

Lesson 5 The Workload Scheduler production process

file to indicate the status of the job. Workload Scheduler mirrors the contents of the Symphony file into the Workload Scheduler database.

Lesson 6 Workload Scheduler user interfaces

Workload Scheduler user interfaces

© Copyright IBM Corporation 2017

In this lesson, you learn about the user interfaces that are available with IBM Workload Scheduler. The types of interfaces that are provided are as follows:

- Web-based, graphical user interfaces
- Command-line, text-oriented interfaces
- Application programming interfaces

Workload Scheduler graphical user interfaces

1. Dynamic Workload Console

- Dashboard
- Workload designer
- What-if
- Plan view
- Reports

2. Application Lab

3. Self-Service Catalog

4. Mobile Dashboard

Screenshot 1: Dynamic Workload Console

The screenshot shows the Dynamic Workload Console interface. It features a top navigation bar with links for Default, Administration, Planning, Reporting, System Configuration, System Status and Health, and Troubleshooting and Support. Below the navigation is a "Hello range, welcome to the Dynamic Workload Console" message. A "Take the tour" button is highlighted with a yellow circle labeled "1". To its right is a "Quick start" section with a "Set your preferences" button. Further down are sections for "Manage settings", "Connect your engines", "Define your workstations", "Design your workload", and "Monitor".

Screenshot 2: Application Lab

This screenshot shows the Application Lab interface. It has a left sidebar titled "PROCESS LIBRARY" containing a tree view of processes under "MSP". A specific node named "Syncrono" is selected and expanded, showing its details. The main area displays tabs for "General", "Triggers", "Variables", "Steps", and "History". A table at the bottom lists "AGENTS" with columns for Name, Operating System, Status, Start Date, Stage Completed, Total Steps, and Elapsed Time.

Screenshot 3: Self-Service Catalog

The Self-Service Catalog interface shows a "First catalog" entry. It includes a search bar and a list of items. A yellow circle labeled "3" highlights the catalog title.

Screenshot 4: Mobile Dashboard

The Mobile Dashboard provides a summary of job status. It includes a pie chart showing the distribution of successful, waiting, running, and ready jobs, and a table below with detailed counts for each category. The table also includes columns for Job, Workstation, and Critical status.

© Copyright IBM Corporation 2017

Workload Scheduler graphical interfaces

IBM Workload Scheduler users four unique web-based user interfaces. They are built on the *IBM Jazz for Service Management* platform, and *Dashboard Application Services Hub*.

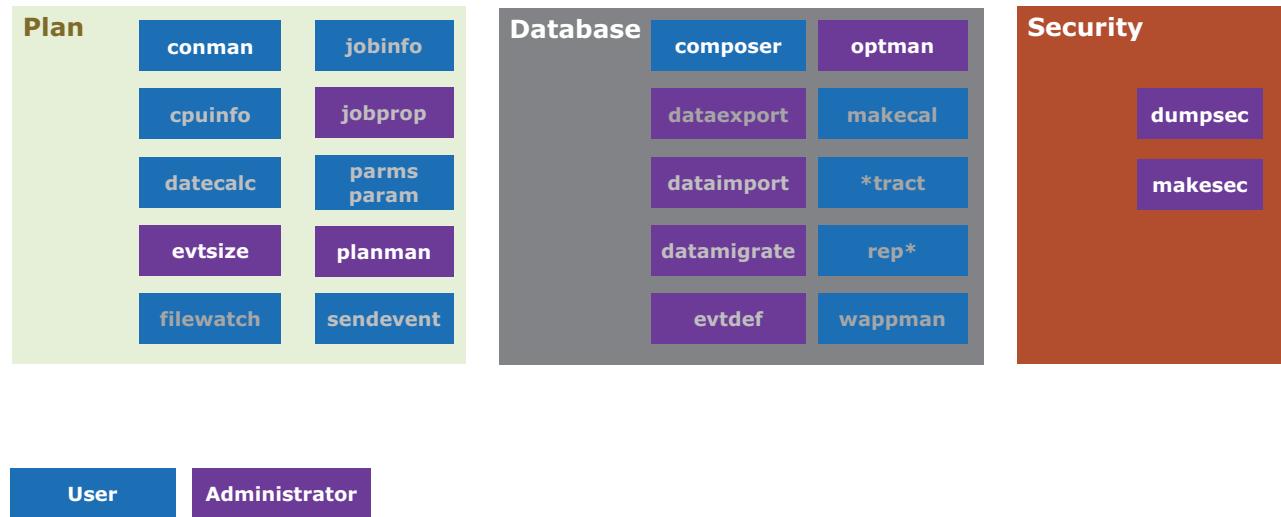
Jazz for Service Management links data and other shared integration services, including administrative, dashboard, reporting, and security services. Jazz for Service Management accelerates deployment, integration, and automation services across IBM, partner, and third-party tools by using open and standardized approaches to linking data.

The **Dashboard Application Services Hub** is a web-based console component that provides common navigation, aggregates data, and passes messages from views of different products.

The four web-based interfaces that you use with Workload Scheduler are listed here:

1. Use the **Dynamic Workload Console** manage and control IBM Workload Scheduler production for z/OS and distributed environments.
2. Use the **Application Lab** to create, run, and monitor simple business processes from a single web browser page.
3. From the **Self-Service catalog** you can define services that correspond to Workload Scheduler job streams and submit them to run from your mobile device, even if you do not have experience with IBM Workload Scheduler.
4. Use the **Self-Service Dashboard** to view dashboards on your mobile device and drill down to more detailed jobs and workstations that match the criteria you select.

Workload Scheduler command-line interfaces



© Copyright IBM Corporation 2017

Workload Scheduler command-line interfaces

In addition to web-based graphical user interfaces, IBM Workload Scheduler provides command-line user interfaces and utility programs that you use to manage specific parts of Workload Scheduler production. Commands that are highlighted in violet are used by systems administrators, and commands that are highlighted in blue are used by Workload Scheduler users.

Command-line interfaces for managing the plan

The main interfaces that you can use to manage the Workload Scheduler plan are listed in [Table 1](#).

Table 1 Command-line interfaces for managing the plan

Command	Description
conman	You can manage the IBM Workload Scheduler production plan environment by using the <code>conman</code> command-line program. You use the <code>conman</code> program to start and stop processing, alter and display the Symphony production plan, and control workstations that link in a network. You can use <code>conman</code> from the master domain manager and from any fault-tolerant agent in the IBM Workload Scheduler network.
cpuinfo	You can use the <code>cpuinfo</code> command to retrieve workstation information such as the node name, port number, and time zone.
datecalc	Use the <code>datecalc</code> command to resolve date expressions and report them in a format you choose.

Table 1 Command-line interfaces for managing the plan (continued)

Command	Description
evtsize	The IBM Workload Scheduler administrator uses <code>evtsize</code> to increase the size of a message file, or to monitor the size of the queue of messages that are contained in the message file.
filewatch	Use the <code>filewatch</code> command in your job scripts to cause the script to pause until a file is created, modified, or completed.
jobinfo	On the master domain manager and on fault-tolerant agents, you use the <code>jobinfo</code> command in a job script to retrieve information about the job, such as its job stream, priority, or rerun status.
jobprop	Use the <code>jobprop</code> command to set and retrieve variables for jobs that run on dynamic agents.
parms	Use the <code>parms</code> command to define and query locally managed parameters.
param	Use the <code>param</code> command to define and manage locally managed variables and passwords on dynamic agents.
planman	Workload Scheduler administrators use the <code>planman</code> command.
sendevent	Use the <code>sendevent</code> command to send custom events that are defined with the <code>evtdef</code> command to the event processor.

Command-line interfaces for managing the database

The main interfaces that you can use to manage the Workload Scheduler database, and create reports are listed in [Table 2](#) .

Table 2 Command-line interfaces for managing the database

Command	Description
composer	You can manage scheduling objects in the IBM Workload Scheduler database by using the <code>composer</code> command-line program. You can install the <code>composer</code> command-line program on any computer that is connected to the system where the master domain manager runs. It does not require the installation of a IBM Workload Scheduler workstation as a prerequisite.
dataexport	You use the <code>dataexport</code> command to create text files that contain all of the scheduling object definitions and global settings from the Workload Scheduler database.
dataimport	You use the <code>dataimport</code> command to import all scheduling object definitions and global settings that the <code>dataexport</code> command saves into the Workload Scheduler database.
datamigrate	Use the <code>datamigrate</code> utility on the master domain manager to import data from the text files that were created by using <code>composer extract</code> commands or <code>dataexport</code> utility into the Workload Scheduler database.

Table 2 Command-line interfaces for managing the database (continued)

Command	Description
evtdef	In addition to the already defined event types and event classes (known as providers) listed in detail in Event providers and definitions, IBM Workload Scheduler supplies the template of a generic event provider that is named <code>GenericEventPlugIn</code> . Programmers with specific application and XML programming skills can modify the plug-in to define custom event types that might be of use to the organization. The <code>evtdef</code> command imports and exports a generic event provider XML definition file that you can edit to add and modify custom event types. You can then use the <code>sendevent</code> command to send these events to the event processing server.
optman	IBM Workload Scheduler administrators use the <code>optman</code> command to list, show, and change Workload Scheduler global options.
makecal	Use the <code>makecal</code> command to create custom calendars in the Workload Scheduler database.
Extract programs	Use the data extraction programs to generate text files that contain information about Workload Scheduler database scheduling objects and production scheduling.
Report programs	Use the report programs to format information about scheduling objects and production scheduling and history into text-based output.
wappman	Use the <code>wappman</code> command to create, replace, delete, display, or list a workload application.

Command-line interfaces for managing security

IBM Workload Scheduler administrators control how Workload Scheduler manages security by using the classic security model, or the role-based security model. When you use the classic security model, you manage security by extracting, editing, and compiling a configuration file. The `dumpsec` command extracts the security definitions into a text-based configuration file. The `makesec` command compiles the configuration file into a format that Workload Scheduler uses to manage user authorization. When you use role-based authorization, the `makesec` and `dumpsec` commands are disabled, and you use the `composer` command to model security domains, security roles, and access control lists.

Application programming interfaces

Many features of IBM Workload Scheduler can be programmed. You can automate Workload Scheduler functions by using the Java API, Web Services, and REST interfaces to manage objects in the database and plan.

Lesson 6 Workload Scheduler user interfaces

IBM Workload Scheduler includes an Eclipse-based Integration workbench. With this SDK, you can create and develop API projects and references to the Java classes and methods.

You can extend IBM Workload Automation by creating Java plug-ins to implement events and actions for event-driven automation. You can extend Workload Scheduler to new job types by creating job plug-ins for running dynamic job types, and create Java jobs.

You can use application programming interfaces to create your own user interfaces to perform the functions of command-line programs and the Dynamic Workload Console. You can perform the following tasks:

- Modify objects in the database
- Submit workload
- Monitor the plan
- Act on the plan, such as rerun jobs

Lesson 7 Using the Dynamic Workload Console

Using the Dynamic Workload Console

© Copyright IBM Corporation 2017

The **Dynamic Workload Console** is a web-based user interface for IBM Workload Scheduler and IBM Workload Scheduler for z/OS. You create, display, and monitor jobs and job streams in IBM Workload Scheduler by using the Dynamic Workload Console.

You can access the Dynamic Workload Console from a computer in your environment by using a web browser through either the secure HTTPS or HTTP protocol. The Dynamic Workload Console is built on the IBM Dashboard Application Services Hub (DASH).

When you log in to the IBM Dashboard Application Services Hub, you see icons on the top that contain entries for each product that is installed and integrated with it.

To access the Dynamic Workload Console online documentation, click the question mark at the upper right of the window.

With Dynamic Workload Console, you can perform these tasks:

- 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 implement secure authorization to scheduling database and plan objects.
- 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.

Tasks that use the Dynamic Workload Console

When you install and first connect to the Dynamic Workload Console, you create a connection to an IBM Workload Scheduler engine, define security authorizations, define your scheduling environment, and create monitoring tasks.

Creating connections to Workload Scheduler engines

To manage scheduling objects, you must connect from the Dynamic Workload Console to a Workload Scheduler environment. In the Dynamic Workload Console, you connect by defining engine connections. You can connect to both Workload Scheduler distributed and z/OS environments, and you can create as many engine connections as you need.

Defining the scheduling environment

You define the Workload Scheduler network by creating workstation definitions in the Workload Scheduler database. Workstations represent the computers where a workload is scheduled to run. A Workload Scheduler network consists of workstations where job processing occurs.

Managing Workload Scheduler auditing and security authorization settings

You can change the auditing preferences and justification categories by using the Dynamic Workload Console. When you enable role-based security authorization, you can also manage Workload Scheduler security by using the Dynamic Workload Console.

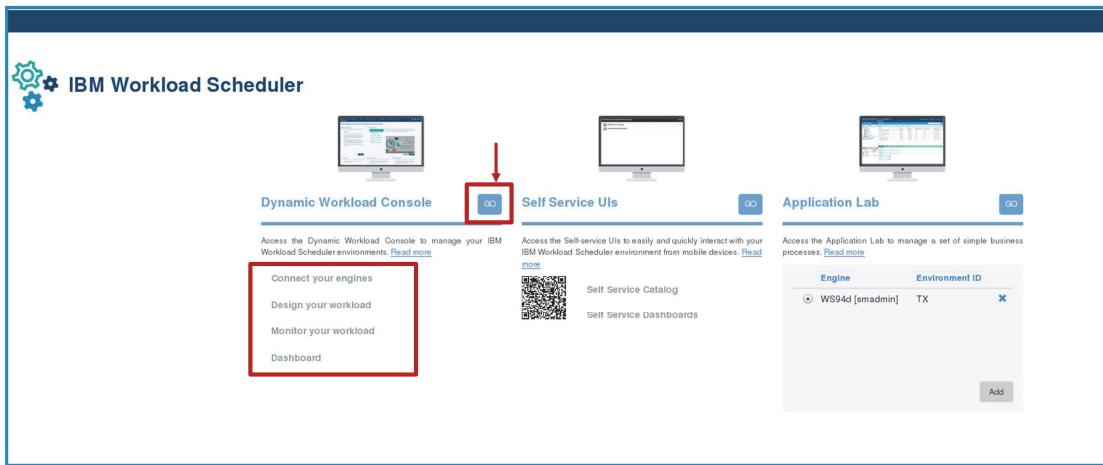
Defining objects in the scheduling database

You define the workload for Workload Scheduler to schedule. The definitions include calendars, jobs, run cycles, and job streams. You also specify dependencies between jobs and job streams to act as conditions for other jobs to run. Defining scheduling objects is covered in the *IBM Workload Scheduler 9.4 Scheduling and Operations* class.

Creating tasks to manage objects in the plan

You build task definitions in the Dynamic Workload Console for monitoring the progress of a production plan that is running. Each task specifies filtering criteria to query the list of scheduling object instances whose attributes satisfy your specifications. For example, you can create a task that specifies all jobs that ended in error. Each task can be shared with other users who use the Dynamic Workload Console.

Opening the Dynamic Workload Console



© Copyright IBM Corporation 2017

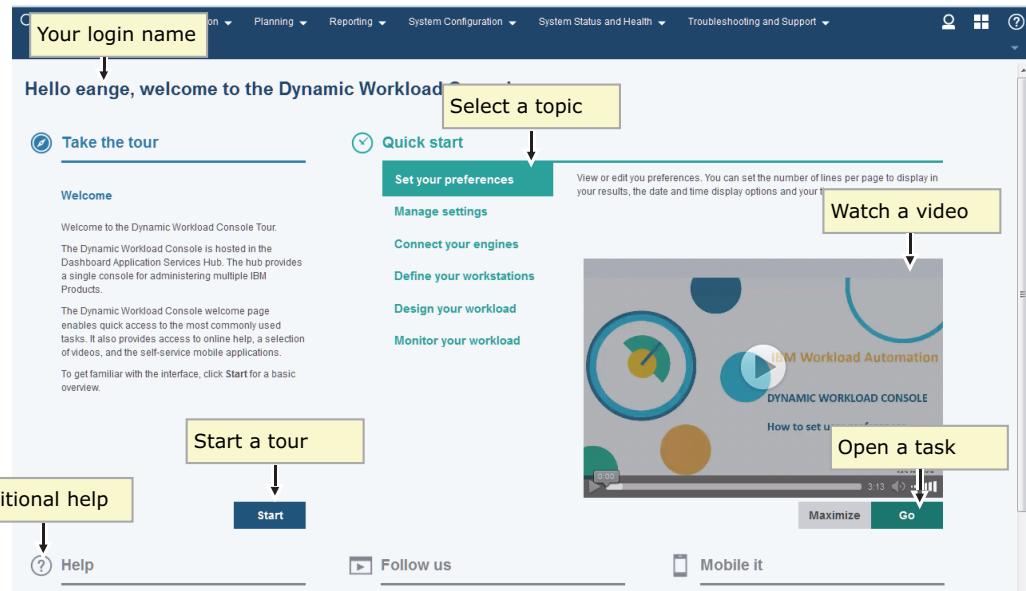
Opening the Dynamic Workload Console

Follow these steps to access the console.

1. Address your web browser to the URL of the server where the console is installed and running. The default address ends in `/ibm/console`, but it might be configured differently for your console. For example, if the console is installed to a server on `www.example.com` that uses a secure port of 16311 and with the default context root, enter this address in the browser:
`https://www.example.com:16311/ibm/console`
2. After the login page for the console is displayed, enter your user ID and password and click **Login**. When you are finished, an introduction page is displayed. In the Dynamic Workload Console section, click **GO** to open the Dynamic Workload Console. On the Welcome page, you can take a tour of the Dynamic Workload Console. To access the Welcome page, click the **Help** icon and then **Welcome**.
3. Be sure to use the User icon, **Logout** link in the navigation bar when you are finished with the console to prevent unauthorized access. If no activity occurs during the session for an extended period, the session expires and you must log in again.

Note: The browser **Back** button is not reliable to navigate the console and is not supported. Use the navigation controls and links in the console as an alternative.

Using the Welcome page



Using the Welcome page

The Welcome page is first page that you see when you log on to the Dynamic Workload Console. The Welcome page contains the following elements:

- A welcome message, with your login name
- An introduction to a built-in tour of the Dynamic Workload Console
- A Quick start section that contains links to commonly used tasks, along with video demonstrations of those tasks
- Links to extra help, social media sites that focus on IBM Workload Automation, and Workload Scheduler interfaces that are designed for mobile devices

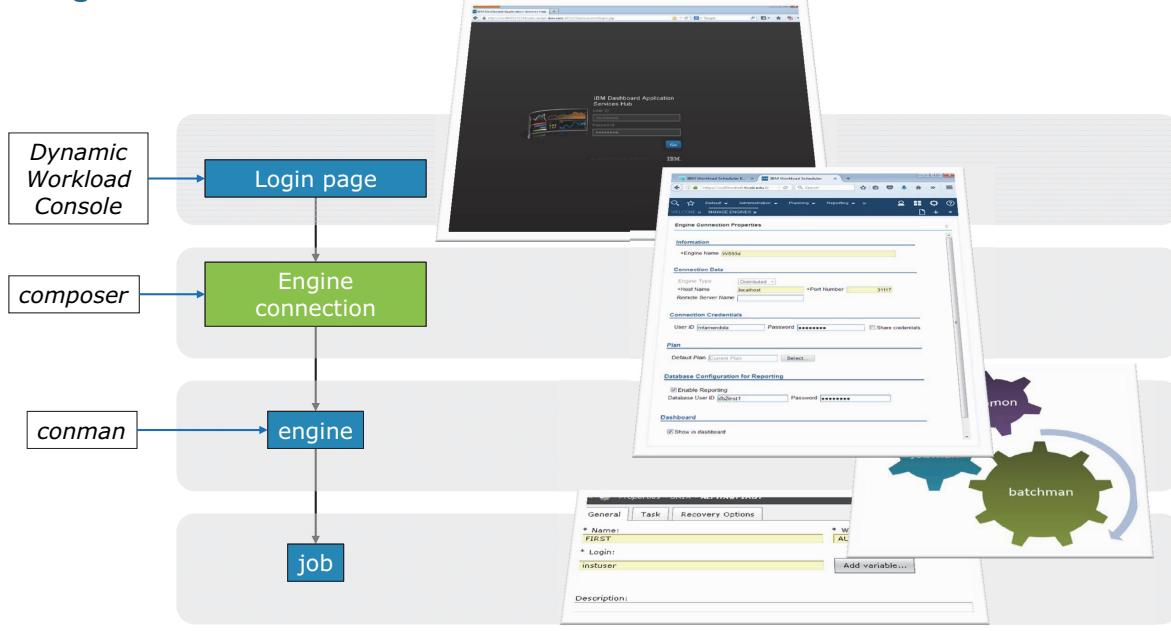
Lesson 8 Setting up the Dynamic Workload Console

Setting up the Dynamic Workload Console

© Copyright IBM Corporation 2017

In this lesson, you learn how to configure the Dynamic Workload Console to connect to one or more Workload Scheduler or Workload Scheduler for z/OS scheduling engines.

The engine connection



© Copyright IBM Corporation 2017

The engine connection

An engine connection contains configuration information that identifies a specific computer in the Workload Scheduler network that runs the Workload Scheduler connector software. The master domain manager runs the connector software, and it can also be installed on other computers in the Workload Scheduler network.

When you create a connection to a Workload Scheduler engine, you specify the details that are required to access Workload Scheduler. You also can specify the credentials that are required to access a database for creating and viewing historical reports. You can create connections to more than one Workload Scheduler or Workload Scheduler for z/OS engine. For example, if you have separate test and production environments, you can create connections to both of them. When you build job monitoring tasks, you can combine the information found on the engines that you create and view that information in a single view.

When you create an engine connection, you give it a name that is familiar to you, and select the plan that you want it to access. The selected plan must be accessible from that computer.

You can access different types of plans and connect to different types of engines. For example, you can create an engine definition that uses a forecast plan to review and report on plan objects predicted to run in the future. Based on the type of plan that you select and the engine that you connect, you can get different results when you run your tasks in the same Workload Scheduler distributed environment. You can connect to the following locations:

- The master domain manager workstation is the top management hub. Select this workstation if you want to access the entire set of objects that are involved in the current plan, or if you want

to access a trial plan, a forecast plan, or an archived plan. You can define and use different engine connections to the master domain manager, each accessing a different plan.

- A domain manager where the connector software is installed. Select this workstation (where jobs and job streams are run) if you want to access the set of objects that are involved in the current plan and scheduled to run in that domain. You choose this option if you need up-to-date information about job status on this workstation.

Creating and managing engine connections

Engine connection properties

- Engine name
- Engine type
- Host name (localhost)
- Port number (31117)
- Remote server name (z/OS engines only)
- User ID (blank for SSO)
- Password (blank for SSO)
- Share credentials (cleared)
- Default plan (current plan)
- Enable reporting (selected)
- Database user ID and password
- Show in dashboard (selected)

The screenshot shows the 'Engine Connection Properties' dialog box. It includes sections for Information (Engine Name: WS93d), Connection Data (Engine Type: Distributed, Host Name: localhost, Port Number: 31117), Connection Credentials (User ID: mlamendola, Password: [redacted], Share credentials checked), Plan (Default Plan: Current Plan, Select... button), Database Configuration for Reporting (Enable Reporting checked, Database User ID: db2inst1, Password: [redacted]), and Dashboard (Show in dashboard checked). A copyright notice at the bottom right reads: © Copyright IBM Corporation 2017.

Creating and managing engine connections

You can modify or delete engine connections that you created. To create, modify, or delete an engine connection, from the navigation toolbar, click **System Configuration > Manage Engines**.

From the displayed page, you can create, edit, delete, share an engine connection, and test the connection to the remote server where Workload Scheduler is installed. You can order the list of engine connections that are displayed on the page by using sorting criteria that you select with the buttons at the upper left corner of the table.

Specifying details of the engine connection

Creating an engine connection means defining the details of the Workload Scheduler engine to which you want to connect.

Engine Name	The name for the connection that you are creating. It is a familiar name for the Workload Scheduler environment to which you are connecting, for example, <i>Production</i> , <i>QA</i> , or <i>jmulberry@prodmdm(31117)</i> . The name cannot contain spaces and must start with a letter.
Engine Type	The type of engine connection you are creating. Choose Distributed or z/OS .
Host Name	For distributed environments, enter the host name or TCP/IP address of the computer where the Workload Scheduler engine is installed. For z/OS environments, it is the host name or TCP/IP address of the computer where the z/OS connector is installed. The default is localhost .

Port Number	The port number that is used to connect to the computer where the Workload Scheduler engine or the z/OS connector is installed. The default port number is 31117 for Workload Scheduler distributed engines, or 16312 for Workload Scheduler for z/OS engines at version 9.1 or later.
Remote Server Name	The name of the remote server of the engine as it was specified in the z/OS connector configuration. This setting applies only to z/OS systems and is required when you select a z/OS engine type.
User ID	The user name that you want to use to access Workload Scheduler through your engine connection.
Password	The password of the user that you want to use to access Workload Scheduler through your engine connection.
Share credentials	Select the check box to share the user ID and password to all the users who share this engine. Then, users can use the engine connection without entering any credentials, automatically accessing it with your credentials. When this check box is not selected, users who share this engine connection must enter credentials to use it. Note: Users cannot modify engine connection properties with the shared connection.
Default Plan	Click Select to view a list of available plans. Select one plan from the list, and click OK .
Enable Reporting	Check this field if you want this engine connection to be able to run reports.
Database User ID and Password	Specify the database user and password that is authorized to access reports. This user must have access to directly query the database. This option is enabled only for reports.
Show in dashboard	Select the check box to include information about the current plan on this engine in the dashboard graphical views. The check box is selected by default. Clear the check box if you do not want information about this engine to be included in the dashboard.



Attention: In some cases, the user ID and password are not required in the engine definition because both the Dynamic Workload Console and Workload Scheduler can share authentication protocols.

When you are finished, click the following options:

Show Data	To view more details about the engine
Test Connection	To verify whether the connection works

OK	To save your settings and exit
Cancel	To exit without saving your settings

Lesson 9 Getting started with the command-line interfaces

Getting started with the command-line interfaces

© Copyright IBM Corporation 2017

The conman and composer programs are command-line user interfaces for IBM Workload Scheduler. You monitor jobs and job streams in IBM Workload Scheduler by using the conman program (also known as conman). You can access conman from the master domain manager in your Workload Scheduler environment by using a login shell. You use composer to manage objects in the Workload Scheduler database. The composer program is installed on the master domain manager, but you can also install it as part of the Workload Scheduler Command Line Client feature on other computers.

When you use the command-line interfaces, you can quickly access specific objects in the database or plan by using English-like keywords. You can also add or modify large quantities of objects with few commands. The command-line interfaces can also be used within batch files and shell scripts to perform Workload Scheduler related functions unattended.

Preparing to use the command-line interfaces

```
[wsuser@ws94mdm0 ~]$ . /opt/IBM/TWA/TWS/tws_env.sh
IBM Workload Scheduler Environment Successfully Set !!!
[wsuser@ws94mdm0 ~]$ export MAESTRO_OUTPUT_STYLE=LONG
[wsuser@ws94mdm0 ~]$ conman
IBM Workload Scheduler(UNIX)/CONMAN 9.4.0.00 (20161202)
Licensed Materials - Property of IBM* and HCL**
5698-WSH
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.
(C) Copyright HCL Technologies Ltd. 2016 All rights reserved.
* Trademark of International Business Machines
** Trademark of HCL Technologies Limited
Installed for user "wsuser".
Locale LANG set to the following: "en"
Scheduled for (Exp) 05/02/17 (#45) on MDM0. Batchman LIVES. Limit: 11,
Fence: 0, Audit Level: 1
% exit
[wsuser@ws94mdm0 ~]$
```

1. Log on using **xterm**, **PuTTY**, or other
2. Source the Workload Scheduler environment
3. Set variables to customize your display
4. Run **conman** or **composer**
5. Type **exit** to return to the shell

© Copyright IBM Corporation 2017

Preparing to use the command-line interfaces

You use the command-line interfaces from a login shell or command prompt on Microsoft Windows. Before you run the command-line interfaces, set your **PATH** and **TWS_TISDIR** variables by sourcing one of the following scripts:

- In UNIX and Linux:
 - `./WSHome/tws_env.sh` for Bourne, Korn, and Bash shells
 - `source./WSHome/tws_env.csh` for C shell
- In Windows:
 - `WSHome\tws_env.cmd`

Setting variables

Before you run the **composer** or **conman** command-line interfaces, you can set other environment variables to change the behavior of the interface.

The **MAESTROLINES** and **MAESTROCOLUMNS** variables determine the number of rows and columns sent to your screen at a time. If either variable is not set, the standard variables **LINES** and **COLUMNS** are used. At the end of each screen page, after displaying a number of lines of data, the interface prompts to continue. If **MAESTROLINES** is set to zero or a negative number, the interface does not pause.

The columns of output you see displayed in **conman** and **composer** changes, based on the value of **MAESTROCOLUMNS**. If **MAESTROCOLUMNS** is less than 120, you see fewer columns of data.

The `MAESTRO_OUTPUT_STYLE` variable specifies how object names are displayed in conman. If set to `LONG`, full names are displayed. If not set, or set to any value other than `LONG`, names longer than eight characters are truncated to seven characters followed by a plus sign (+).

You use the `;offline` option in composer and conman commands to print the output of a command. When you use the `;offline` option, these variables affect the output:

- `MAESTROLP` specifies the destination of the output of the command.
- `MAESTROLPCOLUMNS` specifies the number of characters per line. The default is 132.
- `MAESTROLPLINES` specifies the number of lines per page. The default is 60.

Setting your connection profile

When you run composer or conman, your user name to log in to the operating system where you run the commands is the authenticated user for Workload Scheduler functions. However, for database functions you must supply connection parameters to the master domain manager application server infrastructure. You supply the connection criteria in one of these ways:

- **Stored in the `localopts` file.** The `localopts` file contains a set of connection parameters that apply only to the local workstation for a specific instance of the product and its installed user.
- **Stored in the `useropts` file.** There can be at least one `useropts` file for each user who uses Workload Scheduler command-line interfaces. The `useropts` file contains custom connection values specific to the user, and the instance of the Workload Scheduler product that is installed. The `useropts` file is stored within the user's home directory. Its name reflects the instance name of the Workload Scheduler product installation.
- **Stored in a user-defined options file specified on the command line.** You can create a custom connection parameters file to use when you run a command-line interface. You specify the name of your custom parameters file on the conman or composer command line.
- **Supplied to the command as arguments to the command string.** You can pass the connection information for connecting and authenticating to the Workload Scheduler infrastructure on the command line.

Running the command line in batch versus interactive mode

You can use conman and composer command lines in both *batch* and *interactive* modes. When you run conman or composer in interactive mode, you run the command-line interface program. From its command prompt, you run commands, typically one at a time. When you run conman or composer in batch mode, you supply commands as input arguments to the program, for example:

```
composer DISPLAY WS=AGENT1
```

Using wildcards

In the command-line interfaces, you can use wildcards to select objects.

@	Replaces one or more alphanumeric characters
?	Replaces one alphanumeric character
%	Replaces one numeric character



Hint: The composer and comman commands are not case-sensitive.

Using the command-line interfaces

```
User: wsuser, Host:127.0.0.1, Port:31116
- help commands
```

COMPOSER COMMANDS

Note: Command names and keywords can be entered in either uppercase or lowercase characters, and can be abbreviated to as few leading characters as are needed to uniquely distinguish them from each other. Some of the command names also have short forms.

However there are some abbreviations, such as v, that point to a specific command, version in this case, even though they do not uniquely identify that command in the list. This happens when the abbreviation is hard coded in the product and so mismatches in invoking the wrong command are automatically bypassed.

Command	Short Name	Description
add	a	Adds a scheduling objects definition to the database from a text file.
authenticate	au	Changes the credentials of the user running composer.

- Help is available by using the **help** command
- Type **help commands** for a list of available commands

© Copyright IBM Corporation 2017

Using the command-line interfaces

You can use the **conman** command for monitoring all production plan functions except Broker jobs, Broker Computers, and Event Rules. With **conman**, you can start and stop processing, alter and display objects in the production plan, and control workstation linking. You can use **conman** from the master domain manager and from any fault-tolerant agent in the Workload Scheduler network.

You run the **conman** command from a shell prompt or the command window on Microsoft Windows. When you run **conman**, you see a banner that contains information about your Workload Scheduler environment, and a prompt, which is by default a percent sign (%). In the banner, you see the following information:

- The version of Workload Scheduler, for example, 9.4.0.00 (20161202)
- The IBM license statement and part number
- The instance name, for example, “**installed for user wsuser**”
- The current scheduled plan date, for example, “**Scheduled for (Exp) 09/06/17**”
- The current run number
- The status of the batch manager process, **batchman**, either **LIVES** or **down**.
- The current limit, fence, and audit level of the workstation where you are running **conman**.

The information that you see when you run monitoring commands is only updated while the Workload Scheduler batch manager process (**batchman**) is running on the workstations. If **batchman** is running, or not is confirmed on the screen by either **Batchman LIVES** or **Batchman down** message when you issue the **conman status** command.

The Workload Scheduler administrator can restrict which objects you can list or modify in the plan. You use the `composer` command to create, modify, or delete objects in the database.

Getting help with conman or composer

The `conman HELP` command provides online help to run `conman` and use its commands. The `composer HELP` command provides online help to run `composer` and use its commands.



Note: Using `conman HELP`

```
[wsuser@ws94mdm0 ~]$ conman
IBM Workload Scheduler(UNIX) /CONMAN 9.4.0.00 (20161202)
Licensed Materials - Property of IBM* and HCL**
5698-WSH
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.
(C) Copyright HCL Technologies Ltd. 2016 All rights reserved.
* Trademark of International Business Machines
** Trademark of HCL Technologies Limited
Installed for user "wsuser".
Locale LANG set to the following: "en"
Scheduled for (Exp) 05/02/17 (#45) on MDM0. Batchman LIVES. Limit: 11, Fence:
0, Audit Level: 1
%
%HELP
```

Displays help information about commands. Not available in Windows.

Syntax

```
{help | h} {command|keyword}
```

Arguments

`command` Specifies the name of a `conman` or system command. For `conman` commands, enter the full command name; abbreviations and short forms are not supported. For commands consisting of two words, enter the first word, and help for all versions of the command is displayed. For example, entering `help display` displays information about the `display` file, `display job`, and `display sched` commands.

Supply a topic title or command with the HELP command for specific help about the topic or command. For example, type HELP COMMANDS in conman or composer to see a list of commands that you can use. Most commands have both long forms and short forms. Both forms are listed in the *User's Guide and Reference* and in the HELP COMMANDS section. For example, instead of typing SHOWSCHEDULES to display the job streams in the plan, you can type ss.

Commands

Command	Short Name	Description
add	a	Adds a scheduling objects definition to the database from a text file.
authenticate	au	Changes the credentials of the user running composer.
continue	c	Ignores the next error.
create	cr	Extracts an object definition from the database and writes it in a text file. Synonym for the extract command.
delete	de	Deletes scheduling objects.
display	di	Displays the details of the specified scheduling object.
edit	ed	Edits a file.
exit	e	Exits composer.
extract	ext	Extracts an object definition from the database and writes it in a text file.
help	h	Invoke the help on line for a command.
list	l	Lists scheduling objects.
lock	lo	Locks the access to database objects.
modify	m	Modifies scheduling objects.

Instructor demonstration

Logging in to the learning environment
Starting the Dynamic Workload Console
Starting a command line interface



© Copyright IBM Corporation 2017

Instructor demonstration



Exercises

- Log in to the computer
- Start the Dynamic Workload Console interface
- Define an engine connection
- Start the command interface

© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. What are the four primary IBM Workload Automation configurations?
2. How many domains are required in the Workload Scheduler network?
3. What are three benefits of deploying a multiple-domain Workload Scheduler topology?
4. What is the primary purpose and use of the Dynamic Workload Broker and dynamic scheduling?
5. What are the three primary categories of Workload Scheduler agents?
6. What is the main function of a workstation class?
7. What is the name of the production control file that Workload Scheduler creates to manage the planned production period (usually 24 hours)?
8. What types of agents automatically receive a copy of the production control file?
9. What three types (categories) of user interfaces does Workload Scheduler provide?
10. On what IBM management platforms are the Workload Scheduler web-based interfaces built?
11. What are the primary command-line interfaces for managing the plan and the database?
12. What is the function of an *engine connection*?
13. Which wildcard character replaces one or more alphanumeric characters in conman and composer?

Review answers

1. What are the four primary IBM Workload Automation configurations?

The primary configurations are mainframe, distributed, cloud, and peer-to-peer (or “Mixed mode”).

2. How many domains are required in the Workload Scheduler network?

A Workload Scheduler network has at least one domain: the master domain.

3. What are three benefits of deploying a multiple-domain Workload Scheduler topology?

Three benefits of a multiple-domain topology are decreased network traffic, better security, and optimized fault tolerance.

4. What is the primary purpose and use of the Dynamic Workload Broker and dynamic scheduling?

With dynamic scheduling, you can choose to set Workload Scheduler to dynamically associate your submitted workload or part of it to the best available resources at run time.

5. What are the three primary categories of Workload Scheduler agents?

Workload Scheduler agents are fault-tolerant agents, dynamic agents, and extended agents.

6. What is the main function of a workstation class?

Workload Scheduler replicates the jobs and job streams that you created for the workstation class to each member of the class.

7. What is the name of the production control file that Workload Scheduler creates to manage the planned production period (usually 24 hours)?

*The production control file is named **Symphony**.*

8. What types of agents automatically receive a copy of the production control file?

Domain managers and fault-tolerant agents receive a copy of the Symphony file.

9. What three types of user interfaces does Workload Scheduler provide?

Workload Scheduler provides web-based, graphical user interfaces, command-line, text-oriented interfaces, and application programming interfaces.

10. On what IBM management platforms are the Workload Scheduler web-based interfaces built?

The four unique web-based user interfaces for IBM Workload Scheduler are built on the IBM Jazz for Service Management platform, and Dashboard Application Services Hub.

11. What are the primary command-line interfaces for managing the plan and the database?

You can manage the IBM Workload Scheduler production plan environment by using the conman command-line program. You can manage scheduling objects in the IBM Workload Scheduler database by using the composer command-line program.

12. What is the function of an *engine connection*?

The engine connection specifies computer and connection information to connect the Dynamic Workload Console to the master domain manager.

13. Which wildcard character replaces one or more alphanumeric characters in conman and composer?

The at-sign (@) character replaces one or more alphanumeric characters in conman and composer.

Summary

You should now be able to perform the following tasks:

- Describe the concepts, needs, and terminologies of dynamic batch computing, and how batch computing silos are reduced through common infrastructures
- Plan to use a Workload Scheduler solution that fits your requirements
 - Mainframe-based
 - Distributed-based
 - Peer-to-peer
- Describe typical Workload Scheduler architectures
- Outline Workload Scheduler agent and workstation types
- Set up and log in to the Dynamic Workload Console
- Start using the command line interfaces

© Copyright IBM Corporation 2017

Unit summary

Unit 2 Planning the Workload Scheduler implementation

IBM Training



Planning the Workload Scheduler implementation

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to gather and apply information about your infrastructure. You can then use this information to build an effective Workload Scheduler network environment. This unit includes lessons about finding hardware and software prerequisites and planning Workload Scheduler networks, including networks with firewalls.

Objectives

In this unit, you learn how to perform the following tasks:

- Design a Workload Scheduler network topology
- Create a domain hierarchy
- Prepare a database
- Identify hardware and software prerequisites
- Create workstation and domain definitions
- Identify user privileges

© Copyright IBM Corporation 2017

Objectives

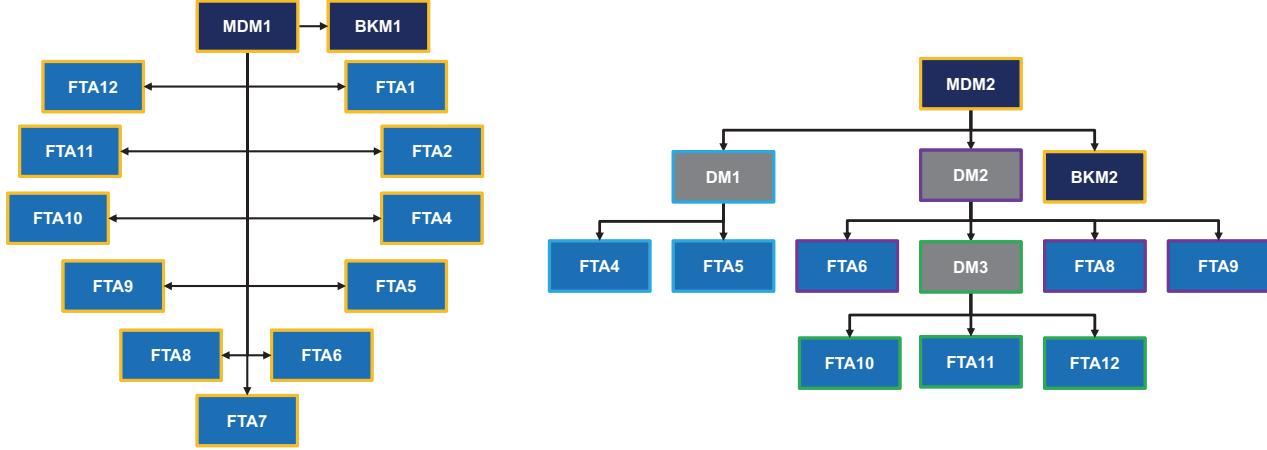
Lesson 1 Planning the Workload Scheduler topology

Planning the Workload Scheduler topology

© Copyright IBM Corporation 2017

A Workload Scheduler network consists of a linked set of workstations on which jobs run. The network has one or more domains. The domains have a domain manager, agents, and one or more backup domain managers. Workload Scheduler runs jobs either statically or dynamically. You can configure domains, depending on the type and nature of workload that you schedule and the groupings of computers where jobs run. You can combine fault-tolerant domains and dynamic domains.

Domain hierarchies for fault-tolerant agents

**Single-domain topology****Multiple-domain topology**

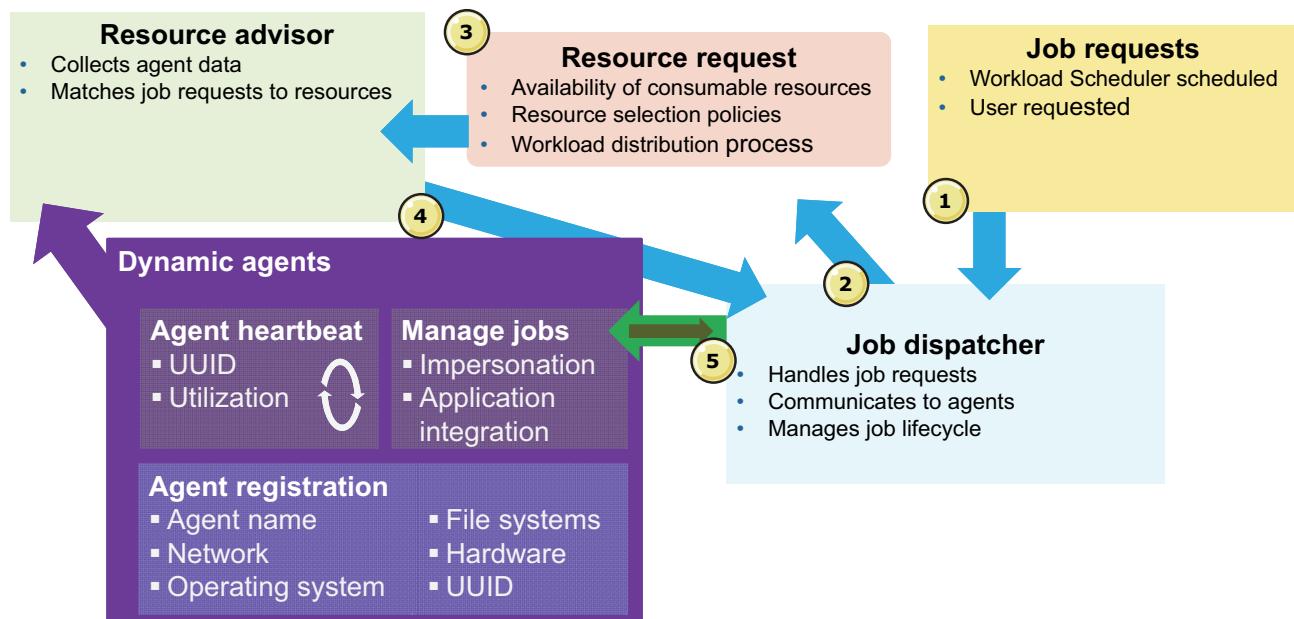
© Copyright IBM Corporation 2017

Domain hierarchies for fault-tolerant agents

In the slide, you see 13 fault-tolerant agents, including the master domain manager (MDM). They are organized first in a single domain, on the left, and then into four domains, on the right. Consider the following scenarios, by using the domain hierarchies on this slide as an example.

- **Daily plan distribution:** In a single-domain topology, the master domain manager directly connects to 12 agents. The master domain manager distributes the daily production plan to all 12 agents in order. In the multiple-domain topology, only three agents are directly connected. The production plan is distributed to DM1, DM2, and BKM2. The plan is distributed in parallel to the fault-tolerant agents in the domains of DM1 and DM2 and to the domain manager DM3.
- **Dependency resolution:** In the single-domain topology, the FTA10 jobs that have dependencies on the FTA6 jobs must wait for communication with the master domain manager. Under heavy workload conditions or during an outage, this wait can be delayed. In the multiple-domain topology, the domain manager handles dependencies between workstations in the same domain. DM2 resolves dependencies between FTA10 and FTA6, even if a disconnection or outage with the MDM occurs.
- **Firewalled locations:** Consider the case when FTA7, FTA11, and FTA12 are in a remote location, behind an IP firewall. In a single-domain topology, the firewall configuration of the computers that are running FTA7, FTA11, and FTA12 must allow IP connections from MDM. In a multiple-domain topology, FTA7 becomes domain manager DM3 for FTA11 and FTA12, and it has routing capabilities for Workload Scheduler traffic. All three workstation definitions have the firewall attribute. Only the firewall configuration of DM3 requires IP connections with MDM. For Workload Scheduler traffic, FTA11 and FTA12 communicate only with DM3.

Dynamic Workload Broker



© Copyright IBM Corporation 2017

Dynamic Workload Broker

The *Dynamic Workload Broker* is a component of the Workload Scheduler master domain manager and dynamic domain managers. The dynamic workload broker manages the allocation of jobs to dynamic agents. It consists of two main parts:

- The *job dispatcher* manages interactions between job requests and the dynamic agents, and interacts with the job based on users' requests, such as retrieving the job log.
- The *resource advisor* matches job requests to resources.

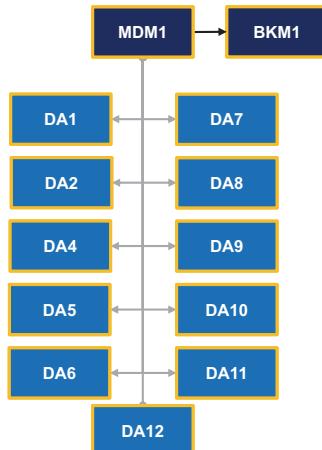
Workload brokering process

The following list describes the process for managing job requests, applying them to the correct resource, and running the job on an agent.

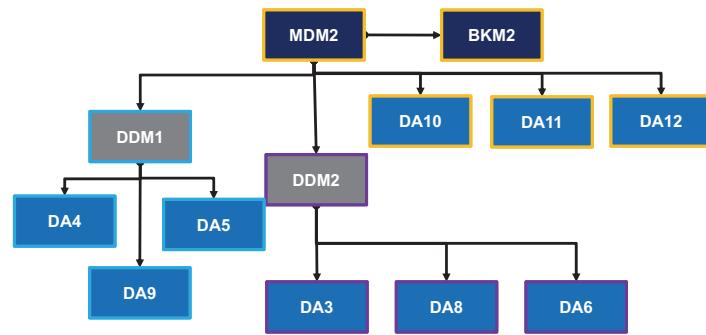
1. On an ongoing basis, dynamic agents populate the resource repository, which is part of the Workload Scheduler database, with the availability and performance information, such as CPU usage and file system availability.
2. When a job start is requested, the job dispatcher passes information from the job requests to the resource advisor.

3. The resource advisor's matching process selects the most capable resource to run jobs on these criteria:
 - **Availability of consumable resources:** Which resources are available according to information provided by the agent and which jobs are concurrently running.
 - **Resource selection policies:** User configuration can determine whether jobs run on the most available resource, the one with fewest jobs that are running, or least available resource, the one with the most jobs that are running.
 - **Workload distribution process:** Priorities set on the job definition give privileged assignment when multiple jobs compete for the same resource.
4. After the resource advisor returns an allocation request, the job dispatcher manages the job's lifecycle. The job dispatcher communicates with the dynamic agent, monitors the status of the job, and interacts with the job based on users' requests, such as retrieving the job log.

Hierarchies for dynamic agent domains



Single-domain topology



Multiple-domain topology

© Copyright IBM Corporation 2017

Hierarchies for dynamic agent domains

You can create domains of dynamic agents by installing *Dynamic Domain Managers*. A dynamic domain manager adds dynamic workload brokering capabilities to a domain manager. When you install a dynamic agent, you configure the dynamic domain manager or the master domain manager with which the agent communicates. When the agent connects to its domain manager, the agent definition is registered in the Workload Scheduler database, and a new workstation definition is automatically stored.

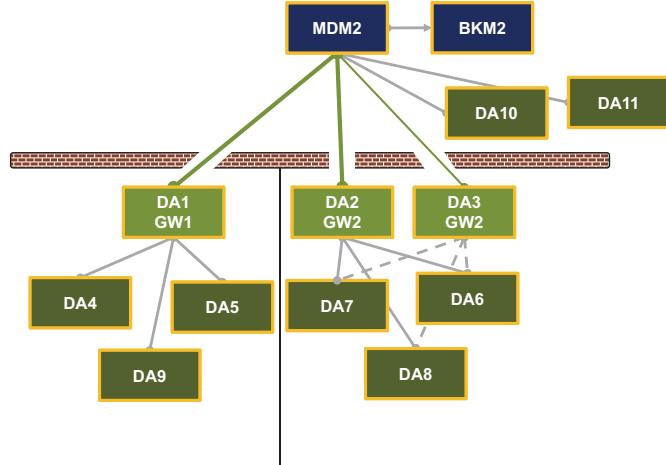
When a dynamic domain manager is installed, all the communication to and from dynamic agents in the domain are routed through the dynamic domain manager. A dynamic agent connects to only one dynamic domain manager or its backup either directly, or indirectly through a gateway.



Important: When you create dynamic pool workstations, all of the agents that subscribe to the pool must be in the same dynamic domain.

Because dynamic agents do not receive copies of the production plan and do not resolve dependencies, the dynamic domain manager performs these duties.

Using gateway agents



Agents as gateways

- You can configure agents to act as network gateways for other agents that cannot directly connect to the master domain manager or dynamic domain manager
- Parallel gateways provide redundancy for routing communications to the domain manager

© Copyright IBM Corporation 2017

Using gateway agents

Typically, you configure dynamic agents to connect directly to the dynamic domain manager or master domain manager. In more complex topologies, if the network configuration prevents the domain manager from connecting with the agents, you can configure a dynamic agent to act as a gateway or to use another dynamic agent as a gateway.

When you configure an agent to use a gateway, its network communication with the domain manager is through the gateway, and the gateway *initiates* connections to the domain manager. The gateway agents must be able to communicate directly with the master domain manager or dynamic domain manager and the other agents.

Lesson 2 Creating workstation and domain definitions

Creating workstation and domain definitions

© Copyright IBM Corporation 2017

You create workstation and domain definitions by using either the Dynamic Workload Console or the command-line interface, called **composer**. In the database, a domain definition is a variant of a workstation definition. You create workstations first and then add them to domains. You can create domains as part of creating or editing workstations. To create new workstations, from the Dynamic Workload Console navigation bar, go to **Administration > Create Workstations**.



Note: Alternatively, go to **Administration > List Workstations** to preview a list of workstation definitions that exist. Then, click **New** to create a new workstation definition.

To edit workstations, from the Dynamic Workload Console navigation bar, go to **Administration > List Workstations**. Select the engine that you want to use from the **Engine name** menu, optionally enter a workstation filter, and click **Display**. From the list, you can create new definitions, edit existing ones, or copy an existing definition into a new one.

Workstation definition attributes

Field	Attribute
Workstation name	CPUNAME
Description	DESCRIPTION
Variable table	VARTABLE
Operating system type	OS
Time zone	TIMEZONE
Workstation type	TYPE
TCP address	TCPADDR
Secure (SSL) address	SECUREADDR
Workload Scheduler domain	DOMAIN

Field	Attribute
Hosting workstation	HOST
Extended agent access method	ACCESS
Agent unique ID	AGENTID
Automatically link	AUTOLINK
Behind a firewall	BEHINDFIREWALL
Security level (SSL)	SECURITYLEVEL
Full status	FULLSTATUS
Mailman server ID	SERVER
Agent protocol (http/https)	PROTOCOL

© Copyright IBM Corporation 2017

Workstation definitions attributes

The following fields are found on the page that you use to create or modify workstation definitions. Some fields are populated with different options or disabled, depending on which type of definition you specify. The attribute is the keyword found in the workstation definition when you use composer to edit the configuration.

Table 1 Workstation definition fields

Field	Attribute	Description
Name	CPUNAME	The name of the workstation
Description	DESCRIPTION	A description of the workstation
Variable table	VARTABLE	The name of a variable table that is associated with the workstation. Variables that are used with the workstation are defined in this table.
Operating system	OS	The operating system that is installed on the system. Specify one of the following values: <ul style="list-style-type: none"> • UNIX (UNIX) • Windows (WNT) • z/OS (ZOS) • IBM i (IBM_I) • Other (OTHER)
Node name	NODE	The host name or IP address of the system where the agent is running.

Table 1 Workstation definition fields

Field	Attribute	Description
TCP/IP port	TCPADDR	The netman TCP port number, as installed on the agent. The default is 31111 . This value is assigned to nm port in the localopts file on the computer where the agent is installed.
SSL communication	SECURITYLEVEL	The SSL communication security level that you want. Choose one of the following settings: <ul style="list-style-type: none"> Disabled. The workstation does not use SSL. This level is the default. Allow incoming (ENABLED). The workstation uses SSL authentication only if its domain manager or a subordinate workstation requires it. Upward forced (ON). The workstation uses SSL authentication when it connects to a parent domain manager. The workstation refuses any non-SSL connection from a domain manager. All forced (FORCE). The workstation accepts only SSL authenticated connections. Other connections are refused.
SSL port	SECUREADDR	The SSL port number, as installed on the agent. The value must match the value that is assigned to nm ssl port in the localopts file on the computer where the agent is installed. Specify this setting if security level is set to on , force , or enabled .
Time zone	TIMEZONE, TZ	The time zone in which the system is located. The value must match the value set on the operating system. Configuring this setting incorrectly might cause jobs to run at the wrong time.
Domain	DOMAIN	The name of the workstation's Workload Scheduler domain. MASTERDM is the default name.
Extended agent options (Host)	HOST	For extended agents, the workstation of the fault-tolerant, or standard agent that is hosting the extended agent. For standard agents, the domain manager in the selected domain is always the host. The host can be set to one of these options: <ul style="list-style-type: none"> Host: another agent that controls jobs for the extended agent (<i>WORKSTATION_ID</i>) Hosted by domain manager (\$MANAGER) or Hosted by master domain manager (\$MASTER)
Access method	ACCESS	For extended agents, the access method that is used, for example, <i>r3batch</i> .

Table 1 Workstation definition fields

Field	Attribute	Description
Workload broker	HOST	For pools, dynamic pools, remote engines, and agent types, the name of the workload broker workstation.
Workstation type	TYPE	<p>One of the following selections:</p> <ul style="list-style-type: none"> • Domain manager (MANAGER) • Fault-tolerant agent (FTA) • Standard agent (S-AGENT) • Extended Agent (X-AGENT) • Dynamic agent (AGENT) • Pool (POOL) • Dynamic pool (D-POOL) • Remote engine (REM-ENG)
Ignore during plan generation	IGNORE	Include this attribute if you do not want this workstation to participate in the next production plan.
Auto link	AUTOLINK	<p>Indicates whether a link between workstations is automatically opened at start. The autolink attribute is optional. The default value is ON. Specify one of the following values:</p> <ul style="list-style-type: none"> • ON: The workstation is automatically linked when the domain manager initializes. • OFF: The workstation must be linked manually
Behind firewall	BEHINDFIREWALL	<p>Indicates whether a firewall is between the workstation and the master domain manager. The default value is OFF. Specify one of the following values:</p> <ul style="list-style-type: none"> • ON: All communications between the master domain manager and this workstation are routed through intermediate domain managers (the domain hierarchy). • OFF: Communications for controlling the workstation (starting, stopping) and retrieving job logs are routed directly to the workstation.
Full status	FULLSTATUS	<p>Only for fault-tolerant agents, specifies that the domain manager of this workstation sends all status message to this workstation. Specify one of the following values:</p> <ul style="list-style-type: none"> • ON: The agent is updated with the status of jobs and job streams on all workstations in its domain. • OFF: The agent is updated with only the status of jobs and job streams that directly affect its own jobs and job streams.

Table 1 Workstation definition fields

Field	Attribute	Description
Mailman server	SERVER	<p>This option is available only for standard agents and fault-tolerant agents. It specifies which mailman server on the domain manager services was used for the messages that are exchanged with the workstation.</p> <p>When the domain manager starts, it creates a separate mailman server process for each unique server ID. If the same ID is specified for multiple agents, a single-server process is created to handle their communications. Your domain manager must be able to create enough system processes as the <i>wsuser</i> to accommodate any additional mailman server processes.</p> <p>Because the IDs are unique to each domain manager, you can use the same IDs in other domains without conflict. When you do not specify a server, the main mailman server process on the domain manager handles communication with agents.</p>
Members	MEMBERS	For a pool workstation or workstation class, add members to the pool or class by specifying the workstations you want to add.
Protocol	PROTOCOL	Only for dynamic agent workstations, specifies the communication protocol that the connection to the agent uses. Specify http or https .
Agent ID	AGENTID	Only for dynamic agent workstations, the universally unique identifier that the agent software on the computer where the dynamic agent is running generated. Do not edit this field.
Requirement	Requirements in JSSDL format	For dynamic pools, this field specifies the requirements, in JSSDL format, that agents must satisfy to be members of the pool.

Workstation attributes requirements

REQUIRED	OPTIONAL	NOT APPLICABLE		CPU NAME	DESCRIPTION	VARIABLE	OS	NODE	TIMEZONE	TYPE
Attribute	MDM	DM	BDM	FTA/SA	BROKER	X-AGENT	AGENT	REM-AGENT	POOL	D-POOL
TCPADDR	R	R	R	R	R	R	R	R	N	N
SECUREADDR	O	O	O	O	N	N	O	O	N	N
DOMAIN	R	R	R	R	R	R	N	N	N	N
HOST	N	N	N	N	N	R	R	R	R	R
ACCESS	N	N	N	N	N	R	N	N	N	N
AGENTID	N	N	N	N	N	N	R	N	N	N
AUTOLINK	O	O	O	O	O	N	N	N	N	N
BEHINDFIREWALL	N	O	O	O	O	N	N	N	N	N
SECURITYLEVEL	O	O	O	O	N	N	N	N	N	N
FULLSTATUS	R	R	R	O	N	N	N	N	N	N
SERVER	N	N	N	O	O	N	N	N	N	N
PROTOCOL	N	N	N	N	N	N	R	R	N	N
MEMBERS	N	N	N	N	N	N	N	N	R	N
REQUIREMENTS	N	N	N	N	N	N	N	N	N	R

© Copyright IBM Corporation 2017

Workstations attributes requirements

The characteristics of each type of workstation and what kinds of agents are used for each type are shown on the slide. Some attributes are required or optional, depending on which type of workstation you define.

When you create a workstation definition, you define the characteristics that uniquely identify the system and how it controls how jobs that are run on it. These settings are extra attributes that you can configure.

- The **master domain manager** (MDM) workstation is created when you install Workload Scheduler. It is a fault-tolerant agent with added characteristics.



Attention: Being prepared for network problems makes recovery easier. Set up a backup domain manager or backup dynamic domain manager for each domain manager or dynamic domain manager in your network. This way, you ensure that Workload Scheduler peak job scheduling loads are met. Choose any fault-tolerant agent in the domain to be a backup domain manager.

- A **backup master domain manager** (BKM) acts as a backup for the master domain manager when problems occur. The backup domain manager is installed like the master domain manager.



Important: Verify that the master domain manager and the backup master domain manager have **Full Status** turned on in the workstation definition. Also, verify that the backup master domain manager is synchronized in time with the master domain manager. The secure way is to use a *Network Time Protocol Server* to control the time on both systems, with the same repeat interval.

Full status is important if you must resort to long-term recovery, where the backup master domain manager creates the production plan. If full status is not turned on, the former master domain manager is a regular fault-tolerant agent after the production plan is extended.

During typical operations, the production plan extension process automatically turns on the Full Status flag for the active master domain manager, if it is not already turned on. When a new master domain manager extends the plan, it does not recognize the former master domain manager as a backup master domain manager unless the full status flag is enabled. If the full status flag is not set, the former master domain manager does not have an accurate production plan when it switches back.



Note: A backup master domain manager is more effective at keeping your Workload Scheduler environment available than placing a master domain manager into a clustered environment. The backup master domain manager is also useful for maintaining the Workload Scheduler network while you upgrade Workload Scheduler on the master domain manager.

- A **domain manager** (DM) controls a Workload Scheduler domain and shares management responsibilities for part of the network. Domain managers are installed as fault-tolerant agents and then configured as domain managers in the Workload Scheduler database.
- A **backup domain manager** (BDM) is a backup for the domain manager. The backup domain manager is configured in the same domain as its domain manager.
- A **dynamic domain manager** adds dynamic workload brokering capabilities to a domain manager. When a dynamic domain manager is installed, all the communication to and from dynamic agents in the domain are routed through the dynamic domain manager. At least two workstations are configured to run on the dynamic domain manager computer: the domain manager and a workload broker workstation. Extra components are installed on the domain manager that add workload brokering capabilities. The workstation definition for dynamic domain managers is the same as domain managers.
- The **backup dynamic domain manager** acts as a backup for the dynamic domain manager when problems occur. Its use is optional.
- The **dynamic agent** (DA) manages various job types. The dynamic agent workstation is automatically created in the Workload Scheduler database when you install the dynamic agent. You can view properties of dynamic agents in the Dynamic Workload Console. The dynamic agent registers with the dynamic domain manager periodically and updates the resource

advisor database with current activity. Depending on the global option setting for `enAddWorkstation`, dynamic agents are also automatically added to the production plan. Do not change the attributes of dynamic agent workstations without consulting IBM software support.

- A **pool workstation** groups sets of dynamic agents into one workstation. Workload Scheduler balances the number of jobs that run on each agent within the pool. When an agent in the pool becomes unavailable, Workload Scheduler automatically runs jobs on other agents in the pool. You create pool workstations in the database and select which dynamic agents are members of the pool. You can create as many pool workstations as you need, and dynamic agents can be members of many pools. The number of agents you assign to the pool determine its size.
- A **dynamic pool workstation** is similar to a pool workstation. The workstation grouping for dynamic pools is defined dynamically based on resource requirements that you select in the workstation definition. Workload Scheduler chooses which agent in the pool runs each job based on characteristics that you define in the workstation object. The number of agents that meet the requirements you specify determine the size of the pool; it can grow or shrink automatically. For example, you select one or more of the following characteristics:
 - The operating system type, for example, Windows, Linux, or AIX
 - Maximum processor usage, for example, not over 80% processor usage
 - Ordered workstations, which is a list of dynamic workstations

You select how Workload Scheduler selects an agent from the pool for each job that it runs:

- Balancing workload
- Optimizing processor usage
- Workstation with the most or least logical resources
- Workstation that uses the most or least logical resources
- **IBM Workload Scheduler agent for z/OS** runs on mainframe computers that run z/OS. The z/OS agent is like a dynamic agent. It registers itself to the Dynamic Workload Broker and automatically creates a workstation definition in the Workload Scheduler database. When you create jobs for the dynamic agent workstation for z/OS, you include the JCL that the job runs. Workload Scheduler agent for z/OS workstations can be members of pools.
- **Remote engine** workstations represent a remote Workload Scheduler engine, like the one you create in the Dynamic Workload Console. You use the remote engine workstation to run shadow jobs. A shadow job runs locally and maps to another job that runs on a remote engine. The status of the shadow job reflects the actual status of the job that runs on the remote engine. You create job definitions on the remote engine workstation. You create relationships between jobs that run locally and jobs that run on other Workload Scheduler or Workload Scheduler for z/OS networks. These relationships are called cross dependencies. The remote engine workstation uses a connection based on http or https protocols for the environments to communicate.

Creating workstation classes

You create workstation classes by using the Workload Designer. Follow these steps to create a workstation class:

1. From the Dynamic Workload Console portfolio, go to **Administration > Manage Workload Definitions** to open the workload designer.
2. Select the engine from the menu and click **Go**. The Workload Designer window opens.
3. Select **New > Workstation Class** from the **Working List** menu.
4. On the **General** tab, enter the name of the workstation class.
5. On the **Workstations** tab, click the green plus icon to add workstations to the class.
6. Click **Save** to save the workstation class to the database.

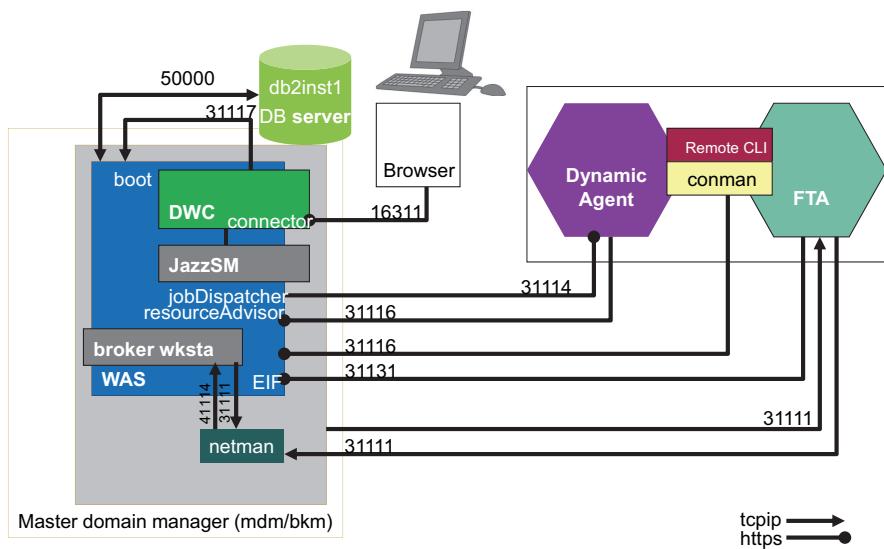
Lesson 3 Workload Scheduler networks

Workload Scheduler networks

© Copyright IBM Corporation 2017

In this lesson, you learn about some of the TCP/IP ports that IBM Workload Automation uses. You can plan your installations more easily when you understand which ports are used and where to change the port numbers. You learn how to change the port numbers during and after installation later in this class.

Workstation Scheduler component ports



© Copyright IBM Corporation 2017

Workload Scheduler component ports

The following table has information about external network ports.

Port	Listener	Description
16310	WAS (JazzSM)	http transport, connection as needed from web browsers for Dynamic Workload Console
16311	WAS (JazzSM)	https transport, connection as needed from web browser for Dynamic Workload Console
16314	WAS (JazzSM)	Administrative console (http) transport
16315	WAS (JazzSM)	Administrative console (https) transport, connection as needed from web browsers
31111	MDM and FTA	Netman port, persistent connection
31113	MDM and FTA	Secure netman port (optional), persistent connection
31114	Dynamic agents	https transport, connection as needed
31115	WAS (MDM)	http transport, connection as needed from dynamic agents, remote command line when configured to use http
31116	WAS (MDM)	https transport, connection as needed from dynamic agents, remote command line, when configured to use https
31117	WAS (MDM)	Bootstrap (RMI) port, connection as needed from graphical interfaces such as the Dynamic Workload Console
31118	WAS (MDM)	Optional SOAP port, which is used by web services interface

Port	Listener	Description
31123	WAS (MDM)	Administration http transport, used to access administration panes
31124	WAS (MDM)	Administration https transport, used to access WAS administration panes
31131	WAS (MDM)	EIF transport, which is used by the event processor to receive data about event rules
50000	DB2 instance server	Default database port, DB2 instance

Lesson 4 Checking hardware and software prerequisites

Checking hardware and software prerequisites

© Copyright IBM Corporation 2017

In this lesson, you learn about what is required on your computers before you install Workload Scheduler version 9.4. Computers on which you install Workload Scheduler must meet some basic requirements.

Only the minimum requirements are identified here and in the product documentation because only the users and administrators of any particular computer know the configuration of that computer. A computer's configuration might include other software that affects the installation and running of Workload Scheduler.



Important: The most common cause of installation failures is that the computer where the installation runs does not meet the required conditions.

The prerequisites for running Workload Scheduler and its various components change often as IBM adds support for new systems or removes support for other systems. For updated detailed systems

Lesson 4 Checking hardware and software prerequisites

requirements, links to download electronic images of the software, IBM Workload Scheduler Release Notes, and other important information, consult the following page.

<http://www-01.ibm.com/support/docview.wss?uid=swg27048858>

Workload Scheduler is installed by using **IBM Installation Manager**. The installation manager runs a scan on the computer before it installs or upgrades Workload Scheduler. The scan verifies certain facts:

- The operating system is supported.
- The necessary libraries are installed.
- Enough permanent and temporary disk space to install the product is available.
- Enough memory and virtual memory is available.

You can use some of the characteristics of Workload Scheduler to decide what hardware upgrades are more effective. For example, consider the following characteristics:

- The planning engine, event rule engine, and dynamic scheduling use memory on the master domain manager and backup domain manager. Therefore, the master domain manager has higher memory requirements.
- The internal processing of Workload Scheduler mechanisms uses I/O bandwidth on the file system, on agents and on the master domain manager. Therefore, installing Workload Scheduler on a file system with fast I/O is beneficial.
- Workload Scheduler runs on recent operating systems that are under support of their vendors. The list of supported operating systems online is updated monthly.

Software prerequisites

Master domain manager or dynamic domain manager

- A supported relational database
 - DB2
 - Oracle
 - Informix Dynamic Server
 - Microsoft SQL
- WebSphere Application Server
- WebSphere SDK Java Technology Edition

Dynamic Workload Console

- WebSphere Application Server
- Jazz for Service Management
- Dashboard Application Services Hub
- WebSphere SDK Java Technology Edition

© Copyright IBM Corporation 2017

Software prerequisites

When you install a master domain manager or dynamic domain manager, the following prerequisites apply:

- A supported *relational database*. At the release of version 9.4, Workload Scheduler supports DB2, Oracle (Enterprise and Standard Editions), Informix database server, and Microsoft SQL Server. You must install the database separately.
- If WebSphere Application Server is not installed, the installation manager automatically installs it for you.
- If WebSphere SDK Java Technology Edition is not installed, the installation manager automatically installs it for you.

When you install the Dynamic Workload Console, the following prerequisites apply:

- If WebSphere Application Server is not installed, the installation manager automatically installs it for you.
- If Jazz for Service Management is not installed, you can select to have the installation manager install it.

If you plan to use Tivoli Common Reporting, you select **Reporting Services**, an extra component of Jazz for Service Management. Reporting Services requires added system

Lesson 4 Checking hardware and software prerequisites

resources. Run the **IBM Prerequisite Scanner** before you install Reporting Services to ensure a successful installation.

- If **Dashboard Application Services Hub** is not installed, the installation manager automatically installs it for you.
- If **WebSphere SDK Java Technology Edition** is not installed, the installation manager automatically installs it for you.

Lesson 5 Preparing a database

Preparing the database

© Copyright IBM Corporation 2017

The master domain manager, backup master domain manager, dynamic domain managers, and their backups connect to the relational database. In this lesson, you learn about requirements for running Workload Scheduler with a database. Installing and configuring the database server are outside the scope of this class.

Choosing which database to use depends on the environment and situation in which the Workload Scheduler network is situated. The database can be installed on the same computer as Workload Scheduler, or on a different computer. DB2 V11 is prepackaged with Workload Scheduler. You can install it by using the Workload Scheduler launchpad. These are also supported database systems, but they must be installed separately:

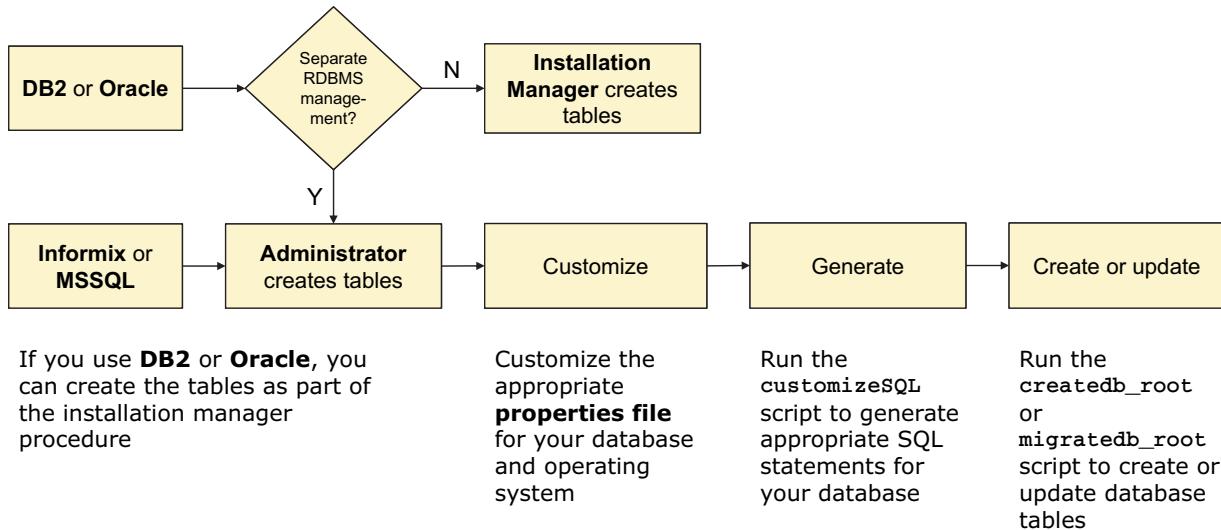
- Oracle RDBMS
- Informix database server
- Microsoft SQL



Note: You can install *DB2 Server* and the master domain manager on the same computer. You can alternatively install *DB2 Client* on the computer that acts as the master domain manager and install *DB2 Server* on a different computer. The advantage of this configuration is that you can more easily switch between your master domain manager and the backup master domain manager.

You can install *Oracle Enterprise Edition* or *Oracle Standard Edition*. With Oracle Enterprise Edition, you can implement the partitioning feature to improve the performance of event-driven workload automation.

Preparing the databases and tables



© Copyright IBM Corporation 2017

Preparing the databases and tables

You can create the Workload Scheduler database tables before you install or upgrade the master domain manager. In this way, a database administrator can run the procedure to create or upgrade the database tables for Workload Scheduler without sharing the database user ID and password.

If you use DB2 or Oracle database management systems, you can choose to have IBM Installation Manager create or update the Workload Scheduler tables. You can instead manually create the tables. If you use Informix or Microsoft SQL Server, you must create the tables manually.

To manually create or update the database tables, perform the following steps.

1. Open the properties file that is found on the installation media, in the `dbtools` folder. Customize the SQL properties with the values that match your requirements.
2. Generate SQL files for your database by running the `customizesQL` script, passing the name of the properties file you customized.
3. Run the `createdb_root` script to create the database tables, or `migratedb_root` script to update database tables.

Lesson 6 IBM Workload Automation users

IBM Workload Automation users

© Copyright IBM Corporation 2017

When you plan to install and run Workload Scheduler, you must understand the user accounts that are associated with installing the product, users that own the installation, and people who use the product.

Workload Automation user types



- **The installer:** The user who installs the product
- **The wsuser:** The Workload Scheduler user that owns the components that are installed on the systems
- **The principals:** The primary account for managing the WebSphere Application Server profiles
- **The users:** The users who create definitions, monitor the system, and manage Workload Scheduler
- **The stream logon users:** The users who run jobs under the control of Workload Scheduler

© Copyright IBM Corporation 2017

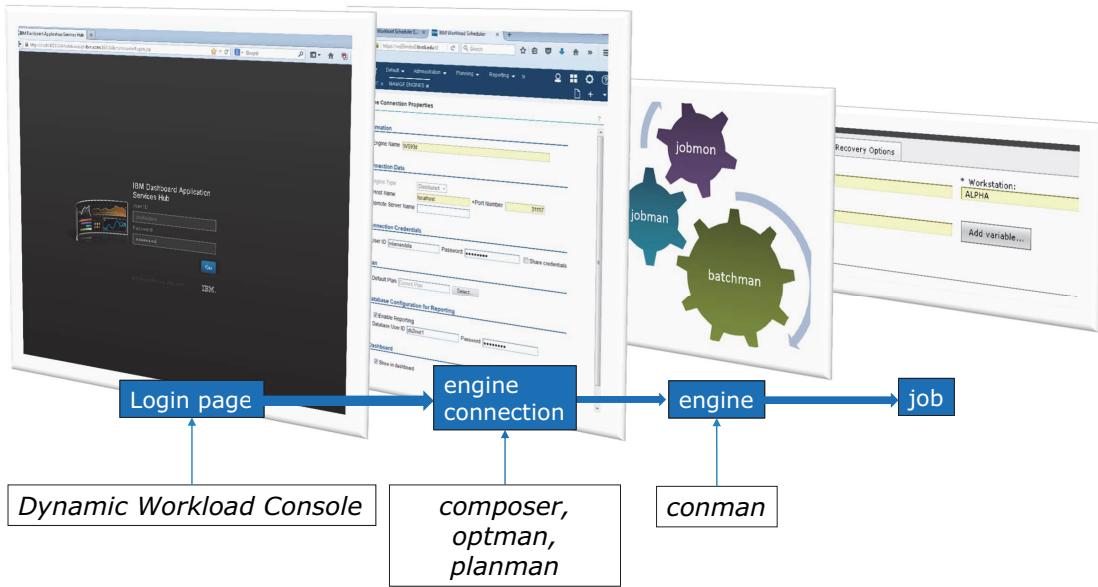
Workload Automation user types

Users who interact with IBM Workload Automation install, run, and use Workload Scheduler. Workload Scheduler manages jobs that run as users. The types of users that interact with Workload Scheduler are listed in the following table.

Table 2 Workload Automation users

User type	Description
The installer	The user who installs the product.
The wsuser	The Workload Scheduler user that owns the components that are installed on the systems
The principals	The primary account for managing the WebSphere Application Server profiles. For Workload Scheduler application server, the primary account is the same as the wsuser. For the Dynamic Workload Console application server, you specify the principal name during the installation process.
The users	Users who create definitions, monitor the system, and manage Workload Scheduler
The stream logon users	Users who run jobs under the control of Workload Scheduler

Users' entry points



Users' entry points

Depending on the method a user employs to access the Workload Scheduler environment, their authentication to Workload Scheduler might change.

The installer

When you install Workload Scheduler, you must have the authorization and privileges necessary to install software. These authorizations are required to install or upgrade Workload Scheduler:

- For UNIX and Linux operating systems, **root** access is required.
- For Windows operating systems, the login you use to install Workload Scheduler must be one of these types:
 - A member of the Windows **Administrators** group
 - A domain administrator with **Act as Part of the Operating System**

The wsuser

Workload Scheduler components run on the computer where they are installed under the permissions and environment of a user that you assign. This Workload Scheduler user, or **wsuser**, has a unique user ID from other users on the computer where you install Workload Scheduler. The Workload Scheduler user owns the Workload Automation instance that is created for each installation. The Workload Scheduler user ID with few exceptions owns the files in the instance, and most processes run under the permissions of the Workload Scheduler user.

On UNIX and Linux operating systems, regardless of the method of installation you choose, the Workload Scheduler user must be created manually before you start the installation. Use the appropriate UNIX and Linux operating system commands to create the user.



Important: Some operating systems require that for users with a password, the password must be changed at the first login. In that situation, for a successful installation, you must log in as the user and change the password for the first time.

When you install on Windows operating systems, you must not install two agents on the same system with the same user name or account. In other words, do not install one on that is defined on the local system and one that is defined on the domain with the same name.

The Windows user who installs the product must be a member of the local administrative group, if the **wsuser** is a local Windows user. The Windows user who installs the product must be a member of the domain administrative groups in the domain controller if the **wsuser** is a domain user.

Workload Scheduler users who use Windows operating systems require specific user rights. You can grant them locally. Domain-level policies always override local policies. You might have to grant the rights from the domain. The following user rights are required:

- Act as part of the operating system
- Allow logon locally
- Impersonate a client after authentication
- Logon as a batch job and as a service
- Replace process level token

These rights are granted during the installation, but you can confirm them manually.

The principals

When you install the application server on the Workload Scheduler master domain manager, the *principal* contains the credential data for managing the services. The principal name is the same as the **wsuser**, and is automatically configured.

When you install the Dynamic Workload Console, you must provide an account name for the principal that manages the application server and Jazz for Service Management .

You cannot use the same user ID more than once in a WebSphere Application Server authentication realm. For example, you cannot have the same user ID in different LDAP directories, even under different organizational structures. You cannot have the same user ID in different repositories, for example in a file repository and in an LDAP directory.



Important: When you install the Dynamic Workload Console, specify a principal (account) name that does not exist in the local operating system, or in any LDAP directories you intend to reference for authentication.

The users

People who log on to a Workload Scheduler user interface and create definitions, manage scheduling activities, and monitor them must be defined to the Workload Scheduler security system.

The *stream logon* users

The ***stream logon*** user is part of the job definition in Workload Scheduler. The *stream logon* user must exist on the computer where the agent runs the job.

Exercises

- Create a domain
- Create a workstation definition
- Monitor dynamic workload broker computers
- View or modify workstations by using composer



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. Describe how the plan is distributed in a multiple-domain Workload Scheduler network.
2. What are the two main parts of the dynamic workload broker?
3. What three criteria does the resource advisor use to select the most capable resource to run a job?
4. What is the function of a gateway agent?
5. What are the benefits of setting up a backup domain manager?
6. What are the default TCP ports for the netman service and the WAS https transports to which dynamic agents connect?
7. What is the most common cause of Workload Scheduler installation failures?
8. Describe the functions of the Workload Scheduler user (wsuser or twsuser).

Review answers

1. Describe how the plan is distributed in a multiple-domain Workload Scheduler network.

The master domain manager distributes the plan to all of the first-level domain managers. The domain managers then distribute the plan to the agents in their domains.

2. What are the two main parts of the dynamic workload broker?

The job dispatcher and the resource advisor are the two main parts of the workload broker.

3. What three criteria does the resource advisor use to select the most capable resource to run a job?

Availability of consumable resources, resource selection policies, and workload distribution process are three criteria the resource advisor uses.

4. What is the function of a gateway agent?

Gateway agents manage network connections between other agents and the dynamic domain manager.

5. What are the benefits of setting up a backup domain manager?

Set up a backup domain manager to prepare for network or other problems with the domain manager.

6. What are the default TCP ports for the netman service and the WAS https transports (to which dynamic agents connect)?

The netman port is 31111, and the WAS port for dynamic agents is 31116.

7. What is the most common cause of Workload Scheduler installation failures?

The most common cause of installation failures is that the computer where the installation runs does not meet the required conditions.

8. Describe the functions of the Workload Scheduler user (**wsuser** or **twsuser**).

The Workload Scheduler user has a unique user ID from other users on the computer where you install Workload Scheduler, and owns the components that are installed.

Summary

You should now be able to perform the following tasks:

- Design a Workload Scheduler network topology
- Identify TCP ports occupied by a Workload Scheduler installation
- Create a domain hierarchy
- Prepare a database
- Identify hardware and software prerequisites
- Create workstation and domain definitions
- Identify user privileges

© Copyright IBM Corporation 2017

Unit summary

Unit 3 Installing and configuring Workload Scheduler

IBM Training



Installing and configuring Workload Scheduler

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to plan, prepare, and install Workload Scheduler.

References:

- *IBM Workload Scheduler Planning and Installation, Version 9.4*
- *IBM Workload Scheduler Troubleshooting, Version 9.4*
- *IBM Workload Scheduler User's Guide and Reference, Version 9.4*
- *IBM Workload Scheduler Download Documents, System Requirements, Release Notes, Version 9.4*

Objectives

In this unit, you learn how to perform the following tasks:

- Plan your installation of Workload Scheduler
- Choose the best method for installing Workload Scheduler and other workload automation products
- Describe the components of a Workload Scheduler implementation
- Configure global options

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Preparing for the installation

Preparing for the installation

© Copyright IBM Corporation 2017

Preinstallation planning involves checking some configurations of the computers and collecting some information that you require during the installation.



Important: Read the *IBM Workload Scheduler Planning and Installation* guide. It contains instructions that are specific to different operating systems and tips for avoiding failed installations.

Workload automation related service

- A supported **relational database**
- A **master domain manager** or backup master domain manager, which includes these components:
 - IBM WebSphere Application Server
 - WebSphere SDK Java Technology Edition
 - IBM Workload Scheduler
- **Dynamic Workload Console**, which includes these components:
 - Jazz for Service Management
 - Dashboard Application Services Hub
 - WebSphere SDK Java Technology Edition
 - Workload Scheduler for z/OS connector

© Copyright IBM Corporation 2017

Workload Automation related services

The Workload Automation products and components use **IBM WebSphere Application Server** as the communication infrastructure and **Jazz for Service Management** for user interfaces. You can use one instance of WebSphere Application Server with Workload Scheduler and the Jazz for Service Management components that connect the Dynamic Workload Console to run those components on the same computer. Other components of Workload Scheduler, such as the command-line client, do not use the WebSphere Application Server.

Jazz for Service Management overview

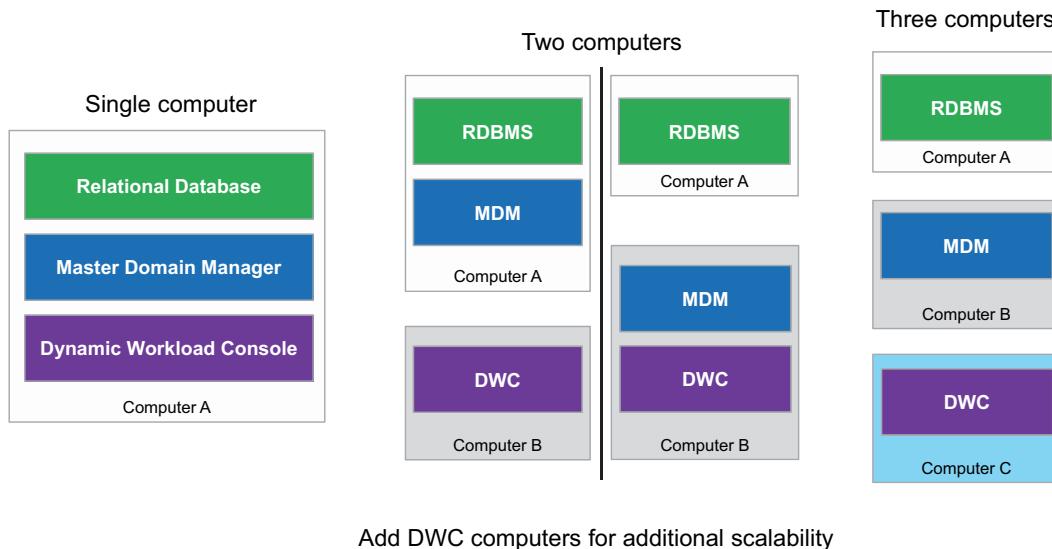
Jazz for Service Management (JazzSM) brings together open specifications for linking data and other shared integration services, including a linked data registry service, administration services, dashboard service, reporting service, and security services. It underpins client-defined management scenarios such as cloud, performance monitoring, and IT Service Management. You

install Jazz for Service Management as part of the Dynamic Workload Console. Jazz for Service Management provides these integration services:

- **Registry Services**, providing a shared repository of your applications and resources, enabling linked data while you use *Open Services for Lifecycle Collaboration* (OSLC)
- On demand, self-service reporting through **Tivoli Common Reporting**
- Consistent UI experience through **Dashboard Application Services Hub** (DASH)
- Simplified Administration of products and solutions that integrate through Jazz for Service Management
- Security authentication to enable single sign-on support across both WebSphere and non-WebSphere applications

By using Jazz for Service Management and DASH, you can combine the user experience of several IBM software products in a single user interface.

Workload Automation Collocation



© Copyright IBM Corporation 2017

Workload Automation collocation

You can install one of each the following items on a single computer:

- A supported relational database
- A master domain manager or backup master domain manager, which includes these components:
 - IBM WebSphere Application Server
 - WebSphere SDK Java Technology Edition
 - IBM Workload Scheduler
- Dynamic Workload Console, which includes these components:
 - Jazz for Service Management
 - Dashboard Application Services Hub
 - WebSphere SDK Java Technology Edition
 - Workload Scheduler for z/OS connector

You can also run the database, master domain manager, and Dynamic Workload Console on separate computers. You can install more than one agent on a single computer, and you can combine fault-tolerant agents and dynamic agents on one computer.

Product installation paths

WAhome/TWS	Workload Scheduler master domain manager and fault-tolerant agents
└── TWS/ITA/cpa	Dynamic agent
└── TDWB	Dynamic Workload Broker components
└── WAS	WebSphere Application Server
└── wastools	Application server tools

WAhome is a path you choose, which is by default one of these options:

- /opt/IBM/TWA on UNIX and Linux
- C:\Program Files\IBM\TWA on Windows

/opt/IBM/JazzSM	Jazz for Service Management
/opt/IBM/TWAUI	Dynamic Workload Console

Some files are installed outside of these directories

© Copyright IBM Corporation 2017

Product installation paths

The installation of Workload Scheduler has a **wsuser** and an installation directory, **WAhome**.



Important: The home directory of the **wsuser** should be different from the **WAhome**.

Component locations

You install files for Workload Scheduler components and their prerequisite components in the **WAhome** directory that you choose. The default directory differs with the operating system:

- On Windows operating systems, **C:\Program Files\IBM\TWA**
- On UNIX operating systems, **/opt/IBM/TWA**
- On Linux operating systems, **/opt/ibm/TWA**

Lesson 1 Preparing for the installation

If you install a new instance onto a computer that has an existing **TWA** directory, a new default installation directory is created, for example, **TWA1** or **TWA2**. You can change the installation directory to a different one during the installation.

UNIX and Linux installation path	Windows installation path	Components installed
WAhome/TWS	WAhome\TWS	IBM Workload Scheduler master domain manager and fault-tolerant agents
WAhome/TWS/ITA/cpa	WAhome\TWS\ITA\cpa	Workload Scheduler dynamic agent uses the default path structure and also installs files here. The agent also installs some files outside this path.
WAhome/TDWB	WAhome\TDWB	Files for the dynamic workload broker capability are installed here.
/opt/IBM/TWAUI	C:\Program Files\IBM\TWAUI	The Dynamic Workload Console is installed here. You can choose a different location.
WAhome/WAS	WAhome\WAS	The WebSphere Application Server is automatically installed when you create a new Workload Automation instance. You can choose a different location.
/opt/IBM/JazzSM	C:\Program Files\IBM\JazzSM	For the Dynamic Workload Console, the Jazz components are installed here.
WAhome/wastools	WAhome\wastools	The application server tools are installed here. You can use the tools that are provided in this directory to do WebSphere Application Server administration tasks.

Files that are installed outside the *WAhome* path

When you install Workload Scheduler, some files are created outside of the *WAhome* path.

Table 1 Other directories that you create when you install Workload Scheduler

Windows operating systems	UNIX and Linux operating systems
<ul style="list-style-type: none"> • %WINDIR%\TWA • %WINDIR%\TWS • %WINDIR%\system32\TWSRegistry.dat (32-bit installations on 32-bit computers) • %WINDIR%\sysWOW64\TWSRegistry.dat (32-bit installations on 64-bit computers) • %WINDIR%\TWSRegistry.dat (64-bit installation on 64-bit computers) • %WINDIR%\teb • %WINDIR%\cit • %ProgramFiles%\tivoli\cit (or the path that is specified in %WINDIR%\cit\cit.ini) 	<ul style="list-style-type: none"> • /etc/TWA • /etc/TWS • /etc/teb • /etc/cit • /etc/init.d/tebclt-tws_cpa_agent_instance • /usr/Tivoli/TWS • /usr/ibm/tivoli/common/CIT/logs • /opt/tivoli/cit (or the path that is specified in /etc/cit/cit.ini).

The following list shows the directories that are created outside of *WAhome* when you install the Dynamic Workload Console and Workload Scheduler for z/OS connector.

Table 2 Other directories that you create when you install the Dynamic Workload Console

Windows operating systems	UNIX and Linux operating systems
Dynamic Workload Console: <ul style="list-style-type: none"> • %WINDIR%\TWA z/OS connector: <ul style="list-style-type: none"> • %WINDIR%\TWA • %WINDIR%\system32\TWSRegistry.dat (32 bits) • %WINDIR%\sysWOW64\TWSRegistry.dat (32-bit software on 64-bit computers) • %WINDIR%\TWSRegistry.dat (64-bit software on 64-bits computers) 	Dynamic Workload Console: <ul style="list-style-type: none"> • /etc/TWA z/OS connector: <ul style="list-style-type: none"> • /etc/TWA • /etc/TWS

The following list shows the directories that are created outside of *WAhome* when you install a dynamic agent.

Table 3 Other directories that are created when you install dynamic agents

Windows operating systems	UNIX and Linux operating systems
<ul style="list-style-type: none">•%windir%\teb\teb_tws_cpa_agent_twsuser.ini•%ALLUSERSPROFILE%\Application Data\ibm\CAP\EMICPA_default.xml	<ul style="list-style-type: none">•/etc/teb/teb_tws_cpa_agent_twsuser.ini•/opt/IBM/CAP/EMICPA_default.xml•/etc/init.d/tebctl-tws_cpa_agent_twsuser (on Linux and Solaris)•/etc/rc.d/init.d/tebctl-tws_cpa_agent_twsuser (on AIX)•/sbin/init.d/tebctl-tws_cpa_agent_twsuser (on HP-UX)

Installation registries

- Information about the installation locations and component versions and settings are kept in the installation registries
- Upgrade and fix pack tools use this information when upgrading or applying fixes

UNIX/Linux file	Windows file	Registry contents
/etc/TWA/twainstance0.TWA.properties	%WINDIR%\TWA\twainstance0.TWA.properties	Installation location of components
/etc/TWS/TWSRegistry.dat	%WINDIR%\system32\TWS\TWSRegistry.dat	Information about Workload Scheduler base components

© Copyright IBM Corporation 2017

Installation registries

After you install a new Workload Automation instance, a registry of instance information can be found in the following files:

- On UNIX and Linux, in `/etc/TWA/twainstance0.TWA.properties`, where **0** is incremented for each new instance.
- On Windows, in `%WINDIR%\TWA\twainstance0.TWA.properties`, where **0** is incremented for each new instance.

When you run the installation program, it checks to determine whether Workload Scheduler is already installed, and if so, the version and location of components. After installation, a registry of what was installed is in the following files:

- On Linux and UNIX:

`/etc/TWS/TWSRegistry.dat`

- On Windows:

`%WINDIR%\system32\TWS\TWSRegistry.dat`

The registry contains the following items:

Table 4 TWSRegistry.dat

Attribute	Value
ProductID	TWS_ENGINE
PackageName	The name of the software package that is installed

Table 4 TWSRegistry.dat (continued)

Attribute	Value
InstallationPath	The absolute path of the Workload Scheduler instance
UserOwner	The owner of the installation
MajorVersion	Workload Scheduler version number
MinorVersion	Workload Scheduler release number
MaintenanceVersion	Workload Scheduler maintenance version number
PatchVersion	The last product patch number installed
Agent	The type of agent that is installed, for example, standard agent (SA), fault-tolerant agent (FTA), for master domain manager (MDM)
FeatureList	The list of other features installed

Example: A TWSRegistry.dat file on a master domain manager:

```
/Tivoli/Workload_Scheduler/twsuser_DN_objectClass=OU
/Tivoli/Workload_Scheduler/twsuser_DN_PackageName=FP_TWS_LINUX_X86_64_twsuser.9
.4.0.00
/Tivoli/Workload_Scheduler/twsuser_DN_MajorVersion=9
/Tivoli/Workload_Scheduler/twsuser_DN_MinorVersion=4
/Tivoli/Workload_Scheduler/twsuser_DN_PatchVersion=0
/Tivoli/Workload_Scheduler/twsuser_DN_FeatureList=
/Tivoli/Workload_Scheduler/twsuser_DN_ProductID=TWS_ENGINE
/Tivoli/Workload_Scheduler/twsuser_DN_ou=twsuser
/Tivoli/Workload_Scheduler/twsuser_DN_InstallationPath=/opt/IBM/TWA/TWS
/Tivoli/Workload_Scheduler/twsuser_DN_UserOwner=twsuser
/Tivoli/Workload_Scheduler/twsuser_DN_MaintenanceVersion=0
/Tivoli/Workload_Scheduler/twsuser_DN_Agent=MDM
```

Lesson 2 Choosing an installation method

Choosing an installation method

© Copyright IBM Corporation 2017

To install a fault-tolerant agent or dynamic agent, you use the **twsinst** script. For components other than agents, you have three installation methods from which to choose. The method that you use depends on the type of installation, your access to the computer where you run the tasks, and other factors. You can install a master domain manager, its backup, a dynamic domain manager, or its backup by using one of the following methods:

- Launchpad
- Installation wizard
- Silent installation



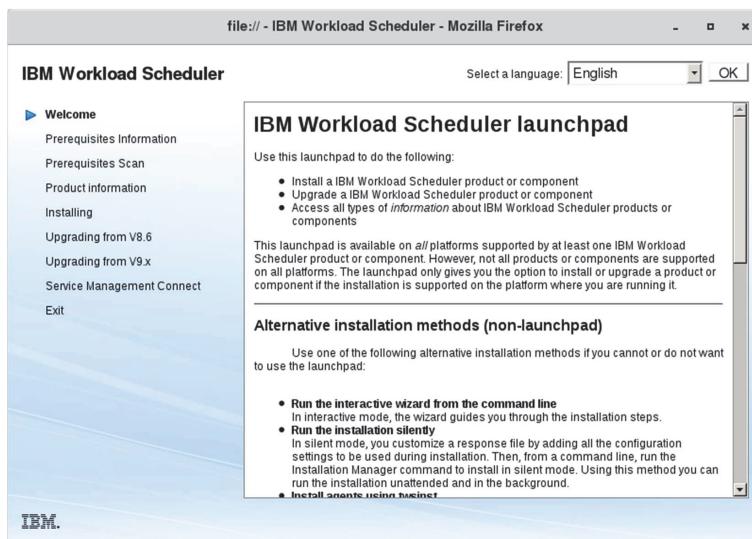
Important: Install patches to Workload Scheduler with the same method that you used to install the product.

The installation wizard and silent mode installations start the IBM Installation Manager. You use IBM Installation Manager to install and maintain software packages. Within Installation Manager, you can configure repositories from which to install software packages and software updates. You

Lesson 2 Choosing an installation method

can integrate Installation Manager with IBM Passport Advantage and IBM Fix Central to provide service repositories to keep your software up to date.

Using the launchpad



Use the Launchpad for these tasks

- **Check** the system to verify that it meets **system prerequisites** for installing Workload Scheduler
- **Install** or **upgrade** one or all Workload Scheduler components
- **Install** or **upgrade** the Dynamic Workload Console and the Workload Scheduler for z/OS Connector
- **Access** product information

© Copyright IBM Corporation 2017

Using the launchpad

The launchpad is a starting point for installing products that are part of Workload Automation. From the launchpad, you can perform these tasks:

- Check the system to verify that it meets system prerequisites for installing Workload Scheduler.
- Install or upgrade one or all Workload Scheduler components, including WebSphere Application Server and Jazz for Service Management.
- Install or upgrade the Dynamic Workload Console and the Workload Scheduler for z/OS Connector.
- Access product information.

The launchpad requires some additional installation prerequisites, such as a supported web browser. For more information, see the *IBM Workload Scheduler System Requirements Document* at the following website:

<http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27045181>

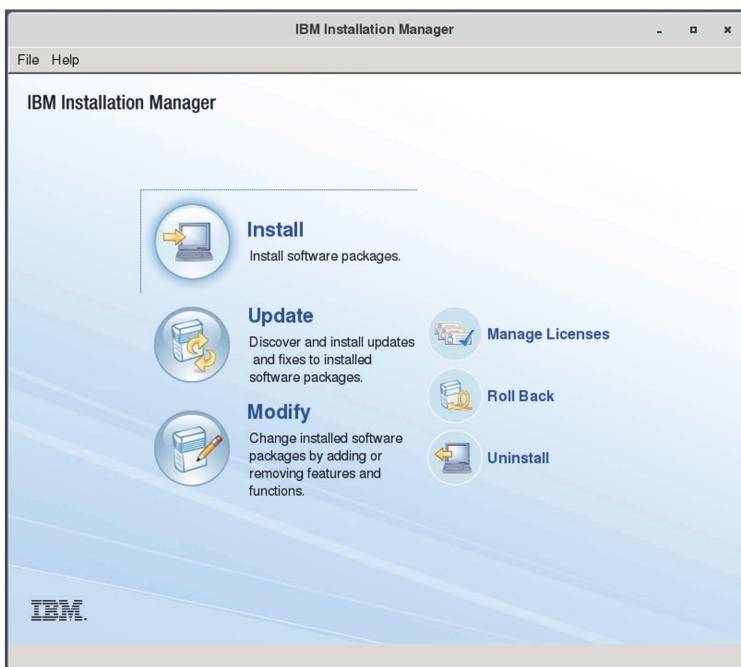


Note: Use the launchpad on systems where a web browser is installed and you want to read detailed information about installations. The launchpad starts the installation wizard that shows the installations that are provided.

Using the installation wizard

Use the Installation wizard for these tasks

- **Install** or **upgrade** one or all Workload Scheduler components
- **Install** or **upgrade** the Dynamic Workload Console and the Workload Scheduler for z/OS Connector
- **Record** a response file for silent installations
- **Download** fixes and updates



Using the installation wizard

You can use the installation wizard to install Workload Scheduler master domain managers, backup masters, dynamic domain managers, and the Dynamic Workload Console. Start the individual setup files for each supported operating system and use the wizard interactively through each of the installation steps. You can use the installation wizard where you have a graphical user environment and you want to interactively enter configuration information manually.

You can also use the installation wizard to record a response file that you later use to install Workload Scheduler silently.

Using silent mode

```
<profile id='IBM Workload Scheduler'  
installLocation='/opt/IBM/TWA'>  
  <data key='user.offeringId,com.ibm.tws'  
value='com.ibm.tws' />  
  <data  
key='user.isSilentUpgrade,com.ibm.tws'  
value='false' />  
  <data key='user.userName,com.ibm.tws'  
value='wsuser' />  
  <data key='user.password,com.ibm.tws'  
value='xjuUsEX6tQjDsiiVZuOeqg==' />  
  <data key='user.mdmlsBackup,com.ibm.tws'  
value='FALSE' />  
  <data  
key='user.mdmCompany,com.ibm.tws'  
value='IBM' />  
  <data key='user.ftaThisCpu,com.ibm.tws'  
value='WS94MDM0' />  
  <data key='user.ftaMasterCpu,com.ibm.tws'  
value='WS94MDM0' />  
...  
</profile>
```

When to use silent mode

- When no graphical desktop environment is available
- When you want to repeat similar installations

Follow these steps

- To use silent mode, you must install IBM Installation Manager (IBMIM)
- Install **IBM Installation Manager**
- Edit a copy of a suitable response file template, or use Installation Manager to record a new one
- Run the **imcl** tool specifying the response file that you create

© Copyright IBM Corporation 2017

Using silent mode

A silent installation runs according to settings configured in a response file. The response file includes all of the information that is required to complete the installation without user intervention.



Important: You must first install *IBM Installation Manager* before you can silently install Workload Scheduler.

You can create the response file by using a text editor. Examples of different types of response files are included in the **response_files** directory on the installation media. You can use one of the examples that most closely matches the type of installation or update that you want as a template. A complete description of the response file entries is available in the *IBM Workload Scheduler Planning and Installation* guide.

You must encrypt the passwords that you include in the response file before the installation. Use the IBM Installation Manager command to encrypt a text string:

`IBMIM encryptString secretPassword`

Installing silently with a response file

To install Workload Scheduler by using a response file, complete these steps:

1. Copy the response file to a local directory, and edit its contents to match your configuration.
2. Change to the IBM Installation Manager tools directory.

3. Run the installer command.

- On Windows

```
imcl.exe input directory\response_file.xml -log directory\log_file.xml  
-acceptLicense
```

- On UNIX and Linux

```
./imcl input directory/response_file.xml -log directory/log_file.xml  
-acceptLicense
```

The installation continues without user intervention. When the installation is complete, you can view the results in the **summary.log** file in the system temporary directory.

Using twsinst to install agents

Syntax:

```
twsinst -new -uname username
  [-acceptlicense yes|no]
  [-addjruntime true|false]
  [-agent dynamic|fta|both]
  [-company company_name]
  [-create_link]
  [-displayname agentname]
  [-gateway local|remote|none]
  [-gwid gateway_id]
  [-hostname hostname]
  [-inst_dir install_dir]
  [-jimport port_number]
  [-jimportssl true|false]
  [-lang lang_id]
  [-master master_cpu_name]
  [-port port_number]
  [-reset_perm]
  [-skip_usercheck]
  [-stoponcheckprereq]
  [-tdwbnhostname host_name]
  [-tdwbpport tdwbpport_number]
  [-thiscpu workstation]
  [-work_dir working_dir]
```

The **twsinst** command is the preferred way to install agents

Example

```
./twsinst -new -uname wsuser
  -inst_dir /opt/ibm/TWA -agent both
  -master WS94MDM0
  -hostname here.example.com
  -displayname MYAGENT0
  -company AnyBank -addjruntime true
  -tdwbnhostname ws94mdm0.example.com
  -tdwbpport 31116
  -thiscpu FTAGENT0 -acceptlicense yes
```

© Copyright IBM Corporation 2017

Using twsinst to install agents

You use the **twsinst** command to install the Workload Scheduler agents. If you are installing the dynamic agent, you can use the **twsinst** command to add the Java runtime necessary to run job types with advanced options.

The user that runs the installation command must meet the following authorization requirements:

- On Windows operating systems, if you set the Windows User Account Control (UAC), your login account must be a member of the Windows **Administrators** group or domain administrators with the rights **Act as Part of the Operating System**. If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **Administrator**.
- On UNIX and Linux operating systems, you must have **root** access.

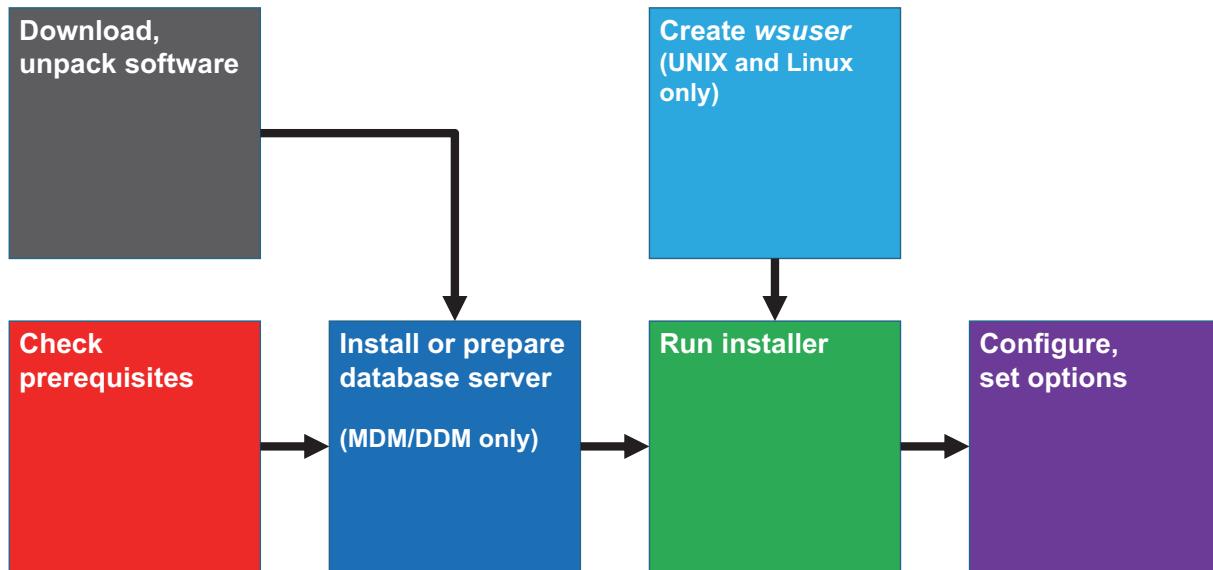
The **twsinst** command parameters and options are listed in the *IBM Workload Scheduler Planning and Installation* guide. A sample installation of a UNIX agent has syntax similar to the following one.



Hint: Using the **twsinst** command

```
twsinst -new -uname username
  -acceptlicense yes|no [
  -addjruntime true|false] [
  -agent dynamic|fta|both] [
  -company company_name] [
  -create_link] [
  -displayname agentname] [
  -gateway local|remote|none] [
  -gwid gateway_id] [
  -hostname hostname] [
  -inst_dir install_dir] [
  -jimport port_number] [
  -jimportssl true|false] [
  -lang lang_id] [
  -master master_cpu_name] [
  -port port_number] [
  -reset_perm] [
  -skip_usercheck] [
  -stoponcheckprereq] [
  -tdwbhostname host_name] [
  -tdwbport tdwbport_number] [
  -thiscpu workstation] [
  -work_dir working_dir]
```

Installation scenario



© Copyright IBM Corporation 2017

Installation scenario

The steps that outline an installation of all of the components of Workload Scheduler are listed on the slide. First, ensure that the computer meets the prerequisites for the software that you are installing. Install the database and create a user for the Workload Automation instance. Then, install Workload Scheduler and the Dynamic Workload Console.



Note: IBM Workload Scheduler V9.4 also packages *IBM Workload Scheduler for Applications* to control your SAP or PeopleSoft workloads.

For the agents, ensure that the computer meets prerequisites for the software that you are installing.

Lesson 3 Configuring Workload Scheduler

Configuring Workload Scheduler

© Copyright IBM Corporation 2017

In this lesson, you learn how to change configuration options for Workload Scheduler.

Setting global configuration options

```
optman chg option = value
optman ls
optman show option
```

Auditing and security

- auditStore / as = BOTH
- enPlanAudit / pa = 1
- enDbAudit / da = 1
- enListSecChk / sc = YES
- enRoleBasedSecurityFileCreation / rs = YES|NO
- enLogonBatch / lb = YES

Planning and scheduling

- enCarryForward / cf = ALL
- startOfDay / sd = 0000
- enForecastStartTime / st = YES
- enAddUser / au = YES
- enAddWorkstation / aw = YES
- workstationLimit / wl = YES

Reporting

- statsHistory / sh = 365

Sending electronic mail

- mailSenderName / ms = user@dom
- smtpServerName / sn = smtphost
- smtpServerPort / sp = 25
- smtpUseAuthentication / ua = YES|NO
- smtpUseSSL / us = YES|NO
- smtpUseTLS / tl = YES|NO
- smtpUserName / un = user
- smtpUserPassword / up = secret

© Copyright IBM Corporation 2017

Setting global configuration options

When you install Workload Scheduler, the installer configures and sets global options. There are over 50 global options to set. They change the way Workload Scheduler functions in many ways, and they affect the system globally. In other words, they do not affect only one workstation, but rather all workloads on all workstations. The installation uses some default values that you might change according to your requirements. In this lesson, you learn about some important settings to change.

You show or change the configuration of a global option by using the **optman** command. To show the settings of the global options, use the **optman ls** command. To show help text for a particular option, use the **optman show option** command, where *option* is the configuration for which you want to show help.

Auditing and security settings

You can change five auditing and security-related settings, depending on your requirements:

- The **auditStore** option (**as**) sets the type of storage to be used to log database audit records. Enter one of the following values:
 - **file**: To specify that a text file in the **WAhome/TWS/audit/database** directory is used to store the audit records (the default value)
 - **db**: To specify that the Workload Scheduler database itself is used to store the audit records
 - **both**: To have audit records that are logged in both the file and the database

Any change of this value is effective immediately.

- The **enPlanAudit** option (**pa**) enables or disables plan auditing.
 - To enable plan auditing, specify **1**.
 - To disable plan auditing, specify **0**.

Auditing information is logged to a text file in the **WAhome/TWS/audit/plan** directory. Each Workload Scheduler workstation maintains its own log. For the plan, only actions are logged in the auditing file, not the success, or failure of any action. The default value is 1. Run **JnextPlan** to activate this change.

- Use the **enDbAudit** option (**da**) to enable or disable database auditing.
 - To disable database auditing, specify **0**.
 - To activate database auditing, specify **1**.

Auditing information is logged to a text file in the **WAhome/TWS/audit/database** directory, to the Workload Scheduler database itself, or to both. To choose one, set the **optman** property **auditStore**. Each Workload Scheduler workstation maintains its own log. Only actions are logged, not the success or failure of the action. Installation of dynamic domain managers and dynamic agents is not recorded in audit logs.

For more information about using this feature, see the section about auditing facilities in the *IBM Workload Scheduler Troubleshooting Guide*. The default value is 1. Changes to this parameter are effective immediately.

- The **enRoleBasedSecurityFileCreation** option (**rs**) specifies whether to use the role-based security model. The role-based security model is explained in [Unit 7, “Securing Workload Scheduler”](#). The default is no.
 - To enable role-based security modeling, specify **yes**.
 - To use the classic security model, specify **no**.
- Change the **enListSecChk** setting (**sc**). You change it to control the objects in the plan that a user can list when they run a query on the Dynamic Workload Console or a **conman show object** command.
 - If set to **yes**, objects in the plan that are returned from a query or show command are shown to the user only if the user is granted the list permission in the security file.
 - If set to **no**, all objects are shown, regardless of the settings in the security file.

Setting this option to yes affects how the graphical user interfaces function for the users who are defined in the security file. The default value is no. Run **JnextPlan** to activate this change for the plan. For the database, this option takes immediate effect.

- The **enLogonBatch** option (**lb**) is for Windows jobs only.
 - If set to **yes**, the logon users for Windows jobs are automatically granted the right to Log on as batch job.
 - If set to **no** or omitted, the right must be granted manually to each user or group.

Because the right cannot be granted automatically for users that run jobs on a backup domain controller, you must grant those rights manually. The default value is **no**. Run **JnextPlan** to activate this change.

Planning and scheduling

The following settings affect how Workload Scheduler creates production plans. Their effects are covered in the production planning unit. Review the following three global options settings:

- The **enCarryForward** option (**cf**) is used to enable *carry forward*. This preproduction option affects the operation of the **stageman** command. Change the enCarryForward setting to specify whether job streams that did not complete are carried forward from the old to the new production plan (Symphony).
 - Enter **yes** to have incomplete job streams that carried forward only if the Carry Forward option is enabled in the job stream definition.
 - Enter **all** to have all incomplete job streams that carried forward, regardless of the Carry Forward option.
 - Enter **no** to completely disable the Carry Forward function.

If you run the **JnextPlan -for 0000** command and the Carry Forward option is set to either **yes** or **no**, a message is displayed. The message indicates that incomplete job streams might not be carried forward. When the **stageman -carryforward** command is used, it overrides enCarryForward. For more information, see *IBM Workload Scheduler User's Guide and Reference*. If this option is set to **no**, running jobs are moved to the **USERJOBS** job stream. The default value is **all**. Run JnextPlan to activate this change.

- Change the **startOfDay** option (**sd**) to modify the start time of processing day. Specify the start time of the Workload Scheduler processing day in 24-hour format: **hhmm** (0000 - 2359). The default value is 0000, or 12:00 a.m. midnight. Run **JnextPlan** to activate the startOfDay setting.
- Set the **enForecastStartTime** option (**st**) to enable forecast start time. The option is only applicable when workload service assurance is enabled (see **enWorkloadServiceAssurance**).
 - Enter **yes** to enable the calculation of the predicted start time of each job when you run a forecast plan. Enabling this feature might negatively affect the time that is taken to generate the forecast plan.
 - Enter **no** to disable the calculation of the predicted start time of each job when you run a forecast plan. When this option is set to **yes**, the **enPreventStart** global option is ignored during the creation of forecast plans.

The default value is **no**. Any change of this value is effective immediately.

- Set the **enAddUser** option (**au**) to configure whether to automatically add a Windows user to the production plan (Symphony) after you create or modify the user in the Workload Scheduler

database. Valid options are **yes**, to enable automatically adding users, or **no**, to disable this feature. The default value is **yes**. Changes to this option are effective immediately.

- Set the **enAddWorkstation** option (**aw**) to configure whether to automatically add dynamic agent workstations to the production plan (Symphony) after you add them to the database. Specify **yes** to enable this feature, or **no** to disable it. The default value is **no**. Changes to this option are effective immediately.
- Change the **workstationLimit** option (**wl**) to set the job limit a new dynamic agent assumes when it is first added to the production plan. The default value is **100**. Changes to this option are effective immediately. You can change the workstation limit by using the command-line interface or the Dynamic Workload Console.

Reporting

You might want to change the **statsHistory** option (**sh**), which sets the job statistics history period. This setting specifies the number of days for which you want to maintain job statistics. Statistics are discarded on a first-in, first-out (FIFO) basis. For example, if you use 10, statistics are maintained for the last 10 days. The statsHistory setting has no effect on job standard list files, which must be removed with the **rmstdlist** command. For information about the **rmstdlist** command, see the *IBM Workload Scheduler User's Guide and Reference*

Run **JnextPlan** to activate this change in the plan. For the database, this option takes effect immediately.

Sending electronic mail

With the event-driven workload automation feature, you define event rules, which send an electronic mail in response to activity in the network. For example, an event rule can send email when any job ends in error.

To use the email sender plug-in for event rules, you must correctly configure the SMTP options to match the settings of your local mail servers. You can change these settings:

- The **smtpServerName** option (**sn**) sets the SMTP server name. Specify the DNS name or IP address of the SMTP server to be used by the mail plug-in.
The default value is **localhost**. Changes to this parameter are effective immediately.
- Set the **smtpServerPort** option (**sp**) to the SMTP Server port on the mail server. Valid values are in the range 0–65535. The default value is 25.
Changes to this parameter are effective for the next mail send action.
- The **smtpUseAuthentication** option (**ua**) defines whether the mail plug-in uses SMTP authentication. Specify **yes** if the SMTP connection must be authenticated. The default is **no**.

Changes to this parameter are effective immediately.

- The **mailSenderName** option (**ms**) sets the mail sender name. Specify a string to be used as the sender of the emails.

The default value is **TWS**. Changes to this parameter are effective for the next mail send action.

- The **smtpUserName** option (**un**) defines the SMTP server user name. The default value is the name of the Workload Scheduler user (the *wsuser*) on the master domain manager.

Changes to this parameter are effective immediately.

- Set the **smtpUserPassword** option (**up**) to the password of the *smtpUserName* user. The password is stored in an encrypted form. Changes to this parameter are effective immediately.
- Set the **smtpUseSSL** option (**us**) to yes if the mail plug-in uses SSL. Values are **yes** or **no**.

The default is no. Changes to this parameter are effective immediately.

- Set the **smtpUseTLS** option (**tl**) to **yes** if the mail plug-in SMTP connection authenticates by using the Transport Layer Security (TLS) protocol. Accepted values are **yes** or **no**.

The default is **no**. Changes to this parameter are effective immediately.

Lesson 4 Customizing the Dynamic Workload Console

Customizing the Dynamic Workload Console

© Copyright IBM Corporation 2017

In this lesson, you learn how to override the default configuration options for the Dynamic Workload Broker.

When you install the Dynamic Workload Console, the installer configures and sets default options. The options that you set change the way the Dynamic Workload Console functions in many ways, and they affect the system globally. In other words, they do not affect only one user, but rather all users who connect to the console, or all members that have the role that you specify in the configuration. Some settings can be limited to specific Dashboard Application Services Hub roles. The installation uses some default values that you might change according to your requirements. In this lesson, you learn about the settings you can change.

Setting Dynamic Workload Console configuration options

1. Create **JazzSMprofileDir/profile/registry/TdwcGlobalSettings.xml**
2. Add settings and properties you want to configure
3. Log in to the Dynamic Workload Console

You can:

- Limit the objects that are displayed in graphical views
- Choose whether to automatically create monitoring tasks
- Audit the mobile interfaces (SSC/SSD)
- Limit who shares monitoring tasks and engine connections
- Display or hide resolved dependencies
- Limit the display in the What-If Analysis interface
- Link the monitoring tasks to online job and job stream documentation

Template

```
<?xml version="1.0"?>
<tdwc>
  <settings>
    <!-- settings that apply to all users -->
  </settings>
  <settings role="ops2">
    <!-- settings that apply to ops2 users -->
  </settings>
</tdwc>
```

© Copyright IBM Corporation 2017

Setting Dynamic Workload Console configuration options

To change the console global settings, you edit the following XML file, or create it if it does not exist.

JazzSMprofileDir/profile/registry/TdwcGlobalSettings.xml



Note: By default, **JazzSMprofileDir** is **/opt/IBM/JazzSM** on Linux and UNIX environments, and **C:\Program Files\IBM\JazzSM** on Windows environments.

Settings that you change become effective when a user logs in to the Dynamic Workload Console.

The **TdwcGlobalSettings.xml** has the following basic format.



Note: A basic global settings file

```
<?xml version="1.0"?>
<tdwc>
  <settings>
    <!-- settings that apply to all users -->
  </settings>
  <settings role="ops01">
    <!-- settings that apply to only users with role ops01 -->
  </settings>
</tdwc>
```

Settings that you can customize

In the following section, you learn about some of the console settings you can customize.

Limiting the graphical views

You can limit how many objects a user sees when they open the Plan View view. The default value for all limits is 1000. 1000 is also the maximum value for all limits.

```
<graphViews>
  <property name="planViewMaxJobstreams" value="497"></property>
</graphViews>
```

Automatically creating monitoring tasks

When a user first logs in to the Dynamic Workload Console, monitoring tasks are automatically created. You see the tasks when you choose, **System Status and Health > All Configured Tasks** from the menus.

You can change the way monitoring tasks are created by configuring the `precannedTaskCreation` settings as follows:

Table 5 `precannedTaskCreation` settings

Setting	Description
<code>all</code>	Tasks are created for distributed and z/OS engines. <code>all</code> is the default setting.
<code>distributed</code>	Predefined tasks are created only for distributed engines
<code>zos</code>	Predefined tasks are created only for z/OS engines
<code>none</code>	No tasks are created.

In the following example, the Workload Console creates monitoring tasks only for distributed engines.

```
<application>
  <property name="precannedTaskCreation" value="distributed"/>
</application>
```

Auditing the mobile interfaces

You can configure the Dynamic Workload Console to log activities that users who use the **Self-Service Catalog** (SSC) and **Self-Service Dashboard** (SSD) applications perform to a file. You change how the applications are audited by configuring `SSCMAuditing` settings in the `TdwcGlobalSettings.xml` file.

The properties that you can configure are shown in the following table.

Setting	Value	Description
SSAuditing	true	Operations users perform in the Self-Service Catalog and Self-Service Dashboards applications are tracked in an auditing log file. The log files are at the following directory. <i>JazzSMprofileDir/logs/server1/audit_SSC_SSD.log</i> . The log files contain the creation, modification and deletion dates, the operations performed in the mobile applications, and the user who completes the operations.
	false	Do not record Self-Service Catalog and Self-Service Dashboards actions.
SSAuditingLogSize	<i>num</i>	The maximum size of a log file in KB. When a log file reaches the maximum size, the system rolls that log file over and creates a new file. By default, the maximum size of a log file is 100 KB.
SSAuditingLogFile	<i>num</i>	The default number of log files to create. When this number is met and the latest log file reaches its maximum size, the system deletes the oldest log file and rolls the latest file over and creates a new file.

The following example shows how to enable auditing, set the file size to 249 kb, and maintain three files.

```
<SSCMIAuditing>
  <property name = "SSAuditing"      value="true"></property>
  <property name = "SSAuditingLogSize"  value="249"></property>
  <property name = "SSAuditingLogFile" value="3"></property>
</settings>
```

Sharing tasks and engine connections

You can configure the Workload Console to prevent users from sharing their tasks and engine connections. By default users can share their tasks and engine connections. To change the task and engine connection sharing behavior, configure the following properties in the **security** section:

Table 6 Task and engine connection sharing options

Property	Option setting description
limitShareTask	Set to true to prevent users from sharing tasks. The default is false .
limitShareEngine	Set to true to prevent users from sharing engine connections. The default is false .
groupIdMap	Set to the LDAP query mapping you use to display the names of groups with which you can share tasks. By default, the common name (cn) is displayed.

The following example shows how to prevent users from sharing tasks and engine connections.

```
<security>
  <property name="limitShareTask"      value="true" />
  <property name="limitShareEngine"    value="true" />
</security>
```

Displaying all dependencies

When you view the list of dependencies in **Monitor jobs** and **Monitor job streams** query tasks, by default, only dependencies that are not satisfied are shown. To display all dependencies, regardless of their status, change properties in the `showDependencies` section.

Set the value of the `AlwaysShowAllDependencies` property to **true** to show all dependencies, or **false** to show only unsatisfied dependencies. The default is **false**.

The following example shows how to always display all dependencies.

```
<ShowDependencies>
  <property name = "AlwaysShowAllDependencies" value="true"></property>
</ShowDependencies>
```

Customizing the What-If Analysis interface

You can disable the What-if Analysis interface by setting the `enWhatIfAnalysis` global option to **no**. The default value is **yes**. Use the `optman` command to change the setting, and run `JnextPlan` (or await the completion of the `FINAL` job stream) to make this change effective.

When you open the What-If Analysis interface, all of the predecessors that are related to the object you select are loaded into the Gantt view. You can hide the predecessors by setting the property `whatIfAutoLoadPreds` to **true**. To revert to the default behavior, set the property to **false**, or comment the `WhatifAnalysis` section in the `TdwcGlobalSettings.xml` file.

You can set the maximum number of jobs that are displayed in the Gantt view by setting the `whatifJobsMaxNumber` property in the `WhatifAnalysis` section of the `TdwcGlobalSettings.xml` file.

The following example shows how to hide all dependencies, and set the maximum number of jobs that are displayed to 497.

```
<WhatifAnalysis>
  <property name = "whatIfAutoLoadPreds" value="true"></property>
  <property name = "whatIfJobsMaxNumber" value="497"></property>
</WhatifAnalysis>
```

Linking to online job and job stream documentation

1. Store proprietary information about your jobs and job streams in a document management tool such as a wiki
2. Configure the Workload Console **twsObjectDoc** setting
3. Access documents directly from the job and job stream query tasks

The screenshot shows the 'Monitor Workload' interface. The 'More Actions' menu is open, displaying options like 'Predecessors...', 'Successors...', 'Hold', 'Release', 'Priority...', 'Kill', 'Cancel Pending Critical Path', 'Confirm SUCC', 'Confirm ABEND', 'Download Job Log', 'Properties...', 'Properties (New Window)...', 'Show Rerun Instances', 'Job Run History', 'Show Spool List', 'Restart Process Chain', 'Recent Activity', and 'Santralinius V'. The 'Recent Activity' option is highlighted with a red box.

Status	Internal Status	Job	Type	Workstation	Scheduled Time
<input type="checkbox"/> Successful	SUCC	J1DFW_CREAK_32547	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Successful	SUCC	J1DFW_FLUID_3579	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Successful	SUCC	J1DFW_FPOST_35718	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Successful	SUCC	J1DFW_MOREL_33509	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Successful	SUCC	J1DFW_SALLY_35736	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Successful	SUCC	J1DFW_SHRED_33521	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Successful	SUCC	J1DFW_SOUSE_32502	executable	ALPH	8/14/17 12:00 AM
<input checked="" type="checkbox"/> Successful	SUCC	J1DFW_TAPIP_33624	executable	ALPH	8/14/17 12:00 AM
<input type="checkbox"/> Waiting	HOLD	J1DFW_AROMA_32530	executable	ALPH	8/15/17 12:00 AM
<input type="checkbox"/> Waiting	HOLD	J1DFW_ATILT_32490	executable	ALPH	8/15/17 12:00 AM
<input type="checkbox"/> Waiting	HOLD	J1DFW_BORNE_32579	executable	ALPH	8/15/17 12:00 AM
<input type="checkbox"/> Waiting	HOLD	J1DFW_CHEEP_33560	executable	ALPH	8/15/17 12:00 AM
<input type="checkbox"/> Waiting	HOLD	J1DFW_CONST_33619	executable	ALPH	8/15/17 12:00 AM

© Copyright IBM Corporation 2017

Linking to online job and job stream documentation

You can link from the Dynamic Workload Console monitoring tasks for jobs and job streams to online proprietary documentation by configuring the **twsObjectDoc** settings.

The **twsObjectDoc** settings form URLs where you can access stored, customized documentation about your jobs or job streams. You also customize the name of the menu item that appears in the **More actions** menu of job and job stream monitoring tasks.

To form the URLs, you create URL templates in the `TdwcGlobalSettings.xml` file. In each URL template, you include variables that the Dynamic Workload Console replaces with data from the job or job stream that you select in the query.

The variables that you can use in the template are shown in the following table.

Object type	Variable name	Description
Job	<code>job_name_w</code>	The name of the job
	<code>job_number_w</code>	The job number
	<code>job_wkst_w</code>	The name of the workstation on which the job runs
	<code>job_jsname_w</code>	The name of the job stream that contains the job
	<code>job_jswkst_w</code>	The name of the workstation on which the job stream that contains the job runs
	<code>job_id_w</code>	The ID of the job, in the format <code>WORKSTATION;SCHEDID;JOBNAME</code> or a UUID, for example, <code>0caf4156-8109-37f4-846a-2e6408f43ee4</code>
	<code>job_returncode_w</code>	The return code of the job, or blank for jobs that did not run.
	<code>job_starttime_w</code>	The start date and hour of the job, in the format: <code>YYYY-MM-DDThh</code>
Job stream	<code>js_name_w</code>	The name of the job stream
	<code>js_wkst_w</code>	The name of the workstation on which the job stream runs
	<code>js_id_w</code>	The job stream ID, for example, <code>0AAAAAAAAAAAAGQS</code>
	<code>js_latest_start_w</code>	The latest time at which a job stream can start in epoch time with milliseconds. For example, <code>1502337840000</code>
Job and job stream	<code>engine_name_w</code>	The name of the engine connection
	<code>engine_host_w</code>	The host name of the engine connection
	<code>engine_port_w</code>	The port number of the engine connection
	<code>engine_plan_w</code>	The name of the select plan, or blank for the current plan. For example, <code>M201708150202</code>

You set the `customActionLabel` property to the label you want to appear on the **More Actions** menu.

In the following example, you view a **Monitor Jobs** query task, and select the job `J1DFW_FROST_35718` in job stream `SK1DFW_TMK` on workstation `ALPHA`. When you select **More Actions >Documents**, a new browser opens with the following address.

`http://docsrv/jobs/ALPHA.SK1DFW_TMK.J1DFW_FROST_35718`



Hint: Example of Workload Console dynamic links to online documentation

```
<twsObjectDoc>
<property name="jobstreamUrlTemplate"
value="http://docsrv/jobstreams/${js_wkst_w}.${js_name_w}" />
<property name="jobUrlTemplate"
value="http://docsrv/jobs/${job_wkst_w}.${job_jsname_w}.${job_name_w}" />
<property name="customActionLabel" value="Documents" />
</twsObjectDoc>
```

Managing settings per role

You can configure Workload Console settings that are applied only to users with a Dynamic Workload Console role. For example, you can allow users with the **TWSWEBUIAdministrators** role to share engine connections and monitoring tasks, but prohibit other users from sharing.

To configure settings for a Workload Console role, create a **settings** section with a **role** attribute. Include settings properties for that role in the new section.

In the following example, the following rules apply:

- Predefined distributed tasks are created only for users with the **TWSWEBUIOperator** role.
- Only users with the **TWSWEBUIAdministrator** role can share tasks and engine connections.

```
<?xml version="1.0"?>
<tdwc>
    <settings>
        <!-- settings that apply to all users -->
        <application>
            <property name="precannedTaskCreation" value="none"/>
        </application>
        <security>
            <property name="limitShareTask"      value="true" />
            <property name="limitShareEngine"   value="true" />
        </security>
    </settings>
    <settings role="TWSWEBUIAdministrator">
        <!-- settings that apply to only users with role TWSWEBUIAdministrator -->
        <security>
            <property name="limitShareTask"      value="false" />
            <property name="limitShareEngine"   value="false" />
        </security>
    </settings>
    <settings role="TWSWEBUIOperator">
        <!-- settings that apply to only users with role TWSWEBUIOperator -->
        <application>
            <property name="precannedTaskCreation" value="distributed"/>
        </application>
    </settings>
</tdwc>
```


Exercises

- Find instance and installation information
- Run IBM Installation Manager
- Check global options settings



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. Which (major) Workload Scheduler components can be installed on a single computer?
2. On UNIX and Linux systems, where is the Workload Scheduler registry installed?
3. What package must you install before silently installing Workload Scheduler?
4. Which command do you use to install Workload Scheduler agents?
5. Which command do you use to display all of the global option settings?

Review answers

1. Which (major) Workload Scheduler components can be installed on a single computer?

The relational database, the master domain manager, and the Dynamic Workload Console can be collocated, or installed on separate computers.

2. On UNIX and Linux systems, where is the Workload Scheduler registry installed?

The registry is at /etc/TWS/TWSRegistry.dat.

3. What package must you install before silently installing Workload Scheduler?

You must install IBM Installation Manager before you can silently install Workload Scheduler.

4. Which command do you use to install Workload Scheduler agents?

Use the twsinst command to install and update Workload Scheduler agents.

5. Which command do you use to display all of the global option settings?

Use "optman ls" to display the global options settings

6. Which command do you use to change the user name that the event-driven workload automation feature uses to authenticate to the SMTP server for sending email alerts?

Use "optman chg smtpUserName = newname" to set the SMTP user name.

Summary

You should now be able to perform the following tasks:

- Plan your installation of Workload Scheduler
- Choose the best installation method to install Workload Scheduler and other workload automation products
- Describe the components of a Workload Scheduler implementation
- Configure global options

© Copyright IBM Corporation 2017

Unit summary

Unit 4 Upgrading Workload Scheduler

IBM Training



Upgrading Workload Scheduler

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Workload Scheduler developers continually work to improve the software in each version. In this unit, you learn how to upgrade and maintain the IBM Workload Scheduler server, agent, and Dynamic Workload Console.

Reference: *IBM Workload Scheduler Planning and Installation*

Objectives

In this unit, you learn how to upgrade and maintain Workload Scheduler. You learn how to perform the following tasks:

- Plan and prepare your next Workload Scheduler upgrade
- Choose whether to upgrade top-down or bottom-up, directly or in parallel
- Upgrade domain managers and master domain managers
- Safely upgrade agents
- Use the agent upgrade job type to schedule agent upgrades
- Perform ongoing maintenance

© Copyright IBM Corporation 2017

Objectives

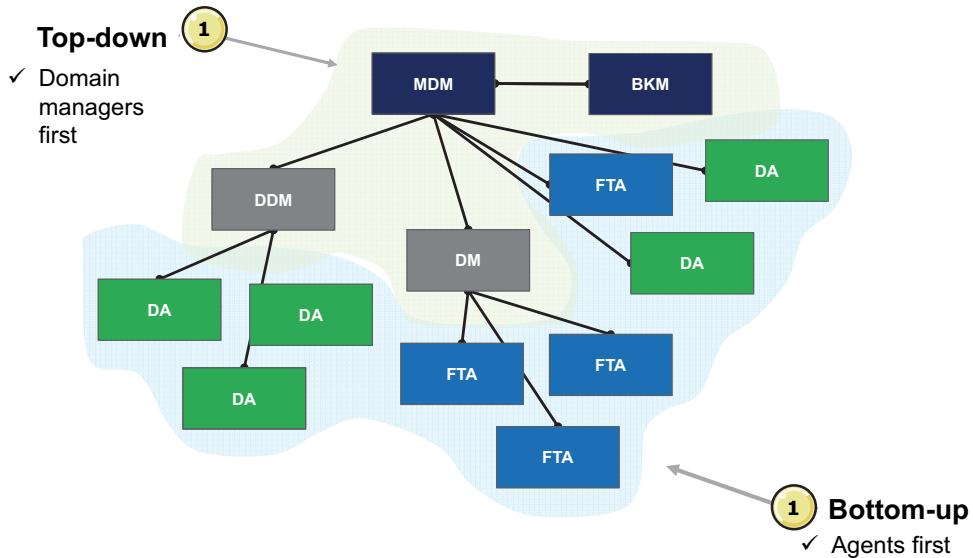
Lesson 1 Planning and preparing the upgrade

Planning and preparing for the upgrade

© Copyright IBM Corporation 2017

In this lesson, you learn how to choose which upgrade strategies to use for your next Workload Scheduler product upgrade. You can choose to upgrade agents or managers first, and whether to upgrade directly or in parallel.

Choosing top-down or bottom-up upgrade strategies



© Copyright IBM Corporation 2017

Choosing top-down or bottom-up network upgrades

Workload Scheduler supports compatibility with earlier versions of the master domain manager and agents. You can decide whether to upgrade the agents first or the master domain manager first. Master domain managers and fault-tolerant agents that run Tivoli Workload Scheduler 8.6, and higher can be upgraded to Workload Scheduler versions 9.2, 9.3, or 9.4.

Top-down

With the top-down strategy, you upgrade the master domain manager and the backup master domain manager first. Then, you upgrade the agents. The advantage is that many new features become available immediately, and others become available as the agents are upgraded. The disadvantage is that not all features are available to all the agents simultaneously.

Bottom-up

With the bottom-up strategy, you upgrade the agents first and then upgrade the master domain manager and backup master domain manager. The advantage is that new features are available to the entire Workload Scheduler network simultaneously. The disadvantage is that you cannot use new features until the entire network is upgraded.

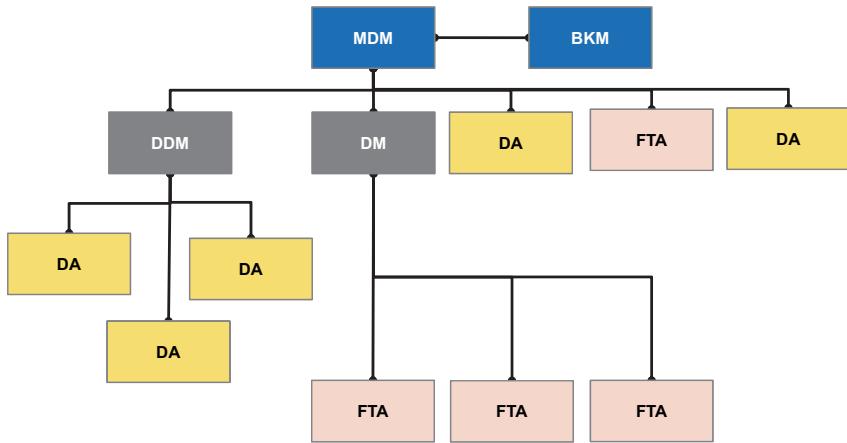
Hybrid

You might choose to use a hybrid approach, upgrading a mix of agents and domain managers. In some cases, a hybrid approach is necessary because the version of Workload Scheduler does not support the operating systems on which the software runs.

Choosing parallel or direct upgrades

Parallel

1. Upgrade backup domain manager
2. Switch to backup manager
3. Upgrade primary domain manager
4. Switch to primary domain manager



Direct

1. Upgrade domain managers *in place*

© Copyright IBM Corporation 2017

Choosing parallel or direct upgrades

By upgrading in parallel, you keep your existing version of Workload Scheduler while you upgrade to the new version. In a parallel upgrade, you install the new version of Workload Scheduler alongside the existing version. You migrate your settings and objects to the new version before you switch. You change the configuration to switch to using the new version after you are confident that everything was correctly installed. If the new version does not function correctly, you can revert your configuration and use the older version until the problem is resolved.

The advantage of the parallel upgrade is that you can switch from the old release to the new one quickly. The agents and domain managers are always at the same release level. This technique minimizes downtime to a few minutes or less. The disadvantage of the parallel upgrade is that you must also carefully copy all of the customizations from the existing version to the new version. The setup program does not inherit your settings in a parallel upgrade. The computer on which you upgrade in parallel must have enough resources to meet the requirements of both installed versions at the same time. These resources include disk space, network ports, and an extra **wsuser**.

A direct upgrade replaces the existing version with the new one, overwriting the existing software during the upgrade process. The upgraded version of Workload Scheduler inherits the settings and objects from the older version automatically. The only changes might pertain to new features found in the new release version.

The advantage of the direct upgrade is that the requirements of the computer are limited to one version of Workload Scheduler at a time. The process to upgrade the software is much simpler to complete. One disadvantage of the direct upgrade is that no jobs can be running while you upgrade the software.

Performing a safe upgrade

During a direct upgrade, the installer runs the installation tasks in *safe mode*. Safe mode ensures that the interruption during the upgrade does not risk the integrity of the running processes. The installer runs the following tasks in safe mode before it starts the upgrade:

- Checks if any Workload Scheduler command-line interfaces are running.
- Prevents any new jobs from starting, by setting the job fence on the workstation to GO (101).
- Checks whether any jobs are running. If any jobs are running, it waits for them to complete. You can specify the maximum wait time when you run the upgrade.
- Checks whether any of the agent's processes are running. If any processes are running, it stops them and waits for them to finish.

When all of these checks are complete, Workload Scheduler starts the upgrade. When the upgrade is complete, the upgrade sets the job fence back to its original value. If the upgrade is not successful or if the other checks are not successful, the job fence is not reset. In that case, you must resolve the checks and resume the upgrade manually or reset the job fence.

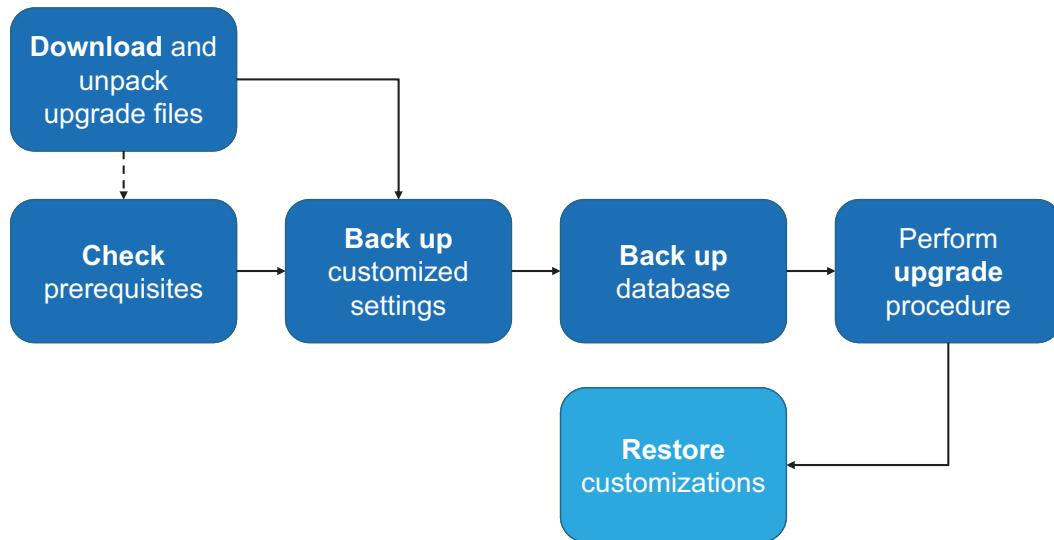
Lesson 2 Upgrading Workload Scheduler

Upgrading Workload Scheduler

© Copyright IBM Corporation 2017

In this lesson, you learn how to upgrade Workload Scheduler. The procedure for upgrading the master domain manager and backup master domain manager varies mainly in the preparation for running the upgrade commands.

Upgrading Workload Scheduler



© Copyright IBM Corporation 2017

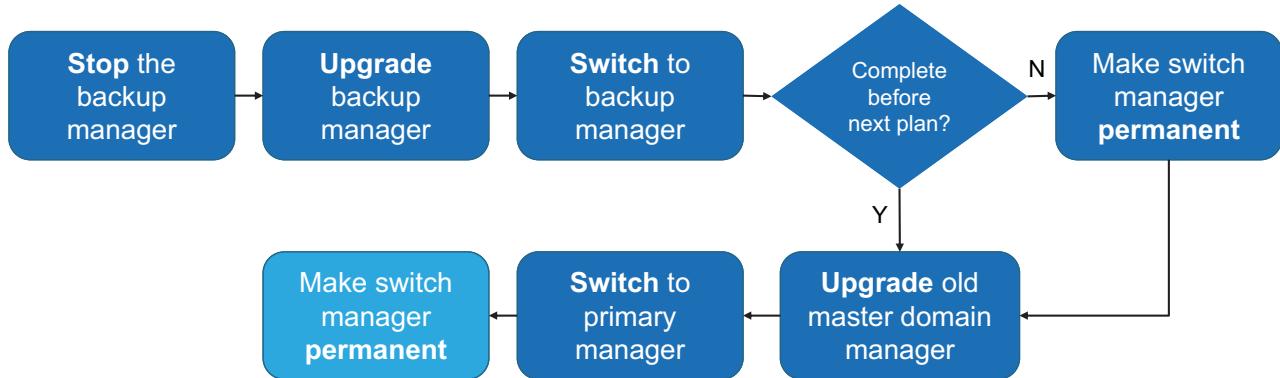
Upgrading Workload Scheduler

Before you upgrade Workload Scheduler, check that the computer has the minimum required versions of the operating system, products that are installed, and database if applicable. You can go to the IBM Software Product Compatibility website to determine the product requirements. Go to <http://www.ibm.com/software/reports/compatibility/clarity/index.html> to run a report.

Before you start the upgrade process, you can make a backup of any customized agent settings and files. The IBM Installation Manager upgrade utility makes a backup of critical information in earlier versions. You can also use the `tws_info_pull_info` to back up critical information.

Upgrading the master domain manager

Requires a backup master domain manager



© Copyright IBM Corporation 2017

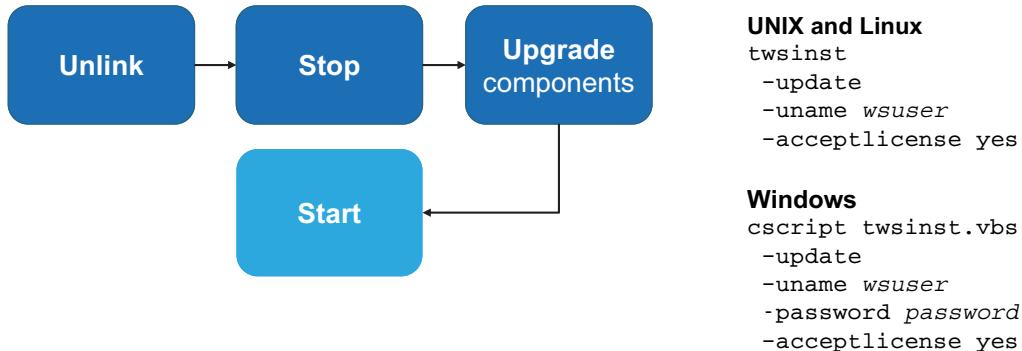
Upgrading the master domain manager

Before you start the upgrade process, make a backup of any settings and files that were customized. Also, make a backup of the Workload Scheduler database. The backup master domain manager is upgraded first. Before you upgrade Workload Scheduler on the master domain manager, switch the domain manager to the upgraded backup domain manager by using the **conman switchmgr** command. Then, perform the upgrade by using the same method that you used to install Workload Scheduler. After the upgrade is complete, switch back to the primary master domain manager by using the **conman switchmgr** command again.



Hint: Plan your upgrade of the master domain manager near the time when the FINAL schedule runs to minimize the effects to the users.

Upgrading agents



© Copyright IBM Corporation 2017

Upgrading agents

To upgrade a Workload Scheduler agent, complete the following steps.

1. You can manually unlink and stop the agent from its domain manager before you start the upgrade. You can also allow the installation program to stop and unlink the agent for you.
2. Run the upgrade procedure by starting the `twsinst` script, passing the update arguments.
For example, on UNIX and Linux systems, run the following command.
`twsinst -update -uname wsuser -acceptlicense yes`
On Windows systems, run the following command.
`cscript twsinst.vbs -update -uname wsuser -password password -acceptlicense yes`
3. When the update procedure completes, start the agent, and, if necessary, link to it from the domain manager.

Automating agent upgrades

You can upgrade agents by using **IBM Endpoint Manager**. IBM provides **Fixlets** to automatically find all the IBM Workload Scheduler agents on which to install upgrades. You can choose to schedule or immediately run an IBM Workload Scheduler upgrade.

IBM Workload Scheduler 9.4 also provides a **Centralized Agent Update** advanced job type that you can use to create jobs that upgrade Workload Scheduler agents. You learn more about centralized agent updates in [“Maintaining Workload Scheduler”](#).

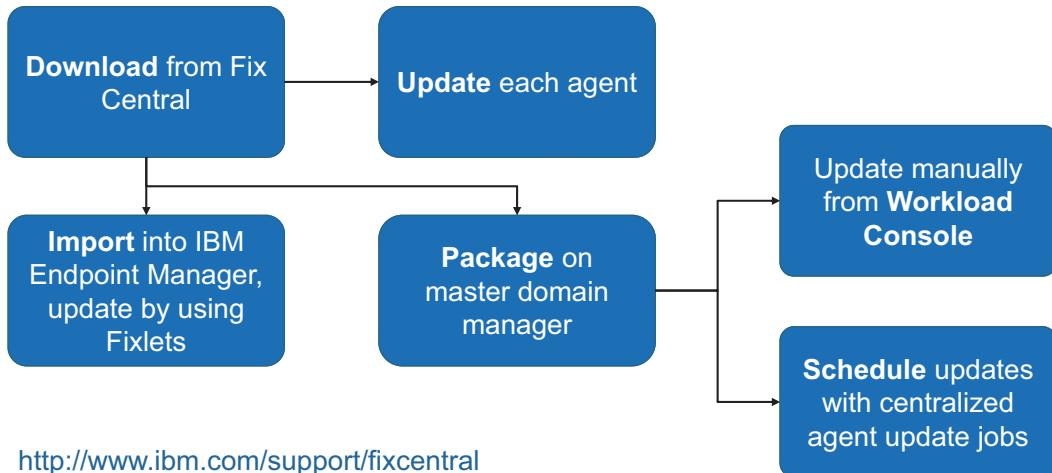
Lesson 3 Maintaining Workload Scheduler

Maintaining Workload Scheduler

© Copyright IBM Corporation 2017

In this lesson, you learn about how to maintain Workload Scheduler and apply vendor-provided fixes.

Applying product fixes



© Copyright IBM Corporation 2017

Applying product fixes

IBM periodically releases fix packs for its products on its **Fix Central** website. Go to <http://www.ibm.com/support/fixcentral/> to find and download fixes for Workload Scheduler.

The procedure for deploying new fixes is similar to upgrading to a new release. Some fix packs also add new features to Workload Scheduler. However, the contents of a fix vary from one fix release to the next. To determine which fix level is applied to your Workload Scheduler environment, use the **version** command.

Each fix package includes its own installation instructions and setup commands. You must follow the instructions that are included in the **readme file** that accompanies the fix pack.

You can install fix packs for multiple agents by downloading the packages on the master domain manager computer, and updating multiple agents by running an action from the Dynamic Workload Console. You can also schedule a task to centrally update multiple agents by using a type of job that is designed to update agents. The procedure for performing centralized updates is described in

the *Workload Scheduler Planning and Installation* guide, and in the *readme files* for the fix packs related to Workload Scheduler version 9.4. In general, you perform the following steps.

1. Download the necessary fix packs from **IBM Fix Central**, and copy them to the designated depot directory that is created (or another directory that you optionally create and configure) on the master domain manager computer.
2. Update the agents in one of the following ways. To manually update the agents manually from the Dynamic Workload Console, perform the following procedure.
 - a. Run a **monitor workstations** task in the Dynamic Workload Console, and select the agents that you want to update.
 - b. Choose **More Actions > Update agent** to update the agents.

To create an agent update job to update agents based on other scheduling criteria, perform the following procedure.

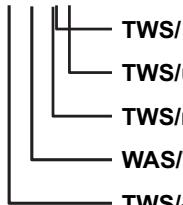
- a. From the **Workload Designer**, select **Job Definition**.
- b. Choose **Centralized Agent Update** from the list of job types.
- c. Complete the job definition according to your requirements.
- d. Submit the job manually, or schedule it by including the job in a job stream.

Maintaining the data

Back up the database

Directories that contain configuration files to preserve

WAhome/TWS



- TWS/ localopts, Security, and *msg files
- TWS/useropts/ User options files
- TWS/mozart/ Run number and global options configurations
- WAS/TWSprofile/properties Application server configuration file **TWSConfig.properties**
- TWS/{schedForecast, schedlog, schedTrial} Forecast, history, and trial plan files

Use the `tws_inst_pull_info` script to create backups

© Copyright IBM Corporation 2017

Maintaining the data

To maintain the integrity of Workload Scheduler data, follow these recommendations.

- Frequently back up the database to auxiliary storage or a backup master domain manager, following the instructions in the database documentation.
- Install and use a backup master domain manager. Update the backup master domain manager as frequently as possible by using a separate database that mirrors the database on the backup master domain manager.
- Use the supplied `tws_inst_pull_info` utility to back up the database, configuration files, and log files. `tws_inst_pull_info` is described in [Unit , “Gathering data for IBM support”](#).
- Regularly back up the log files.
- Periodically optimize the data structures in the database. DB2 and Oracle RDBMS automatically optimize their data structures by periodically running internally scheduled procedures to reorganize the data storage if necessary. You can also manually start the database reorganization by following instructions that are provided with your database management system.

Maintaining the file system

Directories that grow, and must be maintained

WAhome/TWS

TWS/stdlist/	Job logs and agent traces
TWS/audit/	Database and plan audit text files
TWS/schedlog/	History plan files
TWS/ssm/Log/	Event rule monitoring log files
WAS/TWSProfile/logs/	WebSphere Application Server log files

© Copyright IBM Corporation 2017

Maintaining the file system

The file systems where IBM Workload Scheduler and its supporting components are installed require periodic maintenance and continuous monitoring. As daily workload increases, IBM Workload Scheduler has many files and directories that grow in size.

- Each production day, Workload Scheduler refreshes the **Symphony file** to include the planned workload that runs throughout the day. The Symphony file grows to track new jobs that enter the plan, either by being scheduled, or by being submitted. You can estimate the approximate size of the Symphony file by counting the number of items of each type of data in the plan (workstations, jobs, file dependencies, and so on) and referencing the *IBM Workload Scheduler Administration* guide.
- **Message files** track interprocess and inter-workstation communication. When network communication is lost, the message files can grow quickly.

You must monitor the space on the file system where the Symphony and message files are generated. There must always be enough space to allow those files to expand to meet the demands of workload peaks or network outages. If the files cannot expand, they might become corrupted, and in some cases unrecoverable.

On the domain managers and fault-tolerant agents, jobs and processes that run create and leave log files on the file systems where the agents run. You can maintain them by scheduling the `rmstdlist` utility to remove aged log files.

If file system space on the master domain manager is too limited, and you cannot expand it, you must switch the master domain manager's duties to the backup master domain manager.

Monitoring the disk space that is used by IBM Workload Scheduler

In [Unit , “Event rules for systems administrators”](#), you learn how to use Workload Scheduler event rules to monitor the file system that is used by Workload Scheduler and to evaluate the usage of the message queues.

Exercises

- Find IBM Fix Central
- Search for product fixes



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. Which upgrade strategy, top-down, or bottom-up, do you use to upgrade agents?
2. What is an advantage of using a parallel upgrade of the master domain manager?
3. What two procedures must you complete before you start a Workload Scheduler upgrade?
4. Which command do you use to upgrade Workload Scheduler agents?

Review answers

1. Which upgrade strategy, top-down, or bottom-up, should you use to upgrade agents?

You decide which strategy to use based on your environment and its requirements.

2. What is an advantage of using a parallel upgrade of the master domain manager?

The advantage of the parallel upgrade is that you can switch from the old release to the new one quickly.

3. What two procedures must you complete before you start a Workload Scheduler upgrade?

Make a backup of the database and any customizations, and check that the computer meets the minimum prerequisites for the new software release before you upgrade.

4. Which command do you use to upgrade Workload Scheduler agents?

*Use the **twsinst** command, with the **-update** option, to upgrade Workload Scheduler agents.*

Summary

You should now be able to perform the following tasks:

- Plan and prepare your next Workload Scheduler upgrade
- Choose whether to upgrade top-down or bottom-up, directly or in parallel
- Upgrade domain managers and master domain managers
- Safely upgrade agents
- Use the agent upgrade job type to schedule agent upgrades
- Perform ongoing maintenance

© Copyright IBM Corporation 2017

Unit summary

Unit 5 Managing the production cycle

IBM Training



Managing the production cycle

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how the production planning cycle in Workload Scheduler functions. You learn how to create and extend production plans, how to create trial and forecast plans, and how to recover plans that are not correctly functioning.

Reference: *IBM Workload Scheduler Version 9.4 Administration Guide*

Objectives

In this unit, you learn how to manage the production and preproduction planning functions. You learn how to:

- Describe the production plan and preproduction plan
- Generate a preproduction plan
- Update and extend the preproduction plan
- Configure carry forward and start of day options to meet your needs
- Reset the production plan and preproduction plan
- Recover a corrupted production plan
- Schedule regular updates of the production plan

© Copyright IBM Corporation 2017

Objectives

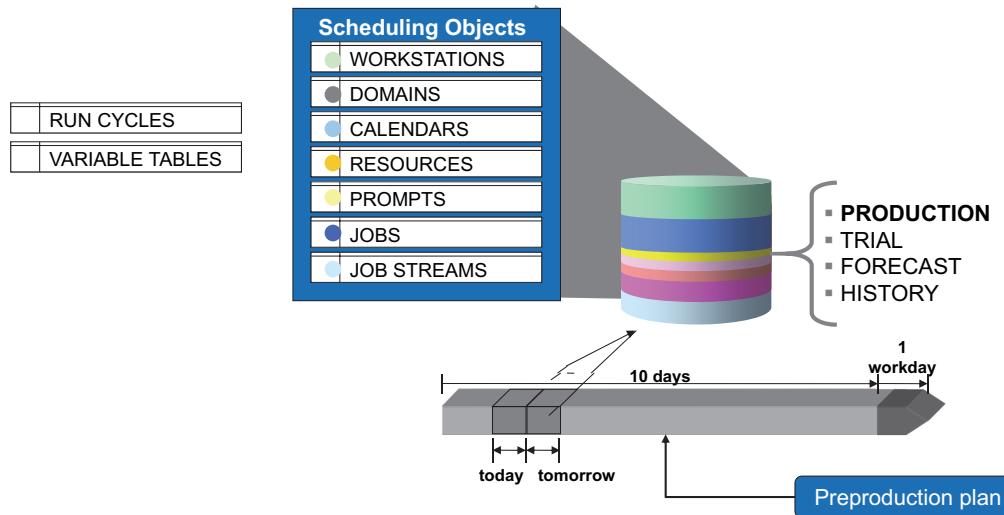
Lesson 1 Production and preproduction plans

Preproduction and production plans

© Copyright IBM Corporation 2017

Production and preproduction plans are essential parts of Workload Scheduler. Workload Scheduler automatically refreshes the plans, but you can also update them manually.

Production and preproduction plans



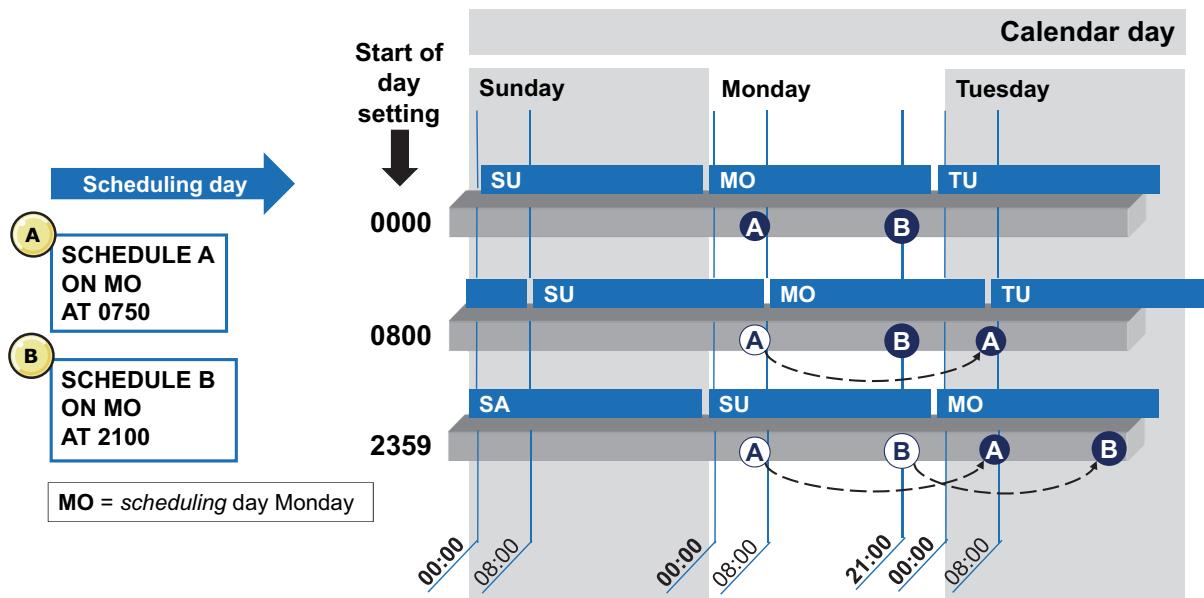
© Copyright IBM Corporation 2017

Production and preproduction plans

The *production plan* is the list of tasks to be completed within a stated interval of time on the workstations in the network. It contains all the relationships and restrictions and workload definitions that the agents need to complete the defined work.

The *preproduction plan* is an intermediate plan that Workload Scheduler automatically creates and manages. Starting from the modeling data that is stored in the database, the preproduction plan contains information that is used to compile the production plan when needed. The preproduction plan is stored in a table in the Workload Scheduler database.

Changing the start of day setting



© Copyright IBM Corporation 2017

Changing the start of day setting

You can specify the start time of the Workload Scheduler processing day by changing the **startOfDay** global option. By changing the start of day, you shift processing times from a midnight-based day to a time you choose. If you set the **startOfDay** option to 08:00, Workload Scheduler calculates the start of each processing day to start at 08:00 AM. The time and date dependencies are thus shifted 8 hours. Monday's processing day starts at 8:00 AM on Monday and continues through 7:59 AM on Tuesday.

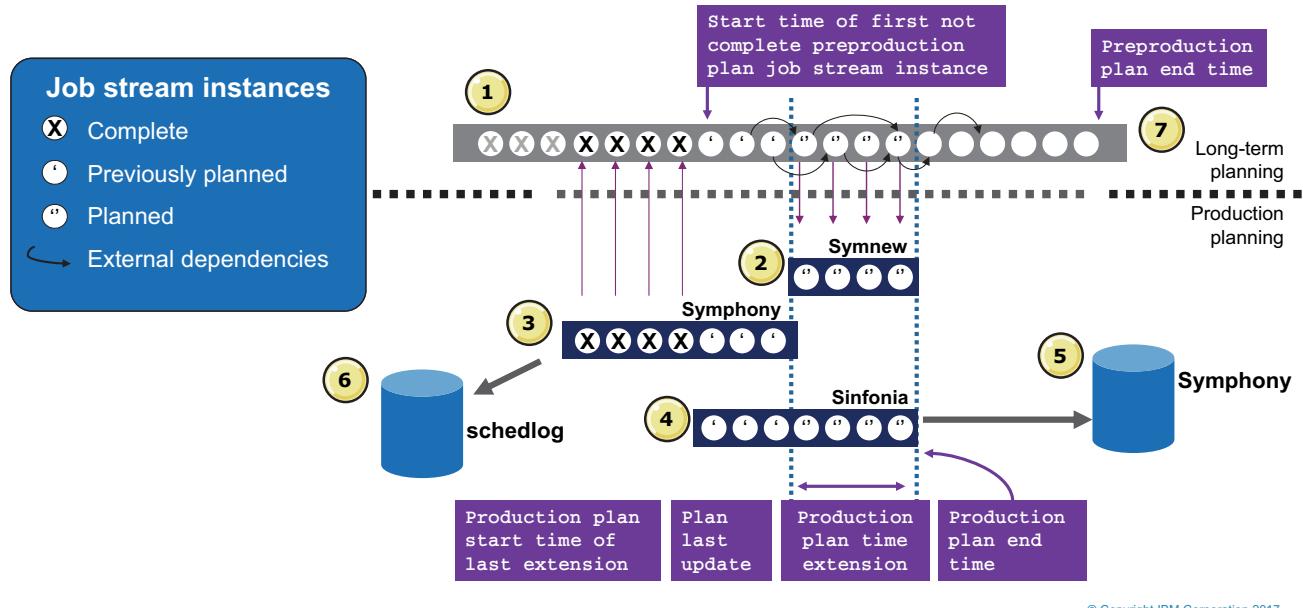
For example:

- If the start of day setting is 0800, any job stream you schedule to run between midnight and 8:00 AM starts the next calendar day.
- If the start of day setting is 0800, and you schedule a job stream to run on Monday at 7:50 AM, the job stream starts at 7:50 AM on Tuesday.
- If the start of day setting is 2359, any job stream you schedule runs the following calendar day.

No link exists between the time you set for the **startOfDay** and the time when **JnextPlan** is run, or the time you schedule the FINAL job stream. You can run **JnextPlan** at any time.

By default, for a new Workload Scheduler installation, the start of day setting is 0:00, or midnight. If you upgrade Workload Scheduler from a previous version, the previous setting is maintained.

Generating the preproduction plan



© Copyright IBM Corporation 2017

Generating the preproduction plan

You use the **JnextPlan** script on the master domain manager to update the preproduction plan, generate the production plan, and distribute it across the network. To generate and start a new production plan, Workload Scheduler performs the following steps:

- Updates the preproduction plan with the objects that are defined in the database that were added or updated since the last time the plan was created or extended.
- Retrieves information from the preproduction plan about which job streams run in the specified time period. Saves this information in an intermediate production plan called **Symnew**.
- Selects incomplete job streams from the previous production plan to be included in the production plan, depending on the *Carry Forward* policy set.
- Creates a production plan and stores it in a file named **Sinfonia**.
- Distributes a copy of the Sinfonia file to the workstations involved in the new plan processing. The agents customize the Sinfonia file and store it as **Symphony**.
- Logs all the statistics of the previous production plan into an archive.
- Updates the job stream states in the preproduction plan.

During processing, each fault-tolerant agent has its own copy of the Symphony file that is updated with the information about jobs that are running. Job status changes are propagated through the domain topology to the master domain manager. In this way, all domain managers contain updated statuses of the agents in their domains. The Symphony file on the master domain manager is kept up to date with the jobs that must run, the jobs that are running, and the ones that are complete.

Using the preproduction plan

The preproduction plan is used to identify which job streams and job stream dependencies are involved during the specified time period. Using a preproduction plan improves performance when Workload Scheduler generates the production plan because a high-level schedule of the anticipated production workload is prepared in advance.

The list of job streams to be run and their external dependencies are included in the preproduction plan. External dependencies are predecessors and successors of jobs or job streams on jobs that are part of another (external) job stream. Internal dependencies (that is, predecessors of jobs within a job stream), calendars, prompts, domains, resources, files, and other scheduling objects in production planning are not kept in the preproduction plan. They are retrieved from the database during the production planning process.

When the production plan is extended, old job stream instances are automatically removed from the preproduction plan. This method ensures that the preproduction plan size does not increase continuously. The algorithm that is used to decide which job streams to remove uses the first job stream instance that is not complete. It uses the end time of the old production plan. The first non-complete job stream instance, or FNCJSI, can be a planned job stream or one that was submitted on demand. Which job stream instances are removed depend on the age of the first non-complete job stream instance.

Carrying incomplete work forward

Job streams that did not complete in the old production plan are carried forward from the old to the new production plan (Symphony). How the job stream is carried forward depends on these items:

- The **carryforward** keyword in the job stream.
- The **enCarryForward** global option.
- The **stageman -carryforward** keyword.
- The **untilDays** global option.

In the following table, you see how the carry forward global option works.

Global option	Carry forward operation
enCarryForward=all	Job streams are carried forward only if they did not complete. All jobs are carried forward with the job streams. All is the default setting.
enCarryForward=no	No job streams are carried forward. If this option is set to no , running jobs are moved to the USERJOBS job stream.

Global option	Carry forward operation
enCarryForward=yes	Job streams are carried forward only if they did not complete <i>and</i> have the carryforward keyword set in the job stream definition. All jobs are carried forward with the job streams
untilDays=0	<p>Change the <code>untilDays</code> global option to have Workload Scheduler remove obsolete job stream instances from the plan during the planning cycle. When no latest start time (until time) is defined for a job stream, Workload Scheduler calculates a default latest start time by adding the number of days you specify in the <code>untilDays</code> option to the scheduled time for the job stream.</p> <p>If the <code>enCarryForward</code> option is set to <code>all</code>, and the number of days that are specified for <code>untilDays</code> is reached, any incomplete job stream instances that are in the plan are automatically removed from the plan and are not added to the new production plan.</p> <p>The default value is 0. If the default value is used, then no default time is set for the latest start time (until time). Run <code>JnextPlan</code> to make this change effective.</p>

The carry forward option set in the job stream definition (and the `enCarryForward` global option) is persistent. An unsuccessful job stream that is marked as carryforward continues to be carried forward until one of the following events occurs:

- It ends in a SUCC state.
- Its UNTIL time is reached.
- It is canceled.



Important: Regardless of how carry forward options are set, job streams that do not contain jobs are not carried forward.

The decision to carry forward a repetitive job is based on the state of its most recent run. A repetitive job is one that contains an `every time` setting in its definition or a chain of rerun jobs. Only the first job and the last job of the chain are carried forward.



Important: If you use the `enCarryForward=yes` option and you run `JnextPlan` manually, all the job streams that do not have the **carryforward** keyword are lost. In other words, they are not carried forward.

Checking preproduction plan status

planman showinfo

Retrieves information related to production plan status

```
wsuser@ws94mdm0:~$ planman showinfo
IBM Workload Scheduler(UNIX)/PLANMAN 9.4.0.00 (20161202)
Licensed Materials - Property of IBM* and HCL**
5698-WSH
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.
(C) Copyright HCL Technologies Ltd. 2016 All rights reserved.
* Trademark of International Business Machines
** Trademark of HCL Technologies Limited
Installed for user "wsuser".
Locale LANG set to the following: "en"
Plan creation start time: 05/30/2017 00:00 TZ America/New_York
Production plan start time of last extension: 06/01/2017 19:21 TZ America/New_York
Production plan end time: 06/02/2017 19:20 TZ America/New_York
Production plan time extention: 024:00
Plan last update: 06/01/2017 15:53 TZ America/New_York
Preproduction plan end time: 06/14/2017 19:19 TZ America/New_York
Start time of first not complete preproduction plan job stream instance: 06/01/2017 17:00 TZ America/New_York
Run number: 61
Confirm run number: 61
```

© Copyright IBM Corporation 2017

Checking preproduction plan status

You check the status of the preproduction plan by using the **planman showinfo** command. This command shows the start and end times of the preproduction and production plans, and the start time of the first not-complete job stream instance. You can see when the production plan was last updated and for how long.

Managing the production plan generation process

JnextPlan syntax

JnextPlan

```
[ -from mm/dd/[yy]yy [hhmm [tz | timezone tzname]]]
[ -to mm/dd/[yy]yy [ hhmm [tz | timezone tzname]]] |
[ -for [h]hhmm] | -days n | -days n -for [h]hhmm ]
```

- The **JnextPlan** script manages the production plan generation process
- It runs the same scripts as the jobs in the **FINAL** job stream

Plan manager planning function syntax

planman crt | ext | crtrial | exttrial | crtfc

```
[ -from mm/dd/[yy]yy [hhmm [tz | timezone tzname]]]
[ -to mm/dd/[yy]yy [ hhmm [tz | timezone tzname]]] |
[ -for [h]hhmm] | -days n | -days n -for [h]hhmm ]
```

© Copyright IBM Corporation 2017

Managing the production plan generation process

You can manage the entire production plan process with the **JnextPlan** script. You can run JnextPlan at any time during the processing day. The new production plan that is generated is immediately activated on the target workstations. The plan is activated regardless of the time set as the Workload Scheduler start of day. During the activation, the processing on all the workstations is stopped and restarted. The arguments to the JnextPlan script are like the **planman** command arguments.

Using the **planman** command

You use the **planman** command to manage intermediate production plans, trial plans, and forecast plans. You can also use **planman** to collect information about the plan, unlock the database entries that are locked by the planning process, and deploy scheduling event rules. The **planman** command runs on the master domain manager. You can create three kinds of plans:

- A *trial plan* is a projection of what a production plan is if it covers a longer period. The start date and time of a trial plan matches the preproduction plan start date and production plan end date. A trial plan has the same format as a production plan, but it cannot manage production. You can use it to see what impact new job stream definitions have when the production plan is extended. Use the **planman exttrial** command to extend a trial plan.
- A *forecast plan* is a projection of what the production plan is in any chosen time frame. It can cover any time, even in the past or overlapping the current production plan. A forecast plan has the same format as a production plan, but it cannot manage production.

When workload service assurance is enabled, it can calculate the predicted start time of each job in the job stream. Workload Scheduler calculates the average run duration for each job

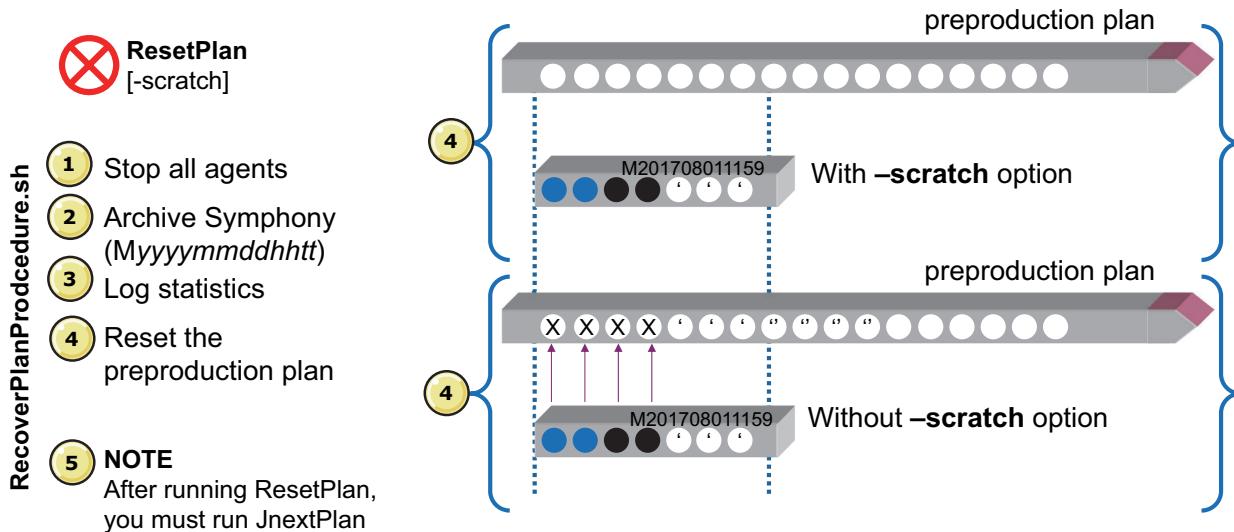
based on all previous runs. For complex plans, enabling this feature negatively affects the time that is taken to generate the forecast plan. While Workload Scheduler creates a forecast plan, the database is locked, and only unlocked when the operation completes.

No restriction on the time period that is selected to generate a forecast plan exists. However, you must consider the size of the resulting file that contains all forecast plan information.

- A *production plan* can be created or extended with the **JnextPlan** command. Using JnextPlan runs the **planman** command but also automates the initialization and deployment of the new plan to the workstations in the Workload Scheduler network.

When you create a trial or production plan, you specify either an end date and time, or a duration. Additionally, when you create a forecast plan, you specify a start time.

Resetting the plan



© Copyright IBM Corporation 2017

Resetting the plans

To build a new production plan that recovers corrupted production plans or to start planning from scratch, you must reset the preproduction plan.

You can reset both plans by using the **ResetPlan** command. If you *reset* the production plan, the preproduction plan is kept and updated with the job status. With it, you can generate a new production plan. When you generate a new production plan, it contains all the job stream instances that were not complete when you reset the plan. Resetting the plan stops the processes on the master domain manager and archives the current production plan, the **Symphony** file. It also updates job statistics from the archived production plan.

If you delete the preproduction also, by running **ResetPlan -scratch**, the preproduction plan is deleted also. The preproduction plan is created again based on the modeling information that is stored in the Workload Scheduler database when you generate a new production plan. When you generate a new plan, it contains all job stream instances for the selected time frame. It contains these instances regardless of whether they were complete when you reset the plan.



Important: If you use the **-scratch** option, run the DB2 **RUNSTATS** utility to update the database catalog statistics. A script is provided in Workload Scheduler that is named **dbrunstats** that sets the correct values and runs the appropriate DB2 commands. Consult the *IBM Workload Scheduler Version 9.4 Administration Guide* for instructions.

You can periodically use the **logman** command with the **-prod** option to record all completed job stream instances into the preproduction plan. When you run **logman -prod**, the job stream

instances that completed are marked in the preproduction plan. This method is useful in the rare condition that the Symphony file later becomes corrupted and cannot be logged.

Recovering a corrupted plan

1. Run **logman -prod** to mark the job streams instances that are completed
Perform this step periodically throughout the day
2. Run **planman showinfo** and capture the **Start time of first not complete preproduction plan job stream instance**
3. Run **ResetPlan** without the **-scratch** to avoid resetting the preproduction plan
4. Run **JnextPlan** with **-from** set to the time captured above, **-to** set to tomorrow, at the end of the corrupted Symphony plan

For example, if on 08/26 there is a Symphony corruption, you see a message similar to this one:

```
Start time of first not complete preproduction plan job stream instance: 07/31/2017 06:00 TZ GMT-03:00
```

ResetPlan

```
JnextPlan -from 07/31/2017 -to 08/27/2017 1200 TZ America/New_York
```

Specify a time zone when using times to avoid ambiguity

© Copyright IBM Corporation 2017

Recovering a corrupted plan

If the production plan is corrupted, you can use this recovery procedure to recover the plan and lose as little work as possible. A possible reason for corruption might be a hardware failure that causes read failures on the Symphony file.

1. Run **logman -prod** to mark the job streams instances completed. Run this command periodically throughout the day.
2. Run **planman showinfo** and capture the *Start time of first not complete preproduction plan job stream instance*.
3. Run **ResetPlan** without the **-scratch** option to avoid resetting the preproduction plan.
4. Run **JnextPlan** with the **-from** set to the time captured in Step 2, and **-to** set to the end time of the old production plan.

These steps are contained in a shell script that is installed on the master domain manager. The script that is named **RecoveryPlanProcedure.sh** starts the steps that are listed.

When you run **planman** or **JnextPlan**, specify a time zone with the time to avoid any ambiguity.

Lesson 2 Automating the production planning cycle

IBM Training

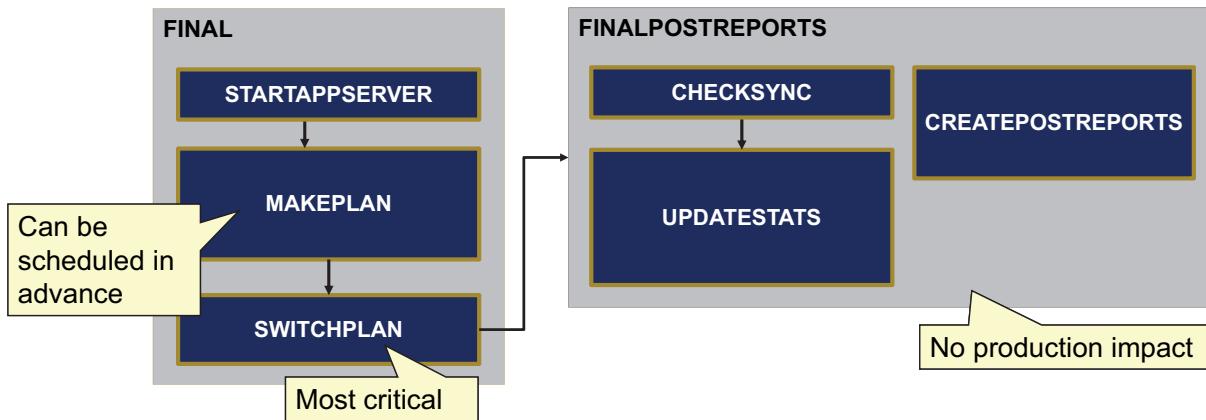
IBM

Automating the production planning cycle

© Copyright IBM Corporation 2017

In this lesson, you learn how to automate the production planning cycle by scheduling a set of jobs that maintain the preproduction, production, and postproduction plans.

Scheduling the production planning cycle



© Copyright IBM Corporation 2017

Scheduling the production planning cycle

You automate the production planning cycle by scheduling jobs that update the preproduction plan and create production plan. When you install Workload Scheduler, a text file that is named **Sfinal** is created, which contains the definitions of these jobs and their dependencies. To import the definitions that are found in Sfinal, use the **composer add** command. This command creates a job stream that is named **FINAL** in the database from the definitions in the text file:

```
composer add Sfinal
```

As installed, the FINAL job stream is scheduled to run at the default start of day, which is midnight. You can modify the FINAL job stream to customize it to your needs by using the Dynamic Workload Console or the **composer** command-line interface.

Five jobs are defined in the FINAL job stream:

- The embedded application server must be running for the planning process to work. The **STARTAPPSERVER** job ensures that the embedded application server is started. It must complete successfully for the next job to work.
- The **MAKEPLAN** job links all the workstations in the network to gather the most recent job information. It then runs the **planman** command to extend the production plan. At the end of the MAKEPLAN job, the **Symnew** file is created, and a preproduction report is generated. You can read the results of the preproduction report by viewing the job log of the MAKEPLAN job.



Note: The MAKEPLAN job can be scheduled in advance of the other jobs that follow it. The duration of the MAKEPLAN job can vary from its average more than the other jobs in the FINAL job stream.

- The **SWITCHPLAN** job is the most critical job in the FINAL job stream. Its purpose is to switch out the expiring production plan for the new one. The SWITCHPLAN job stops all of the workstations in the Workload Scheduler network and then runs the **stageman** command to perform the following functions:
 - Add workstations and calendars to the production plan
 - Add incomplete job streams from the old production plan into the new production plan, according to the *carry forward* policies in effect
 - Add job streams and their jobs, resources, prompts, and dependencies from the **Symnew** file to the production plan
 - Stage the production data into a new file that is named **Sinfonia**
 - Store the old production plan into a file that is stored in the **schedlog** directory

Finally, the SWITCHPLAN job starts all of the workstations and deploys the new production plan to them through the Workload Scheduler network.

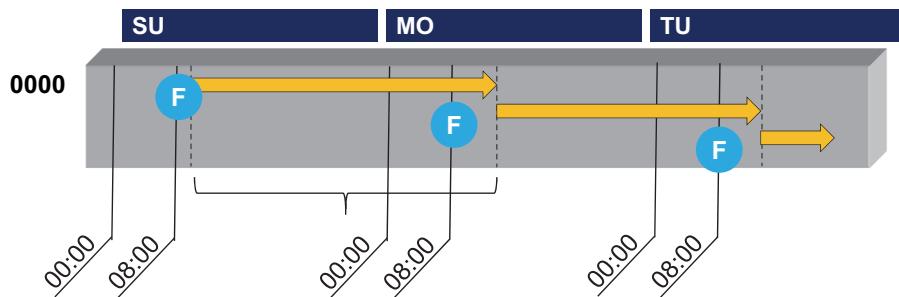
- The **CHECKSYNC** job runs the **planman checksync** command, which monitors the outcome of the initial synchronization of the Symphony file into the database.
- The **CREATEPOSTREPORTS** job runs the **reprt** and **rep8** commands to produce postproduction reports. You can view the output of those reports in the job log of the CREATEPOSTREPORTS job.
- The **UPDATESTATS** job runs the **logman** command to extract the job status information from the old production plan. The job logs the data into the Workload Scheduler database. The data can then be used to create reports and provide more accurate forecasting.

Customizing the FINAL job stream

**SCHEDULE FINAL
ON EVERYDAY
AT 0759**



Each run of FINAL (JnextPlan) extends the production plan by 24 hours, starting at the end of the current plan



Note: Be sure to extend the plan past the next scheduled time of the FINAL job stream.

© Copyright IBM Corporation 2017

Customizing the FINAL job stream

You can customize the scheduled production planning process. With this customization, you schedule the production planning process for a more convenient time and provide extra buffer after the planning cycle process.



Attention: If the plan is not extended past the current time, and you submit a job stream into the plan, you receive an error similar to the following message:

AWSJPL506E The job stream "ALPHA#SK1DFW_TMK" cannot be submitted because the specified scheduled time "7/7/17 3:35 PM" on the master domain manager is later than the end time of the production plan "7/7/17 12:00 PM" on the master domain manager.

Customizing the planning process to add a buffer can prevent this error.

To make this customization, check the following settings:

- The **enCarryForward** global option must be set to **all**.
- The **ignoreCals** global option must be set to **no**, which is the default.
- The **HOLIDAYS** calendar must be defined, and contain at least one date.
- The **FINAL** job stream is scheduled to run at a time when fewer jobs run and during normal office hours.
- Set *until times*, also called *Latest Start Time*, on job streams that might never complete to remove them from the plan automatically if left unattended.
- Run **JnextPlan -for 0100** to add an extra hour to the production plan. In this way, processing continues if any error in the FINAL job stream occurs.

Exercises



- Retrieving production and preproduction plan information
- Extending the production plan
- Resetting the plan, and building a new plan
- Using the scratch option
- Extending the plan for more than one day

© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. Where is the preproduction plan stored?
2. What happens if you set the `startOfDay` global option to 2359?
3. When the `enCarryForward` option is set to `all`, which job streams are *not* carried forward?
4. Which command do you use to check the status of the preproduction plan?
5. What three types of plans can you create by using the `planman` command?
6. What is the primary difference between running the `ResetPlan` command with and without the `-scratch` option?
7. Which job in the FINAL job stream is most critical?

Review answers

1. Where is the preproduction plan stored?

The preproduction plan is stored in the Workload Scheduler database.

2. What happens if you set the startOfDay global option to 2359?

If you set the start of day setting to 2359, any job stream you schedule runs the following calendar day.

3. When the enCarryForward option is set to all, which job streams are not carried forward?

Job streams that are complete, and job streams that do not contain any jobs are not carried forward, regardless of the enCarryForward setting.

4. Which command do you use to check the status of the preproduction plan?

*Use the **planman showinfo** command to check the status of the preproduction plan.*

5. What three types of plans can you create by using the planman command?

*You use the **planman** command to create trial plans, forecast plans, and production plans.*

6. What is the primary difference between running the **ResetPlan** command with and without the **-scratch** option?

*Running **ResetPlan** with the **-scratch** option deletes information about which job streams completed in the current production plan.*

7. Which job in the FINAL job stream is most critical?

*The **SWITCHPLAN** job is the most critical job in the FINAL job stream. Its purpose is to switch out the expiring production plan for the new one.*

Summary

You should now be able to perform the following tasks:

- Describe the production plan and preproduction plan
- Generate a preproduction plan
- Update and extend the preproduction plan
- Configure carry forward and start of day options to meet your needs
- Reset the production plan and preproduction plan
- Recover a corrupted production plan
- Schedule regular updates of the production plan

© Copyright IBM Corporation 2017

Unit summary

Unit 6 Running agent processes

Running agent processes

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn about the processes on each workstation and how the agents that run on a workstation manage jobs that run there.

In the Workload Scheduler network, on each agent, a group of specialized scheduling processes manages the jobs that run. The processes also send the information about job processing throughout the hierarchical tree until the master domain manager is reached. Using the information that is received from the agents, the master domain manager updates its copy of the Symphony file.

The Symphony file contains the records that describe the job processing activities to be managed across the Workload Scheduler network during the current production plan. The master domain manager sends the updates on its activities to the workstations involved. In this unit, you learn both the job processing that each workstation completes, and the network communication that is established across the hierarchical tree.

Objectives

In this unit, you learn how to manage Workload Scheduler agents. You learn how to perform the following tasks.

- Describe the agent process tree
- Describe what processes and services run on agents and domain managers
- Describe the interprocess communication files
- Configure and customize the job monitor to set user environments for jobs to run

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Agent processes and process communication

Agent processes and process communication

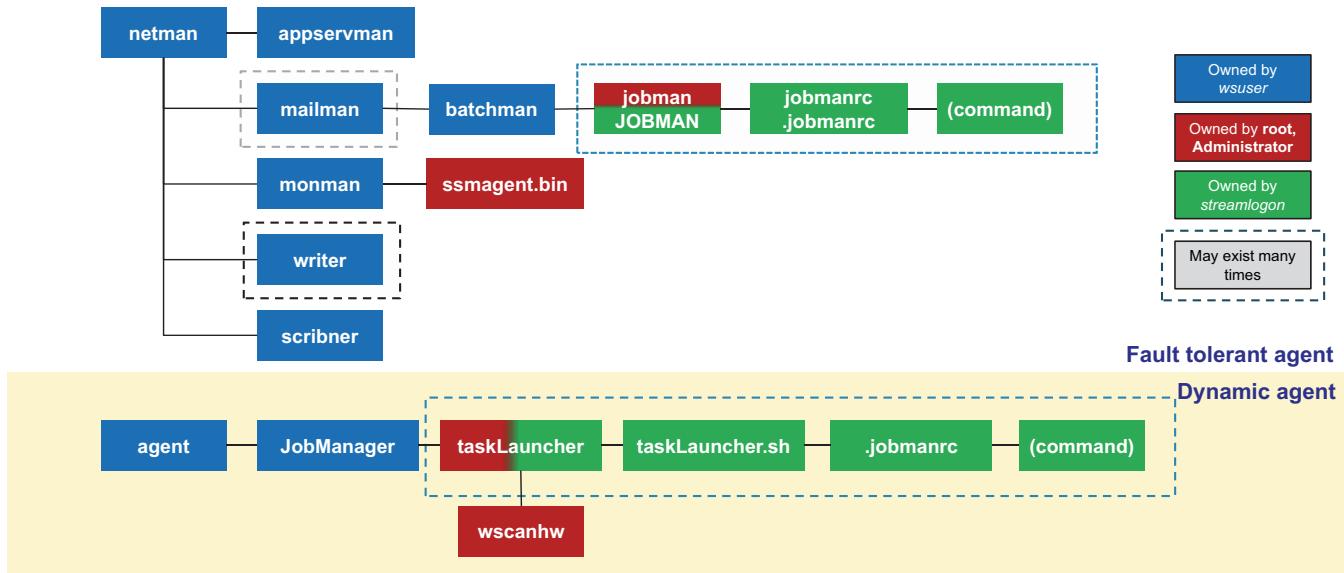
© Copyright IBM Corporation 2017

Each Workload Scheduler agent manages the communication between workstations and local job processes, and notifies domain managers about the status. A series of management processes are automatically installed with the workstation. These processes allow Workload Scheduler to manage these tasks:

- Communicate across the Workload Scheduler network.
- Manage authentication mechanisms for remote clients, such as command-line programs, connecting to the master domain manager by using the HTTP or HTTPS protocols.

The *WebSphere Application Server* infrastructure that runs on the master domain manager and its backup is not apparent when you use Workload Scheduler. The only part that is apparent is the manual start and stop of the WebSphere Application Server and managing the connection parameters when the server communicates across the network.

Agent processes



© Copyright IBM Corporation 2017

Agent processes

The following processes are started on the Workload Scheduler fault-tolerant agents:

- The **netman** process is the *network management* process. The **StartUp** command starts netman. The netman process is a network listener program that receives start, stop, link, or unlink requests from the network. The netman process examines each incoming request and creates a local Workload Scheduler process to handle the requests. Use the **conman shutdown** command to stop netman and all other processes.
- The **monman** process is a process that netman starts and is used in event management. It starts monitoring and **ssmagent** services that detect the events that are defined when rules, which are deployed and activated on the specific workstation. When these services detect any such events, after a preliminary filtering action, they send them to the event processing server. This server runs in the master domain manager. If no event rule configurations are downloaded to the workstation, the monitoring services stay idle.

The communication process between the monitoring agents and the event processing server is independent of the Workload Scheduler network topology. It is based directly on the EIF port number of the event processor. The event information flows directly from the monitoring agents without passing through intermediate domain managers. Local cache memories provide a degree of fault tolerance. These memories temporarily store the event occurrences on the agents if communication with the event processor is down.

- Use the **conman startmon** command to start the monman process and the **conman stopmon** command to stop it.
- The **writer** process is a process that *netman* starts to pass incoming messages to the local **mailman** process. Link requests start the writer processes. There might be more than one on a

domain manager workstation. Writer processes stop by unlink requests or when the communicating mailman ends.

- The **mailman** process is the *mail management* process. It routes messages to either local or remote workstations. On a domain manager, more mailman processes can be created. They divide the load on mailman because of the initialization of agents and to improve the timeliness of messages. When the domain manager starts, it creates a separate mailman process instance for each Server ID that is specified in the workstation definitions.
- The **batchman** process is the *production control* process. It interacts directly with the copy of the **Symphony** file that is distributed to the workstations at the beginning of the production period and updates it. Batchman performs several functions:
 - Locally manages processing the plan and updating other entities
 - Resolves dependencies of jobs and job streams
 - Selects jobs to be run
 - Updates the plan with the results of job processing

Batchman is the only process that can update the **Symphony** file.

- The **jobman** process is the *job management* process. It starts jobs under the direction of batchman and reports job status back to mailman. It is responsible for tracking job states and for setting the environment as defined in the **jobmanrc** and **.jobmanrc** scripts when it requests to start jobs. When the jobman process receives a start job message from batchman, it creates a job monitor process. The maximum number of job monitor processes that can be created on a workstation is set by using the **limit cpu** command from the conman command-line prompt.

The job monitor is **JOBMAN** on UNIX or **JOBMON.exe** and **jobInch.exe** on Windows operating systems. The job monitor process first completes a set of actions that set the environment before the job is started. It then starts the job by running the script file or command that is specified in the job definition. The setup activities consist of starting the standard configuration file, which is **WSHome/jobmanrc** in UNIX and Linux and **WSHome/jobmanrc.cmd** in Windows). This file contains settings that apply to all jobs that run on the workstation. In addition, on UNIX workstations a local configuration script **streamlogon_user/.jobmanrc** is started, if it exists in the home directory of the user that starts the job. This local configuration file contains settings that apply only to jobs started by the specific user. If any of these steps fails, the job ends in the **FAIL** state.



Important: If, on Windows systems, a system variable that is called **TEMP** exists, **wsuser** must be authorized to create files in the directory to which the variable is set. If this requirement is not met, the **JOBMON.exe** binary file fails to start successfully.

All processes, except **jobman**, run as the Workload Scheduler user. Jobman runs as **root**.

On standard agent workstations, the **batchman** process is not started because this type of workstation does not manage job scheduling. These workstations start jobs only under the direction of their domain manager. Locally on the workstation, the management processes wait for a request to start a job from the domain manager in LISTEN mode. When the request is received, the job is started locally and the result is sent back to the domain manager.

On UNIX and Linux operating systems, the Workload Scheduler user owns and runs the processes. On Windows operating systems, an extra service is the IBM Token Service. This service enables Workload Scheduler processes to be started as if the Workload Scheduler user started them.

To stop all of the processes except `netman`, use the **conman stop** command. To start the processing, use the **conman start** command.



Important: On UNIX operating systems, all jobs inherit their **ulimits** from the **WSuser**. To be sure jobs run successfully, set the **ulimits** of the **WSuser** to **unlimited**.

Dynamic agent processes

The dynamic agent starts the agent, JobManager, and taskLauncher processes.

The **agent** process manages network communication with the workload broker service that runs on the master domain manager and starts the **JobManager** service.

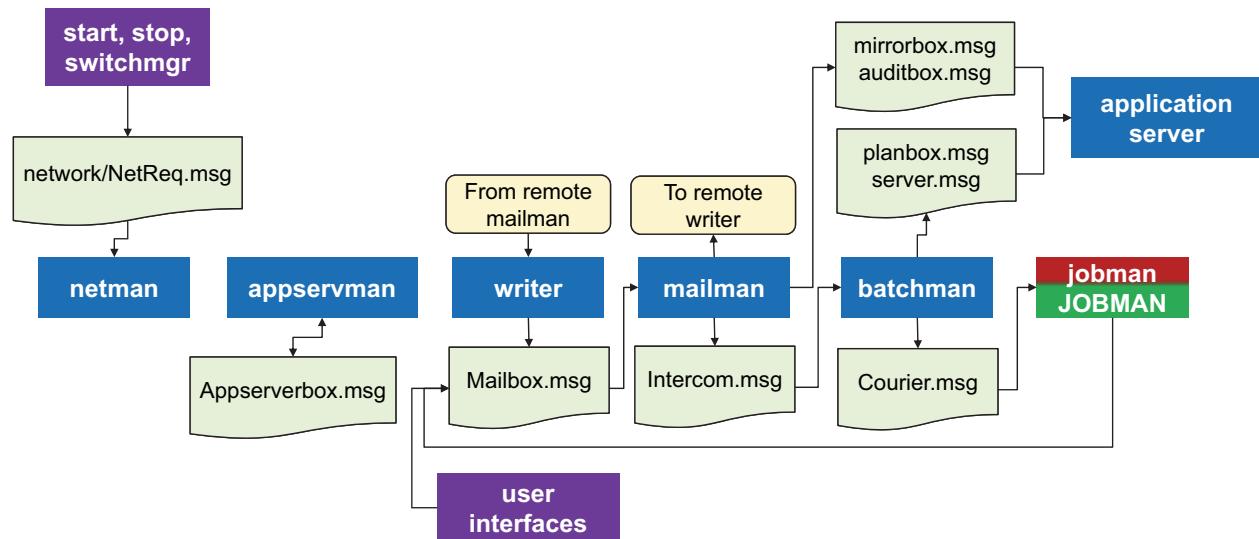
The **JobManager** process performs these tasks:

- Collects computer, operating system, network, and file system information about the computer where the agent is running
- Manages jobs that are running on the agent

The **taskLauncher** first starts a set of actions that set the environment before the job is started. It then starts the job by running the script file or command that is specified in the job definition. The setup activities consist of starting the standard configuration file, which is **WSHome/bin/taskLauncher.sh** in UNIX and Linux and **WSHome\bin\taskLauncher.cmd** in Windows). This file contains settings that apply to all jobs that run on the workstation. In addition, on UNIX workstations a local configuration script **streamlogon_user!.jobmanrc** is started, if it exists in the home directory of the user that runs the job. This local configuration file contains settings that apply only to jobs started by the specific user. If any of these steps fails, the job ends in the **FAIL** state.

On UNIX and Linux, you must be the **root** user to start or stop the dynamic agent process. Use the **StartUpLwa** command to start the processes and the **ShutDownLwa** command to stop them.

Interprocess communication



© Copyright IBM Corporation 2017

Interprocess communication

Workload Scheduler uses message queues for local interprocess communication. These message files are in the **WSHome** directory:

NetReq.msg	The netman process reads this message file for local commands. It receives messages such as START, STOP, LINK, and UNLINK.
Mailbox.msg	The mailman process reads this message file. It receives messages, through the graphical user interface (Dynamic Workload Console) or the console manager (conman). It receives messages from the local batchman and jobman processes and from other Workload Scheduler workstations in the network.
Intercom.msg	The batchman process reads this message file. It contains instructions that the local mailman process sends.
Courier.msg	The batchman process reads this message file that the jobman process sends.
planbox.msg	The batchman process writes message file, and the application server reads it.
Server.msg	The batchman process writes message file, and the application server reads it.

These files have a default maximum size of 10 MB.

Using evtsize

The **evtsize** utility defines the size of the Workload Scheduler message files. To change the maximum size of a message file, specify the file name and its new size, for example:

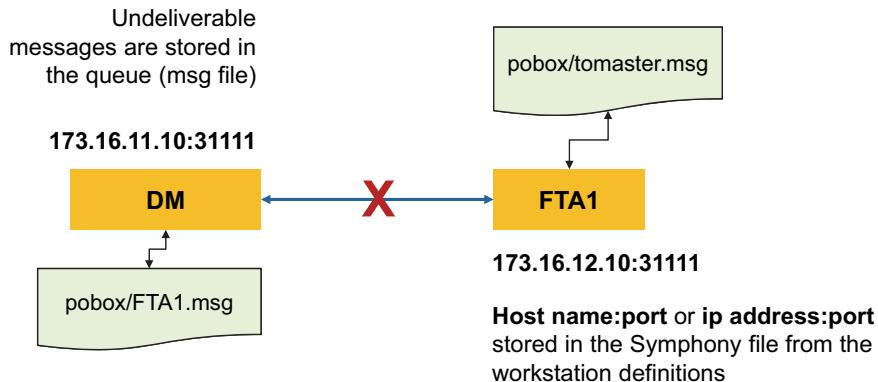
```
evtsize Intercom.msg 20000000
```

These evtsize options show information about message files:

- The **-info** option displays the percentage of the queue file in use.
- The **-show** option displays the size of the queue of messages that are contained in the file.

The **evtsize -compact** option reduces the message file size to equal the size that is occupied by the messages present at the time you run the command.

Network communications store and forward



When the network connection is established, the queues are drained.

© Copyright IBM Corporation 2017

Network communication store and forward

Workload Scheduler uses the TCP/IP protocol for network communication. The host name and the port number that is used to establish the TCP/IP connection are set for each workstation in its workstation definition.

A **store-and-forward** technology is used to maintain consistency and fault-tolerance always across the network by queuing messages in message files while the connection is not active. When TCP/IP communication is established between systems, Workload Scheduler provides bidirectional communication between workstations that use links. The **autolink** flag set in the workstation definition and the **link** and **unlink** commands that are entered from the **conman** command-line program control the links. When a link is opened, messages are passed between two workstations. When a link is closed, the sending workstation stores messages in a local message file and sends them to the destination workstation when the link is open again.

Lesson 2 Using the console manager

Using the console manager

© Copyright IBM Corporation 2017

In this lesson, you learn how to use the console manager to start and stop processes and find problems. The Workload Scheduler production plan environment is managed by using the **conman** command-line program.

The **conman** program is used to start and stop processing, alter the Symphony production plan, show the plan, and control workstation links in a network. It can be used from the master domain manager and from any fault-tolerant agent in the Workload Scheduler network.

Using the console manager

- Use conman to manage most production plan activities
- Use conman interactively or in shell scripts
- Help is available inside conman with the HELP command
- Commands are processed asynchronously in the background
- Effects of commands might not be immediate because they flow through the Workload Scheduler queues

© Copyright IBM Corporation 2017

Using the console manager

To configure the environment for using conman, set the **PATH** and **TWS_TISDIR** variables by running one of the following scripts:

In UNIX, run one of these scripts:

`../WSHome/tws_env.sh` for Bourne and Korn shells
`../WSHome/tws_env.csh` for C shells

In Windows, run the following script:

`WSHome\tws_env.cmd`



Hint: Include the contents of the **tws_env.sh** script in your own profile initialization scripts to avoid sourcing the TWS_env scripts.

Using conman commands

You can use **conman** commands interactively or in shell scripts. When used interactively, **conman** becomes a command processor. You start **conman** and enter commands at the **conman** prompt, which is by default a percent sign (%). For example, a **conman** session is shown in the following paragraph:

```
wsuser@ws94mdm0:~> . ./tws_env.sh
IBM Workload Scheduler(UNIX) /CONMAN 9.4.0.00 (20161202)
Licensed Materials - Property of IBM* and HCL**
5698-WSH
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.
(C) Copyright HCL Technologies Ltd. 2016 All rights reserved.
* Trademark of International Business Machines
** Trademark of HCL Technologies Limited
Installed for user "wsuser".
Locale LANG set to the following: "en"
Scheduled for (Exp) 05/03/17 (#47) on MDM0. Batchman LIVES. Limit: 11, Fence:
0, Audit Level: 1
%showcpus;info
```

Commands can be entered in uppercase or lowercase. When used in shell scripts, you can pass **conman** commands as arguments to the **conman** command. For example, to show a list of workstations, you place the following command in a script:

```
conman 'showcpus;info'
```

To see a list of commands, use the **conman HELP COMMANDS** command. You can abbreviate most **conman** commands to two- or three-letter short forms. For example, you can shorten the **SHOWJOBS** command to **SJ**. A list of abbreviations is shown in the output of the **HELP COMMANDS** command.

Starting, stopping, linking, and unlinking workstations

```

conman      start [domain!]workstation
              [;noask]

          stop [domain!]workstation
              [;wait]
              [;noask]

          {link | lk} [domain!]workstation
              [;noask]

          unlink [domain!]workstation
              [;noask]

          {shutdown | shut} [;wait]

```

© Copyright IBM Corporation 2017

Starting, stopping, linking, and unlinking workstations

You can use the console manager command **conman** to start, stop, link, and unlink Workload Scheduler workstations.

Starting workstations

The **conman start** command starts Workload Scheduler production processes, except for the event monitoring engine and WebSphere Application Server.



Important: Make sure that **conman start** is not entered while either **JnextPlan** or **stageman** runs.

```
start [domain!]workstation
      [;noask]
```

The **domain** option specifies the name of the domain in which workstations are started. Wildcard characters are allowed. This argument is useful when you start more than one workstation in a domain. If domain is omitted and workstation contains wildcard characters, the default domain is the one in which conman is running.

The **workstation** option specifies the name of the workstation to be started. Wildcard characters are allowed. This command is not supported on remote engine workstations.

The **noask** option specifies the command must not prompt for confirmation before it acts on each qualifying workstation.



Hint: When you type `conman` commands without the **noask** option, the `conman` program interactively prompts you to confirm the action for each matching object. The commands can act upon several objects, for example, with wildcards.

You use the **start** command at the start of each production period to restart Workload Scheduler following preproduction processing. Then, it causes the autolinked fault-tolerant agents, that is, the workstation definitions that contain the **autolink** parameter, and standard agents to be initialized and started automatically. Agents that are not autolinked are initialized and started when you run a link command.

Assuming the user has *start* access to the workstations to start, the following rules apply:

- A user who runs `conman` on the master domain manager can start any workstation in the network.
- A user who runs `conman` on a domain manager other than the master can start any workstation in that domain and subordinate domains. The user cannot start workstations in peer domains.
- A user who runs `conman` on an agent can start workstations that agent hosts.

Stopping workstations

The **conman stop** command stops Workload Scheduler production processes. To stop the `netman` process also, use the **shutdown** command.

```
stop [domain!] workstation
      [,wait]
      [,noask]
```

The **domain** option specifies the name of the domain in which workstations are stopped. Because workstations have unique names, the domain is not needed when you stop a specific workstation. Wildcard characters are allowed.

This argument is useful when you stop more than one workstation in a domain. For example, to stop all the agents in domain NEWDM, use the following command:

```
stop newdm!@
```

If domain is omitted and `workstation` contains wildcard characters, the default domain is the one in which `conman` is running.

The **workstation** option specifies the name of the workstation to be stopped. Wildcard characters are allowed. This command is not supported on remote engine workstations.

The **wait** option specifies not to accept another command until all processes stop. With the wait option, you use the **conman stop** command synchronously.

The **noask** option specifies the command must not to prompt for confirmation before it acts on each qualifying workstation.



Note: You might not be able to apply the **stop** command to a distant workstation. For example, the TCP/IP path might not be available. The command is stored locally in a message file in the **pobox** directory and is sent to the workstation when it becomes linked.

When you issue a **stop @** command on a domain manager, a local **conman stop** command runs on the remote workstations. Finally, the command starts running on the lowest stations in the network hierarchy and then runs on the domain manager. However, the Symphony file is not updated before the workstations stop. Therefore, if you type a **conman sc @!@** command from any CPU, the resulting information might not be an up-to-date picture of the states of the CPUs, even of the domain manager.

Linking workstations

The **conman link** command opens communication links between workstations. In a Workload Scheduler network, fault-tolerant and standard agents are linked to their domain managers, and domain managers are linked to their parent domain managers. Extended agents are not linked; they communicate through a host.

```
{link | lk} [domain!]workstation  
[;noask]
```

The **domain** option specifies the name of the domain in which links are opened. Wildcard characters are allowed. This argument is useful when you link more than one workstation in a domain. The domain is not needed if you do not include wildcard characters in workstation. If you do not include domain and you include wildcard characters in workstation, the default domain is the one in which conman is running.

The **workstation** option specifies the name of the workstation to be linked. Wildcard characters are allowed. This command is not supported on remote engine workstations.

The **noask** option specifies the command must not to prompt for confirmation before it acts on each qualifying workstation.

If the **autolink** option is set to **on** in a workstation definition, its link is opened automatically each time Workload Scheduler is started. If autolink is set to off, you must use link and unlink commands to control linking.

Assuming that a user has *link* access to the workstations to link, the following rules apply:

- A user who runs `conman` on the master domain manager can link any workstation in the network.
- A user who runs `conman` on a domain manager other than the master can link any workstation in its own domain and subordinate domains. The user cannot link workstations in peer domains.
- A user who runs `conman` on an agent can link any workstation in its local domain if the workstation is a domain manager or host. A peer agent in the local domain cannot be linked.
- To link a subordinate domain when you run `conman` in a higher domain, it is not necessary that the intervening links be open.

Unlinking workstations

The `conman unlink` command closes communication links between workstations:

```
unlink [domain!]workstation  
[;noask]
```

The **domain** option specifies the name of the domain in which to close links. It is not necessary to specify the domain name of a workstation in the master domain. Wildcard characters are allowed.



Important: You must always specify the domain name when you unlink a workstation that is not in the master domain.

If you do not specify a domain and workstation includes wildcard characters, the default domain is the one in which `conman` is running.

The **workstation** option specifies the name of the workstation to be unlinked. Wildcard characters are allowed. This command is not supported on remote engine workstations.

The **noask** option specifies the command must not to prompt for confirmation before it acts.

Starting and stopping other processes

```
conman startappserver [domain!]workstation  
[;wait]  
  
{stopappserver | stopapps} [domain!]workstation  
[;wait]  
  
{starteventprocessor | startevtp} [domain!]workstation  
  
{stopeventprocessor | stopevtp} [domain!][workstation]  
  
{startmon | startm} [domain!]workstation  
[,noask]  
  
{stopmon | stopm} [domain!]workstation  
[;wait]  
[;noask]
```

© Copyright IBM Corporation 2017

Starting and stopping other processes

You can use the console manager program, **conman**, to manage other processes and the agents.

The **conman startappserver** command starts the WebSphere Application Server profile on the workstation.

```
startappserver [domain!] workstation  
[;wait]
```

The **domain** option specifies the name of the domain of the workstation. Because workstations have unique names, the domain is not needed when you start the WebSphere Application Server on a specific workstation. Wildcard characters are allowed. If domain is omitted and workstation contains wildcard characters, the default domain is the one in which conman is running.

The **workstation** option specifies the name of the workstation where you want to start the application server. Wildcard characters are allowed. If no domain and workstations are specified, the action is on the local workstation.

The **wait** option causes conman to wait until WebSphere Application Server starts before it prompts for another command. You can start WebSphere Application Server also with the **StartUp** utility command.

Stopping the application server

The **conman stopappserver** command stops the WebSphere Application Server on the workstation.

```
{stopappserver | stopapps} [domain!] workstation  
[;wait]
```

The optional **domain** option specifies the name of the domain of the workstation.

The **workstation** option specifies the name of the workstation where you want to stop the application server. Wildcard characters are allowed. If no domain and workstations are specified, the action is on the local workstation.

The **wait** option specifies that conman wait until WebSphere Application Server stops before it prompts for another command.



Important: On Windows systems, refrain from using Windows services to stop WebSphere Application Server. If you use Windows services, the **appserverman** process, which continues to run, starts WebSphere Application Server again. Use this command or the **stopWas** command without the **-direct** option instead.

When you run the command, the **appserverman** process first checks if WebSphere Application Server can retrieve the user's credentials (user name and password) from the **soap.client.props** file. This file is in the WebSphere Application Server profile. If the check is negative, **appserverman** reads them from the **useropts** file of the user. The **appserverman** process runs the **stopServer.sh** (bat) script to pass them to WebSphere Application Server.

To run the command, you must complete one of the following two customization procedures to provide the user credentials to WebSphere Application Server:

- Customize the user name (**com.ibm.SOAP.loginUserId**) and password (**com.ibm.SOAP.loginPassword**) properties in the **soap.client.props** file that is in the following directory:

TWAhome/WAS/TWSPProfile/properties/

You must also perform these steps:

- a. Set property **com.ibm.SOAP.securityEnabled** to **true** in the same file to enable the SOAP client security.
 - b. Run the **encryptProfileProperties.sh** script to encrypt the password. For more information about this application server tool, see the *IBM Workload Scheduler 9.4 Administration* guide.
- Customize the attributes for conman connections section in the **localopts** file by specifying the details of the connector or of the master domain manager.

You must also perform these steps:

- a. Create or customize, if already present, the **useropts** file manually, adding the USERNAME, and PASSWORD attributes for the user who runs stopappserver. Make sure that the **useropts** file name is entered in the USEROPTS key in the attributes for conman (CLI) connections section. See the administration guide for further details.
- b. Encrypt the password in the **useropts** file by running conman.

Starting the event processor

The **conman starteventprocessor** starts the event processing server on the master domain manager, backup master, or on a workstation that is installed as a backup master that functions as a plain fault-tolerant agent.

```
{starteventprocessor | startevtp} [domain!] workstation
```

The **domain** option specifies the name of the domain of the workstation.

The **workstation** option specifies the name of the workstation where you want to start the event processing server. Wildcard characters are *not* allowed.

You can omit the workstation name if you run the command locally.

Stopping the event processor

The **conman stopeventprocessor** stops the event processing server.

```
{stopeventprocessor | stopevtp} [domain!] [workstation]
```

The **domain** option specifies the name of the domain of the workstation.

The **workstation** option specifies where you want to stop the event processing server. Wildcard characters are *not* allowed. You can omit the workstation name if you run the command locally.

This command cannot be started in an asynchronous way.

If you issue the command from a workstation other than the one where the event processor is configured, the command uses the command-line client. The user credentials for the command-line client must be set correctly.

Starting the event monitoring processes

The **conman startmon** command starts the **monman** process that turns on the event monitoring engine on the workstation.

```
{startmon | startm} [domain!] workstation  
[;noask]
```

The optional **domain** option specifies the name of the domain of the workstation.

The required **workstation** option specifies the name of the workstation where you want to start the monitoring engine. Wildcard characters are allowed.

The **noask** option specifies the command must not to prompt for confirmation before it acts on each qualifying workstation.

Stopping the event monitoring processes

The **conman stopmon** command stops the event monitoring engine on the workstation.

```
{stopmon | stopm} [domain!] workstation  
[;wait]  
[;noask]
```

The **domain** option specifies the name of the domain of the workstation.

The **workstation** option specifies the name of the workstation where you want to stop the monitoring engine.

The **wait** option specifies not to accept another command until the monitoring engine stops.

The **noask** option specifies the command must not to prompt for confirmation before it acts on each qualifying workstation.

The monitoring engine is restarted automatically when the next production plan is activated. The monitoring engine restarts on Windows also when Workload Scheduler is restarted. It does not restart if you disable the autostart monman local option. The command is asynchronous, unless you specify the **wait** keyword.

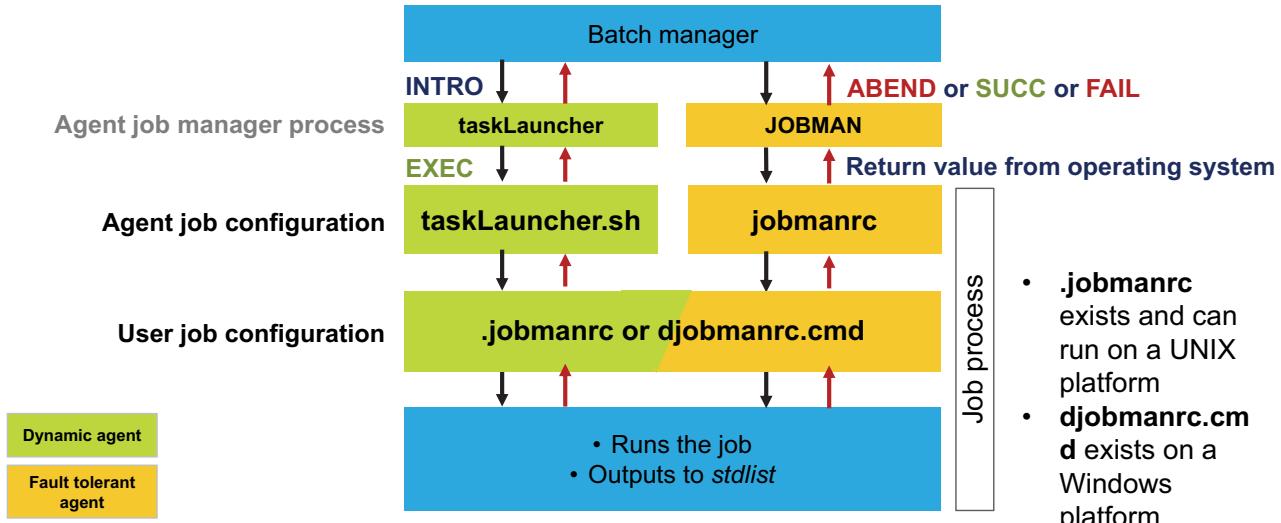
Lesson 3 Configuring the job environment

Configuring the job environment

© Copyright IBM Corporation 2017

On each fault-tolerant agent workstation, the `batchman` production control process starts jobs. The `batchman` process resolves all job dependencies to ensure the correct order of job processing and then queues a job start message to the `jobman` process. For dynamic agent workstations, the domain manager resolves all job dependencies, then queues a job request to the `JobManager` process on the agent.

Configuring the job environment



© Copyright IBM Corporation 2017

Configuring the job environment

The job manager process starts the job process, including the configuration scripts and the jobs, and retains the user name that is recorded with the definition of the job. These jobs include jobs, files, or commands submitted not through a scheduled plan, but manually by a user. Submitted jobs retain the submitting user's name unless otherwise specified.

The **agent job manager** process starts a monitoring process that begins by setting a group of environment variables. It then runs an **agent job configuration** script, which you can customize. The **agent job configuration** script sets variables that are used to configure the way jobs are started, regardless of the user.

On UNIX workstations, the **user job configuration** script **.jobmanrc** also runs if all these conditions are true:

- The user can use a local configuration script.
- The script **\$HOME/.jobmanrc** exists.
- The script is an executable file.

Then, the standard configuration script or the user one runs the job. The agent job manager process ports the results of job processing and then updates the batch manager with the information about job completion status. To have jobs that are run with the user's environment, add the following instruction in the local configuration script:

`. $HOME/.profile`



Note: Before you source the **.profile** in the **.jobmanrc** file, make sure that it does not contain any **stty** setting or any step that requires user manual intervention. In case it does, add in only the necessary steps that are contained in the **.profile** to the **.jobmanrc** script.

On Windows workstations, the user job configuration script **djobmanrc.cmd** is run if it exists in the user's *Documents and Settings* directory. On workstations that run Windows 2008, **djobmanrc.cmd** must be in the **\users\username** directory. The environment variable **%USERPROFILE%** represents the directory and depends on the Windows language installation. The **jobmanrc.cmd** runs the **djobmanrc.cmd** script.



Hint: If a script or command runs from a shell prompt, but does not run correctly when processed by Workload Scheduler, you can submit an ad hoc job. This job runs the **env** command, which is set on Windows. Then, run the same command outside of Workload Scheduler. Compare the output of the two commands. If they are different, modify the **.jobmanrc** script to match them.

Job environment variables

- Submit a job that runs `env` on UNIX, `set` on Windows
- See the actual environment in the job output

Variable	Value or description
PATH	/bin:/usr/bin
TWS_PROMOTED_JOB	If YES, the job is in a critical path and is at risk
UNISON_CPU	This workstation
UNISON_DIR	Where Workload Scheduler is installed
UNISON_JCL	The script or command to run
UNISON_JOB	The fully qualified job name
UNISON_JOBNUM	The job number
UNISON_SCHED	The job stream name
UNISON_STDLIST	The file storing this job's output

© Copyright IBM Corporation 2017

Job environment variables

The job monitor process sets the following environment variables on Windows workstations before it starts the job.

Table 1 Environment variables on Windows workstations

Variable name	Value
COMPUTERNAME	The value of %COMPUTERNAME% from the user environment.
HOMEDRIVE	The value of the HOMEDRIVE set in the user environment.
HOMEPATH	The value of the HOMEPATH set in the user environment.
LANG	The value of the LANG set in the user environment. If not set, its value is set to C.
MAESTRO_OUTPUT_STYLE	The setting for output style for long object names.
TEMP	The value of the TEMP set in the user environment. If not specified, its value is set to c:\temp.
TIVOLI_JOB_DATE	The scheduled date for a job.
TWS_PROMOTED_JOB	Applies to the Workload Service Assurance functions. Can be YES or NO. When the value is YES, it means that the critical job or one of its predecessors was promoted.
UNISON_CPU	The name of this workstation.
UNISON_DIR	The directory where Workload Scheduler is installed.
UNISON_EXEC_PATH	The jobmanrc fully qualified path.
UNISONHOME	The path where the Workload Scheduler instance was installed.

Table 1 Environment variables on Windows workstations

Variable name	Value
UNISON_HOST	The name of the host workstation.
UNISON_JCL	The script or command to be run, as configured in the job definition.
UNISON_JOB	The absolute job identifier: <i>workstation#sched_id.job</i> .
UNISON_JOBNUM	The job number.
UNISON_MASTER	The name of the master domain manager workstation.
UNISON_RUN	The Workload Scheduler current production run number.
UNISON_SCHED	The job stream name.
UNISON_SCHED_DATE	The Workload Scheduler production date (<i>yyymmdd</i>).
UNISON_SCHED_ID	The job stream ID of the job stream that contains the job in process.
UNISON_SCHED_IA	The job stream schedule time of the job stream that contains the job in process.
UNISON_SCHED_EPOCH	The Workload Scheduler production date that is expressed in epoch form.
UNISON_SHELL	The login shell of the user that runs the job.
UNISON_STDLIST	The path name of the standard list file of the job.
UNISON_SYM	The Symphony record number of the started job.

The job monitor process on UNIX and Linux workstations sets following environment variables before it starts the job.

Table 2 Environment variables on UNIX and Linux workstations

Variable Name	Value
MAESTRO_OUTPUT_STYLE	The setting for output style for long object names.
PATH	/bin:/usr/bin
TIVOLI_JOB_DATE	The scheduled date for the job.
TWS_PROMOTED_JOB	Applies to the Workload Service Assurance functions. Can be YES or NO. When the value is YES, it means that the critical job or one of its predecessors was promoted.
UNISON_CPU	The name of this workstation.
UNISON_DIR	The directory where Workload Scheduler is installed.
UNISON_EXEC_PATH	The <code>jobmanrc</code> fully qualified path.
UNISON_HOST	The name of the host workstation.
UNISON_JCL	The script or command to be run, as configured in the job definition.
UNISON_JOB	The absolute job identifier: <i>workstation#sched_id.job</i> .
UNISON_JOBNUM	The job number.
UNISON_MASTER	The name of the master domain manager workstation.

Table 2 Environment variables on UNIX and Linux workstations

Variable Name	Value
UNISON_RUN	The Workload Scheduler current production run number.
UNISON_SCHED	The job stream name.
UNISON_SCHED_DATE	The Workload Scheduler production date (<i>yyymmdd</i>) reported in the Symphony file.
UNISON_SCHED_ID	The jobstreamID of the job stream that contains the job in process.
UNISON_SCHED_IA	The StartTime of the job stream that contains the job in process.
UNISON_SCHED_EPOCH	The Workload Scheduler production date, expressed in epoch form.
UNISON_SHELL	The login shell of the user that runs the job.
UNISON_STDLIST	The path name of the standard list file of the job, where its output is being written.
UNISON_SYM	The Symphony record number of the started job.

Exercises

- View the system process tree to see Workload Scheduler processes
- Stop and start Workload Scheduler agents
- Configure the job environment



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. On fault-tolerant agents, what is the parent network management process?
2. On fault-tolerant agents, which process runs under the root user?
3. Which message queue file feeds the mailman process and is thus the most active?
4. Which command do you use to query and define the size of the Workload Scheduler message queues?
5. What is the function of the **wait** option on conman commands?
6. Which file can you create and customize to modify jobs' environments for a user?
7. Which environment variable points to the file that contains the job's output?

Review answers

1. On fault-tolerant agents, what is the parent network management process?

The netman process is the network management process, and the parent of other Workload Scheduler processes on fault-tolerant agents.

2. On fault-tolerant agents, which process runs under the root user?

The jobman process runs as root. Other processes run as the Workload Scheduler user.

3. Which message queue file feeds the mailman process and is thus the most active?

The mailman process reads Mailbox.msg, which receives messages from several processes.

4. Which command do you use to query and define the size of the Workload Scheduler message queues?

*Use the **evtsize** command to show and change the size of message queue files.*

5. What is the function of the **wait** option on conman commands?

*Use the **wait** option to perform conman stop commands synchronously. The **wait** option causes the conman command to not accept another command until the previous function completes.*

6. Which file can you create and customize to modify jobs' environments for a user?

*Create a file that is named **.jobmanrc** in the user's home directory to customize the jobs' environments.*

7. Which environment variable points to the file that contains the job's output?

***\$UNISON_STDLIST** references the job's output.*

Summary

You should now be able to perform the following tasks:

- Describe the agent process tree
- Describe what processes run on agents and domain managers
- Describe the interprocess communication files
- Configure and customize the job monitor to set user environments for jobs to run

© Copyright IBM Corporation 2017

Unit summary

Unit 7 Securing Workload Scheduler

IBM Training



Securing Workload Scheduler

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to secure access to Workload Scheduler. You learn about how to grant or deny access to the objects and how users are authenticated to use Workload Scheduler.

References: *IBM Workload Scheduler 9.4 Administration*

Objectives

In this unit, you learn how to perform the following tasks.

- Configure security to protect your environment
- Configure authentication using the WebSphere Administrative Console and provided scripts
- Modify the Security configuration and compile its definitions
- Configure these attributes
 - User definitions
 - Object access
 - Access privileges
- Create security roles, security domains, and access control lists
- Configure tracking, auditing, and justification requirements

© Copyright IBM Corporation 2017

Objectives

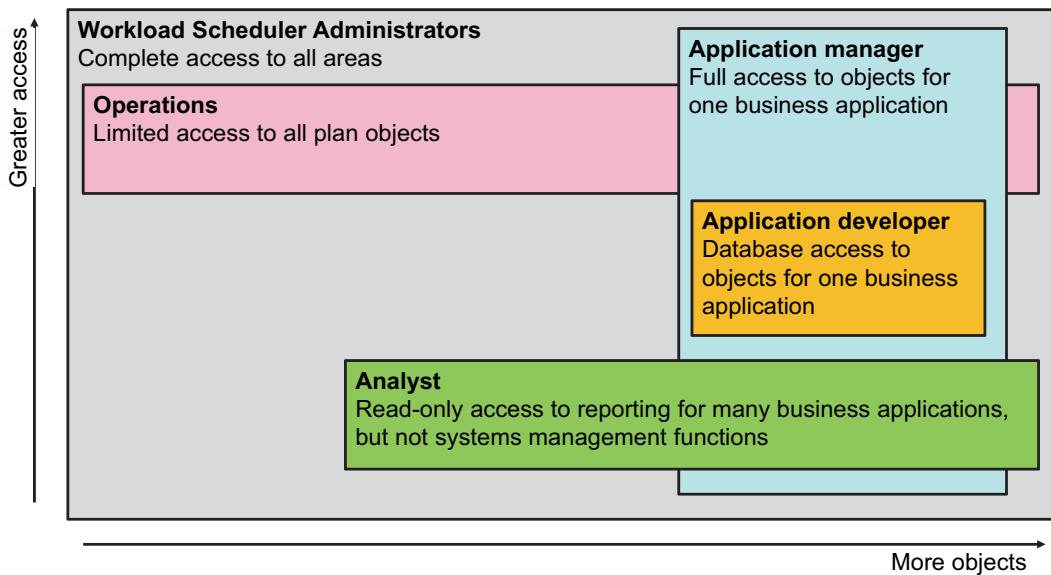
Lesson 1 Providing specific access

Providing specific access

© Copyright IBM Corporation 2017

The object of defining a security model for your Workload Scheduler users is to segregate users into groups that have specific access to specific objects in the database and plan. In this lesson, you learn how types of users can have different security requirements.

User access types

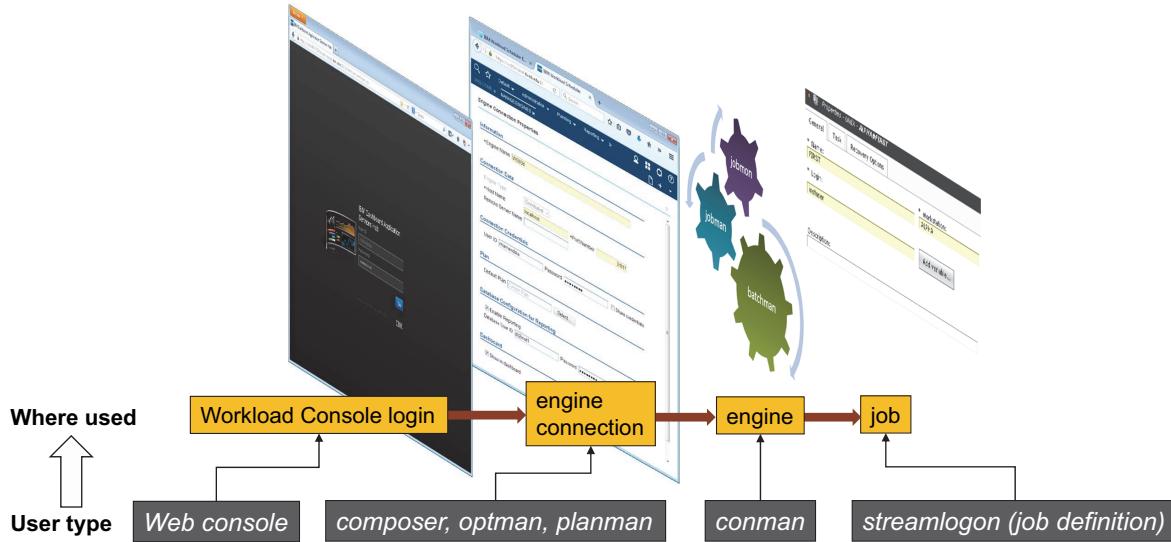


© Copyright IBM Corporation 2017

User access types

For example, developers of business applications need full access to scheduling objects that relate to the applications they develop. They do not need access to scheduling objects for other applications. Operations staff need certain specific access to almost all objects in the plan. For example, operators can rerun and cancel jobs in the plan, but they cannot define new ones or delete objects in the database.

Access entry points



© Copyright IBM Corporation 2017

Access entry points

Users can access the Workload Scheduler plan and database in different ways. You can modify your security model according to how a user accesses Workload Scheduler. Users who access Workload Scheduler by using the Workload Console might have different permissions than the same user who accesses the Workload Scheduler plan by using the **conman** command.

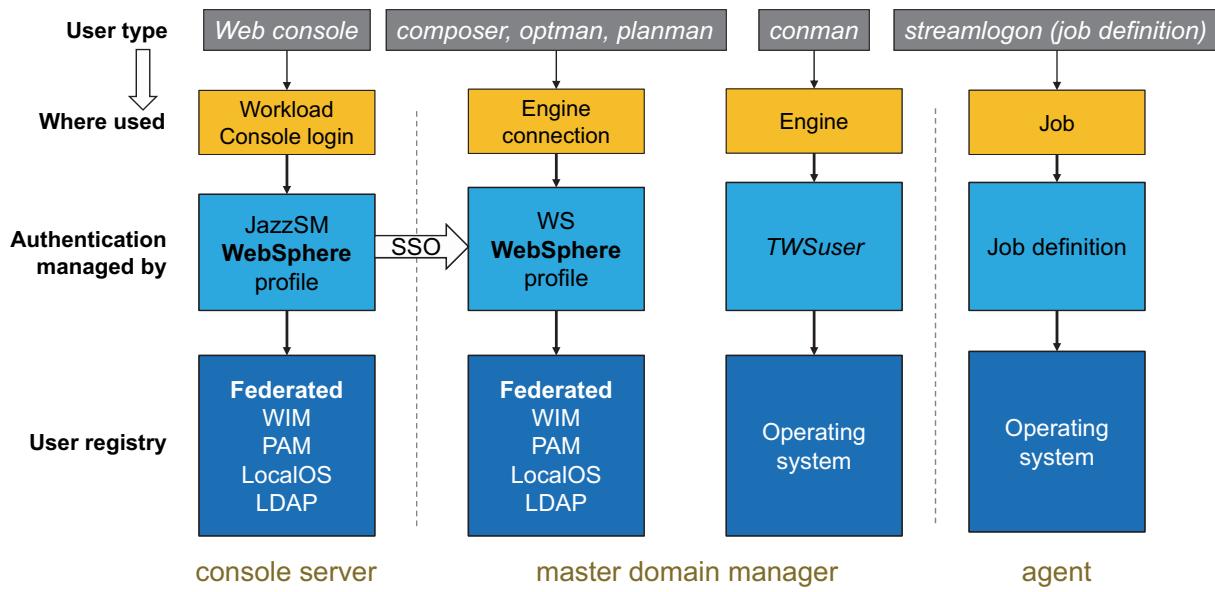
Lesson 2 Configuring authentication

Configuring authentication

© Copyright IBM Corporation 2017

The first level of security in Workload Scheduler is authentication. Authentication checks that the person or process that connects to the Workload Scheduler system is in fact who or what it declares itself to be. In this lesson, you learn about configuring authentication with Workload Scheduler.

Authentication configuration points



© Copyright IBM Corporation 2017

Authentication configuration points

You must configure authentication for each user type and entry point so that the Workload Scheduler computers and networks are protected.

Configuring authentication for jobs and conman users

Jobs that are run with the rights and privileges that are assigned to the user that is specified in the job definition.

- On UNIX and Linux computers, the Workload Scheduler agent impersonates the specified user. No further authentication is needed.
- On Windows computers, the user name and password are saved in the Workload Scheduler database, which is compiled into the Symphony file and accessed by the agent when jobs run. The user name and password that is defined in the Workload Scheduler database must match the user that is defined in Windows.

Configuring authentication for the Dynamic Workload Console and engine connection users

When you install the Dynamic Workload Console, only the user that was defined during installation can access the console. The installation user is defined in the WebSphere Application Server file registry (WIM). Before other users can access the Dynamic Workload Console, the administrator must configure a user registry, in the *JazzSM WebSphere Application Server* profile. Then, assign roles to users and groups.

When you install the Workload Scheduler master domain manager and backup master domain manager, the user that you specified during installation, the **WSuser**, can access the Workload Scheduler database, plan, and settings with administrative privileges. On UNIX and Linux computers, the WebSphere Application Server registry uses a custom PAM registry that is installed with Workload Scheduler. You can also add other registries that are described in this lesson.

By default, the WebSphere Application Server profile that is installed with the Dynamic Workload Console is configured to use the file-based (WIM) user registry. You create users and groups by using the WebSphere Application Server administrative console.

You can configure WebSphere Application Server profile that is installed with the Dynamic Workload Console to use the local operating system or PAM authentication. You can also configure WebSphere Application Server to use Lightweight Directory Access Protocol (LDAP) as the user registry.

Authentication must be configured on each instance of the WebSphere Application Server that you are installing. The same authentication method must be configured for the following components:

- Master domain manager
- Backup master domain manager
- Agents that have a connector installed



Important: You must configure authentication for the Dynamic Workload Console separately. The authentication configuration for one instance can be different from the other, unless you plan to use single sign-on, in which case all authentication configurations must be identical.

When you install Workload Scheduler, all components that use the WebSphere Application Server are configured for authentication in Virtual Member Manager (VMM) mode. This method creates a

federated user registry, which you can use with one or more of the following authentication systems:

- **Local operating system:** The default authentication system at installation on Windows operating systems
- **Custom through the pluggable authentication module (PAM):** The default authentication system at installation on UNIX and Linux operating systems
- **LDAP**
- **File registry**

If you choose to enable LDAP, you can use one of the following servers. Sample configuration templates are supplied in the documentation *IBM Workload Scheduler 9.4 Administration Guide*.

- IBM Security Directory Server
- Sun Java Director Server
- Microsoft Windows Active Directory
- z/OS Integrated Security Services LDAP Server

Using application server configuration methods

Use the administrative console

- Dynamic Workload Console
<https://dwc.example.com:16316/ibm/console>
- Workload Scheduler
<https://mdm.example.com:31124/ibm/console>
- Security > Global security > Configure

Use the wastools scripts

- On the Dynamic Workload Console server
WAhome/.../TWAUI/wastools
/opt/IBM/TWAUI/wastools
- On the Workload Scheduler Master Domain Manager server
WAhome/wastools
/opt/IBM/TWA/wastools
 - `showSecurityProperties.sh > file`
 - Edit `file`
 - `changeSecurityProperties.sh file`

© Copyright IBM Corporation 2017

Using Application Server configuration methods

Authentication can be configured by using the **administrative console** or manually, by using the WebSphere Application Server tools that are supplied with Workload Scheduler. When you configure by using the administrative console, you use a web browser to log in to the console and make changes from the Security pages. When you configure manually by using the supplied scripts, you must log in to the WebSphere server, gain administrative privileges, and edit configuration text files.

Before you change the configuration, create a backup of the current configuration by using the `backupConfig` utility.

To configure with the administrative console, perform the following steps:

1. On each instance of the WebSphere Application Server where you want to modify the default authentication configuration, open the WebSphere Administrative Console.
2. From the console, configure **Global Security**.
3. Choose and configure the authentication mechanism or mechanisms you use in your environment.

For more information about configuring authentication, see “*Configuring authentication using the WebSphere Administrative Console*” in the *IBM Workload Scheduler 9.4 Administration Guide*.

To configure manually, you can use the WebSphere Application Server tools that are supplied with Workload Scheduler. The server configuration tools locations are provided here:

- `WAhome/.../TWAUI/wastools`, for the Dynamic Workload Console
- `WAhome/wastools`, for Workload Scheduler

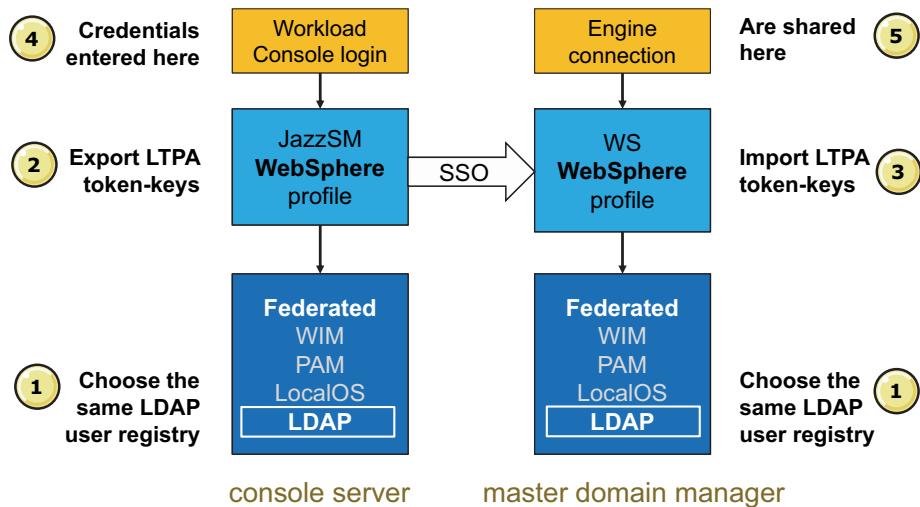
On each instance of the WebSphere Application Server where you want to modify the default authentication configuration, you run the `showSecurityProperties` script to create a template that contains the current security configuration. You modify the template by adding or changing the properties that define the authentication mechanisms you use in your environment. Finally, you run the `changeSecurityProperties` script to update the WebSphere Application Server security configuration.

Configuration scenario

In a complex environment, you might want to use the following scenario to configure your chosen authentication mechanism across your workload scheduling environment:

1. Use the `changeSecurityProperties` script to configure the mechanism on one instance of WebSphere Application Server, for example, that installed with the master domain manager.
2. Test that you can log in using the configured authentication with several user IDs.
3. On that instance, run the `showSecurityProperties` script and save the output to create a text template file that contains the configuration. The `showSecurityProperties` script extracts only those configurations that are created by using the `changeSecurityProperties` script.
4. On each of these systems where you want to configure authentication, perform these steps:
 - a. Copy the text template file that is created in the previous step.
 - b. Run `showSecurityProperties` and save the file.
 - c. Merge this file with the configuration template file.
 - d. Run `changeSecurityProperties` to update the WebSphere Application Server configuration.
 - e. Test that you can log in using the configured authentication with several user IDs.

Configuring the use of Single Sign-On (SSO)



© Copyright IBM Corporation 2017

Configuring the use of Single Sign-On (SSO)

You can configure the Dynamic Workload Console to use single sign-on (SSO) with Workload Scheduler. When you use SSO, you can monitor the plan or manage database object definitions with the same credentials you use to logon to the Dynamic Workload Console. Using Single Sign-On (SSO) allows users to authenticate once to access multiple applications that share the user registry.

When you use the *Self-Service Catalog* and *Self-Service Dashboards* apps from a mobile device, these apps automatically use the same credentials that you used to log in to the Dynamic Workload Console, if the Dynamic Workload Console is configured to use SSO.

To use SSO, both the Dynamic Workload Console and the IBM Workload Scheduler engine must be configured to share the LDAP user registry. Also, be sure that the same value is defined for the `WMMRealm` property in both the Dynamic Workload Console and master domain manager.

To use single sign-on, you must configure both instances of WebSphere Application Server to use the same *Lightweight Third-Party Authentication* (LTPA) token keys and disable their automatic regeneration. You export LTPA token-keys from the WebSphere Application Server where the Dynamic Workload Console is installed, and import them into the other instances of WebSphere Application Server.



Hint: When you configure SSO, you do not enter connection credentials in the Engine Connection Properties. You use the credentials that you use to log in to the Dynamic Workload Console.

When you configure Dynamic Workload Console to use Single Sign-On with an engine, the following conditions apply:

- If a user specifies credentials in an engine connection definition, those credentials are used, ignoring credentials shared through SSO. This behavior also includes shared engine connections that contain saved user credentials.
- If user credentials are not specified in the engine connection, the credentials you used to log in to Dynamic Workload Console are used. This behavior also includes shared engine connections with no saved user credentials.

Lesson 3 Configuring Dynamic Workload Console roles

Configuring Dynamic Workload Console roles

© Copyright IBM Corporation 2017

When you install the Dynamic Workload Console, roles are created in the Dashboard Application Services Hub for you to use. The roles determine which menu options and dashboard panes are available to a user and which activities a user can perform.

Adding users and groups to roles

Dynamic Workload Console users are granted access to resources based on their assigned roles. From the console settings menu, select **Roles** to add and remove roles and to assign access to Dynamic Workload Console portfolio sets. Only users with **admininsecuritymanager** and **Administrator** role can create, delete, or change the properties of a role. If you assign access for any other role to the Roles portlet, users in that role can view only roles and change access to views and pages.

To manage users and groups and assign them to roles, perform these steps:

1. Click the role to which you want to add users or groups, and click **Users and Groups**.
2. Click the plus sign (+) or minus sign (-) in the Users or Groups section to add or delete users or groups from the role. Enter a user ID or group name to add, and click **Search**.
3. Select the check box beside any users or groups that you want to add, and click **Add**.
4. Click **Save** to save the role.

Dynamic Workload Console rules: TWSWEBUI

Menu option	Role	TWSWEBUI Administrator	TWSWEBUI Operator	TWSWEBUI Developer	TWSWEBUI Configurator	TWSWEBUI Analyst	TWSWEBUI Business Developer
Quick Start	Y						
All configured tasks	Y	Y					
Workload design	Y		Y				
Workload environment design	Y		Y	Y			
Workload forecast	Y	Y					
• View preproduction plan	Y		Y				
Workload submission	Y	Y					
Monitor	Y	Y					
Workload reporting	Y				Y		
Manage engines	Y			Y			
Set user preferences	Y	Y	Y	Y	Y		
Manage settings	Y						
Self-service UI							Y
Troubleshooting and Support	Y	Y	Y	Y	Y		

© Copyright IBM Corporation 2017

Dynamic Workload Console roles: TWSWEBUI

The roles for accessing the Workload Scheduler views are listed in the following table.

Role	Description
TWSWEBUIAdministrator	Users with this role can see the entire portfolio and use all features of the Dynamic Workload Console.
TWSWEBUIConfigurator	Users with this role can manage Dynamic Workload Console scheduler connections, user preferences, and scheduling environment design.
TWSWEBUIOperator	Users with this role can see Dynamic Workload Console: <ul style="list-style-type: none"> • All monitor tasks • Jobs and job streams to be submitted on request • User preferences
TWSWEBUIBusinessDeveloper	Users with this role can access and use the Self-service Catalog and the Self-service dashboard mobile applications.
TWSWEBUIDeveloper	Users with this role can create, list, and edit workload definitions, workstations, and event rule definitions in the Workload Scheduler database.
TWSWEBUIAnalyst	Users with this role can manage Dynamic Workload Console reports and user preferences. Users with this role can also use the Self-service Catalog and Self-service dashboards for submitting service requests and displaying the dashboard.

Dynamic Workload Console rules: Dynamic Workload Broker

	TDWB Administrator	TDWB Operator	TDWB Developer	TDWB Configurator
Broker Design				
• Create jobs	Y		Y	
• Manage jobs	Y	Y	Y	
• Create logical resources, groups	Y	Y		Y
Broker settings	Y	Y	Y	Y
Broker monitoring				
• Monitor jobs	Y	Y		
• Monitor computers, resources, groups	Y	Y		Y

© Copyright IBM Corporation 2017

Dynamic Workload Console roles: Dynamic Workload Broker

Additional roles are created for accessing the Dynamic Workload Broker views in the Dynamic Workload Console. The DWB roles are listed in the following table.

Role	Description
TDWBAdministrator	Users with this role can see the entire Workload Broker portfolio and use all Workload Broker features of the Dynamic Workload Console.
TDWBConfigurator	Users with this role can create broker logical resource and resource groups and monitor broker computers and resources. Users with this role can also manage Dynamic Workload Console broker server connections and user preferences.
TDWBOperator	Users with this role can create broker logical resource and resource groups and manage broker jobs. Users with this role can also monitor these items: <ul style="list-style-type: none"> • Broker jobs • Broker computers • Broker logical resources • Broker resource groups Users with this role can also manage Dynamic Workload Console broker server connections and user preferences.
TDWBDeveloper	Users with this role can create and manage broker jobs. Users with this role can also manage Dynamic Workload Console broker server connections and user preferences.

Lesson 4 Configuring user authorization

Configuring user authorization

© Copyright IBM Corporation 2017

To control Workload Scheduler security, you use a configuration file that is named the **Security** file. This file controls activities such as these examples:

- Linking workstations
- Accessing command-line interface programs and the Dynamic Workload Console
- Performing operations on scheduling objects in the database or in the plan

In the file, you specify the scheduling objects the user can access and the actions the user can perform on those objects. You can determine access by object type, for example, workstations or resources. You can also determine access within an object type by selected attributes, such as the object name or the workstation in the object definition. You can use wildcards to select related sets of objects. Access rights can be granted on an *included* basis, an *excluded* basis, or a combination of both.

Whenever you must change access permissions, you modify the configuration file and convert it into an encrypted format for performance and security, which replaces the previous file. The system uses this encrypted security file from that point onward.

Each time a user runs Workload Scheduler programs, commands, and user interfaces, the product compares the name of the user with the user definitions in the security file. The comparison is used

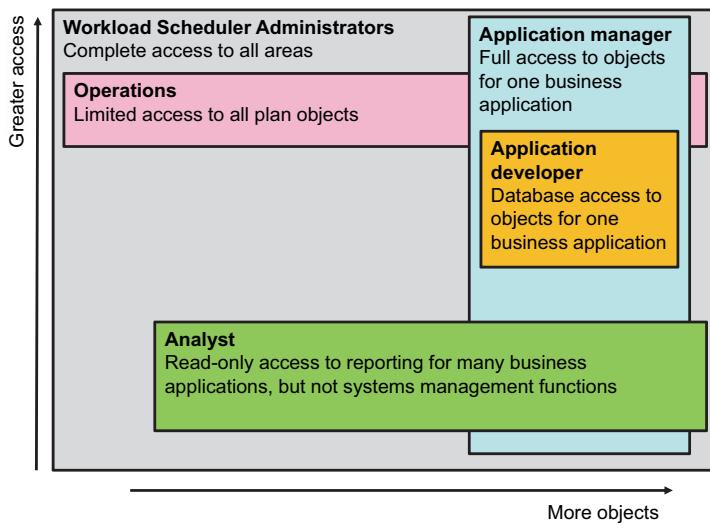
Lesson 4 Configuring user authorization

to determine whether the user has permission to perform those activities on the specified scheduling objects.

By default, the security on scheduling objects is managed locally on each workstation. The system administrator or the **wsuser** on that system can decide user permissions. They can determine which users defined on that system can access which scheduling resources in the network and what actions they can perform.

Alternatively, you can centralize control of how objects are managed on each workstation. This method can be configured by setting a global option. In this scenario, you configure all user permissions in the security file on the master domain manager. The encrypted version of the file is distributed automatically every time you run **JnextPlan**. All workstations have the file locally to determine the permissions of the users on that workstation.

Using the Security file



To which users does this access apply?

- User name
- Group membership

What kinds of objects can these users access?

For example,

- Workstations that are fault-tolerant agents
- Jobs that do not log on as Administrator
- Event rules that send email
- The Security file or the Production Plan

What type of access do these users have to those objects?

- Link and unlink workstations
- Submit, add, modify, delete jobs
- Manage security
- Create event rules
- View reports

© Copyright IBM Corporation 2017

Using the Security file

When you install Workload Scheduler, the default security definitions are stored in a configuration file as **WAhome/TWS/Security.conf**. A compiled, operational copy is installed as **WAhome/TWS/Security**.

This version of the file contains a full access definition for the user who installed the product, **wsuser**, and the system administrator, which is **root** on UNIX or **Administrator** on Windows. Those users are the only users who are defined and allowed to connect to the user interfaces and to perform all operations on all scheduling resources.

Within the network, you can use the security file to distinguish between local root users and the root user on the master domain manager. You permit local root users to perform operations that affect only their login workstations. You provide the master domain manager root user the authorizations to perform operations that affect any workstation across the network. As you work with the product, you might want to add more users with different roles and authorizations. These users can perform specific operations on a defined set of objects.

Workload Scheduler actions and access you can restrict

Design and monitor workload Create, delete, modify, list and view details for jobs, job streams, users, prompts, resources, calendars, run cycle groups, variable tables, workload applications, and parameters	Submit workload • Only existing job definitions • Existing job definitions and ad hoc jobs • Existing job stream definitions	Access IBM Application Lab Permit or deny
Modify current plan • Add or remove job or job stream dependencies • Change job or job stream priorities or limits • Cancel jobs or job streams • Allow actions on job on remote systems • Rerun, kill, confirm, or release jobs • Release job streams • Reply to prompts • Alter user passwords • Change resource quantities	Manage workload environment • List workstations, display details • Create, delete, modify, workstations • Start, stop, shut down a workstation • Change limit or fence • Reset an FTA • Link or unlink a workstation • Use 'console' command from conman • Upgrade a workstation	Manage event rules • List event rules, display details • Create, delete, modify event rules • Display actions in the event rules • Monitor triggered actions • Use action types in the event rules • Use events in the event rules
	Generate workload reports Permit or deny	Administrative tasks • View configuration (dump security and global options) • Change configuration (makesec, optman chg) • Unlock, delete objects definitions • Allow planman deploy, prodsked, and stageman

© Copyright IBM Corporation 2017

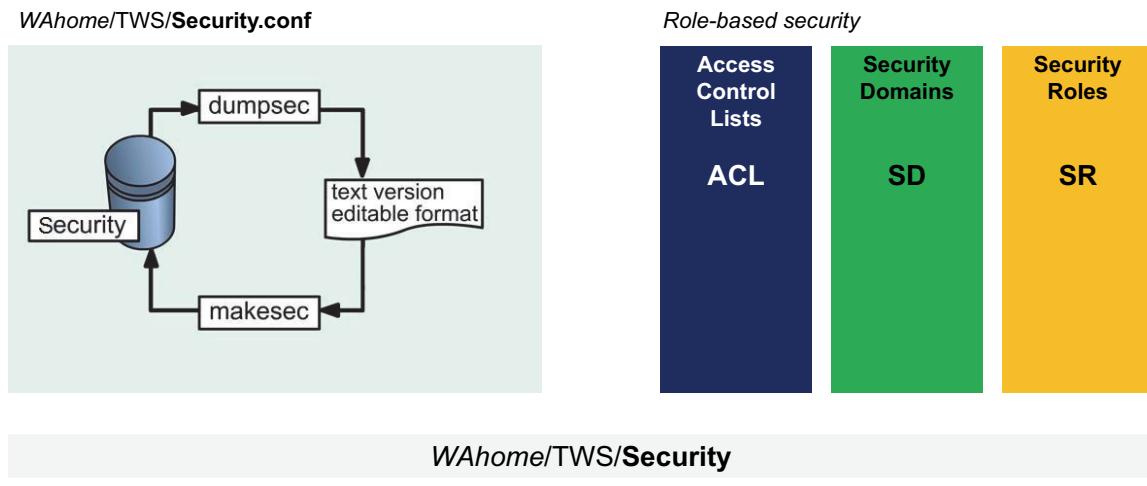
Workload Scheduler actions and access you can restrict

In the Security file, you can specify which scheduling objects users can manage and the way that they can use them. You define these settings by writing user definitions, or access control lists. User definitions and access control lists are associations between a name and a set of users, the objects they can access, and the actions they can perform on the specified objects.

When you define user authorization, consider these items:

- When you issue *composer* commands, the user authorizations are checked in the security file of the master domain manager. The methods that are used to manage the entries in the database are started on the master domain manager. You must use these criteria to define the user:
 - As system user on the system where the master domain manager is installed
 - In the security file on the master domain manager with the authorizations that are required to run the allowed commands on the specific objects
- When you type *conman* commands, the user must be authorized to run the specific commands in the security file on the computer where the *conman* program runs. For commands that access the database, such as **submit job**, the user must have permissions on the connecting workstation and on the master domain manager where the command runs.

Configuring the security file



© Copyright IBM Corporation 2017

Configuring the Security file

You can manage authorization security settings in the Workload Scheduler environment in one of the following ways:

1. Edit the `Security.conf` file, then compile and distribute the compiled Security file as needed. You learn how to edit and maintain the `Security.conf` file in Lesson 5, “Managing the `Security.conf` file,” on page 7-23.
2. Enable role-based security configuration, and create *security roles*, *security domains*, and *access control lists* in the database by using the Dynamic Workload Console or the composer command-line interface. You learn how to use role-based security management in Lesson 6, “Using role-based user authorization,” on page 7-43.



Attention: To create or modify the security configuration, you must have permission to *modify* the `file` object that has the attribute `name=Security`.

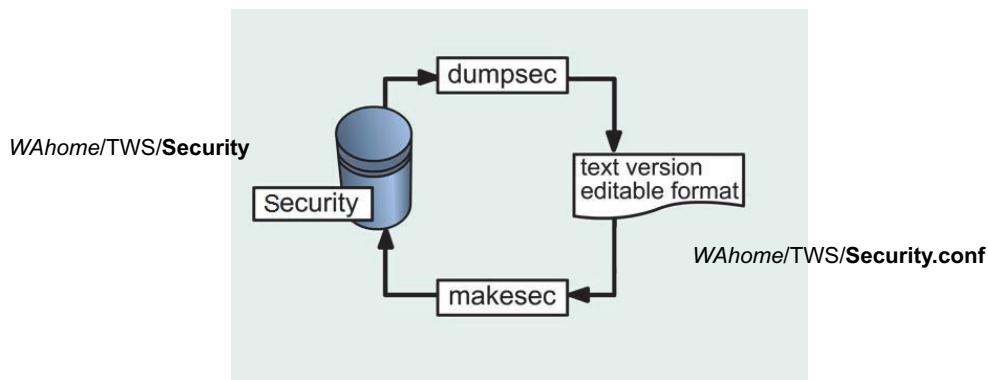
Lesson 5 Managing the Security.conf file

Managing the Security.conf file

© Copyright IBM Corporation 2017

In this lesson, you learn how to create and manage the Security file by editing and compiling a text file. By default, Workload Scheduler authorization permissions can be managed by editing a text file that contains the security definitions. The default file is named **Security.conf**, but you can use any text file instead. When you use the **Security.conf** file to manage the **Security** file and the permissions it configures, you create the text file from the existing Security definitions. You then edit the text file by using any suitable text editor, and compile it to create the replacement Security file.

Using the dumpsec and makesec commands



© Copyright IBM Corporation 2017

Using the dumpsec and makesec commands

When you install Workload Scheduler, the default security definitions are stored in a configuration file as **WAhome/TWS/Security.conf**. A compiled, operational copy is installed as **WAhome/TWS/Security**.

You use the **dumpsec** command to create an editable text file from the compiled security file. To use the **dumpsec** command, change to the directory where the **Security** file is located (**WShome**) and run the following command:

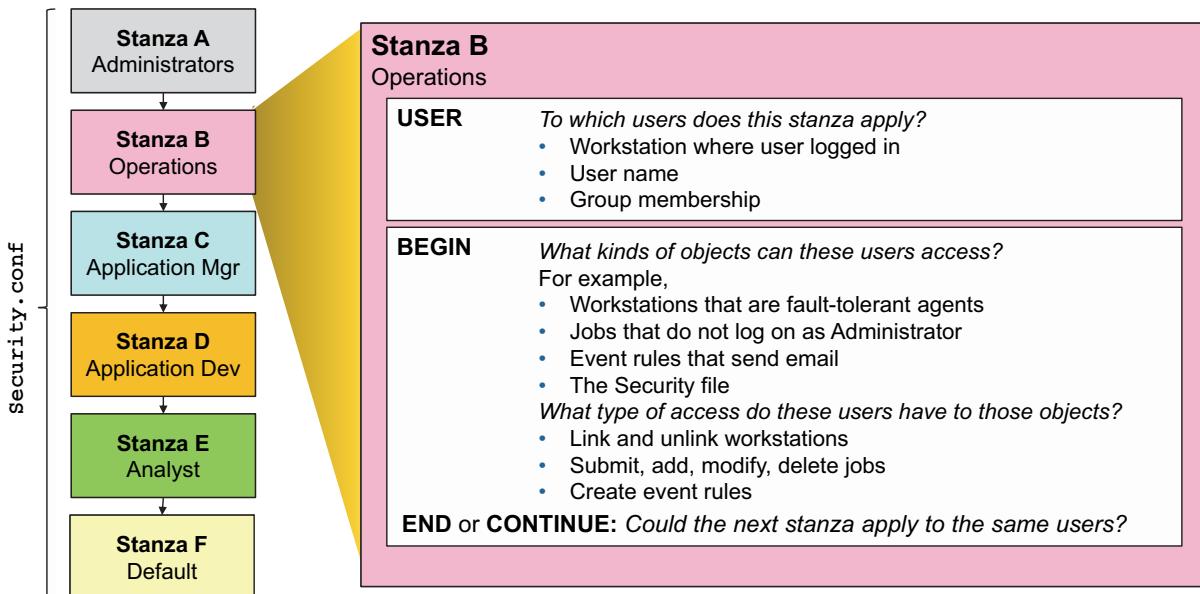
```
dumpsec > Security.conf
```

You can direct the **dumpsec** output to any text file.

Use the **makesec** command to compile the edited text file and install the new security settings. The **-verify** option checks the syntax of the input file without attempting to compile or install its settings. To compile the **Security.conf** text file, run the following command:

```
makesec Security.conf
```

Configuring the Security file



© Copyright IBM Corporation 2017

Configuring the Security file

In the Security.conf file, you can specify which scheduling objects users can manage and the way that they can use them. You define these settings by writing user definitions. A user definition is an association between a name and a set of users, the objects they can access, and the actions they can perform on the specified objects.

When you define user authorization, consider these items:

- When you issue *composer* commands, the user authorizations are checked in the security file of the master domain manager. The methods that are used to manage the entries in the database are started on the master domain manager. You must use these criteria to define the user:
 - As system user on the system where the master domain manager is installed
 - In the security file on the master domain manager with the authorizations that are required to run the allowed commands on the specific objects
- When you enter *conman* commands, the user must be authorized to run the specific commands in the security file on the computer where the conman program runs. For commands that access the database, such as **submit job**, the user must have permissions on the connecting workstation and on the master domain manager where the command runs.

Using variables in attribute definitions

The following variables are supplied with the product. You can use them in these object attributes:

User identifiers

\$jclgroup	The group name of the executable file of a job
\$jclowner	The owner of the executable file for a job
\$owner	The creator of a job stream and its jobs
\$user	The user that runs the Workload Scheduler command or program



Note: You can verify the variables \$jclgroup and \$jclowner only if the user is running a Workload Scheduler program on the workstation where the executable file for the job is present. If the program is being run on a different workstation, the user is denied access.

Workstation identifiers

\$master	The Workload Scheduler master domain manager
\$manager	The Workload Scheduler domain manager
\$remotes	All standard agent workstations
\$slaves	All fault-tolerant agent workstations
\$thiscpu	The workstation on which the user is running the Workload Scheduler command or program

Variable table identifiers

\$default	The name of the current default variable table
-----------	--

User attributes

The user attributes define who has the access that is going to be later defined. They can identify one user, a selection of users, a group of users, a selection of groups of users, or all users. You can also exclude one or more specific users or groups from a selection. You can use logon ID and group name to identify users. You can also describe the user by the workstation from which they log on. You can mix selection criteria. For example, you can select all users in a named group that can access from a set of workstations that are identified by a wildcard and exclude a specific set of users who are identified by their logon IDs.

You make this selection by specifying one or more user attributes. Each attribute can be included or excluded from the selection. Thus, for each attribute type, you can include all or include a selection.

Include all

Include all is the default. For example, if you want to include all groups, you add no user attribute about any group.

Include a selection

Including a selection can be defined in one of these ways:

- Include users that you want to select, such as individuals or one or more sets
- Exclude (from the **include all** default) all users that you do not want to select
- Include a set of users and then exclude some users that are contained in the set

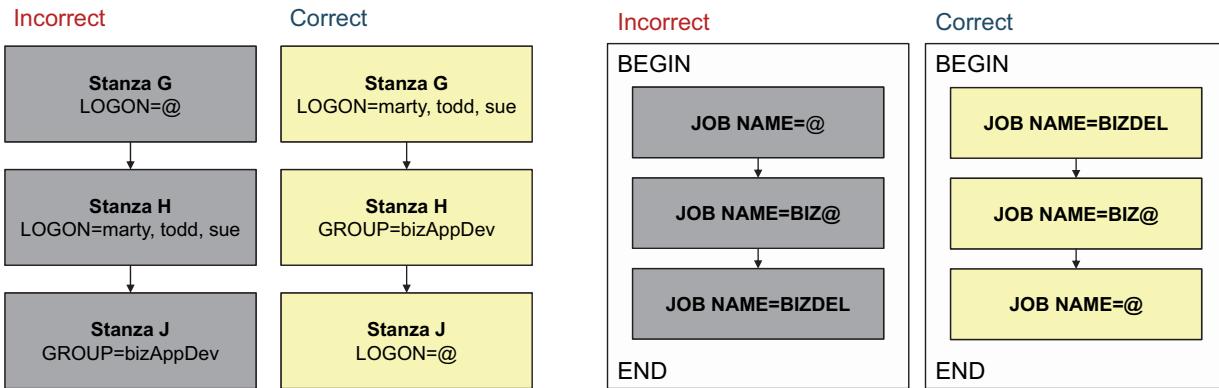
The options that you choose depends on which option is easier to specify. To include an attribute, precede the user attribute expression by a plus (+) sign. All users who are identified by the expression are selected, unless an exclude expression also selects them. If the first attribute in your definition is an *include*, it does not require a plus (+) sign, because the sign is implicit.

If you specify no user attributes, the default is to include all users, on all workstations, in all groups. If you want to define, for example, all users except one named user, you supply the exclude definition for the one user.

To exclude, precede the user attribute expression by a tilde (~) sign. All users who are identified by the expression are never selected, regardless of whether any include expressions select them.

Defining in the correct order

Matching is top down



© Copyright IBM Corporation 2017

Defining in the correct order

You must order definitions from most specific to least specific. Workload Scheduler scans the security file from the top down, with each ID tested against each definition in turn. If the ID is satisfied by the definition, it is selected, and the matching stops at the END statement.

Specifying object types

Specify one or more object types that the user or users in the associated user definition are authorized to access. You can specify the object type, but no attributes. If you do so, the authorized actions that are defined for the user with the access keyword apply to all the objects of that type that are defined in the domain. If an object type from the following list is omitted for a user or users, no objects of that type can be accessed. You can specify the object types that are listed in the following table.

Object type	Description
action	Actions that are defined in scheduling event rules
calendars	User calendars
cpu	Workstations, domains, and workstation classes
event	Event conditions in scheduling event rules
eventrule	Scheduling event rule definitions
file	Workload Scheduler database file

Object type	Description
job	Scheduled jobs and job definitions
parameter	Local parameters. See note about variable tables.
prompt	Global prompts
report	The reports on the Dynamic Workload Console
resource	Scheduling resources
runcygrp	Run cycle groups
schedule	Job streams
userobj	User objects
vartable	Variable tables include authorization to the variable definitions in the tables
wkldappl	Workload application templates

Specifying object attributes

You specify one or more attributes that identify a set of objects that the user of the user definition is authorized to access. You can also specify the object type, without object sets. If you do so, the authorized actions that are defined for the user with the **access** keyword apply to all the objects of that type that is defined in the domain.

General syntax

Each object attribute is specified as follows:

`object_attribute=value`

- `object_attribute` differs according to the object. You can use the name to select all objects. You can also use the workstation name to select some objects, such as jobs. For information about which attributes are available for each object type, see the following “Object attribute” section.
- `value` identifies an individual object or, when you use wildcards, a set of objects.

Object attribute

Object attributes are used to identify a specific set of object within all objects of the same type. For example, access can be restricted to a set of resource objects that have the same name, the same workstation, or both.

You can *include* or *exclude* each attribute from the selection by using the plus (+) and tilde (~) symbols, like the user attributes.

The detailed syntax and use of the selection expressions for objects is the same as the syntax used to select users:

```
[~]object_attribute=value[,value]... [{+|~}object_attribute=value[,value]...]
```

Order of object definition

You must order object definitions from most specific to least specific, like user attributes. In the following example, you see an incorrect and correct method for ordering object definitions.

```
job name=@ access=display
```

```
job name=ar@ access=@
```

In this case, a job with the name beginning with *ar* satisfies the first definition. The job has display access, but not all access.

```
job name=ar@ access=@
```

```
job name=@ access=display
```

Ensure that you order object definitions from most specific to least specific also when you use the **Continue** keyword. With this keyword, you match more user definitions to a single user. The user receives access from more user definition statements. These accesses are then processed in the order they are written in the security file.

Security definition examples

In the following sections, you see examples of security definitions. Note the following information about these examples:

- * Workload Scheduler security examples
- * Lines starting in “**” or “#” are comments

Defining user qualifications

- * This section defines what logons are matched to the object attributes

```

USER BIZDEL
CPU =
$MASTER, # Logged onto the master - I am in this stanza
$MANAGER, # or, logged to any domain manager not the master
$REMOTES, # or, logged on to any FTA
$SLAVES, # or, logged on to any Standard Agent
$THISCPU, # or, if I am logged on locally
@ # or, I am logged in anywhere
+GROUP =
bizAppDev, # My logon group is "bizAppDev"
w@ # or my logon group starts with "w"
~GROUP =
bizAppOps@ # and I am not in the "bizAppOps" groups
+LOGON =
marty, # and I am one of the listed users
todd,
sue

BEGIN
* Statements between BEGIN and END apply to the users that match above.

```

Defining workstation access

```
# This section defines workstation definition and management
#
CPU CPU = $THISCPU, # The workstation where you ran conman/composer
$MASTER, # The master domain manager
$SLAVES, # Any fault tolerant agent
appdevs001 # or TWS workstation "appdevs001"

* Give access only to these types of workstations
TYPE =
AGENT,      # Only "dynamic" agents
BROKER,     # Run JSDL from the Broker library
FTA,        # Fault tolerant agents
D-POOL,      # Dynamic pool
MANAGER,    # Domain managers, including the master
POOL,        # Workstations type POOL
REM-ENG,    # Remote engine workstation
S-AGENT,    # Standard agents
X-AGENT,    # Extended agents

* Give these types of access
ACCESS =
ADD,        # composer add
CONSOLE,    # conman cons
DELETE,     # composer delete
DISPLAY,    # composer display
FENCE,      # conman fence
LIMIT,      # conman limit
LINK,       # conman link
MODIFY,     # composer mod
RESETFTA,   # conman resetfta
RUN,        # composer make new (jobs using JavaExt)
SHUTDOWN,   # conman shut
START,      # conman start
STOP,       # conman stop
UNLINK,     # conman unlink
LIST,       # conman showcpu/composer list
UNLOCK,     # composer unlock
```

Defining job access

```
# The next section defines job objects definition and management

JOB CPU =
    $THISCPU, # The same workstation where the user logs on
    $MASTER, # The master workstation
    $SLAVES, # Any fault tolerant agent
    $REMOTES # Any standard agent
    biz@ # Any workstation whose name starts with "biz"

+ NAME = A@ # Any job with the name starting in A
~ NAME = A2@ # But not if it starts with A2
+ LOGON=
    $USER,      # Streamlogon is the conman/composer user
    $OWNER,     # Streamlogon is the job creator
    $JCLOWNER, # Streamlogon is the OS owner of the file
    $JCLGROUP # Streamlogon is the OS group of the file
~ LOGON = root, twsuser
    # The job does not logon as "root" or "twsuser"
+ JCL = "/usr/local/bin/@" # The script is in /usr/local/bin
~ JCL = "@rm@" # But not any command containing "rm"

* ACCESS defines what kind of access privileges are granted
ACCESS =
    ADD,          # composer new/rename
    ADDDEP,       # conman adj
    ALTPRI,       # conman apj
    CANCEL,       # conman cj
    CONFIRM,      # conman confirm
    DELDEP,       # conman ddj
    DELETE,       # composer delete/rename
    DISPLAY,      # conman, composer display/create/list/print
    KILL,         # conman kill
    MODIFY,       # composer mod
    RELEASE,      # conman rj
    REPLY,        # conman reply (local prompt, recovery)
    RERUN,        # conman rerun
    SUBMIT,       # conman sbj/sbd
    USE,          # composer use in sched
    LIST,         # conman showjobs
    UNLOCK        # composer unlock
```

Configuring access to the Windows user objects

```
# Windows username/passwords are stored as the userobj
USEROBJ

CPU =
$THISCPU, # On the same workstation where the user logs on
$MASTER, # On the master workstation
$SLAVES, # On any fault tolerant agent
$REMOTES # On any standard agent
+ LOGON= "Workgroup\Administrator","wsuser" # Admin or wsuser
~ LOGON= "MyDomain\@" # but not in Domain "mydomain"

# Access keyword defines which privileges are permitted
ACCESS =
ADD,      # composer add
DELETE,   # composer delete
DISPLAY,  # composer display
MODIFY,   # composer modify
ALTPASS,  # conman altpass
UNLOCK    # composer unlock
```

Configuring access to job streams

```
# The SCHEDULE section defines access to job streams
SCHEDULE
    CPU =
        $THISCPU, # The same workstation where the user logs on
        $MASTER, # The master workstation
        $SLAVES, # Any fault tolerant agent
        $REMOTES # Any standard agent
        + NAME = ABC@   # the name must start with ABC
        ~ NAME = @ZZ    # and not end in "ZZ"
# Which accesses are allowed
ACCESS =
    ADD,      # composer add
    ADDDEP,   # conman ads
    ALTPRI,   # conman aps
    CANCEL,   # conman cs
    DELDEP,   # conman dds
    DELETE,   # composer delete
    DISPLAY,  # composer dis
    LIMIT,    # conman ls
    MODIFY,   # composer mod
    RELEASE,  # conman rs
    REPLY,    # conman reply (local prompt)
    SUBMIT,   # conman sbs
    LIST,     # conman ss
    UNLOCK   # composer unlock
```

Configuring access to calendars

```
# Business calendars
CALENDAR NAME = @          # which calendar?
ACCESS =
    ADD,      # composer add
    DELETE,   # composer delete
    DISPLAY,  # composer display
    MODIFY,   # composer modify
    USE,     # use in a job stream
    UNLOCK   # composer unlock
```

Configuring access to event rules (events)

```
# Create event rules
EVENTRULE      NAME=    @
                ACCESS= ADD, DELETE, DISPLAY, MODIFY, LIST, UNLOCK
# Which of the TWS objects events can be used
EVENT PROVIDER=TWSObjectsMonitor
        TYPE =
                JobStatusChanged,
                JobUntil,
                JobSubmit,
                JobCancel,
                JobRestart,
                JobLate,
                JobStreamStatusChanged,
                JobStreamCompleted,
                JobStreamUntil,
                JobStreamSubmit,
                JobStreamCancel,
                JobStreamLate,
                WorkstationStatusChanged,
                ApplicationServerStatusChanged,
                ChildWorkstationLinkChanged,
                ParentWorkstationLinkChanged,
                PromptStatusChanged
        ACCESS= USE

# Which of the file monitor events can be used
EVENT PROVIDER=FileMonitor
        TYPE =
                FileCreated,
                FileDeleted,
                ModificationCompleted,
                LogMessageWritten
        ACCESS = USE
        CUSTOM = SAMPLE
        # Specify security attribute for custom-made event drivers
```

Configuring access to event rules (Actions)

```
# Event rule actions
ACTION PROVIDER = TECEventForwarder      # Forward events to omnibus
  TYPE = TECFWD
  ACCESS = DISPLAY, SUBMIT, USE, LIST
  HOST = omnibushost.ibm.com
  PORT = SNMPListenerport
ACTION PROVIDER=MailSender                # Send email
  TYPE = SendMail
  ACCESS = DISPLAY, SUBMIT, USE, LIST
ACTION PROVIDER = MessageLogger           # Write to the operator log
  TYPE = MSGLOG
  ACCESS = DISPLAY, SUBMIT, USE, LIST
ACTION PROVIDER = TWSAction               # Perform TWS actions
  TYPE = GenericAction
  ACCESS = DISPLAY, SUBMIT, USE, LIST
```

Configuring access to generate reports

```
# Report generation
REPORT
  NAME =
    RUNHIST, # Job Run History
    RUNSTATS, # Job Run Statistics
    WWS,      # Workstation Workload Summary
    WWR,      # Workstation Workload Runtimes
    SQL,      # Custom SQL
    ACTPROD,  # Actual production details (current and archived)
    PLAPROD   # Planned production details
  ACCESS = DISPLAY
```

Configuring access to important Workload Scheduler files

```
# Special purpose TWS files
FILE
  NAME =
    GLOBALOPTS, # optman (ls, show, chg)
    PRODSKED,   # planman (current plan)
    SECURITY,   # dumpsec, makesec
    SYMPHONY,   # stageman
    TRIALSKED   # planman (trial and forecast)

  ACCESS =
    BUILD,     # planman deploy, stageman
    DELETE,    # delete objects from the database
    DISPLAY,   # optman (ls, show), dumpsec, planman showinfo
    MODIFY     # optman chg, makesec, planman crt, ext, reset, crttrial, exttrial,
crtfc)
```

Configuring access to prompts

```
# Prompts
PROMPT NAME = A@           # Prompts with names that start with "A"
ACCESS =
  ADD,      # composer add/rename
  DELETE,   # composer delete/rename
  DISPLAY,  # composer display/list/print/create, conman recall
  MODIFY,   # composer modify
  REPLY,    # conman reply
  USE,      # use in your job stream
  LIST,     # composer list/ conman showprompts
  UNLOCK   # composer unlock
```

Configuring access to resources

```
# Resources
RESOURCE
CPU =
    @,          # All workstations
    $THISCPU,  # The same workstation where the user logs on
    $MASTER,   # The master workstation
    $SLAVES,   # Any fault tolerant agent
    $REMOTES  # Any standard agent
+ NAME = @
ACCESS =
    ADD,        # composer add
    DELETE,     # composer delete
    DISPLAY,    # composer display
    MODIFY,     # composer modify
    RESOURCE,   # conman res/release util.
    USE,        # use in your job stream
    LIST,       # composer list/ conman showprompts
    UNLOCK      # composer unlock
```

Configuring access to run cycle groups

```
# Run cycle groups
RUNCYGRP
NAME = @
ACCESS =
    ADD,        # composer add
    DELETE,     # composer delete
    DISPLAY,    # composer display
    MODIFY,     # composer modify
    USE,        # use in your job stream
    LIST,       # composer list
    UNLOCK      # composer unlock
```

Configuring access to variables and variable tables

```

# Variables
PARAMETER
    CPU = @,           # All workstations
    $THISCPU, # The same workstation where the user logs on
    $MASTER,  # The master workstation
    $SLAVES,   # Any fault tolerant agent
    $REMOTES  # Any standard agent
+ NAME = @
ACCESS =
    ADD,      # composer add/rename or "parms -c"
    DELETE,   # composer delete/rename
    DISPLAY,  # composer display or "parms"
    MODIFY,   # composer mod or "parms -c"
    UNLOCK    # composer unlock

# Variable Tables
# Hint: To authorize users to create their own variable tables, but not
#       touch the default table, use the following:
#       VARTABLE NAME=$DEFAULT ACCESS=DISPLAY,USE,LIST,UNLOCK
#       VARTABLE NAME=@          ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK

VARTABLE
    NAME = A@, # Access to variable tables with name that start in "A"
    $DEFAULT # Access to the default variable table
ACCESS =
    ADD,      # composer new/add
    DELETE,   # composer delete
    DISPLAY,  # composer display, extract
    MODIFY,   # composer modify, lock
    USE,      # use variable tables in run cycles,
              # job streams, and workstations
    LIST,     # composer list; also, list individual variable
              # entries within the table
    UNLOCK    # composer unlock
END

```

Configuring access to workload applications

```
# Workload Applications
WKLDAPPL
  NAME = @
  ACCESS =
    ADD,      # composer add
    DELETE,   # composer delete
    DISPLAY,  # composer display
    MODIFY,   # composer modify
    LIST,     # composer list
    UNLOCK   # composer unlock
```

Using the Continue keyword

```
# The CONTINUE keyword authorizes you to accumulate access permissions
# from as many stanzas as needed
# Remember, order is still important! If you give DISPLAY to everyone
# in the first group they, do not get more in following groups
#
USER BIZAPPADMINS
  CPU=@+GROUP=bizAppDev # If you are a member of the bizAppDev group
BEGIN
  JOB CPU=biza@  ACCESS=DISPLAY,USE,LIST
CONTINUE # Searching continues to the next USER stanza

USER BIZOPSADMINS
  CPU=@+GROUP=bizAppOps # If I am a member of the bisAppOps group
BEGIN
  JOB CPU=bizo@  ACCESS=DISPLAY,USE,LIST
CONTINUE# If my login is a member of coswell and spacely
        # I can use jobs on both workstations

USER BOTHADMINS
  CPU=@+GROUP=bizAppDev,bizAppOps
  # If I am a member of either the bizAppDev or bizAppOps group
BEGIN
  JOB      CPU=@      ACCESS=LIST
END      # End searching combinations
```

Setting a default security group

```
#  
# A Special "Default" group for everybody that is not listed in previous  
# sections cause unlisted users to be accepted through JSC connections  
# but do not have access to any objects when connected  
# (they are "trapped" by the system and logged)  
#  
USER DEFAULT  
CPU = @  
+LOGON = # Everybody that is not listed previously  
@  
BEGIN  
# No access privileges!  
END
```

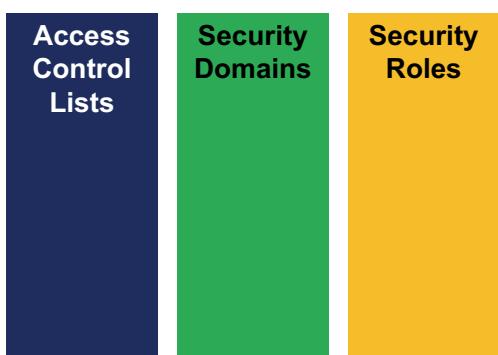
Lesson 6 Using role-based user authorization

Using role-based authorization

© Copyright IBM Corporation 2017

In this lesson, you learn how to create and manage the Security file by using role-based security authorization. Workload Scheduler authorization permissions can be managed by creating objects in the Workload Scheduler database.

Security objects in the database



Security role

A list of *actions* you can perform on a set of object types

Security domain

A list of *objects* that you can manage

Access control list

A list of *users* or *groups* you assign to **security roles** within a **security domain**

Security objects in the database

© Copyright IBM Corporation 2017

When you use role-based security to manage the **Security** file and the permissions it configures, you create **security role**, **security domain**, and **access control list** objects in the database. Workload Scheduler automatically compiles the objects for you, and then replaces the existing Security file.

To enable role-based security in your Workload Scheduler environment, you must set the global option `enRoleBasedSecurityFileCreation` to **yes**. When you enable role-based security, Workload Scheduler creates a Security file that contains the security objects that are contained in the database. You can create security objects in the database before you enable role-based security.



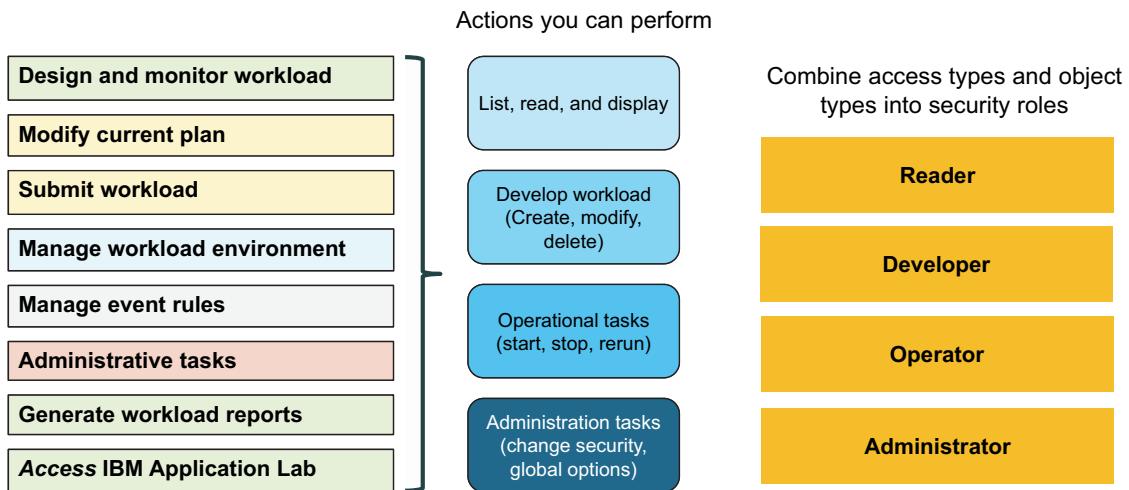
Important: Use caution when you start to use role-based security. Especially be careful about the following points to be sure that the security model doesn't exclude all activity.

- Before you switch to role-based security, make sure that at least one user has permissions to modify and build Workload Scheduler security.
- Changes that you make in security roles, security domains, and access control lists are effective immediately after you save them.

You use the Dynamic Workload Console or the composer command to create security roles, security domains, and access control lists. When you work with role-based security objects in the database, they are not continuously locked. When you update an object, it is locked when you save

the object, and immediately unlocked. If another user edits the same database security object, they see a warning that the first user updated the object.

Configuring security roles



© Copyright IBM Corporation 2017

Configuring security roles

Security roles in the database represent lists of actions that users or groups might start on types of Workload Scheduler objects.

When you create a security role, you combine types of access for each type of Workload Scheduler object into a group that you can manage as a unit. The security role can be specific to one type of database object (only jobs, for example), or it can encompass many types of database objects.

For example, a read-only security role for jobs, would include only DISPLAY and LIST access, while a security role designed for operators would include ALTPRI, CANCEL, CONFIRM, DELDEP, KILL, RELEASE, REPLY, RERUN, and LIST. The same security role might also include operations tasks for job streams (ADDDEP, ALTPRI, CANCEL, DELDEP, LIMIT, RELEASE, REPLY, and LIST) and workstations (CONSOLE, FENCE, LIMIT, LINK, START, STOP, UNLINK, and LIST), or you might choose to separate workstation access and job stream access into separate security roles for more fine-grained control.

[Table 1](#) shows the types of accesses you can use in a security role definition.

Table 1 Access types for Workload Scheduler objects

Object type	Access types
Workstations	ADD, CONSOLE, DELETE, DISPLAY, FENCE, LIMIT, LINK, MODIFY, RESETFTA, RUN, SHUTDOWN, START, STOP, UNLINK, LIST, UNLOCK
Jobs	ADD, ADDDEP, ALTPRI, CANCEL, CONFIRM, DELDEP, DELETE, DISPLAY, KILL, MODIFY, RELEASE, REPLY, RERUN, SUBMIT, USE, LIST, UNLOCK

Table 1 Access types for Workload Scheduler objects

Object type	Access types
Users	ADD, DELETE, DISPLAY, MODIFY, ALTPASS, UNLOCK
Job streams	ADD, ADDDEP, ALTPRI, CANCEL, DELDEP, DELETE, DISPLAY, LIMIT, MODIFY, RELEASE, REPLY, SUBMIT, LIST, UNLOCK
Calendars	ADD, DELETE, DISPLAY, MODIFY, USE, UNLOCK
Event rules	USE
Event rule actions	DISPLAY, SUBMIT, USE, LIST
Reports	DISPLAY
File	BUILD, DELETE, DISPLAY, MODIFY
Prompts	ADD, DELETE, DISPLAY, MODIFY, REPLY, USE, LIST, UNLOCK
Resources	ADD, DELETE, DISPLAY, MODIFY, RESOURCE, USE, LIST, UNLOCK
Run cycle groups	ADD, DELETE, DISPLAY, MODIFY, USE, LIST, UNLOCK
Parameters	ADD, DELETE, DISPLAY, MODIFY, UNLOCK
Variable tables	ADD, DELETE, DISPLAY, MODIFY, USE, LIST, UNLOCK
Workload applications	ADD, DELETE, DISPLAY, MODIFY, LIST, UNLOCK

Configuring security domains

Security Domain All objects <pre>job +names = @ schedule +names = @ cpu resource +names = @ prompt +names = @ calendar +names = @ file +names = @ parameter +names = @ report +names = @ runcygrp +names = @ userobj vartable +names = @ action event eventrule +names = @ lob +names = @ wkldappl +names = @</pre>	Security Domain Non-root jobs <pre>job +logon = @ ~logon = wsuser, root</pre>	<p>Combine sets of objects that you can manage together into security roles</p>
Security Domain Work on BRAVO <pre>job +cpus = BRAVO schedule +cpus = BRAVO cpu +cpus = BRAVO resource +cpus = BRAVO parameter +cpus = BRAVO vartable +name = BRAVO@</pre>		<p>Security roles can define atomic sets of objects, or comprehensive assemblies</p>

© Copyright IBM Corporation 2017

Configuring security domains

Security domains define which sets of objects that users or groups can manage. You define which objects belong to the security domain by using filters. Filters are based on attributes of the type of object that is filtered, similarly to [Specifying object attributes on page 7-29](#). For example, you can filter all types of objects by name, and you can additionally filter job streams by workstation. You can use filters to include objects in the security domain, or to exclude them.

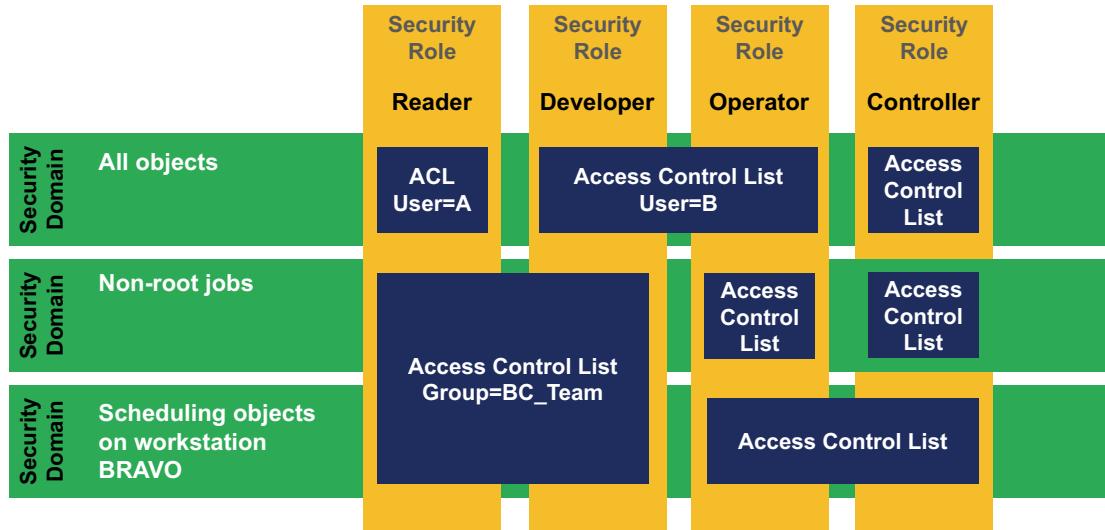
For example, a security domain might encompass all of the objects in the database, or only jobs that do not run as the `root` user. You can choose to create domains with objects based on object name or workstation name (depending on which attributes are available for each type). You can separate different types of objects into separate security domains for more fine-grained control.

Object attributes are used to identify a specific set of object within all objects of the same type. For example, access can be restricted to a set of resource objects that have the same name, the same workstation, or both.

You can *include* or *exclude* each attribute from the selection. The attributes that you can use to declare which objects are referenced by a security domain are shown in table 1 at the following link.

https://www.ibm.com/support/knowledgecenter/SSGSPN_9.4.0/com.ibm.tivoli.itws.doc_9.4/distr/sr_c_tsweb/General_Help/ManagingSecurity_AttributesForObjectTypes_r.htm

Configuring access control lists



© Copyright IBM Corporation 2017

Configuring access control lists

For each **security domain** in the database, you create an **access control list** to define users or groups that access the objects that are represented in the domain. You assign one or more **security roles** to a user or group for each security domain.

For example, in the *All objects* security domain, you might assign the *Controller* role to users who are administrators, and the *Operator* role to the operators group.

Lesson 7 Managing role-based user authorization by using the Dynamic Workload Console

IBM Training



Managing role-based authorization by using the Dynamic Workload Console

© Copyright IBM Corporation 2017

In this lesson, you learn how to create and manage security roles, security domains, and access control lists by using the Dynamic Workload Console.

Managing workload security

Administration > Manage Workload Security



© Copyright IBM Corporation 2017

Managing workload security

To manage workload scheduler security by using the Dynamic Workload Console, click **Administration > Manage Workload Security** from the main menu. Select which Workload Scheduler engine you want to manage from the drop-down list.

In the Manage Workload Security window, you see links to manage access control lists, security domains, and security roles. You can create security objects in any order, remembering that access control lists must reference security domains and security roles that previously exist.

If the Workload Scheduler global option `enRoleBasedSecurityFileCreation` is set to NO, you see the following warning message.



Attention: AWSUI7001W The role-based security is disabled on the selected engine. You can define roles, security domains, and ACL, but the changes are applied only when the security is enabled. For more information on how to enable security, see the Administration Guide.

Managing security roles

To create or manage security roles, click **Create new role**, or **Manage roles** in the **Roles** section. From the manage **Roles** window, you can create, edit and duplicate security roles, and show versions of security roles.

When you create or edit a security role, you can click **Show Details** to select specific levels of access for each type of object you want to control through the roles you create. When details are hidden, you can choose more generic sets of access, such as no access, read-only or full access.

Click **View** to see how the set of permissions that you are assigning compares with the classic security model. Click **Save** to save the role. Click **Save and Exit** to save the role and then exit. Click **Cancel** to discard changes.

Managing security domains

To create or manage security domains, click **Create new Security Domain**, or **Manage Security Domain** in the **Security Domains** section. From the manage **Security Domains** window, you can create, edit and duplicate security domains, and show versions of existing security domains.

When you create or edit a security domain, from the **Type** menu, you can choose to create a **Simple** rule, which provides a filtering rule that applies to all object types. You can instead create a **Complex** rule, in which you specify different filters for each type of object.

You can create as many rules as you need. Rules can include or exclude objects. Each rule contains names, and the names you use can include the asterisk wildcard. Depending on the type of object and its qualifying attributes, the rule might also use workstations, JCL, type, provider, and logon for their filters.

Click **View** to see how the set of filters that you are creating compares with the classic security model. Click **Save** to save the domain. Click **Save and Exit** to save the domain and then exit. Click **Cancel** to discard changes.

Managing access control lists

To create or manage access control lists, click **Give access to user or group**, **View access for user or group**, or **View access for Security Domain** in the **Access Control List** section. From the manage **Access Control Lists** window, you can create, edit and duplicate access control lists, and show versions of existing access control lists.

When you create or edit an access control list, you complete the following steps.

1. Select whether to define access for a user or a group by choosing **User name** or **Group name**. In the related field, type the name of the user or group, or select one from the list.
2. In **Role** field, type the name of an existing security role, or select one or more roles from the menu. You can also create a security role by clicking **Create Role**.
3. In the **Domain** field, select a security domain from the list.
4. Click **View** to see how the set of filters that you are creating compares with the classic security model. Click **Save** to save the domain. Click **Save and Exit** to save the domain and then exit. Click **Cancel** to discard changes.

Lesson 8 Managing role-based user authorization by using the composer command

IBM Training



Managing role-based authorization by using the composer command

© Copyright IBM Corporation 2017

In this lesson, you learn how to create and manage security roles, security domains, and access control lists by using the `composer` command.

Managing workload security objects

Access Control Lists	Security Domains	Security Roles
composer <ul style="list-style-type: none"> • new ACL • modify ACL=<i>name</i> • delete ACL=<i>name</i> <pre>ACCESSCONTROLLIST FOR SECDOM2 USER1 SECROLE1, SECROLE2 USER2 SECROLE2 END</pre>	composer <ul style="list-style-type: none"> • new SD • modify SD=<i>sdname</i> • delete SD=<i>sdname</i> <pre>SECURITYDOMAIN SECDOM2 DESCRIPTION "Sample Security Domain" COMMON CPU=@+NAME=@ USEROBJ CPU=@ JOB CPU=@ SCHEDULE CPU=@+NAME=AP@ RESOURCE CPU=@ PROMPT FILE NAME=@ CPU CPU=@ PARAMETER CPU=@ CALENDAR REPORT NAME=@ EVENTRULE NAME=@ ACTION PROVIDER=@ EVENT PROVIDER=@ VARTABLE NAME=@ WKLDAPPL NAME=@ RUNCYGRP NAME=@ LOB NAME=@</pre>	composer <ul style="list-style-type: none"> • new SR • modify SR=<i>sname</i> • delete SR=<i>sname</i> <pre>SECURITYROLE SECROLE1 DESCRIPTION "Sample Security Role" SCHEDULE ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,DISPLAY,LIMIT,MO RESOURCE ACCESS=ADD,DELETE,DISPLAY,MODIFY,RESOURCE,USE,LIST,UNLOCK PROMPT ACCESS=ADD,DELETE,DISPLAY,MODIFY,REPLY,USE,LIST,UNLOCK FILE ACCESS=BUILD,DELETE,DISPLAY,MODIFY,UNLOCK CPU ACCESS=LIMIT,LINK,MODIFY,SHUTDOWN,START,STOP,UNLINK,LIST,UNLOC PARAMETER ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK CALENDAR ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK REPORT ACCESS=DISPLAY EVENTRULE ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK ACTION ACCESS=DISPLAY,SUBMIT,USE,LIST EVENT ACCESS=USE VARTABLE ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK WKLDAPPL ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK RUNCYGRP ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK LOB ACCESS=USE</pre> <pre>SECURITYROLE SECROLE2 DESCRIPTION "Sample Security Role" SCHEDULE ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,DISPLAY,LIMIT,MO RESOURCE ACCESS=ADD,DELETE,DISPLAY,MODIFY,RESOURCE,USE,LIST,UNLOCK PROMPT ACCESS=ADD,DELETE,DISPLAY,MODIFY,REPLY,USE,LIST,UNLOCK</pre>

© Copyright IBM Corporation 2017

Managing workload security objects

To manage workload scheduler security, run **composer** and use the **new**, **modify**, or **delete** commands to create, edit, or delete objects from the database.

- To manage security roles, use the **SECURITYROLE**, or **SR** object type.
- To manage security domains, use the **SECURITYDOMAIN**, or **SD** object type.
- To manage access control lists, use the **ACCESSCONTROLLIST**, or **ACL** object type.

When you create an object in **composer**, you see a template for the type of object you create. For example, when you use the **new accesscontrollist** **composer** command, a text editor opens with the following sample content. Modify the text to your requirements, and save the file.



Note: Access control list example

```
ACCESSCONTROLLIST FOR SECDOM1
USER1 SECROLE1
USER2 SECROLE2
USER3 SECROLE1, SECROLE2
END

ACCESSCONTROLLIST FOR SECDOM2
USER1 SECROLE1, SECROLE2
USER2 SECROLE2
END
```

Lesson 9 Providing auditing and version controls

Providing auditing and version controls

© Copyright IBM Corporation 2017

Workload Scheduler provides detailed auditing, version control, and release management. In this lesson, you learn how to enable and support these functions.

Tracking changes to scheduling objects

Configure global options for auditing and reporting

- Audit store = both
- DB Audit = 1
- PL Audit = 1
- Audit history = 400

Configure justification and ticketing integration

- Require categories, tickets, descriptions
- Create, change categories
- Integrate existing ticketing system

Configure dynamic workload scheduler auditing

- Edit audit.properties
- Configure event tracking settings

© Copyright IBM Corporation 2017

Tracking changes to scheduling objects

You can track changes to scheduling objects in Workload Scheduler, and create reports to show detailed information about the changes.

You can also create a policy that requires any user who changes objects to provide written justification for the change.

Workload Scheduler can keep an audit trail that consists of detailed information about the user who saved change, the time, and date when the change was completed, the reason why the change was implemented, and the details of the change for every modified item.

Workload Scheduler users can review changes to the database and plan scheduling objects from the Dynamic Workload Console, along with which user performed a specific change, and the time and date when the change was performed.

You can review the history of changes that are made to specific objects, and check previous versions of the object in scheduling language format.

Enabling auditing justification and reporting

Administrators can define a policy that requires each user who changes an object to provide a justification for the change. To define the policies, from the Dynamic Workload Console, perform the following steps:

1. From the navigation toolbar, click **Administration > Auditing Preferences**.
2. You can enable the justification policy for each scheduling engine. After you enable a justification policy, decide which fields (Category, Ticket, and Description) are required.
3. You can change the existing categories, and create categories to add to the default ones in the lower section of the page.
4. Optionally, click the **Ticket required** link to specify the address of the ticketing server and the specific syntax the server supports, for example:

[https://ticket.example.com/tickets/\\${ticketnumber}](https://ticket.example.com/tickets/${ticketnumber}),

Where:

<https://ticket.example.com/tickets/> is the ticketing server address, and \${ticketnumber} is replaced by the ticket number that the user provides.

After you enable a justification policy, the justification pane is displayed each time that a user changes any scheduling object in the Workload Designer. The related changes are stored in the IBM Workload Scheduler audit logs.

Enabling auditing

To enable auditing, you must configure three global options.

auditStore | as

This setting changes the type of store to be used to log database and plan audit records. The following options are available.

- **file:** Specifies that a flat file in the **WAhome/TWS/audit/database** directory is used to store the audit records. You must not set this value if you want to use reporting, version control, or rollback functions.
- **db:** Specifies that the audit records are stored in the Workload Scheduler database.
- **both:** To log audit records in both the file and the database.

The default value is **both**. Any change of this value is effective immediately.

enDbAudit | da

Set the value of this option to **1** (the default) to enable auditing on information about changes to Workload Scheduler database objects. To disable auditing on objects the database, specify **0**. Changes to this parameter are effective immediately.

enPlanAudit | pa

Set the value of this option to **1** to enable auditing on changes to the plan. To disable auditing on changes to the plan, specify **0**.



Attention: When you upgrade the master domain manager from a previous release, the default value for the **auditStore** global option is changed to **both**, **enDbAudit** is changed to **1**, and **enPlanAudit** is changed to **1**. These settings support the auditing feature, which introduces reporting, version control, and rollback functions for database objects.

Other settings

The audit history period setting, **auditHistory (ah)**, is used in audit management. This setting applies only when the **auditStore** option is set to **db** or **both**. Enter the number of days for which you want to save audit record data. Audit records are discarded on a first-in first-out (FIFO) basis. The default value is 180 days. Changes to this option are effective immediately.

Auditing dynamic workload scheduling

When you select the dynamic scheduling capability at installation time, the auditing feature is automatically installed. By default, the auditing feature is disabled. Events that you can audit are listed in [Table 2](#).

Table 2 Auditing settings for dynamic workload scheduling

Setting	Description
JobDefinitionAuditEvent	Maintains a track of operations that are performed on job definitions.
JobLogAuditEvent	Maintains a track of operations that are performed on job logs.
JobAuditEvent	Maintains a track of operations that are performed on jobs.
ResourceAuditEvent	Maintains a track of operations that are performed on resources.
RelationshipAuditEvent	Maintains a track of operations that are performed on relationships between resources.
RecoveryActionAuditEvent	Maintains a track of operations that are performed on recovery actions.
HistoryDataAuditEvent	Maintains a track of operations that are performed on historical data.

To configure the events auditing, enable the auditing feature and change the default values in the configuration file to define event types to be audited. The configuration file is located in **WAhome/TDWB/config/audit.properties**

Configure one or more of the properties in the **audit.properties** file to enable and configure auditing. The properties and configuration settings are listed in the *Workload Scheduler 9.4 Administrators Guide* at the following link:

https://www.ibm.com/support/knowledgecenter/SSGSPN_9.4.0/com.ibm.tivoli.itws.doc_9.4/distr/src_ad/awsadaudit2.htm

Exercises



- Edit and compile the Security.conf file
- Create and edit security roles, security domains, and access control lists
- Change options to enable role-based security
- Test security changes
- Confirm auditing settings
- Enable and manage change justification settings

© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. Where is the authentication for web-based workload console users managed?
2. What two methods can you use to configure authentication on the application servers?
3. To use single sign-on, what two steps must you complete?
4. In what two ways can you manage Workload Scheduler authorization security settings?
5. What command do you use to create a Security.conf file?
6. When you configure the Security.conf file, in which order must the definitions be set?
7. Which global option must you change to enable role-based security configuration?
8. What is the purpose of access control lists in role-based security management?
9. Which three global options must you configure to enable version controls and audit reporting?

Review answers

1. Where is the authentication for web-based workload console users managed?
Before users can access the Dynamic Workload Console, the administrator must configure a user registry, in the JazzSM WebSphere Application Server profile.
2. What two methods can you use to configure authentication on the application servers?
Authentication can be configured by using the administrative console or manually, by using the WebSphere Application Server tools that are supplied with Workload Scheduler.
3. To use single sign-on, what two steps must you complete?
To use single sign-on, both the Dynamic Workload Console and the IBM Workload Scheduler engine must be configured to share an LDAP user registry. You must also configure both instances of WebSphere Application Server to use the same Lightweight Third-Party Authentication (LTPA) token keys and disable their automatic regeneration.
4. In what two ways can you manage Workload Scheduler authorization security settings?
You can edit and compile the Security.conf file, or enable role-based security and edit security roles, security domains, and access control lists.
5. What command do you use to create a Security.conf file?
You use the dumpsec command to create an editable text file from the compiled security file.
6. When you configure the Security.conf file, in which order must the definitions be set?
You must order definitions from most specific to least specific. Workload Scheduler scans the security file from the top down, with each ID tested against each definition in turn.
7. Which global option must you change to enable role-based security configuration?
To enable role-based security in your Workload Scheduler environment, you must set the global option enRoleBasedSecurityFileCreation to yes.
8. What is the purpose of access control lists in role-based security management?
You use access control list to assign users or groups a security role within each security domain.
9. Which three global options must you configure to enable version controls and audit reporting?
To enable auditing, you must configure three global options. The auditStore option must be set to both. The enDbAudit option must be set to 1. The enPlanAudit option must be set to 1.

Summary

You should now be able to perform the following tasks:

- Configure security to protect your environment
- Configure authentication using the WebSphere Administrative Console and provided scripts
- Modify the Security configuration and compile its definitions
- Configure these attributes
 - User definitions
 - Object access
 - Access privileges
- Create security roles, security domains, and access control lists
- Configure tracking, auditing, and justification requirements

© Copyright IBM Corporation 2017

Unit summary

Unit 8 Finding and repairing problems

IBM Training



Finding and repairing problems

© Copyright IBM Corporation 2017
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to troubleshoot problems in the Workload Scheduler environment. You learn how to use the console manager interface, where to find log files, and how to resolve common issues.

References: *IBM Workload Scheduler 9.4 Troubleshooting Guide*

IBM Workload Scheduler 9.4 Administration Guide

Objectives

In this unit, you learn about finding and repairing problems with Workload Scheduler. You learn how to perform the following tasks.

- Describe the built-in troubleshooting features
- Find workstation configuration from the console manager
- Determine current workstation status from the console manager
- Find log files that contain workstation and job failure information
- Find audit logs that contain information about changes to your environment
- Gather information for IBM support to assist with problem determination and resolutions
- Describe event rules that help the administrator

© Copyright IBM Corporation 2017

Unit objectives

Lesson 1 Built-in troubleshooting features

Built-in troubleshooting features

© Copyright IBM Corporation 2017

Workload Scheduler includes features that assist you with troubleshooting. In this unit, you learn how to find problem resolutions and where to find useful messages and configuration.

Available features

- Informational messages
- Error and warning messages
- Message documentation and help
- A logging facility
- Tracing facilities
- In-flight tracing facility
- Engine Log Analyzer
- Auditing facility
- A configuration snapshot Data Capture facility
- First Failure Data Capture (ffdc) configuration snapshot

© Copyright IBM Corporation 2017

Available features

You can use the troubleshooting tools and obtain IBM support for resolution of any issues. You can learn how to use the following tools from the *IBM Workload Scheduler 9.4 Troubleshooting Guide*:

- Informational messages that inform you of expected events.
- Error and warning messages that inform you of unexpected events.
- Message documentation for the most common messages.
- A logging facility that writes all types of messages to log files, which you use to monitor the progress of Workload Scheduler activities. For more information, see “Logs and traces” in the *IBM Workload Scheduler 9.4 Troubleshooting Guide*.
- Various tracing facilities, which record at varying levels of details the Workload Scheduler processes for troubleshooting by IBM Software Support. For more information, see “Logs and traces: the difference” in the *IBM Workload Scheduler 9.4 Troubleshooting Guide*.
- A facility to save a configurable level of log and tracing information in memory. The facility saves all or part of this information to a single, fully integrated file for troubleshooting by IBM Software

Support. For more information, see Chapter 4, “*In-Flight Trace facility for engine*,” in the IBM *Workload Scheduler 9.4 Troubleshooting Guide*.

- A Log Analyzer that you use to read, analyze, and compare log and some trace files. For more information, see “Engine Log Analyzer” in the *IBM Workload Scheduler 9.4 Troubleshooting Guide*.
- An auditing facility that provides an audit trail of changes to the IBM Workload Scheduler database and plan for use in both monitoring and troubleshooting. For more information, see the section about Auditing in the *IBM Workload Scheduler Administration*.
- A configuration snapshot facility that you can use for backup, which provides IBM Software Support with configuration information when unexpected events occur. For more information, see “Capturing product data” in the *IBM Workload Scheduler 9.4 Troubleshooting Guide*.
- A facility that automatically creates a First Failure Data Capture (ffdc) configuration snapshot if the failure of any of the key components that its parent component can detect. For more information, see “First failure data capture (ffdc)” in the *IBM Workload Scheduler 9.4 Troubleshooting Guide*.

Available features

Lesson 2 Checking workstation status

Checking workstation status

© Copyright IBM Corporation 2017

You can use the command-line console manager interface to show workstation information.

Viewing workstation status by using conman

```
Installed for user "wsuser".
Locale LANG set to the following: "en"
Scheduled for (Exp) 06/20/17 (#633) on WS94MDM0. Batchman LIVES. Limit: 11, Fence: 0, Audit Level: 1
%sc
CPUID      RUN NODE    LIMIT FENCE DATE   TIME   STATE          METHOD     DOMAIN
WS94MDM0   633 *UNIX MASTER 11    0 06/20/17 12:29  I J  MDEA    MASTERDM
AGENT0    633 UNIX AGENT 11    0 06/20/17 12:29  LBI J  M      MASTERDM
ALPHA     633 UNIX D-POOL 11    0 06/20/17 12:29  LBI J      MASTERDM
FTA1      633 UNIX FTA   11    0 06/20/17 13:29  I J  MD      MASTERDM
STATIC_POOL 633 UNIX POOL 11    0 06/20/17 12:29  LBI J      MASTERDM
TSMAGENT   633 OTHR X-AGENT 11   0 06/20/17 12:29  LXI JX  tsmxagent  MASTERDM
WS94BROKER 633 OTHR BROKER 11   0 06/20/17 12:29  LTI JW      MASTERDM
%
```

	1	2	3	4	5	6	7	8	9	Description
L										The agent is linked to its domain manager
F										The agent is linked to its domain manager and backup domain managers (enSwFaultTol enabled).
T										The agent is directly linked to the domain manager (by TCPIP connection)
H										The workstation is linked through its hosts
X										The workstation is linked as an extended agent
B										The workstation communicates through the BROKER server.
I										The workstation has completed startup initialization
J										The job manager program is running
W										The writer process is active
M										The monman process is running
D										The workstation is using an up-to-date package monitoring configuration
E										The event processing server is installed and running on the workstation.
e										The event processing server is installed on the workstation but is not running
A										A The WebSphere Application Server started
R										R The WebSphere Application Server is restarting

© Copyright IBM Corporation 2017

Viewing workstation status by using conman

When you enter the console manager, you see a banner that shows critical information. You can repeat this status by entering the **status** command. The status line includes this information:

- The current scheduled production date.
- The current run number. Each time the production plan is extended, the run number is incremented.
- The workstation on which you are running the console manager.
- The status of the current workstation, either **LIVES** (up and running) or **down** (not running).
- The workstation limit (the number of jobs that can be initiated concurrently) on the current workstation
- The workstation job fence (the priority, which any job must exceed to start).
- The audit level (one means plan auditing is enabled, zero means that it is disabled).

Displaying workstation status

The **conman showcpus** command shows information about workstations and links.

The displayed information is updated only while Workload Scheduler (batchman) is running on the workstations. If batchman is up or down is confirmed on screen by the **Batchman LIVES** or **down** message when you run the **conman status** command.

In the listing, the STATE column shows the following information:

- The state of the workstation links and processes. Up to 5 characters are displayed as follows.

The explanation of the characters is divided based on the character scope:

[L|F] [T|H|X|B] [I] [J] [W|H|X] [M] [E|e] [D] [A|R]

Where

-
- L** The workstation is linked. Depending on the type of workstation, the link can be different.
- If the workstation is a fault-tolerant or domain manager, it is linked to its domain manager.
 - If the workstation is a dynamic agent or remote engine, this flag indicates that the workstation is connected to the workload broker server.
 - If the workstation is a pool or dynamic pool, this flag indicates that the workload broker workstation is linked to its domain or upper manager.
-
- F** The workstation is fully linked through primary and all secondary connections. This flag is shown only if the **enSwfaultTol** global option is set to YES. It indicates that the workstation is directly linked to its domain manager and to all its backup domain managers.
-
- T** This flag is displayed if the fault-tolerant agent is directly linked to the domain manager from where you run the command.
-
- H** The workstation is linked through its host.
-
- X** The workstation is linked as an extended agent (x-agent).
-
- B** The workstation communicates through the workload broker server.
-
- I** If the workstation is of type agent MASTER, MANAGER, FTA, S-AGENT, X-AGENT, this flag indicates that the **jobman** program completed its initialization. If the workstation is of type dynamic agent, pool or dynamic pool, this flag indicates that the agent is correctly initialized. If the workstation is of type remote engine, this flag indicates that the communication between the remote engine workstation and the remote engine is correctly initialized.
-
- J** If the workstation is of type agent MASTER, MANAGER, FTA, S-AGENT, X-AGENT, this flag indicates that **jobman** program is running. If the workstation is of type dynamic agent, this flag indicates that **JobManager** is running. Because dynamic pool workstations are not monitored, for this workstation type the J character is always shown. If the workstation is of type pool, this flag indicates that the **JobManager** process is running on at least one agent that is registered to the pool. If the workstation is of type remote engine, this flag indicates that the ping command to the remote engine is successful.
-
- W** The writer process is active on the workstation.
-

If the workstation that runs **conman** is the extended agent host, the state of the extended agent is **LXI JX**. If the workstation that runs **conman** is not the extended agent host, the state of the extended agent is **LHI JH**.

- The state of the monitoring agent. Up to 3 characters are displayed as follows:

[M] [E|e] [D]

where:

-
- M** The **monman** process is running. This flag is displayed for all the workstations in the network when the event-driven workload automation feature is enabled. That means that the global option **enEventDrivenWorkloadAutomation** is set to **yes**. The flag is not displayed for those workstations where **monman** was manually stopped from conman or the Dynamic Workload Console.
-
- E** The event processing server is installed and running on the workstation.
-
- e** The event processing server is installed on the workstation but is not running.
-
- D** The workstation is using an up-to-date package monitoring configuration. This flag is displayed for the workstations on which the last package of event rules was deployed. It was deployed either manually with the **planman deploy** command or automatically with the frequency that is specified by the **deploymentFrequency** global option.
-

- The state of the WebSphere Application Server. A one-character flag is displayed, if the application server is installed:

[A|R]

where:

A The WebSphere Application Server was started.

R The WebSphere Application Server is restarting.

The flag is blank if the application server is down or if it was not installed.

Checking the health of fault-tolerant agents

The **conman checkhealthstatus (chs)** command checks and shows the health of fault-tolerant agent workstations. When you run this command for a particular workstation, you see this information:

- If the domain manager can connect to the agent
- If the Symphony file on the agent is not corrupted
- If the mailbox files can be successfully read by the mailman process without error
- That the mailbox is not full

The **checkhealthstatus** command is useful to help you diagnose the reason for an unlinked workstation and to get suggestions about how to recover the problem.

Alternative workstation views

conman showcpus					
:info	CPUID	VERSION	TIME ZONE		INFO
	WS94MDMO	9.4.0.01	America/Chicago		Linux 3.10.0-514.21.1 #1 SM
	AGENT2	9.4.0.01	America/Los_Angeles		Red Hat Enterprise Linux Ser
	BRAVO	9.4.0.01	America/Chicago		Dynamic Pool
	FTA1	9.4.0.01	Europe/Rome		Linux 3.10.0-514.21.1 #1 SM
	STATIC_POOL	9.4.0.01	America/Chicago		Pool
	WS94BROKER	9.4.0.01	America/Chicago		Dynamic Workload Broker
:link	CPUID	HOST	FLAGS	ADDR	NODE
	WS94MDMO	WS94MDMO	AF T	31111	ws94mdm0
	AGENT0	WS94BROKER		31114	ws94mdm0
	CHARLIE	WS94BROKER		31111	
	FTA1	FTA1	A T	31111	ws94agent1
	WS94BROKER	WS93MDMO	A T	41114	localhost
:getmon	Monitoring configuration for FTA1: ***** *** Package date : 2017/06/20 20:05 GMT *** *****				
	FILE_INCOMING_FTA1::FileMonitor#FileCreated:/var/tmp/incoming/*.* on FTA1; J5MAXDUR::TWSObjectsMonitor#JobMaxDurationExceeded:bravo # SE* . *; J5MINDUR::TWSObjectsMonitor#JobMinDurationNotReached:BRAVO # SE* . *;				

© Copyright IBM Corporation 2017

Alternative workstation views

You can see other views of workstation information in the console manager **showcpus** command. These views are **info**, **link**, and **getmon**.

Info

The **Info** format of the **showcpus** command lists the version of Workload Scheduler, the time zone, and the operating system version for each workstation in the plan.

Link

The **Link** format of the **showcpus** command lists the scheduling host, node name and TCP address, and service flags for each workstation.

The FLAG column shows workstation settings set in the workstation definition. Up to 5 characters are displayed as follows:

[A] [B] [F] [s] [T]

- A** *Autolink* is turned on in the workstation definition.
- B** This flag is used only in end-to-end environment and it indicates whether the *deactivate job starting* flag is enabled.
- F** *Full Status* mode is turned on in the workstation definition.
- s** The ID of mailman server for the workstation (A-Z or 0-9, or blank).
- T** The link is defined as TCP/IP.

Getmon

The **Getmon** format shows all of the event rule monitors deployed to the workstations.

Using the Link format to troubleshoot networking problems

You can use the output of the command `showcpus;link` command to help you find where networking or configuration problems exist within your Workload Scheduler network. The node name and TCP address that is listed must match actual computers on the network. You can use the `ncat` command to detect whether the `netman` process is listening on the node and port:

```
ncat hostname portnum
```

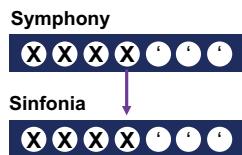
where *hostname* is the name that is listed in the NODE column and *portnum* is the port that is listed in the ADDR column.

If the `ncat` command does not connect, then the node is not available on the network. You can then either resolve the issue outside of Workload Scheduler, or change the Workload Scheduler configuration to match the environment.

Resetting a fault-tolerant agent

RESETFTA FTA1

- 3** Generate a new **Sinfonia** from the existing **Symphony** on the domain manager



- 4** Initialize the agent by linking to it, sending the Sinfonia

- 1** Stop the agent processes

2 Archive into **tmp**

- Appserverbox.msg
- clbox.msg
- Courier.msg
- Intercom.msg
- Mailbox.msg
- Monbox.msg
- Moncmd.msg
- Symphony
- Sinfonia

- 5** Start the agent processes with the new Symphony

On the domain manager

On the agent

© Copyright IBM Corporation 2017

Resetting a fault-tolerant agent

If the production plan on a fault-tolerant agent, which is stored in the **Symphony** file, becomes corrupted, you can reset the agent. Messages in the console log show the corruption. Before you reset a fault-tolerant agent, prepare the system:

- Ensure that enough space exists on the file system to hold the new Symphony and any log files that are produced, and enough memory available for the agent to run.
- Check to see that the **netman** process is running so that the agent's domain manager can communicate with the agent.



Hint: A full file system is the most common cause of Symphony file corruption. Create an event rule for each workstation to monitor the **Workload Scheduler file system filling** event.

Lesson 2 Checking workstation status

During the procedure, the following files are moved to the *TWA_home/TWS/tmp* directory:

- Appserverbox.msg
- clbox.msg
- Courier.msg
- Intercom.msg
- Mailbox.msg
- Monbox.msg
- Moncmd.msg
- Symphony
- Sinfonia

If one of the files is in use by a Workload Scheduler agent process, the procedure is canceled.

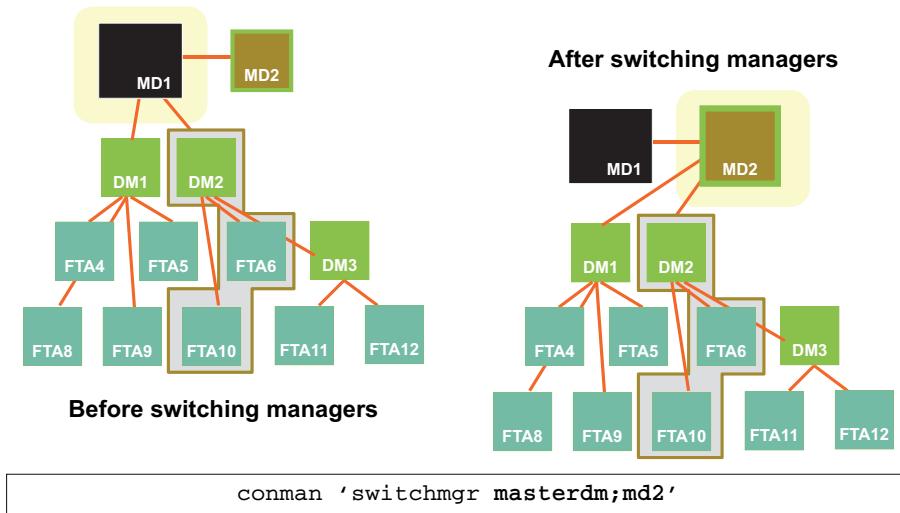
Lesson 3 Managing domains

Managing domains

© Copyright IBM Corporation 2017

In this lesson, you learn how to manage Workload Scheduler domains, and how to switch domain managers.

Switching domain managers



© Copyright IBM Corporation 2017

Switching domain managers

You might switch a domain manager because you want it to run on a different workstation.

Alternatively, it might be the result of network link problems, the failure of the domain manager, or the dynamic domain manager workstation itself. If the domain manager to be changed is a master domain manager, you must run some specific extra steps.

Running without a domain manager has the following effects:

- Agents and subordinate domain managers cannot resolve interworkstation dependencies because activity records that are broadcast by the master domain manager are not being received.
- The upward flow of events is interrupted. This interruption affects events that report the status of jobs, job streams, and dependencies. These items are defined on workstations in the network hierarchy under the failed domain manager.
- Standard agents might be hosted by the failed domain manager. If so, they cannot process work because they depend on the domain manager for all scheduling and job launching.

If the problem is expected to be of short duration, you can wait for the problem to be resolved and Workload Scheduler to recover on its own. For more information, see the section about network link problems in the *IBM Workload Scheduler Troubleshooting Guide*. If you are uncertain about the duration, or if you want to restore normal agent operation, you must switch to a backup.

Network security is enforced by using IP address validation. Workstation linking (**autalink** option or **link** command) might fail if an agent has an old Symphony file that does not contain the new domain manager. If a connection fails, remove the old Symphony file on the agent and try the connection again.

Using the **conman switchmgr** command

The **conman switchmgr** command switches domain management from the current domain manager to a backup domain manager. The **switchmgr** command must be used only as part of specific procedures for switching domain management capabilities from a domain manager to its backup domain manager, either permanently or temporarily. For more information about these procedures, see the *IBM Workload Scheduler 9.4 Administration Guide*.

```
{switchmgr | switchm} domain;newmgr
```

The **domain** option specifies the domain of which you want to switch managers.

The **newmgr** option specifies the name of the new domain manager. This name must be a workstation in the same domain. It must also be defined beforehand as a fault-tolerant agent with the **Full Status** attribute enabled.

The command stops a specified workstation and restarts it as the domain manager. All domain member workstations are informed of the switch, and the old domain manager is converted to a fault-tolerant agent in the domain.

Unless you change the workstation domain definitions in the Workload Scheduler database, the next time **JnextPlan** runs, the old domain manager automatically resumes domain management responsibilities.

Fault-tolerant agents that are defined with **securitylevel = on** might fail to use the SSL port to connect to the new master domain manager after the **switchmgr** command is run. In this case, perform either of the following tasks so that the agent starts correctly:

- Unlink and then link the agent from the new master domain manager.
- Use the **securitylevel = force** option on the agent.

Switching the event processor

The **conman switcheventproc** command switches the event processing server from the master domain manager to the backup master or vice versa.

You can run the event processing server also on a workstation that is installed as a backup master that runs as a plain fault-tolerant agent.

```
{switcheventprocessor | switchevtp} workstation
```

The **workstation** option specifies the name of the master domain manager or of the backup master where you want to switch the event processing server. Wildcard characters are not permitted.

If you enter the command from a workstation other than the one where the event processor is configured, the command uses the command-line client. The user credentials for the command-line client must be set correctly. For backup masters, the workstation must have the full-status attribute set to on.

The correlation state of pending correlation rule instances is lost whenever the server is stopped or migrated. If you enable caching of received events in the configuration file of the EIF listener, the cached events are lost after the event processor is switched. Backup masters workstation definitions must have the full-status attribute set to on.



Important: Before you run this command, run **planman deploy** as a precaution. You run this command to ensure that your last changes or additions to active event rules are deployed before the event processor is switched. Deploying the rules avoids the risk that because of a time mismatch, the old event processor receives the last updates instead of the new one. These updates are sent automatically based on the setup of the **deploymentFrequency** global option.

The master and backup masters that are designated to run the event processor must always have their clocks synchronized to avoid inconsistencies in the calculation of the time interval of running event rules. In fact, if the event processor is switched to a not-synchronized computer, timeout actions that start are triggered might undergo unexpected delays. Use a Network Time Protocol (NTP) server to keep all clocks synchronized.

Lesson 4 Gathering troubleshooting data

Gathering troubleshooting data

© Copyright IBM Corporation 2017

In this lesson, you learn how to gather the data that you need to find resolution to Workload Scheduler problems.

Finding log files

- **Job logs**
 - *WAHome/TWS/stdlist/yyyy.mm.dd/O_{jjjj}.hhmm* (*jjjj* = job number, *hh* = hour, *mm* = min)
- **Agent process** (batchman, mailman, and others) logs
 - *WAHome/TWS/stdlist/logs/yyyymmdd_NETMAN.log*
 - *WAHome/TWS/stdlist/logs/yyyymmdd_TWSMERGE.log*
- **JobManager** (dynamic agent) logs
 - *WAHome/TWS/stdlist/logs/JM*
 - *WAHome/TWS/stdlist/logs/JM/yyyy.mm.dd/archive*
- **Audit logs**
 - *WAHome/TWS/audit/database/yyyymmdd*
 - *WAHome/TWS/audit/plan/yyyymmdd*
- **WebSphere logs**
 - */opt/IBM/TWA/WAS/profiles/TWSProfile/logs/server1*
 - */opt/IBM/JazzSM/profile/logs/server1*

- These log files (except the WebSphere log files) are on the agent computers
- These log files are not removed automatically
- You must schedule the **rmstdlist** utility or operating system function

© Copyright IBM Corporation 2017

Finding log files

Workload Scheduler produces several log files that are useful for monitoring for errors or for diagnosing problems that occur. In this lesson, you learn how to find log files to assist you to find causes of problems.

The following table shows the location of important log files.

Log file type	Locations
Job logs	<i>WAHome/TWS/stdlist/yyyy.mm.dd/O_{jjjj}.hhmm</i> (<i>jjjj</i> = job number, <i>hh</i> = hour, <i>mm</i> = min)
Agent process (batchman, mailman, and so on) logs	<i>WAHome/TWS/stdlist/logs/yyyymmdd_NETMAN.log</i> <i>WAHome/TWS/stdlist/logs/yyyymmdd_TWSMERGE.log</i>
JobManager (dynamic agent) logs	<i>WAHome/TWS/stdlist/logs/JM</i> <i>WAHome/TWS/stdlist/logs/JM/yyyy.mm.dd/archive</i>
Audit logs	<i>WAHome/TWS/audit/database/yyyymmdd</i> <i>WAHome/TWS/audit/plan/yyyymmdd</i>
WebSphere Application Server logs	<i>WAHome/WAS/TWSProfile/logs/server1/SystemErr.log</i> <i>WAHome/WAS/TWSProfile/logs/server1/SystemOut.log</i> <i>WAHome/WAS/TWSProfile/logs/server1/startServer.log</i>

Workload Scheduler does not delete the job logs, agent process logs, JobManager logs, or audit logs. The **rmstdlist** utility is provided for you to automate the removal of these log files. You create a job in Workload Scheduler databases and schedule it to run whenever you want to remove old log

files. To use the rmstdlist command, supply the oldest age, in days, of files to keep. For example, to remove log files older than 14 days, run (or schedule to run) the following command:

```
rmstdlist 14
```

Gathering data for IBM support

```
tws_inst_pull_info.sh -twsuser <userid> -log_dir_base <path>
[-run_db2_module <y/n>][-extract_db_defs <y/n>][-date <yyyymmdd>]
```

```
operation successful.
*****Gathering trace information.*****
*****Gathering Websphere specific information. This may take a few minutes.*****
*****Gathering DB2 specific information. This may take a few minutes.*****
*****Gathering TWS Instance new/existing install root owned information.*****

The /var/tmp/tws_info/TWS_20110826_062103 directory must be removed by the root user
since it contains copies of files owned by root.

Send compressed tarfile /var/tmp/tws_info/TWS_20110826_062103.tar.gz from
/var/tmp/tws_info directory to L2 support see below:

To: tivoli_support@mainz.ibm.com
Subject: pmr#,branch#,countrycode# -nosec

Example:
To: tivoli_support@mainz.ibm.com
Subject: 12345,123,000 -nosec

To view a summary of messages and actions refer to
/var/tmp/tws_info/TWS_20110826_062103/datagather_summary.log
```

© Copyright IBM Corporation 2017

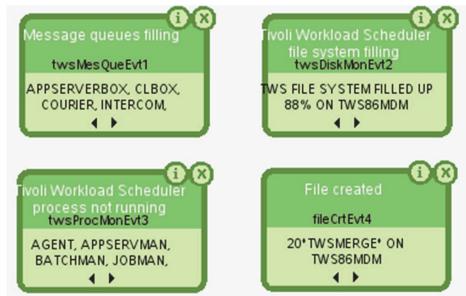
Gathering data for IBM support

IBM provides a utility to gather relevant data and configuration information about your Workload Scheduler installation. You can package them in a file and send them to IBM support. To perform the data gathering, run the following command:

```
tws_inst_pull_info.sh -twsuser userid -log_dir_base path
```

Where, *userid* is the Workload Scheduler installation user, and *path* is a temporary directory where you want to store the files that are created by the utility.

Event rules for systems administrators



Use **event rules** for these purposes:

- Monitoring capacity of the message queues
- Monitoring the utilization of the file system
- Monitoring the presence of the critical processes
- Detecting the changeover of the console and audit logs

© Copyright IBM Corporation 2017

Event rules for systems administrators

Workload Scheduler has several useful event monitors built in. You can create event rules that notify staff when these situations occur:

- A message queue becomes too full
- The file system on which Workload Scheduler is installed starts to fill
- One of the critical processes fails
- A new audit or console log is created

The event rule actions that you create can perform these actions:

- Send an electronic message
- Forward an event to IBM Monitoring
- Run a shell script, passing information from the event

Exercises

- Running conman commands and retrieving workstation status
- Using the console log to find important messages
- Finding Workload Scheduler log files



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this lesson.

Review questions

1. Which command do you use to see the status of the workstations?
2. Which command do you use to reset a fault-tolerant agent?
3. Which command do you use to gather data for IBM customer support?

Review answers

1. Which conman command do you use to see the status of the workstations?

The conman showcpus command shows information about workstations and links.

2. Which command do you use to reset a fault-tolerant agent?

Use the conman resetfta command to reset a fault-tolerant agent.

3. Which command do you use to gather data for IBM customer support?

To perform the data gathering, run the following command: tws_inst_pull_info.sh -twsuser userid -log_dir_base path

Summary

You should now be able to perform the following tasks:

- Describe the built-in troubleshooting features
- Find workstation configuration from the console manager
- Determine current workstation status from the console manager
- Find log files that contain workstation and job failure information
- Find audit logs that contain information about changes to your environment
- Gather information for IBM support to assist with problem determination and resolutions
- Describe event rules that help the administrator

© Copyright IBM Corporation 2017

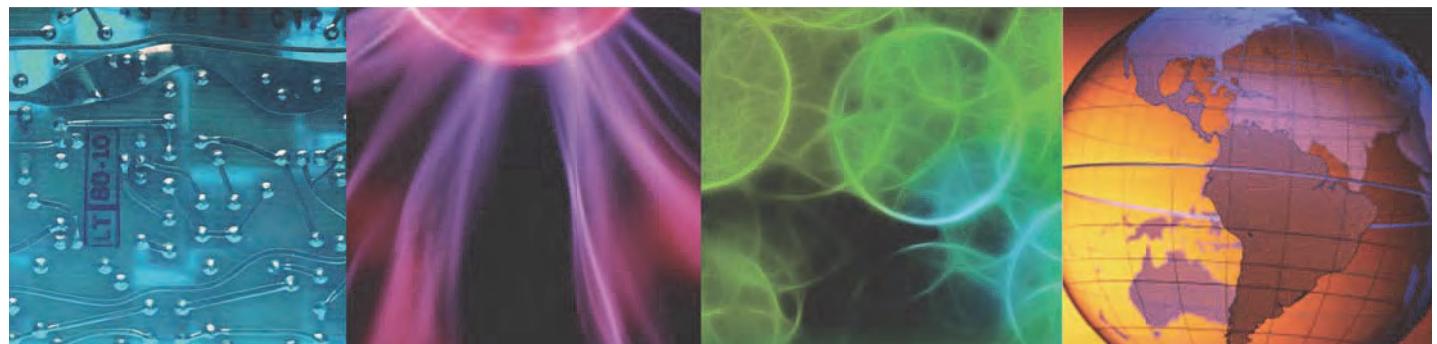
Unit summary



IBM Training



© Copyright IBM Corporation 2017 All Rights Reserved.



ibm.com/training

Authorized
IBM | Training