

Course Guide

IBM Workload Scheduler 9.4 Operations and Scheduling

Course code TX319 ERC 1.0



October 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 10504-1795
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
About the student	xi
Agenda	xii
Course description	xiii
Unit 1 IBM Workload Scheduler introduction	1-1
Unit objectives	1-2
Lesson 1 About Workload Automation	1-3
Workload Automation concepts	1-4
Lesson 2 IBM Workload Automation offerings	1-5
IBM Workload Automation components	1-6
IBM Workload Scheduler for z/OS, the scheduler in z/OS environments	1-6
IBM Workload Automation on Cloud	1-6
IBM Workload Scheduler on Bluemix	1-7
IBM Workload Scheduler V9.4	1-7
Lesson 3 Workload Scheduler terminology	1-9
Workload Scheduler terminology	1-10
Workload Scheduler topologies	1-13
Types of Workload Scheduler agents and workstations	1-15
Dynamic Workload Broker and dynamic scheduling	1-18
The Workload Scheduler production process	1-19
Lesson 4 Workload Scheduler user interfaces	1-21
Workload Scheduler graphical interfaces	1-22
Workload Scheduler command-line interfaces	1-23
Application programming interfaces	1-25
Lesson 5 Opening the Workload Console	1-27
Opening the Dynamic Workload Console	1-28
Lesson 6 Getting started with Application Lab	1-30
The Application Lab interface	1-31
Starting the Application Lab interface	1-32
Opening the Application Lab interface from the start page	1-32
Following the Application Lab tutorial	1-34
Lesson 7 Getting started with the Dynamic Workload Console	1-35
Opening the Dynamic Workload Console	1-37
Using the Welcome page	1-38
Using menus and tabs	1-39
Using the tour and Quick start	1-41
Getting context-sensitive help	1-43
Lesson 8 Connecting the console to Workload Scheduler	1-44
The engine connection	1-45

Creating and managing engine connections	1-47
Lesson 9 Getting started with the command-line interfaces	1-49
The conman and composer programs	1-50
Preparing to use the command-line interfaces	1-51
Using the command-line interfaces	1-54
Getting help with conman or composer	1-55
Instructor demonstration	1-57
Exercises	1-58
Review questions	1-59
Review answers	1-60
Unit summary	1-61
Unit 2 Using the Application Lab interface	2-1
Objectives	2-2
Lesson 1 Using the processes status overview page	2-1
Processes status overview page components	2-2
Lesson 2 Managing processes	2-4
Using the processes library	2-5
Creating a process	2-6
Running and monitoring processes	2-12
Managing and monitoring process steps	2-13
Lesson 3 Using other Application Lab functions	2-15
Exporting and importing process libraries	2-16
Importing crontab and Windows task scheduler definitions	2-17
Publishing a process to the Self-Service Catalog	2-19
Managing a process by using the Dynamic Workload Console	2-20
Enhancing a process	2-21
Exercises	2-23
Review questions	2-24
Review answers	2-25
Unit summary	2-26
Unit 3 Monitoring Workload Scheduler production	3-1
Objectives	3-2
Lesson 1 Monitoring Workload Scheduler production by using queries	3-3
Monitor workload	3-4
Creating and customizing all types of monitoring tasks	3-7
Using monitoring tasks	3-10
Lesson 2 Using the Workload Dashboard	3-14
Workload Dashboard overview	3-15
Lesson 3 Monitoring system status and health	3-18
Monitoring your scheduling environment	3-19
Viewing Dynamic Workload Broker computers	3-21
Using the SHOWCPUS command	3-22
Lesson 4 Monitoring plan objects	3-24
Controlling processing by using dependencies	3-25
Using internal and external dependencies	3-26
Evaluating time restrictions	3-27
Using other dependencies	3-29

Monitoring scheduling objects with conman	3-33
Monitoring job streams	3-35
Job stream status	3-36
Monitoring job streams with conman	3-40
Monitoring jobs	3-44
Job status	3-47
Monitoring jobs with conman	3-49
Using workload service assurance (critical path)	3-52
Monitoring critical jobs	3-54
Lesson 5 Using the What-if Analysis interface	3-55
What-if Analysis interface overview	3-56
Using the Gantt view	3-60
Lesson 6 Monitoring the plan graphically	3-61
Toolbar elements	3-62
Using the Plan View	3-66
Using the Job Stream View	3-67
Instructor demonstration	3-69
Exercises	3-70
Review questions	3-71
Review answers	3-72
Unit summary	3-73
Unit 4 Managing Workload Scheduler production	4-1
Objectives	4-2
Lesson 1 Managing the scheduling environment	4-3
Managing your scheduling environment	4-4
Lesson 2 Performing workstation-related activities with conman	4-6
Managing Dynamic Workload Broker computers	4-8
Lesson 3 Managing scheduling objects in the plan	4-9
Adding or changing time dependencies	4-10
Setting job and job stream priority and limit	4-12
Changing dependencies	4-15
Stopping a job	4-19
Rerunning jobs	4-20
Rerunning jobs with successors	4-22
Confirming jobs	4-23
Canceling jobs and job streams	4-24
Lesson 4 Performing job-related activities with conman	4-26
Submitting new work	4-27
Submitting undefined jobs	4-30
Lesson 5 Submitting new work with the command line	4-32
Instructor demonstration	4-33
Exercises	4-34
Review questions	4-35
Review answers	4-36
Unit summary	4-37

Unit 5 Creating scheduling objects	5-1
Objectives5-2
Lesson 1 Using the Workload Designer5-3
Starting the Workload Designer5-4
Using the Workload Designer5-6
Comparing and restoring previous versions of objects5-8
Lesson 2 Creating scheduling objects5-9
Creating Windows users5-10
Lesson 3 Creating user definitions by using the composer command5-12
Lesson 4 Creating variables and variable tables5-14
Lesson 5 Creating variables and variable tables by using composer5-17
Lesson 6 Creating prompts and resources5-19
Creating resources5-20
Lesson 7 Creating prompts and resources by using composer5-21
Lesson 8 Creating calendars5-23
Lesson 9 Creating calendars by using command-line interfaces5-25
Lesson 10 Creating and using run cycles5-28
Inclusive and exclusive run cycles5-29
Free days and workdays5-30
Run cycle rule types5-31
Creating run cycle groups5-33
Lesson 11 Creating run cycle groups by using composer5-36
Lesson 12 Defining jobs5-38
Creating job definitions5-39
Job types that are available in Workload Scheduler5-41
Job recovery options5-45
Using automation utility jobs5-57
Lesson 13 Creating job definitions by using composer5-59
Instructor demonstration5-63
Exercises5-64
Review questions5-65
Review answers5-66
Unit summary5-67
Unit 6 Scheduling, forecasting, and migrating workloads	6-1
Objectives6-2
Lesson 1 Creating job streams6-3
Planning job streams6-4
Building the job stream6-6
Lesson 2 Applying jobs to job streams6-13
Applying jobs to a job stream6-14
Using no operation jobs6-18
Using conditional dependencies6-19
Analyzing job durations6-21
Lesson 3 Setting time and date restrictions6-24
Selecting run dates6-25
Types of time restrictions6-27
Following the time restriction hierarchy6-31

Repeating work	6-33
Using parallel instance actions	6-35
Lesson 4 Using advanced scheduling options	6-36
Using resolution criteria	6-37
Creating future versions of a job stream	6-38
Using workstation classes	6-40
Lesson 5 Creating job stream definitions by using composer	6-41
Using composer to create job stream definitions	6-42
Lesson 6 Creating and using event rules	6-45
Event rule basics	6-46
Defining the event rule object	6-47
Passing data from the event to the action	6-50
Troubleshooting event rules	6-52
Creating and modifying event rules by using composer	6-54
Lesson 7 Forecasting and previewing future dates	6-59
Alternative plan types	6-60
Using the preproduction plan	6-65
Lesson 8 Exporting and importing scheduling definitions	6-67
Using workload application templates	6-68
Considerations for using workload application templates	6-70
Creating and managing workload application templates	6-72
Lesson 9 Using workload application templates from the command line	6-76
Importing crontab and Windows task scheduler definitions	6-78
Instructor demonstration	6-80
Exercises	6-81
Review questions	6-82
Review answers	6-83
Unit summary	6-85
 Unit 7 Reporting with Workload Scheduler.....	 7-1
Objectives	7-2
Lesson 1 Using reporting features	7-3
Creating reports by using the Dynamic Workload Console	7-4
Creating a report task	7-7
Lesson 2 Using Common Reporting with Workload Scheduler	7-8
Types of Workload Scheduler reports	7-9
Starting the Common Reporting interface	7-10
Customizing the report by filtering data	7-11
Lesson 3 Using the batch reporting utility	7-12
Editing the properties file	7-13
Running the report command line	7-15
Lesson 4 Using the preformatted text-based report utilities	7-17
Running report commands	7-18
Running extract commands	7-20
Instructor demonstration	7-22
Exercises	7-23
Review questions	7-24

Review answers	7-25
Unit summary	7-26
Appendix A Reference material	A-1
Monitor workstation task columns	A-1
Workstation state column flags	A-3
Monitor job streams task columns	A-6
Console manager job selection keywords	A-8
File dependency qualifiers	A-9

About this course



IBM Training

IBM

IBM Workload Scheduler Operations and Scheduling

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

This 3-day instructor-led course is suitable for IBM Workload Scheduler version 9.4 users. This course introduces IBM Workload Scheduler features, environment, and terminology. You learn about the features in IBM Workload Scheduler for distributed environments and how to use both the Dynamic Workload Console and command-line interfaces to manage Workload Scheduler. You also learn how to monitor production workflow, and to plan, implement, and manage Workload Scheduler objects to create a production day plan. Production workflow consists of job and job stream instances, which are designed by using plan objects such as job definitions, prompts, dependencies, and recovery options. You manage changes to objects, the plan, jobs, events, and job streams. You also learn how to optimize production workflow and troubleshoot plan problems. In addition to classic batch scheduling, you learn to use Workload Scheduler to manage dynamic and event-driven workloads. Training occurs in a classroom environment (or online in an instructor-led format) with hands-on labs to reinforce concepts through production scenarios.

The lab environment for this course uses the Red Hat Enterprise Linux 7 operating system.

For more information about other related courses, see the IBM education training paths website:

www.ibm.com/training

Details	
Delivery method	Classroom or instructor-led online (ILO)
Course level	ERC 3.0 This course is an update of the TX317: IBM Workload Scheduler 9.3 Operations and Scheduling ERC 3.0
Product and version	IBM Workload Scheduler 9.4 with Fix Pack 1
Duration	3 days
Skill level	Basic / Intermediate

Course objectives

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

- Explain the production terminology that is used with IBM Workload Scheduler and its production planning process
- Monitor production batch workload processes
- Manage production batch workload
- Create and modify scheduling objects
- Schedule new workloads
- Forecast future workloads
- Migrate workload definitions from test to production environments
- Create and maintain reports about production workloads

© Copyright IBM Corporation 2017

Course objectives

About the student

Before taking this course, make sure that you can use an internet browser such as Mozilla Firefox and perform basic shell commands in a terminal emulator such as PuTTY or xterm.

This course is designed for operators and schedulers who perform duties that are related to batch workload automation. This basic course is for the following audiences:

- New IBM Workload Scheduler users who are seeking proficiency in IBM Workload Scheduler version 9.4 job scheduling and operations
- Existing operators, schedulers, supervisors, and managers who are converting from previous versions to version 9.4
- Anyone needing guidance in managing plans, monitoring workloads, or submitting jobs and job streams
- Anyone needing a refresher course and introduction to the Dynamic Workload Console graphical user interface

Agenda

- IBM Workload Scheduler Introduction
- Using the Application Lab interface
- Monitoring Workload Scheduler production
- Managing Workload Scheduler production
- Creating scheduling objects
- Scheduling, forecasting, and migrating workloads
- Reporting with Workload Scheduler

© Copyright IBM Corporation 2017

Agenda

Course description

The course contains the following units:

1. [IBM Workload Scheduler introduction](#)

In this unit, you learn about the basic needs and functions of workload automation.

In the exercises for this unit, you review the enhancements that are available in Workload Scheduler V9.4, and prepare your environment for the exercises in the next unit. You start the Dynamic Workload Console and use the `conman` and `composer` commands.

2. [Using the Application Lab interface](#)

In this unit, you learn how to use the Application Lab interface to create simple work flows in Workload Scheduler.

In these exercises, you create applications, jobs, and job streams in IBM Workload Scheduler by using the Application Lab interface.

3. [Monitoring Workload Scheduler production](#)

In this unit, you learn how to use the Dynamic Workload Console and the `conman` command to monitor your system and the current production day plan.

In the exercises for this unit, you create and customize tasks in the Dynamic Workload Console. You monitor production plan objects by using Dynamic Workload Console and command-line interfaces.

4. [Managing Workload Scheduler production](#)

In this unit, you learn about managing Workload Scheduler production that uses the Dynamic Workload Console and command-line interfaces. You learn how to perform various tasks that relate to workstations, plans, and jobs.

In the exercises for this unit, you manage objects in the Workload Scheduler production plan. You change objects in the plan by using Dynamic Workload Console and command-line interfaces.

5. [Creating scheduling objects](#)

In this unit, you learn how to define Workload Scheduler objects, such as calendars and resources, and job definitions.

In the exercises for this unit, you create scheduling objects. In the next two units, you document and create a workflow. You are responsible for implementing the workload for one department. In your role as scheduler, you take the job stream requirements from the department. You then create the scheduling objects in Workload Scheduler to implement the workflow.

6. [Scheduling, forecasting, and migrating workloads](#)

In this unit, you learn how to define Workload Scheduler job streams and event rules. You learn how to create various definitions and rules. You also learn how to forecast future work and migrate workload definitions by using workload application templates.

In the exercises for this unit, you create job streams and event rules. You also migrate workload definitions by using workload application templates.

7. [Reporting with Workload Scheduler](#)

In this unit, you learn how to create and run Workload Scheduler reports.

In the exercises for this unit, you create and run reports.

Unit 1 IBM Workload Scheduler introduction

IBM Training



IBM Workload Scheduler introduction

© 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 needs and functions of workload automation.

Unit objectives

In this unit, you learn about the basic needs and functions of Workload Automation. You learn how to perform the following tasks:

- Describe the basic functions of Workload Automation
- Describe the major areas of interest in Workload Scheduler for operators and schedulers
- Explain the production terminology used with workload automation and specifically Workload Scheduler
- Explain the concepts of the Workload Scheduler production planning process
- Start the Application Lab interface
- Start the Dynamic Workload Console
- Start the command line interface

© Copyright IBM Corporation 2017

Unit objectives

Lesson 1 About Workload Automation

About Workload Automation

© Copyright IBM Corporation 2017

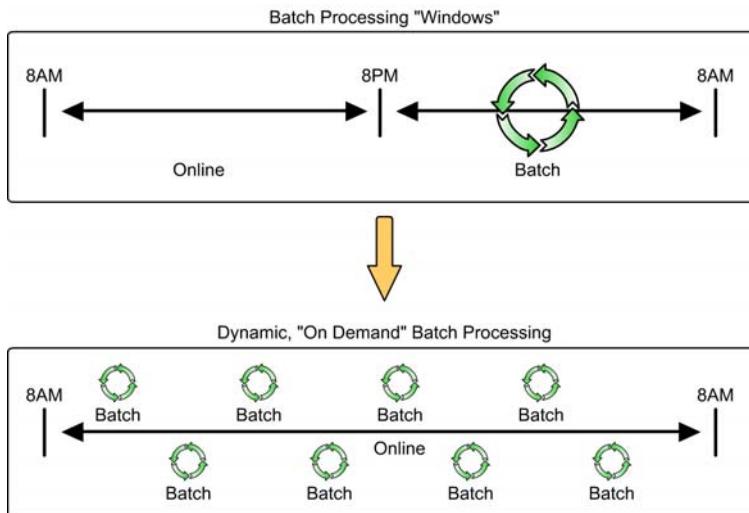
In this lesson, you learn about the basic needs and functions of workload automation. You learn how to describe these items:

- Basic functions of workload automation
- Major areas of interest in Workload Scheduler, for operators and schedulers

Job schedulers control unattended background program execution, commonly known as batch processing. Job schedulers typically provide graphical user interfaces and a single point of control for definition and monitoring of background executions in computer networks.

Increasingly, job schedulers orchestrate the integration of real-time business activities with traditional background IT processing across different operating systems and business application environments. Workload automation is an emerging trend that represents the evolution of traditional batch job schedulers. Tools providing workload automation react to changes in applications and IT infrastructures and the dynamic demands of IT.

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

Job schedulers automate workloads that include these characteristics:

- The work is repeated regularly.
- The work is required to run at a specific time, in a specific order, or both.
- The work happens across several computers in the environment.
- The work is critical to data processing or Systems Management.

By using Workload Scheduler to automate these workloads, you help to ensure these benefits:

- The work is consistently defined and ready to occur as planned.
- Human intervention is reduced or eliminated.
- The work is optimized to occur as quickly as possible while still meeting all dependency requirements.
- Error monitoring is centralized even though the work runs remotely, and you can use a single graphical interface to handle error processing.

Lesson 2 IBM Workload Automation offerings

IBM Workload Automation offerings

© Copyright IBM Corporation 2017

The IBM Workload Automation suite of products is used to simplify systems management across heterogeneous environments by integrating systems management functions. The main components in the IBM portfolio are described in the following section.

IBM Workload Automation components

These are the components that are offered in the IBM Workload Automation suite

- IBM Workload Scheduler for z/OS
- IBM Workload Automation on Cloud
- IBM Workload Scheduler on Bluemix
- IBM Workload Scheduler

© Copyright IBM Corporation 2017

IBM Workload Automation components

The following section describes the components that are offered in the IBM Workload Automation suite.

IBM Workload Scheduler for z/OS, the scheduler in z/OS environments

IBM Workload Scheduler for z/OS provides an infrastructure to run batch and near real-time workloads and activities that support business services. Workload Scheduler for z/OS helps provide the reliable and Service Level Agreement (SLA)-compliant delivery of those business services. Workload Scheduler for z/OS manages the work flows that integrate mainframe application with distributed application, Enterprise Resource Planning, and business intelligence applications. Besides, the traditional ISPF panels, Workload Scheduler for z/OS provides a modern web-based interface for operations, administrative tasks, and reporting. The web-based interface also offers self-service catalog and dashboards on mobile devices for business users to be able to manage automatic tasks and perform health checks in a natural way.

IBM Workload Automation on Cloud

IBM Workload Automation on Cloud, formerly known as IBM Workload Automation (SaaS), is a software-as-a-service (SaaS) solution that allows enterprises to focus on controlling and optimizing running background work flows, and avoid the complications and costs of deploying and managing

a full server-based scheduling infrastructure. You can standardize and optimize the management and workload automation of your processes with automatic promotion of jobs, proactive alerting, and embedded recovery.

Workload Automation on Cloud helps application developers rapidly apply powerful dynamic scheduling to business intelligence applications by employing numerous connectors and ready-to-use process templates. New integrations are made available regularly, which makes the solution valuable for enterprise scheduling exploitation.

IBM Workload Automation on Cloud offers the following benefits:

- Delivers self-service automation in a full software-as-a-service delivery mode and with zero up front investment.
- Accelerates the automation of processes with quick start solutions (templates for typical retail, inventory, and payroll processes).
- Reduces costs by automating work flows with flexible dynamic scheduling.
- Enables application developers' autonomy with a dedicated interface that simplifies the definition and monitoring of their process automation.
- Manages automation on hybrid configurations, where processes are run across on-premises and cloud applications.

IBM Workload Scheduler on Bluemix

With the Workload Scheduler service on IBM Bluemix, you can integrate your applications with the capability to schedule workflows. Your application can use job scheduling within and outside Bluemix. Easily create workflows in your application to run regularly, at a specific time, or when events trigger them (for example, when a file is created), according to your needs.

With IBM Workload Scheduler Bluemix, you can easily automate REST calls, database actions, file transfers, and much more with included interfaces. You can automate applications in a hybrid environment, on cloud or on-premises environment, or a mixture of both. Create your application by integrating processes on different operating systems: Windows, AIX, Linux, HP-UX, and IBM i.

IBM Workload Scheduler V9.4

IBM Workload Scheduler, formerly known as IBM Tivoli Workload Scheduler, is licensed on a managed environment. Workload Scheduler V9.4 includes all the functions that were provided by Tivoli Workload Scheduler and Tivoli Workload Scheduler for Applications.

IBM Workload Scheduler V9.4 automates running batch and near real-time work loads and activities that support business services. It helps provide the reliable and Service Level Agreement (SLA) compliant delivery of these business services. Workload Scheduler extends automation capabilities to heterogeneous systems, and manages work flows that integrate mainframe

applications with distributed applications, Enterprise Resource Planning (ERP), and business intelligence applications.

Workload Scheduler offers mechanisms for resource virtualization and uses cloud technologies, which helps to minimize fixed costs. Workload Scheduler includes a modern web-based interface, called the *Dynamic Workload Console*, for operations, monitoring, scheduling, administrative tasks, and reporting. It offers self-service catalog for mobile devices to show scheduling services to business users in a natural way.

IBM Workload Scheduler V9.4 integrates with existing data center systems and cloud resources and applications. It delivers improved productivity for the daily operations of IT administrators and schedulers.

This class focuses on IBM Workload Scheduler.

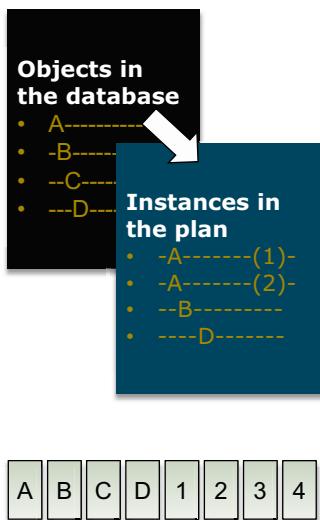
Lesson 3 Workload Scheduler terminology

Workload Scheduler terminology

© Copyright IBM Corporation 2017

In this lesson, you learn about the terminology that is used with Workload Scheduler.

Workload Scheduler terminology



- Production plan, the *to do* list of what to run
- Computers, agents, and workstations
 - Where software is installed
 - Where jobs run
 - What you use to define work
- Jobs and job streams
 - Definitions of tasks to perform
 - Groupings of similar tasks that run together
- Objects versus instances
 - Definitions in the database
 - Occurrences of definitions to run today
- Naming conventions
 - Give objects good names that you can easily filter later

© Copyright IBM Corporation 2017

Workload Scheduler terminology

Workload Scheduler maintains a database of production scheduling objects, such as workstations, jobs, job streams, resources, and calendars, and assembles them into a plan. The minimum set of object definitions that are required to produce a plan consists of a workstation, a job, and a job stream. Other required scheduling objects might be predefined and exist by default.

Production terms

The **production plan** (or just **plan**) is the list tasks that Workload Scheduler runs, and what dependencies must be satisfied before each task is started. Workload Scheduler builds the plan from elements that are stored in the scheduling database. The plan contains the list of the workload to complete within the planned production period. When all conditions and prerequisites are satisfied, the agent runs the workload components and reports the outcome.

While the plan is running, you track it to identify possible problems that affect the production of the workload. It is possible to track the plan from a web-based interface, the Dynamic Workload Console. You can also use the command-line interfaces to access the plan interactively or within shell scripts.

Computers, agents, and workstations

Several terms are used often in the class. Commonly used terms include *computer*, *agent*, *engine*, and *workstation*.

A *computer* is a physical or virtual system with a processing unit and an operating system.

The *agent* is the configuration of the software that runs on a computer to a particular role in the scheduling network. Types of agents include fault-tolerant agent, extended agent, and dynamic agent.

A *workstation* definition in the Workload Scheduler database represents one or more computer systems or entities capable of running tasks and reporting their status to the scheduler. You can also identify physical resources that are associated with workstations.

Jobs and job streams

A *job* is the representation of a task (an executable file, program, or command) that the scheduler schedules and starts. The job is defined to a workstation and run by an agent. After the job runs, it has a status that indicates whether the run was successful. A job definition can specify information on what to do whenever its run was not successful. A job definition does not include any attribute for when to run.

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

Objects versus object instances

Jobs and job streams that are defined in the database are different from jobs and job streams that are scheduled to run within the production period. Each occurrence of a job or a job stream that is scheduled to run in the current plan is called an *instance*. A production plan can contain more than one instance of the same job or job stream. Scheduling definitions in the database are *objects*.

During the production day, each occurrence of an object that is recorded in the plan becomes an *instance* of that database object. For example, during a production day, you can repeatedly submit a job stream that consists of related jobs. These submissions create multiple instances of job streams and job objects within the plan. Other database objects, such as resources and prompts, also correspond to instances in the current plan according to the needs of the job stream.

You can create and manage multiple instances of the same job stream over a number of days or at different times within the same day. You might see more than one instance of the same job stream with the same name in the same plan. The combination of the job stream name, the name of the workstation where it is scheduled to run, and the start time that is defined in the preproduction plan identifies a job stream instance.

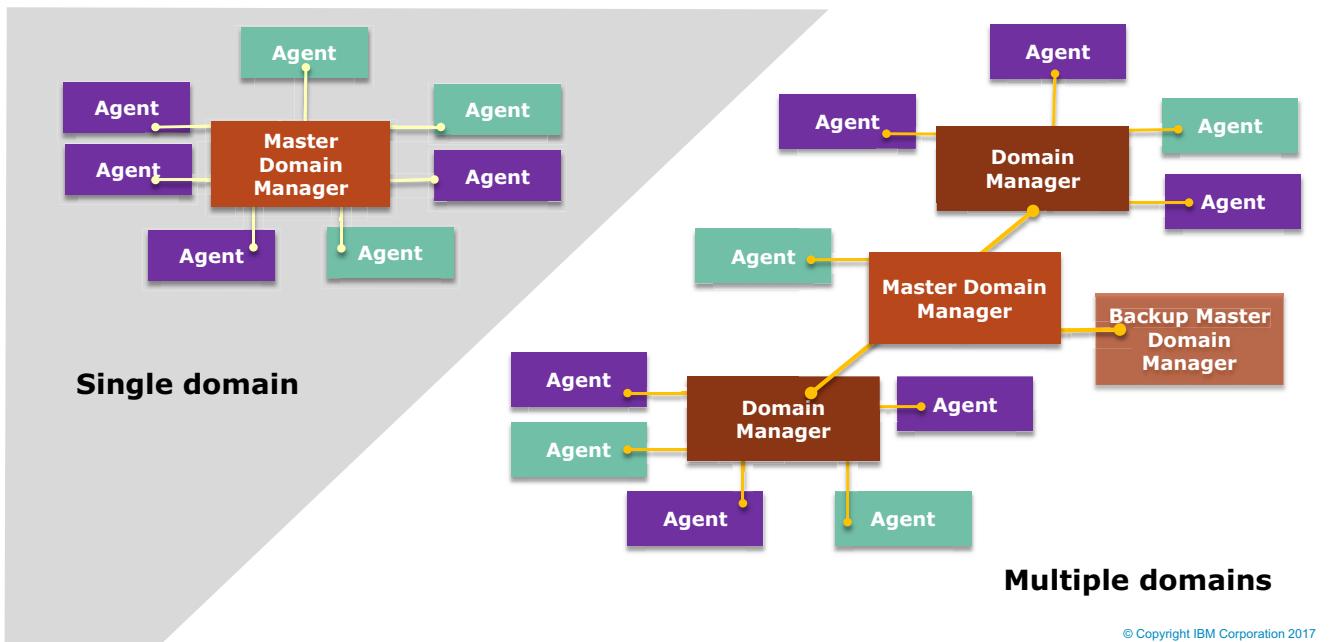


Important: When you change instances of objects within the plan, you are not changing the objects in the database.

Object naming conventions

Workload Scheduler emphasizes organizing your workload by department, division, function, and run cycle for efficient processing. Adopting organization-wide naming standards can help you with this organization. Naming standards establish object and instance ownership for monitoring and organizing databases and planning lists efficiently. Naming standards also control access to processing workstations, operating system groups, and individuals when combined with Workload Scheduler Security file configuration.

Workload Scheduler topologies



Workload Scheduler topologies

© Copyright IBM Corporation 2017

A *Workload Scheduler network* has workstations, where tasks are completed. 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.

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 communicate with workstations in their domains and their subordinate domain managers.

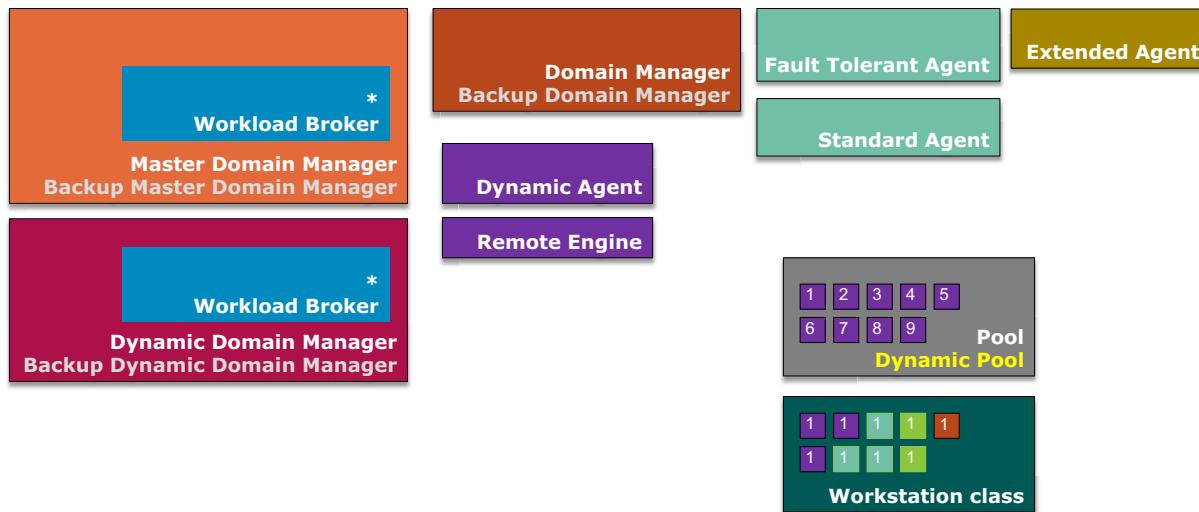
Workload Scheduler domain topology

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. 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.

You can group related processing to limit the amount of interdependency information that must be communicated between domains. Some of the benefits of concentrating processing in domains are listed here:

- **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.

Types of Workload Scheduler agents and workstations



© Copyright IBM Corporation 2017

Types of Workload Scheduler agents and workstations

IBM Workload Scheduler consists of software agents that run jobs and administer work flows. 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. When communication problems between a fault-tolerant agent and its domain manager occur, the 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. You can group dynamic agents in pools and dynamic pools. The Dynamic Workload Broker manages dynamic agents.
- 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.

Agents in a Workload Scheduler network can assume the following *roles*:

- The **master domain manager** is the manager of the topmost domain of a Workload Scheduler network. It connects to the database that stores the object definitions, creates, and updates the plan, distributes the plan, and logs and reports messages.
- A **domain manager** manages communications between agents in a domain.
- A **dynamic domain manager** adds the *dynamic workload broker* role to a domain manager.
- An **agent** starts jobs according to the plan and monitors the progress of these jobs.

You can define **workstations** in the Workload Scheduler database to configure how the tasks you create as job definitions are started on the different types of agents. While creating workstation definitions is outside the scope of this class, it is useful to understand how workstations relate job tasks to the agents where they run.

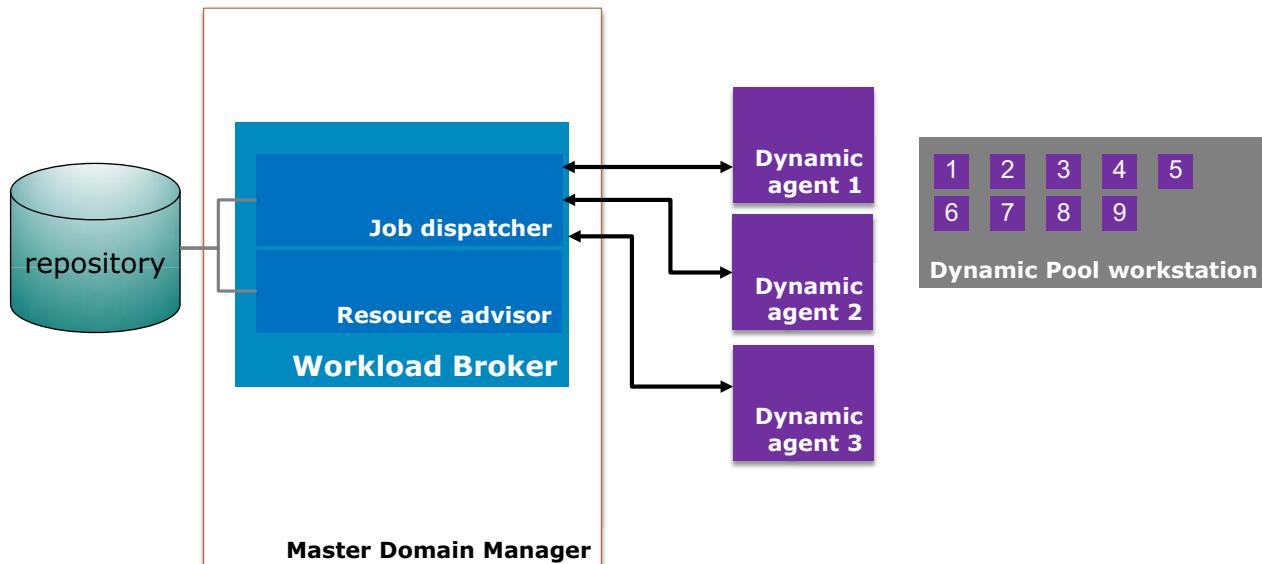
You can create these types of workstation definitions:

- **Domain managers** (DM) manage communications for all or part of the Workload Scheduler network. At least one domain manager, the master domain manager (MDM), is required. It is created during the installation of Workload Scheduler. Domain Managers other than the Master Domain Manager are installed as fault-tolerant agents and configured as domain managers in the workstation definition.
- A **fault-tolerant agent** (FTA) workstation represents each fault tolerant agent. You also use this workstation type for backup domain managers.
- An **extended agent** (X-AGENT) has a host and an access method. The host is any other workstation, except another extended agent. The access method is an IBM-supplied or user-supplied script or program that the host runs whenever the extended agent workstation is referenced. Extended agents are used to extend the job scheduling functions of Workload Scheduler to other systems and applications.
- A **standard agent** (S-AGENT) workstation receives and runs jobs only under the control of its domain manager.
- The **Workload Broker** (BROKER) workstation is created during the installation of Workload Scheduler. This type of workstation manages the lifecycle of Dynamic Workload Broker jobs. You create jobs for the Workload Broker workstation by using a unique format called Job Submission Description Language (JSDL) that is notated in XML. JSDL jobs are outside the scope of this class.
- An **Agent** (AGENT), also called *dynamic agent*, workstation runs a wide variety of job types, for example, database or file transfer jobs, in addition to native scripts and commands. This workstation is automatically created and registered when you install dynamic agent software on a computer that connects to the Master Domain Manager. *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** (POOL) workstation represents a group of dynamic agents with similar hardware or software characteristics. When you create a pool workstation, you assign one or more dynamic agents to be members of the pool. When you create a job that is assigned to a pool workstation, the job can run on any one of the agents that are assigned to the pool. When many jobs are scheduled to run at the same time, jobs are distributed across the agents that are available.
- Similarly, a **dynamic pool** (D-POOL) workstation is a collection of dynamic agents. Rather than directly assigning agents, you specify the requirements that an agent must meet to be a member of the pool. For example, an agent must run the AIX operating system and have more than 25% of its CPU capacity available. When you create a job that is assigned to a dynamic

pool workstation, the job can run on any one of the agents in the network that is qualified to be a member of the pool. When many jobs are scheduled to run on the pool workstation at the same time, the jobs are distributed to the agents that are available according to policies you select. For example, you can define that jobs must run on the least busy agent.

- A **remote engine** (REM-ENG) workstation represents a remote Workload Scheduler network. This workstation runs shadow jobs. You define and run shadow jobs to add and monitor dependencies on jobs that run in other scheduling (distributed or z/OS) environments.
- A **workstation class** is used to replicate its definitions to other workstations in the database. The class workstation includes other workstations in the database as members. When you define a job or job stream to the class workstation, Workload Scheduler replicates each definition to each member of the class when the production plan is extended. The workstation class is useful when you want to schedule the same task to many agents in the network to run on the same days. For example, the same system maintenance or backup task can run on many agents. You can define a single job or job stream to the class workstation, and Workload Scheduler replicates the definition to all members of the class during the planning process. The workstation class definition does not show in the production plans.

Dynamic Workload Broker and dynamic scheduling



© Copyright IBM Corporation 2017

Dynamic Workload Broker and dynamic scheduling

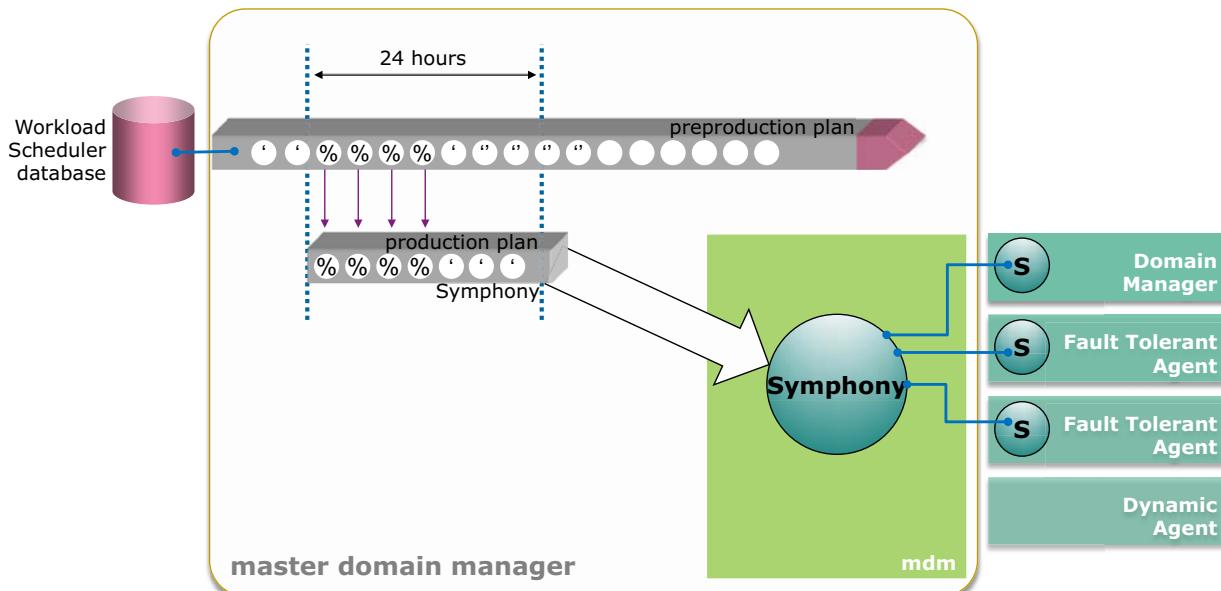
You can choose to set Workload Scheduler to dynamically associate your submitted workload (or part of it) to the best available resources at run time. These characteristics provide high availability and load-balancing potential to your environment and are suited to virtualized environments. To schedule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

- Define pool or dynamic pool workstations that you want to use for running workload as logical resources or groups of resources from the Dynamic Workload Console. 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.

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.

The Workload Scheduler production process



© Copyright IBM Corporation 2017

The Workload Scheduler production process

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. 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.

Creating the production plan

All of the required 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 must 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 production control file and make calls to the operating system to start jobs as required. The operating system runs the job, and informs Workload Scheduler of the job's status. This information is entered into the production plan to indicate the status of the job.

Lesson 4 Workload Scheduler user interfaces

Workload Scheduler user interfaces

© Copyright IBM Corporation 2017

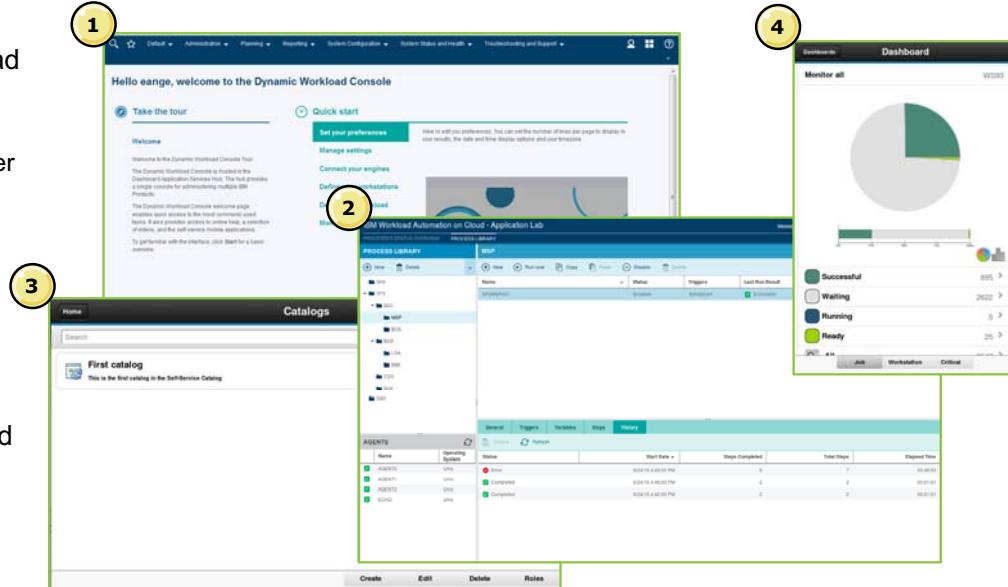
In this lesson, you learn about the user interfaces that are available with IBM Workload Scheduler.

Workload Scheduler provides three types of interfaces:

- 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



© Copyright IBM Corporation 2017

Workload Scheduler graphical interfaces

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

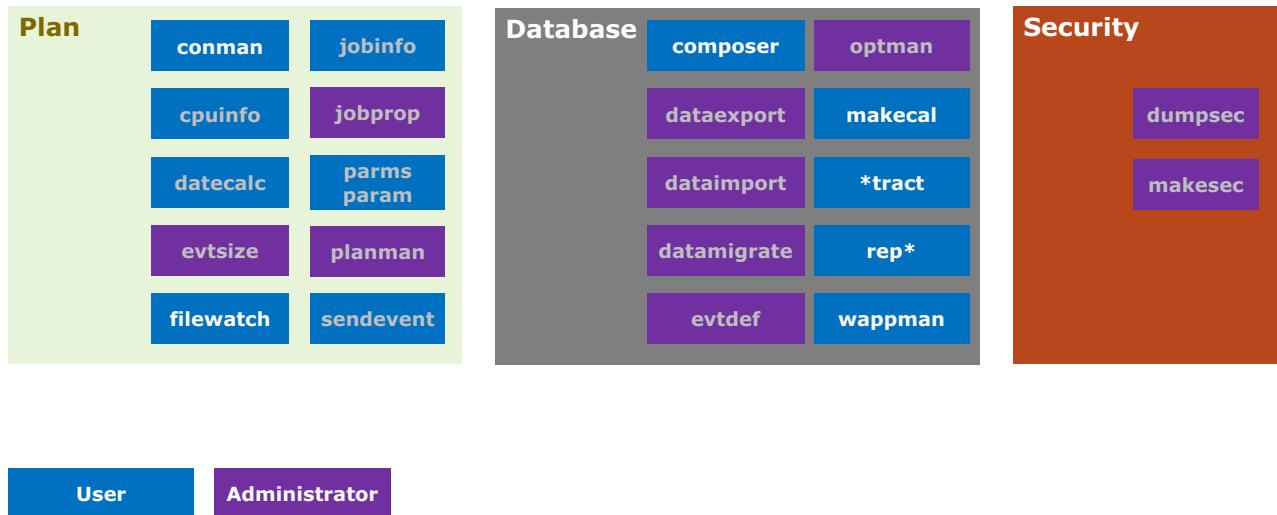
Jazz for Service Management links data and other shared integration services, including administrative, dashboard, reporting, and security services. Jazz for Service Management accelerates deployment, integration, and automation services across IBM, partner, and third-party tools by using open and standardized 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 you use with Workload Scheduler are:

1. **Dynamic Workload Console:** Manage and control IBM Workload Scheduler production for z/OS and distributed environments.
2. **Application Lab:** Create, run, and monitor simple business processes from a single web browser page.
3. **Self-Service catalog:** 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. **Self-Service Dashboard:** 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



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 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 conman command-line program. You use the conman program to start and stop processing, alter, and display the Symphony production plan, and control workstations that link in a network. You can use conman from the master domain manager and from any fault-tolerant agent in the IBM Workload Scheduler network.
cpuinfo	You can use the cpuinfo command to retrieve workstation information such as the node name, port number, and time zone.
datecalc	Use the datecalc 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 evtsize 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 filewatch 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 jobinfo command in a job script to retrieve information about the job, such as its job stream, priority, or rerun status.
jobprop	Use the jobprop command to set and retrieve variables for jobs that run on dynamic agents.
parms	Use the parms command to define and query locally managed parameters.
param	Use the param command to define and manage locally managed variables and passwords on dynamic agents.
planman	Workload Scheduler administrators use the planman command.
sendevent	Use the sendevent command to send custom events that are defined with the evtdef 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 composer command-line program. You can install the composer 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 dataexport 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 dataimport command to import all scheduling object definitions and global settings that the dataexport command saves into the Workload Scheduler database.
datamigrate	Use the datamigrate utility on the master domain manager to import data from the text files that were created by using composer extract commands or dataexport 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 <code>GenericEventPlugIn</code> 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, RESTful, and web Services interfaces to manage objects in the database and plan.

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 5 Opening the Workload Console

Opening the Dynamic Workload Console

© Copyright IBM Corporation 2017

In this lesson, you learn about starting the user interfaces that are available from the Dynamic Workload Console start page.

Opening the Dynamic Workload Console

To skip this page and access to Dynamic Workload Console directly, go to Dynamic Workload Console, click on the "Page Actions" icon on the right and select "Add to My Startup Pages".

© Copyright IBM Corporation 2017

Opening the Dynamic Workload Console

You can access four web-based user interfaces for IBM Workload Scheduler from the Dynamic Workload Console start page. Follow these steps to access the console start page.

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. The login page for the console is displayed. Enter your user ID and password and click **Log in**. When you are finished, the *Start* page is displayed. In the start page, you can link directly to the interfaces that are provided.

Be sure to use the **User** icon, **Log out** link in the navigation bar when you are finished with the console to prevent unauthorized access. The session expires and you must log in again if no activity is detected during the session for an extended period.



Attention: 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.

Depending on the authenticated roles of the user that logs in, you see links to different aspects of the four web-based interfaces you use with Workload Scheduler.

1. Use the **Dynamic Workload Console** to manage and control IBM Workload Scheduler production for z/OS and distributed environments. From this section, you can directly access four areas of the Dynamic Workload Console, depending on your security roles.
2. From the **Self Service UI** section, you can access the IBM Workload Scheduler environment by using mobile devices. You can link to the following interfaces.
 - a. In 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.
 - b. By using the **Self-Service Dashboard**, you can view dashboards on your mobile device and drill down to more detailed jobs and workstations that match the criteria you select.
3. From the **Application Lab** section, you create, run, and monitor simple business processes from a single web browser page.

Lesson 6 Getting started with Application Lab

Getting started with Application Lab

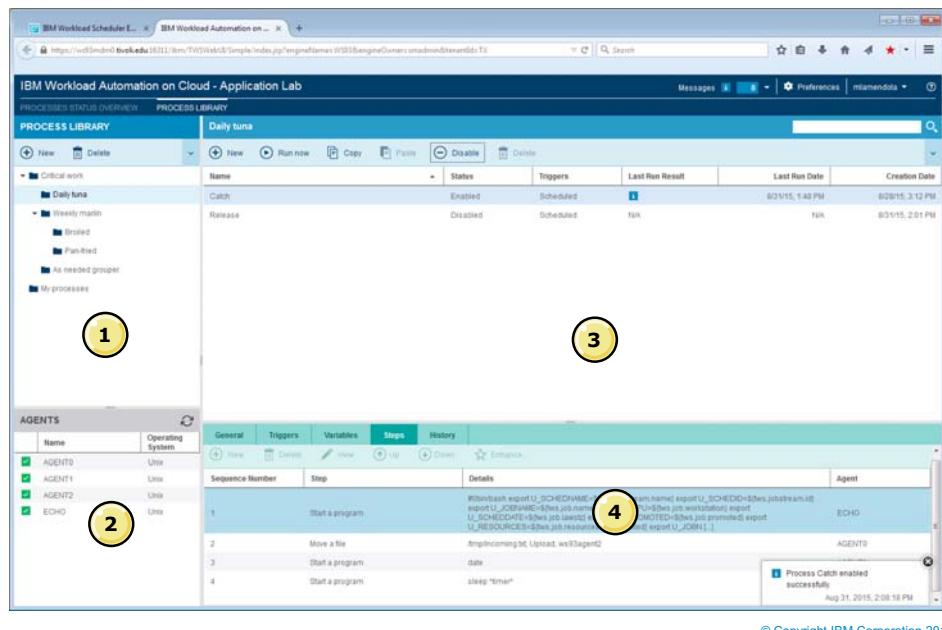
© Copyright IBM Corporation 2017

The Application Lab interface presents, on one page, all of the information you need to create simple scheduled work flows in Workload Scheduler.

In this lesson, you learn how to open the **Application Lab** interface, customize your preferences, and use the overview page.

The Application Lab interface

1. Process library
2. Available agents
3. Status
4. Details



The Application Lab interface

In the Application Lab interface, you see the information that you need to create and monitor simple workload processes. The interface has four panes.

1. The process library pane shows a folder-based structure where the process definitions are stored.
2. The agents list shows the Workload Scheduler agents that are available to use in the Application Lab.
3. The status pane has two modes.
 - a. When **Process Status Overview** is selected, the status pane shows the overall status of all the processes run in the Application Lab.
 - b. When **Process Library** is selected, the status pane shows a list of processes in the selected library.
4. The details pane shows more information about a process you select in the status pane.

Starting the Application Lab interface

To use the Application Lab interface, you start from the Dynamic Workload Console start page, or build a URL that access the Application Lab interface directly.

IBM Training

Opening the Application Lab interface from the start page

1. Click **Add**
 2. Select an engine, enter an environment ID, and click **Add to list**
 3. Click **OK**

© Copyright IBM Corporation 2017

Opening the Application Lab interface from the start page

To use the Application Lab interface from the start page, you must first save an Environment ID, and use the ID to access the Application Lab environment. You can save as many Environment IDs as you need. To save an environment ID, and start the Application Lab interface, complete the following steps.

1. In the Application Lab section of the Dynamic Workload Console start page, click **Add**.
2. In the “Select an engine” window, choose the engine connection that you use to connect to IBM Workload Scheduler. Complete the Environment ID field by entering a two-letter ID, and click **Add to list**.
3. In the Application Lab section of the Dynamic Workload Console start page, select the Environment ID you use for the session, and click **OK**.

Opening the Application Lab interface by using a URL.

The URL you use to open the Application Lab consists of the following parts:

Table 3 Application Lab URL

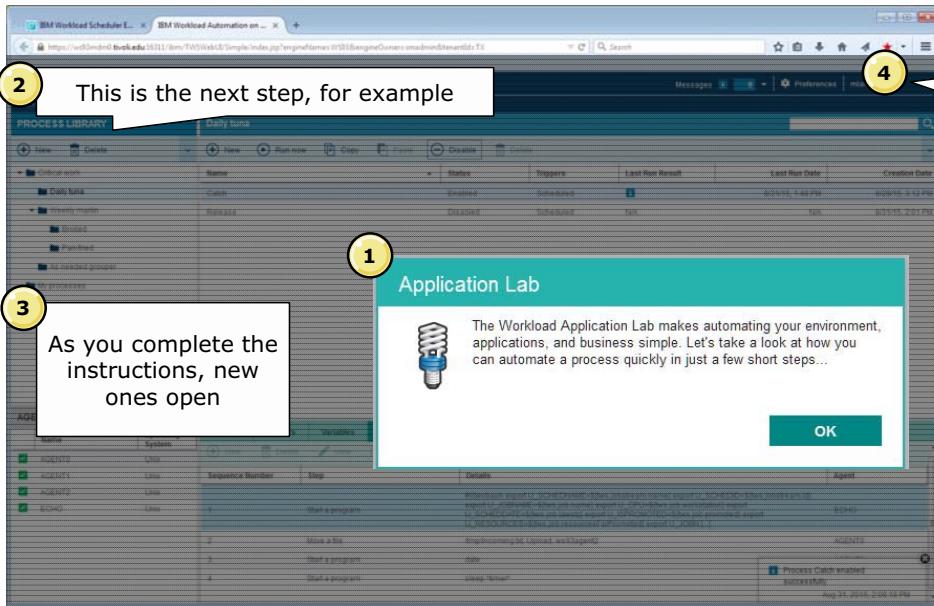
Name	Description and example
Context root	The host name, port number, and path to the Application Lab interface <code>https://host_name:16311/ibm/TWSWebUI/Simple/index.jsp</code>
Engine	The name of the previously defined Dynamic Workload Console engine connection to use. <code>engineName=X123</code>
Owner	The Dynamic Workload Console user that owns the engine connection that is referenced by <code>engineName</code> . <code>engineOwner=smadmin</code>
Environment ID (Tenant ID)	The two-letter environment ID. Users who use the same environment ID share process libraries. The Workload Scheduler administrator can restrict which users access an environment. <code>tenantId=NN</code>

For example, this URL opens the Application Lab with the engine connection named “WS94” and the environment ID “TX”.

`https://dwc.example.com:16311/ibm/TWSWebUI/Simple/index.jsp?engineName=WS94&engineOwner=smadmin&tenantID=TX`

When you open the Application Lab, you must provide your user name and password.

Following the Application Lab tutorial



Start a new tutorial by clicking the question mark (?)

1. A tutorial guides you through the steps to create a process
2. You see a tooltip with instructions for the step
3. New tips guide you to the next step
4. Upon completion, you can start a new tutorial by clicking the question mark
5. Include **skipTutorial=true** in the URL to avoid running the tutorial

© Copyright IBM Corporation 2017

Following the Application Lab tutorial

When you use the Application Lab, you can follow a built-in tutorial that guides you through the steps to create your process. As you complete each step, a tooltip guides you to the next step with instructions about which data is needed for that step.

You can suppress the tutorial by including the **skipTutorial=true** keyword in the URL you use to start the Application Lab interface. For example, the following URL starts the Application Lab without an introductory tutorial.

`https://dwc.example.com:16311/ibm/TWSWebUI/Simple/index.jsp?engineName=WS94&engineOwner=sadmin&tenantID=TX&skipTutorial=true`

Setting user preferences and recalling messages

You can change the time zone that you use to monitor processes by clicking **Preferences** and choosing a time zone from the menu.

As you use the Application Lab, messages that inform you of your progress appear at the lower-right corner of the window. The messages are automatically dismissed after a few seconds. To recall a message that no longer appears, click the **Messages** menu at the upper right to review all of the messages from the current session.

Lesson 7 Getting started with the Dynamic Workload Console

Getting started with the Dynamic Workload Console

© Copyright IBM Corporation 2017

The **Dynamic Workload Console** is a web-based user interface for the following products:

- IBM Workload Scheduler
- IBM Workload Scheduler for z/OS
- IBM Workload Scheduler for Applications

You 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 use menus that contain entries for each product that is installed and integrated with it.

With Dynamic Workload Console, you can perform the following 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 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.

Opening the Dynamic Workload Console

The screenshot shows the start page for the IBM Workload Scheduler. It features three main sections:

- Dynamic Workload Console**: Shows a monitor icon and a brief description: "Access the Dynamic Workload Console to manage your IBM Workload Scheduler environments." It includes links for "Read more", "Connect your engines", "Design your workload", "Monitor your workload", and "Dashboard".
- Self Service UIs**: Shows a monitor icon and a brief description: "Access the Self-service UIs to easily and quickly interact with your IBM Workload Scheduler environment from mobile devices." It includes links for "Read more", "Self Service Catalog", and "Self Service Dashboards".
- Application Lab**: Shows a monitor icon and a brief description: "Access the Application Lab to manage a set of simple business processes." It includes a message: "You don't have any saved configuration for Application Lab. Please add." and a "Add" button.

At the bottom of the page, there is a note: "To skip this page and access the Dynamic Workload Console directly, go to Dynamic Workload Console, click on the "Page Actions" icon on the right and select "Add to My Startup Pages"."

© 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.
For example, <https://www.example.com:16311/ibm/console>
2. The login page for the console is displayed. Enter your user ID and password and click **Log in**.
When you are finished, the Start page is displayed. On the start page, in the Dynamic Workload Console section, click **OK**. The Welcome page is displayed.



Hint: Using start page shortcuts

From the start page, you can go directly to the following pages without passing through the Welcome page.

- Click **Connect your engines** to open the “Manage engines” page.
- Click **Design your workload** to go to the Workload Designer page.
- Click **Monitor your workload** to go to the “Direct query” page.
- Click **Dashboard** to go to the workload monitoring dashboard.

Using the Welcome page

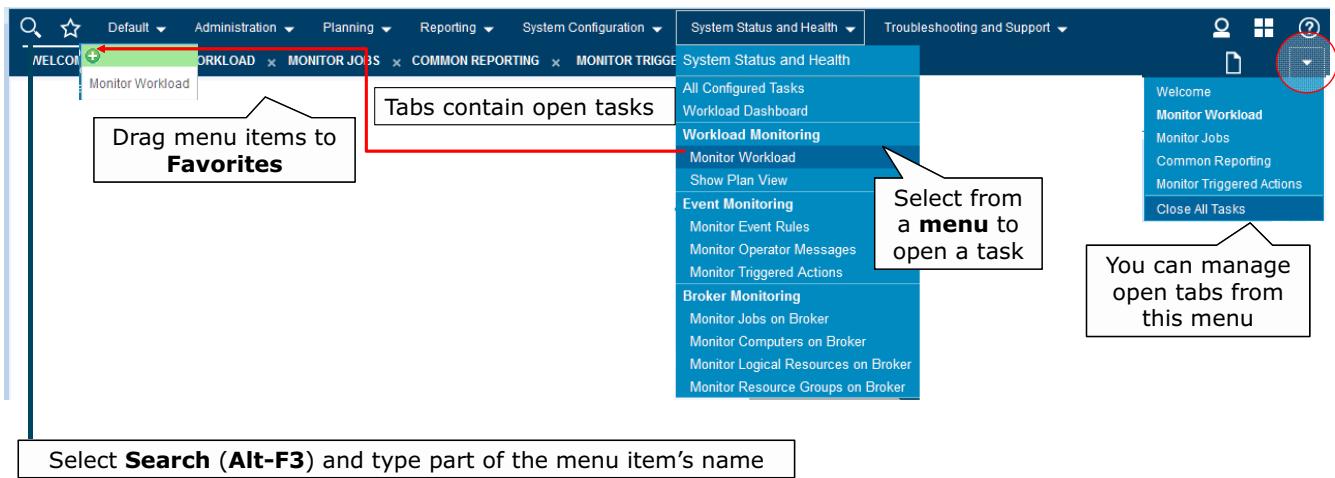
© Copyright IBM Corporation 2017

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.

Using menus and tabs



© Copyright IBM Corporation 2017

Using menus and tabs

The Dynamic Workload Console uses a common interface design that is provided by Dashboard Application Services Hub (DASH). You select tasks from menus across the top of the window. The menus that you see depend on the security roles that are assigned to your account. The menus that are available are listed in [Table 4](#).

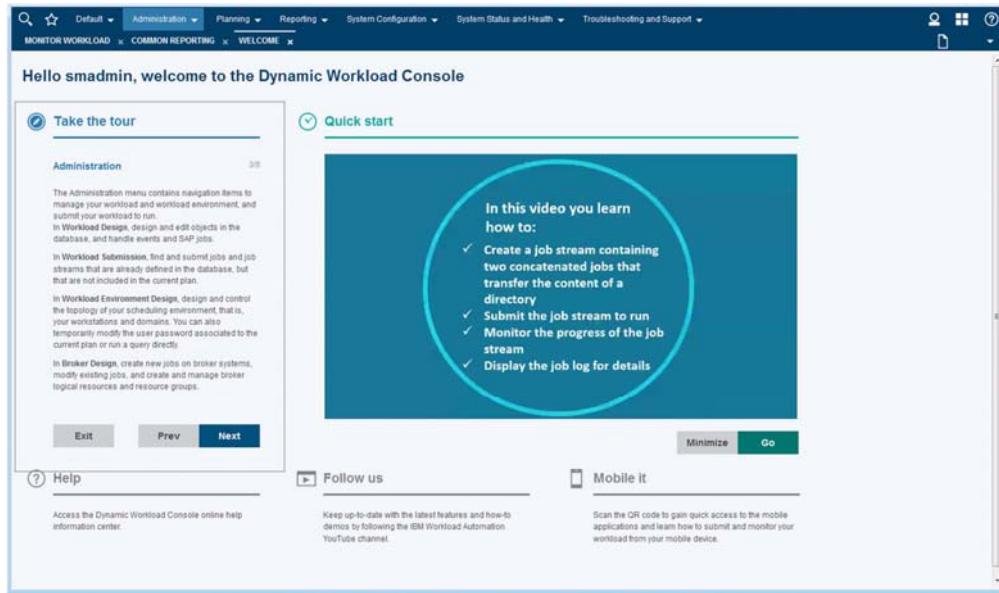
Table 4 Dynamic Workload Console menus

Menu	Description
Search	Enter part of a menu item's name to open the item without using the menus.
Favorites	Drag your favorite menu selections here to save them in a convenient location.
Default	Open the window that starts when you log on to the Dynamic Workload Console.
Administration	Design workload objects, and submit new work from this menu.
Planning	Create and monitor history, trial, and forecast plans.
Reporting	Create and manage workload reports, and use the Common Reporting tool from this menu.
Samples	View sample dashboards that are provided by the DASH interface.
System Configuration	Set user preferences and manage engine connections.
System Status and Health	Monitor workload in various ways from this menu.

Table 4 Dynamic Workload Console menus

Menu	Description
User	Access your favorites, start pages, and user information from this menu. You also log out from this menu.
Views	Depending on which applications are installed in DASH, you can switch to different application-specific views from this menu.
Console Settings	Manage console settings, DASH-related properties, groups, and roles from this menu.
Help	Access help pages from this menu.

Using the tour and Quick start



© Copyright IBM Corporation 2017

Using the tour and Quick start

The Dynamic Workload Console provides a guided tour of its features, and built-in videos that demonstrate how to perform common tasks.

Taking the tour

The tour guides you to use features in the Dynamic Workload Console. As you view each step, you see a menu or option, and text that explains how to use the item.

To start the tour, in the **Take the tour** section, click **Start**. To proceed to the next step, click **Next**. Click **Exit** to complete the tour.

Using the Quick start

The **Quick start** section provides links to commonly used tasks, and video demonstrations about how to use them. From the Quick start, you can perform the following tasks:

- Select a task from the list to the left, and click **Play** to show the video. You can also click **Maximize** to enlarge the video window.
- Select a task from the list, and click **Go** to open the task without using the menus.

Configuration

The configuration area contains icons for user functions, view selection, and console configuration. You see different selections that are based on the assigned roles of your login user.

- Use the **User** icon () to log out and change your password. You can also manage your favorites from the user section.
- Use the **Views** icon () to change views. Views dictate which product tasks, dashboards or pages, are displayed and which pages or dashboards open automatically. The Dynamic Workload Console provides separate views for Workload Scheduler, Workload Scheduler for z/OS, and Dynamic Workload Broker.
- Use the **Configuration** icon () to access console administration and user management functions.

Use the **Help** icon () to access help contents for the console and any installed products and to find details about the console version and installed products. You can also access the Dashboard Application Services Hub community through the **Help** icon, where you find multimedia tutorials and information about upcoming releases.

Working with favorites

You can add frequently used navigation items, such as tasks and dashboards, to your list of favorites. By design, some navigation items, for example, **User icon > Log out**, cannot be added to your list of favorites.

To add a navigation item to your list of favorites, expand the navigation bar to display the item that you want to add and click and drag it to the **Favorites** icon (). When you drag the navigation item to the **Favorites** icon, the cursor changes to show a red box with a plus (+) symbol. When you hover over the **Favorites** icon, the cursor changes to display a green box. Release the mouse button to add the item to your favorites list.



Note: If the item that you drag to the list is already there, the cursor does not change to green. You cannot add an item more than once.

To remove a navigation item from your list of favorites, perform the following tasks:

1. Click the **Favorites** icon to display your list of favorites.
2. Click and drag the item that you want to remove away from the list.
3. Release the mouse button to remove the item from your favorites list.

Managing Startup tasks

Use **My Startup Pages** to add or remove dashboards that are started when you log in to the console. You can set which dashboard opens as the default tab.

To open My Startup Pages dashboard, click the **User Icon** (), and select **My Startup Pages**. From the My Startup Pages dashboard, to open one of your start dashboards, click the link for that page in the Page name column.

You can also define the default view in My Startup Pages. You can select All tasks, Favorites, Tivoli Workload Scheduler, Tivoli Workload Scheduler for z/OS, or Tivoli Dynamic Workload Broker from the list. After you apply your selection, the view that you selected is the default navigation every time you log in.



Hint: To add a page or dashboard to your start pages, go to the dashboard in question. In the top bar, click the **Page Actions** icon, and select **Page Actions > Add to My Startup Pages**.

Using the Search menu

At the left of the navigation menus, use the **Search** () menu to quickly find particular tasks. Click the **Search** icon, and start typing the name of a task that is in the console. Search results are dynamically returned and filtered as you type.

Content area

The content area displays the dashboard that you are using. The dashboard contains one or more web applications or widgets. Pages that contain multiple related widgets are generally called dashboards and invariably include a number of integrated widgets.

Getting context-sensitive help

If you have a function open in the content area, click the question mark () in the upper-right corner of the content area to see task-specific help for the open tab. This help is displayed as a secondary window on top of the main browser window. The **Related Information** link at the bottom of this help window opens the main help index.

Lesson 8 Connecting the console to Workload Scheduler

IBM Training

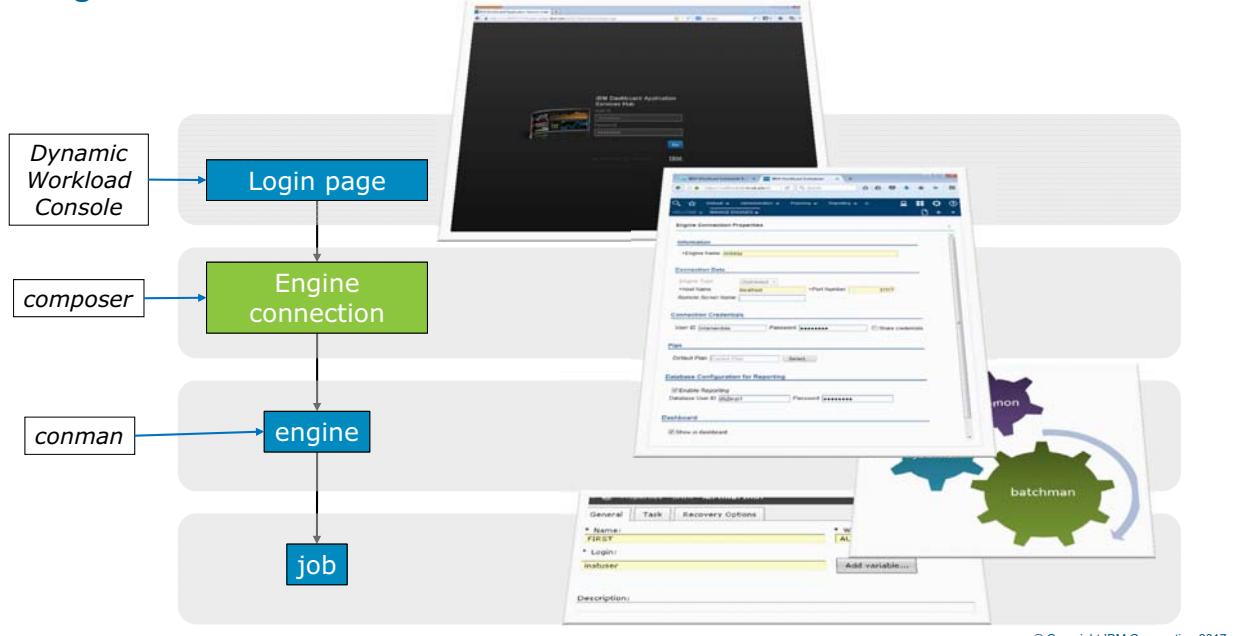
IBM

Connecting the console to Workload Scheduler

© Copyright IBM Corporation 2017

In this lesson, you learn how to connect the Dynamic Workload Console to Workload Scheduler.

The engine connection



The engine connection

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.

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 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 to, 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 **fault-tolerant agent** 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 fault-tolerant agent. You choose this option if you need up-to-date information about job status on this workstation.

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 by using sorting criteria that you select with the toolbar 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, Production, QA, or jmulberry@tws92mdm0(31117). 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.

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

IBM Training



Getting started with the command-line interfaces

© Copyright IBM Corporation 2017

In this lesson, you learn how to start the command-line interfaces.

The conman and composer programs

Plan	conman	Database	composer
	<ul style="list-style-type: none">▪ Monitor jobs and job streams▪ Monitor scheduling environments▪ Manage jobs and job streams▪ Add work to the plan		<ul style="list-style-type: none">▪ Create and modify scheduling objects▪ Create and modify jobs and job stream definitions

© Copyright IBM Corporation 2017

The conman and composer programs

The conman and composer programs are command-line user interfaces for Workload Scheduler. You monitor jobs and job streams in 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 Up!
[wsuser@ws94mdm0 ~]$ export MAESTRO_OUTPUT_STYLE=LONG
[wsuser@ws94mdm0 ~]$ conman
IBM Workload Scheduler(UNIX) CONMAN 9.4.0.0.01 (20170626)
Licensed Materials - Property of IBM* and HCL**
5698-WSH
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.
(C) Copyright HCL Technologies Ltd. 2016,
* Trademark of International Business Machines Corporation
** Trademark of HCL Technologies Limited
Installed for user "wsuser".
Locale LANG set to the following: "en"
Scheduled for (Exp) 09/12/17 (#101) on MDI
% exit
```

1. Log on by using **xterm**, **PuTTY**, or other terminal
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 start the command-line interfaces, set your PATH and TWS_TISDIR variables by sourcing one of the following scripts:

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

Setting optional variables

Before you start the 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. After the interface displays a number of lines of data, the interface prompts to continue at the end of each page. 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 depending on the value of MAESTROCOLUMN. If MAESTROCOLUMN 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 8 characters are truncated to 7 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 start 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 conman commands are not case-sensitive.

Using the command-line interfaces

```
[wsuser@ws94mdm0 ~] $ composer
IBM Workload Scheduler(UNIX) /COMPOSER 9.4.0.01 (20170626)
- help commands
IBM Workload Scheduler(UNIX) /MHELP

COMPOSER COMMANDS

Note: Command names and keywords can be entered in either
      lowercase characters, and can be abbreviated to a
      few characters as are needed to uniquely distinguish
      them. Some of the command names also have short
      aliases.

+-----+-----+
| Command | Short | Description
|          | Name  |
+-----+-----+
| 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 object 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, 9.4.0.01 (20170630)
- 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) 07/01/17**”.
- 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 a **Batchman LIVES** or a **Batchman down** message when you enter 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. You learn more about using the `composer` program in [“Creating job stream definitions by using composer”](#) on page 6-41.

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`

```
[twouser@ws94mdm0 ~]$ conman
IBM Workload Scheduler(UNIX) /CONMAN 9.4.0.01 (20170626)
Licensed Materials - Property of IBM* and HCL**
5698-WSH
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.
(C) Copyright HCL Technologies Ltd. 2016, 2017 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) 08/01/17 (#91) on MDM0. Batchman LIVES. Limit: 97, 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 command 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`.

Instructor demonstration

Starting the Application Lab

Starting the Dynamic Workload Console



© Copyright IBM Corporation 2017

Instructor demonstration

Exercises

- Log in to the computer
- Start the Application Lab interface
- Start the Dynamic Workload Console interface



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this unit.

Review questions

1. What are the main benefits of using Workload Scheduler to automate workloads?
2. What are the four IBM Workload Automation component offerings?
3. What is the difference between an *agent* and a *workstation*?
4. Describe objects versus object instances.
5. What is the purpose of a workstation class?
6. What are the four main graphical user interfaces for Workload Scheduler?
7. What is the main purpose of the *engine connection*?
8. Which are the two main command-line interfaces for accessing the Workload Scheduler plan and database?

Review answers

1. What are the main benefits of using Workload Scheduler to automate workloads?

By using Workload Scheduler, the work is consistently defined and ready to occur as planned, human intervention is reduced or eliminated, the work is optimized to occur as quickly as possible while still meeting all dependency requirements, and error monitoring is centralized.

2. What are the four IBM Workload Automation component offerings?

The components that are offered in the IBM Workload Automation suite are IBM Workload Scheduler for z/OS, IBM Workload Automation on Cloud, IBM Workload Scheduler on Bluemix, and IBM Workload Scheduler.

3. What is the difference between an *agent* and a *workstation*?

The agent is the configuration of the software that runs on a computer to a particular role in the scheduling network. A workstation definition in the Workload Scheduler database represents one or more computer systems or entities capable of running tasks and reporting their status to the scheduler.

4. Describe objects versus object instances.

You define objects, such as jobs and job streams in the database. Each occurrence of a job or a job stream that is scheduled to run in the current plan is called an instance. A production plan can contain more than one instance of the same job or job stream object.

5. What is the purpose of a workstation class?

A workstation class is used to replicate its definitions to other workstations in the database.

6. What are the four main graphical user interfaces for Workload Scheduler?

The four web-based interfaces you use with Workload Scheduler are the Dynamic Workload Console, Application Lab, Self-Service catalog, and Self-Service Dashboard.

7. What is the main purpose of 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. To manage scheduling objects, you must connect from the Dynamic Workload Console to a Workload Scheduler environment with an engine connection.

8. Which are the two main command-line interfaces for accessing the Workload Scheduler plan and database?

*The **conman** and **composer** programs are command-line user interfaces for Workload Scheduler. You monitor jobs and job streams in Workload Scheduler by using the **conman** program. You use **composer** to manage objects in the Workload Scheduler database.*

Summary

You now should be able to perform the following tasks:

- Use Workload Scheduler terminology
- Describe the agent and workstation types that are available in Workload Scheduler
- Start the Application Lab interface
- Start the Dynamic Workload Console
- Start the command line interface

© Copyright IBM Corporation 2017

Unit summary

Unit 2 Using the Application Lab interface

IBM Training



Using the Application Lab interface

© 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 use the Application Lab interface to create simple work flows in Workload Scheduler.

Objectives

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

- Use the process status overview page
- Organize your work into process libraries and sublibraries
- Create a process, including steps, triggers, and variables
- Enable your process to run automatically or manually
- Enhance workload applications built by using the Application Lab
- Import scheduling definitions from cron or Windows Task Scheduler

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Using the processes status overview page

IBM Training

IBM

Using the process status overview page

© Copyright IBM Corporation 2017

In this lesson, you learn how to use the process status overview page.

You use the **processes status overview** page to monitor the progress of all the processes at a high level. From the overview page, you see:

- The process library, where process definitions are organized into folders.
- A list of available agents, from which you can choose to run process steps.
- The processes status overview.

Processes status overview page components

The screenshot shows the 'PROCESSES STATUS OVERVIEW' section of the IBM Workload Automation interface. It includes a sidebar with 'PROCESS LIBRARY' and 'AGENTS' sections. The main area displays a table of processes with columns: Process Name, Status, Start Date, Steps Completed, Total Steps, and Elapsed Time. A 'Step Status' window is overlaid, showing a table of step details with columns: Status, Start Date, Step Type, Details, Agent, Elapsed Time, and Return Code.

© Copyright IBM Corporation 2017

Processes status overview page components

From the processes status overview page, you see all of the enabled processes that are planned for the day. You filter the list of processes according to their age by moving the **View last hours** slider. Moving the selection to the increases how many processes you see in the overview.

The list is not automatically refreshed. Click **Refresh** to update the overview with the most current information.

Select a process instance from the list, and click **Details** to review the information about the steps in the process you select. From the **Step Status** window, you can perform the steps listed in [Table 1](#).

Table 1 Using the Step Status window

Function	Description
	Update the view by clicking Refresh .
	View the output of a completed step by selecting a step from the list, and clicking View Log . You see the log in a new browser window.
	Download the output of a completed step by selecting a step from the list, and clicking Download Log . Depending on the configuration of your browser, you choose to open the log file or save it to your computer.

Table 1 Using the Step Status window (continued)

Function	Description
 Rerun	Rerun a completed step. You can rerun only steps that show Completed or Error status.
 Stop	Stop a running step. You can stop only steps that show Running status.
 Close	Close the Step Status window.

Lesson 2 Managing processes

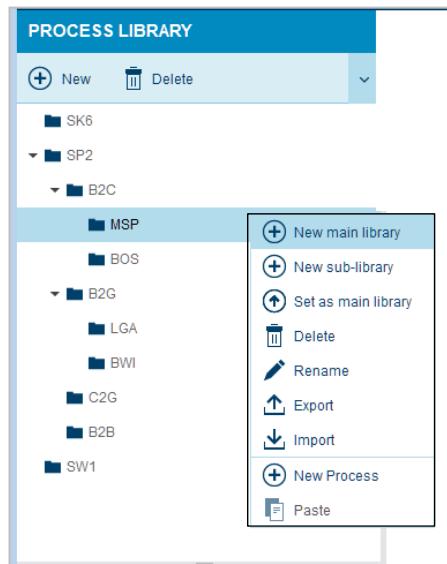
Managing processes

© Copyright IBM Corporation 2017

In this lesson, you learn how to organize and develop your process tasks and steps into process libraries. You learn how to perform the following tasks:

- Create process libraries and sublibraries
- Create new processes in the Process Library
- Create a process, including creating steps, triggers, and variables
- Run and monitor processes

Using the process library



- A folder represents each process library
- Organize processes into libraries and sublibraries (many levels deep)
- Right-click to create, delete, or rename libraries
- Move a process to a different library by dragging it to the library's folder
- All users with the same tenant ID share process libraries

© Copyright IBM Corporation 2017

Using the processes library

By using process libraries, you can organize processes into folders and subfolders for easier access. All users who log in with the same tenant ID share the process libraries.

You create a process library in one of the following ways:

- Click **New** in the process library header.
- Right-click the process library area and click **New main library**.
- Right-click an existing process library, and click **New sublibrary**. You can create as many sublibraries as you need.

Similarly, you delete a process library by clicking **Delete** in the process library header, or right-clicking the library and clicking **Delete** from the menu. Before you can delete a library or sublibrary, you must remove all the processes from the library. When you delete a library, you also delete all of its sublibraries.

You can move a sublibrary to any other main library or sublibrary. Drag the folder to the new library to move it. As you drag the folder, you see its color change from red to green when you reach a valid destination for the folder.

Creating a process

The screenshot shows the 'IBM Workload Automation on Cloud Application Lab' interface. In the top left, there's a tree view under 'Critical work' with nodes like 'Daily tasks', 'As needed groups', and 'My processes'. A yellow circle labeled '1' is around this tree view. In the center, a table lists 'Process Library' items: 'Catch' (Enabled, Scheduled) and 'Release' (Disabled, Scheduled). A yellow circle labeled '2' is around the 'Catch' row. Below the table, a large yellow circle labeled '3' has an arrow pointing to a yellow circle labeled '6' on the right. The bottom section shows a 'Steps' tab with a table of four steps:

Sequence Number	Step	Details	Agent
1	Start program	Wbtask export U_SCHDENAME=\$bsh jobstream named export U_SCHEDID=\$bsh jobstream.id; export U_JOBNAME=\$bsh job name; export U_CPN=\$bsh job priority; export U_SCHEDULED=\$bsh job start; export U_RSCHEDULED=\$bsh job promoted; export U_RESOURCES=\$bsh job resources; export U_JOBBLI... AGENT3	ECHO
2	Move a file	tmpfilecoming.txt, Upload, ws#3agent2 AGENT3	
3	Start a program	date AGENT3	
4	Start a program	sleep 5msec AGENT3	

A message at the bottom right says 'Process Catch enabled successfully! Aug 31, 2015, 2:08:18 PM'.

1. Select the process library
2. Click New
3. Complete the General tab
4. Optionally complete the Triggers tab
5. Optionally complete the Variables tab
6. Complete the Steps tab

Creating a process

To create a process, perform the following steps.

1. Select the library that contains the process.
2. In the status pane, click **New**.
3. Define the process properties and conditions by completing the **General**, **Triggers**, **Variables**, and **Steps** tabs in the details pane. The details for each tab are explained in the next section.
4. Enable the process by selecting it in the status pane, and clicking **Enable**.

Applying general properties

In the details pane, the **General** tab is selected. Complete the following fields in the **General** tab.

- Name the process by completing the **Name** field. This field is required.
- Add a description by completing the **Description** field.
- When you enter a date in the **Valid from** field, the process cannot run before the date you select, either by using triggers or manually requesting it.
- When you enter a date in the **Valid to** date, the process cannot run after the date you select.

Adding triggers to your process

You use triggers to specify when Workload Scheduler should automatically run the processes that you define. You create as many triggers as needed for each process. Triggers are schedule-based

or event-based. You use schedule-based triggers to start the process on a regular interval based on the date, day of week, or day of the month, and the time of day. You use event-based triggers to start the process when a file is created, modified, or deleted on a computer that runs a Workload Scheduler agent. To add a trigger, in the **Triggers** tab, click **New**.

Using schedule-based triggers

Schedule-based triggers define how frequently to run the process. In the Create Trigger window, select **Based on schedule** from the **Start the process** menu. When you use a schedule-based trigger, the fields that are listed in [Table 2](#) are available.

Table 2 Schedule-based trigger options

Field	Description
Frequency	<p>Choose one of the following options:</p> <ul style="list-style-type: none"> • Once: Run the process only once. You also select the start time and date. • Daily: Run the process every day that meets the other criteria that you define. • Weekly: Run the process on the days of the week that you specify. • Monthly: Run the process on the days of the month that you specify. <p>When you select a frequency, other selection criteria on the Create Trigger window change.</p>
Start at	Enter the date and time at which you want the trigger to start.
Valid to	Enter the last date that you want the trigger to be active.
Run every selected number of days, weeks, or months	Depending on the frequency you choose, you can set the trigger to skip consecutive periods by selecting a higher number. For example, to start a trigger that runs a process every other day, choose the Daily frequency, and enter 2 in the Run every selected number of days field.
On the following day	For daily frequencies, select to start the trigger every day, on working days, or on non-working days.
On the following days of the week	For weekly frequencies, select the days of the week you want to start the trigger.
Select days	For monthly frequencies, select that the trigger starts on the first day of the month, the second day of the month, and so on. Use the Advanced Options to choose other days of the month to start.

Table 2 Schedule-based trigger options

Field	Description
	<p>For example, to start a trigger every June 19th, complete the following fields:</p> <ul style="list-style-type: none"> • Select the Monthly frequency. • Choose July 1, 2015 for the Start at field. Optionally, enter a start time. • Enter 12 in the Run every selected number of months field. • In the Advanced options, select On the 19th day.
Repeat every	The trigger starts regularly throughout the day when you complete the Repeat process every field. Complete the Repeat until field to specify when the process should no longer be repeated.

Using event-based triggers

Event-based triggers define how to run the process when a change in the agent's file system occurs. In the Create Trigger window, select **Based on event** from the **Start the process** menu. When you use an event-based trigger, the fields that are listed in [Table 3](#) are available.

Table 3 Event-based trigger options

Field	Description
Event type	<p>Choose one of the following options:</p> <ul style="list-style-type: none"> • File created: The trigger starts when a file exists on the agent's file system that did not exist before. • File deleted: The trigger starts when a file that existed on the agent's file system no longer exists. • File modified: The trigger starts when a file on the agent's file system remains unchanged in two consecutive monitoring periods (as defined in the Advanced options). <p>Note: You can use wildcards as part of the file name for File Created and File Deleted event types.</p>
Agent	Select the agent where the file is monitored.
File Name	Enter the full path and file name of the monitored file.
Advanced options	Enter the monitoring sample interval for testing the event.

Using the triggers rerun option

When you create triggers which run the same process more than once, each instance of the process could start while another instance is still running. To change the behavior of multiple instances running concurrently, choose one of the rerun options listed in [Table 4](#).

Table 4 Trigger rerun options

Rule	Description
Do not start the instance	Choose this rule if Workload Scheduler should suppress a new process instance if an existing instance is still running.
Run the new instance in parallel (default)	Choose this rule if Workload Scheduler should start a new process instance, even if an existing instance is still running.
Queue the new instance	Choose this rule if Workload Scheduler should cause a new process instance to wait for an existing instance to complete successfully.

Defining variables

You can use variables either on the Steps page or in the Self-Service Catalog. Variables are useful when you want to reuse the same definition in different steps. To create variables, click **New** on the **Variables** tab.

In the Edit Variable window, complete the fields as listed in [Table 5](#).

Table 5 Variable options

Field	Description
Name	Enter the name of the variable. You use the name to reference the variable in the steps you create.
Type	Choose one of the available types from the Type menu. The type you select defines what a user of the Self-Service Catalog sees when they select this process.
Description	Optionally, enter a brief description of the variable.
Default value	Optionally, enter a default value for the variable, depending on the Type you selected.
Advanced options	Optionally, depending on the variable type, you can enter further restrictions on the value of the variable that can be entered from the Self-Service Catalog.

Defining steps

You use steps to specify the tasks Workload Scheduler agents should run on the computers where agents are installed. You create as many steps as needed for each process. Steps are started on the agent computers in the order you define them, when the triggers you define start. Each step in your process starts as soon as the previous step in the list completes successfully.

To create a step, you select the type of task to run, select which agent should run the task, and define the properties of the task. Depending on the task type, each task type has different properties which can be defined. The **Create step** window changes according to which type of task you select for the step. The list of task types changes according to the job plugins that are installed on the Workload Scheduler server. In this lesson, you learn how to create two common types of tasks, **Start a program** and **Move a file**.

Creating a Step to start a program

To create a step that starts a program, perform the following steps.

1. To add a step, on the **Steps** tab, click **New**.
2. In the “Create step” window, select **Start a program** from the **Step** menu.
3. Select an agent to run the step from the **Agent** menu.
4. In the **Program** field, type the path to the script or command you want the step to run. You can instead paste or type an entire shell script into the **Program** field.
5. In the **User** field, type the account name of the user that runs the program on the agent.
6. In the **Password** field, type the password of the account you entered in the user field. For programs running on UNIX and Linux computers, this field is optional.
7. Optionally, complete any fields in the **Advanced Options** section.
8. Click **OK** to save your step definition.

Creating a Step to move a file

To create a step that moves a file, perform the following steps.

1. To add a step, on the **Steps** tab, click **New**.
2. In the “Create step” window, select **Move a file** from the **Step** menu.
3. Select an agent to run the step from the **Agent** menu.
4. In the **Agent data** section, in the **File** field, type the path to the file, on the agent, you want to move.
5. In the **User name** field, type the account name of the user that transfers the file on the agent.
6. In the **Password** field, type the password of the account you entered in the user field.
7. Select the direction of transfer from the **Direction** menu. You can select one of the following options.
 - **Server to agent** downloads the file from the server you specify in the **Server** field.
 - **Agent to server** uploads the file to the server you specify in the **Server** field.
8. From the **Protocol** menu, select the protocol of the file transfer mechanism the task uses.

9. In the **Remote server data** section, in the **File** field, type the path to the file, on the remote server, you want to move.
10. In the **Server** field, enter the host name of the remote server.
11. In the **User name** field, enter the account name of the user on the remote server.
12. In the **Password** field, enter the password of the remote user.
13. Optionally, complete the fields in the **Advanced Options** section, including the type of transfer and connection mode
14. Click **OK** to save your step definition.

Using variables in Step definitions

You can use the variables you created on the **Variables** tab in any part of a Step's task definition. To use a variable in your step definition, enter the name of the variable enclosed in caret (^) symbols. For example, if you created a variable named StartDay that contains a date, enter **^StartDay^** in the step definition.



Hint: If you use an event-based trigger to run the process, a variable named **FileName** is automatically created. The value of the variable is the path of the file that caused the trigger to start.

Changing the order of Steps

Steps are started on the agent computers in the order you define them. As you create steps, they are assigned a sequence number, that shows the order the steps run. You can change the order of the steps by moving steps up or down.

To move a step, on the **Steps** tab, highlight the step, and click **Up** or **Down**, depending on which direction you want to move the step.

Running and monitoring processes

The screenshot shows the IBM Workload Automation on Cloud interface. On the left, the 'PROCESS LIBRARY' pane displays a tree view of work items, including 'Critical work' (with 'Daily tuna' selected), 'Weekly martin' (with 'Baked', 'Pan-fried', and 'As needed grouper' children), and 'My processes'. On the right, the main area shows the 'Step Status' window for the 'Daily tuna' process. The 'General' tab is selected, showing a table of step details:

Sequence Number	Step	Status	Start Date	Step Type	Details	Agent	Elapsed Time	Return Code
1	Start a program	Completed	8/18/15 11:21:29 AM	Start a program	who	AGENT0	00:00:01	0
2	Move a file	Completed	8/18/15 1:50:07 PM	Start a program	who	AGENT0	00:00:01	0
3	Start a program							
4	Start a program							

A message box at the bottom right indicates: 'Process Catch enabled successfully'.

- A process must be **enabled** before it can run
- A process must be **disabled** before you can edit it

Running and monitoring processes

After you create a process, you must enable the process to run. When a process is enabled, you can start the process at any time, in addition to any instances that are started by triggers. While a process is enabled, you can view its properties, but it cannot be deleted or edited. To edit a process, it must be disabled.

To deactivate a process and disable it from running, select the process in the process library, and click **Disable**.

To activate a process and enable it to run, select the process in the process library, and click **Enable**.

To run a process, while it is enabled, click **Run now**. The process is immediately submitted to run.

After a process is submitted to run (either manually or by a trigger), you can review its history. To view the history of a process, select the process in the process library, and click the **History** tab in the process details. On the **History** tab, you see the instances of the process that were submitted. Each instance has a status. The status is described in [Table 6](#).

Table 6 Process status

Status	Description
Queued	The process was submitted, and is ready to run.
Running	The process started. One of the steps in the process is running.
Error	One of the steps in the process ended in error.
Completed	All of the steps in the process completed successfully.

Managing and monitoring process steps

The screenshot shows the 'Step Status' window with the following data:

Status	Start Date	Step Type	Details	Agent	Elapsed Time	Return Code
Completed	9/24/15 4:49:26 PM	Start a program	pstree 1	AGENT1	00:00:01	0
Completed	9/24/15 4:51:36 PM	Start a program	exit 0	AGENT1	00:00:01	0
Error	9/24/15 4:49:45 PM	Start a program	exit 1	AGENT1	00:00:01	1
Running	9/24/15 4:51:57 PM	Start a program	sleep 293	AGENT1	00:00:00	0
Queued	N/A	Start a program	systemctl	AGENT0	00:00:00	0

A context menu is open over the 'Running' step, listing the following options:

- Refresh
- View log
- Download log
- Rerun
- Stop

© Copyright IBM Corporation 2017

Managing and monitoring process steps

You can see the status of the steps within a process, and perform actions on the steps from the Step Status window. You can view the step status in one of the following ways.

- From the **process library** that contains the process, click **History**. On the **History** tab, select the process instance you want to review. Click **Details**. You see the Step Status window.
- From the **processes status overview**, select the process instance you want to review. Click **Details**. You see the Step Status window.

Using the Step Status window

From the Step Status window, you can review the status of each step in the process you select. In the status window, you see:

- The status of the step
- The time the step started
- The task type of the step
- Details about the task defined in the step
- Which agent runs the step
- The step's elapsed time
- The return code of the task completed by the step

From the Step Status window, you can perform one of several actions on a step. To perform an action, select the step in the Step Status window, and click the action. The actions you can perform are listed in [Table 7](#)

Table 7 Step status window actions

Action	Description
Refresh	Click Refresh to update the Step Status window
View Log	Click to view the output of the completed step
Download Log	Click to download the file that contains the output of the completed step
Rerun	Click to start the selected step again
Stop	Click to stop a step that is running on an agent

Click **Close** to dismiss the Step Status window.

Lesson 3 Using other Application Lab functions

Using other Application Lab functions

© Copyright IBM Corporation 2017

In this lesson, you learn how to use other features in the Application Lab. You learn how to:

- Use other process library features, such as exporting and importing processes
- Import cron and Windows task scheduler jobs
- Enhance workload applications built with the Application Lab

Exporting and importing process libraries

The screenshot shows two windows side-by-side. On the left is the 'PROCESS LIBRARY' window, which lists several process libraries: SK6, SP2, B2C, B2C, C2C, B2E, and SW1. A context menu is open over the B2C library, with 'Import' highlighted. On the right is the 'Import processes' dialog box. It has a dropdown 'Select the Process Library where to Import the data:' set to 'B2C'. Below it, there's a 'Browse' button, a text input field containing 'crontab.txt', and an 'Upload' button. Under 'Import for user:', there are three radio buttons: 'Specified in the file' (selected), 'Specify', and 'Username'. A 'Username' input field is also present. The main area contains two entries: 'Name: crontab 1' with the command '13 3 * * * /home/twsuser/bin/link_latest.sh' and 'Name: crontab 2' with the command '33 3 * * * systemctl'. Each entry has a 'Do not import' checkbox. At the bottom are 'Cancel' and 'Import' buttons.

© Copyright IBM Corporation 2017

Exporting and importing process libraries

To export a process, complete the following steps:

1. Right-click a process library.
2. Click **Export** from the menu.
3. Save the file.

To import a process, complete the following steps:

1. Right-click a process library.
2. Click **Import** from the menu.
3. Select the process library to accept the data.
4. Click **Browse** to find the file to import. You can choose a JSON file exported from the Application Lab, a text file exported from a crontab library, or an XML file exported from the Windows Task Scheduler.
5. Click **Upload** to verify the file and upload it.
6. Click **Import** to load the definitions.

Importing crontab and Windows task scheduler definitions

The screenshot shows the 'Import processes' dialog box. At the top, it says 'Import processes' and 'Select the Process Library where to Import the data: My processes'. Below that, it says 'You can import files from Cron, Windows Task Scheduler, and Application Lab.' There are three buttons: 'Browse' (highlighted), 'crontab.txt' (selected), and 'Upload'. Under 'Import for user', there are three options: 'Specified in the file' (radio button), 'Specify' (radio button selected), and 'Username' (radio button) followed by a text input field containing 'twsuser'. Two cron entries are listed:

- * Name: rmstldist Do not import Agent: AGENT0
- * Name: dumpsec Do not import Agent: AGENT0

At the bottom are 'Cancel' and 'Import' buttons.

Follow these steps to import task definitions:

1. Right-click a process library, and click **import** from the menu
2. Click **Browse** to find the file to import
3. Click **Upload** to transfer the file to the server
4. Complete the scheduler definitions
5. Click **Import**

© Copyright IBM Corporation 2017

Importing crontab and Windows task scheduler definitions

To import a crontab or Windows task scheduler definition, complete the following steps.

1. Right-click a process library.
2. Click **Import** from the menu.
3. Click **Browse** and find the file on your computer that contains the definitions to import.
4. Click **Upload** to transfer the file to the Workload Scheduler server.
5. You see the processes that are included in the file you transferred. Complete the fields that are described in [Table 8](#).

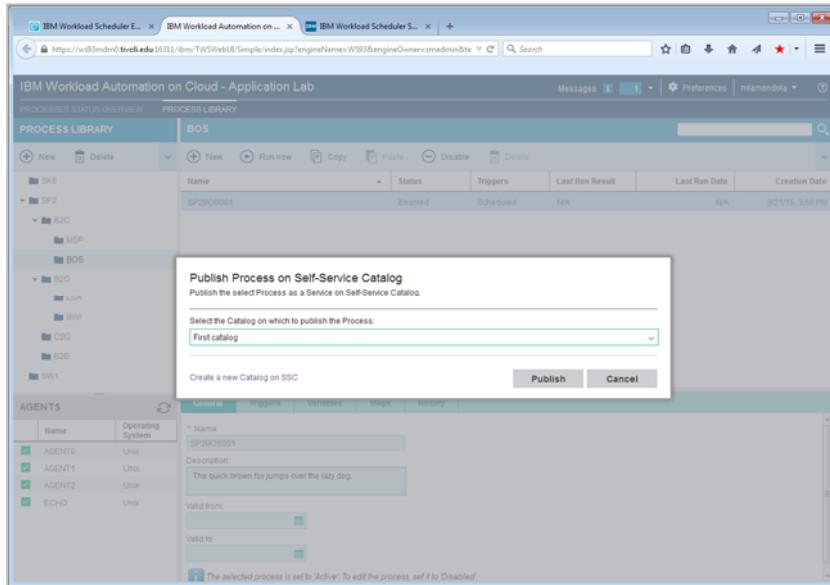
Table 8 Import processes fields

Field name	Description
Import for user	Specify the operating system user that will run the process on the agent you select.
Name	Enter the name that the process is given when the import is complete.
Do not import	Select Do not import if you do not want to import the process
Agent	Select which Workload Scheduler agent runs the process

6. Click **Import**. You see the message, All the processes have been successfully imported.

You can change the properties of the imported processes, and enable them to be started by Workload Scheduler.

Publishing a process to the Self-Service Catalog



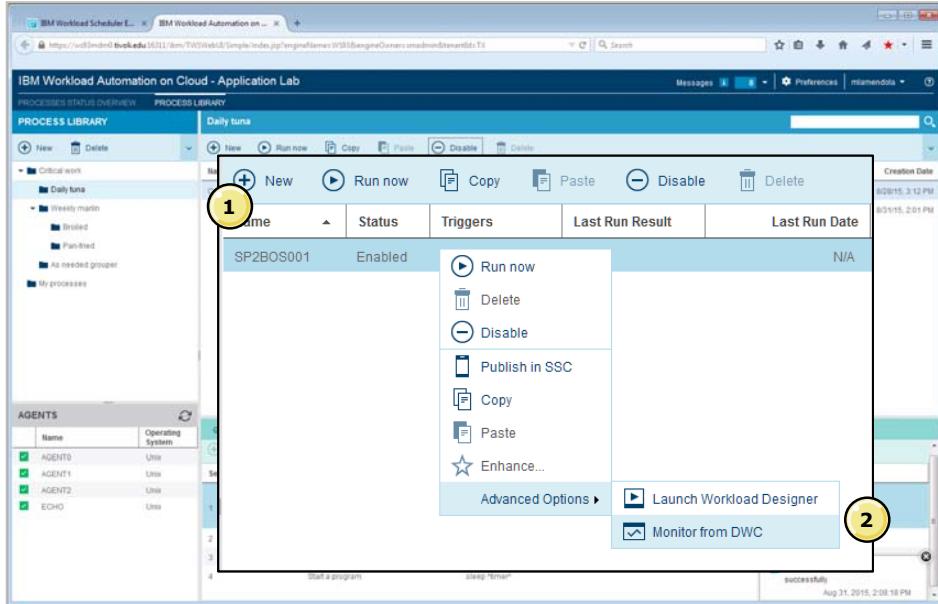
© Copyright IBM Corporation 2017

Publishing a process to the Self-Service Catalog

From the Application Lab, you can publish your process to a library that Self-Service Catalog users access from mobile devices. To publish a process, perform the following steps.

1. Highlight the process you want to publish in the Process Library.
2. For the process to be published, it must be enabled. If the process is disabled, click **Enable** to enable the process. Wait for the “Process enabled successfully” message in the messages window.
3. Right-click the process, and click **Publish in SSC**.
4. In the “Publish Process on Self-Service Catalog” window, select the catalog to contain the process from the menu. You can also click **Create a new Catalog in SSC** to open a new browser window to create a catalog to contain the process.
5. Click **Publish** to publish the process.

Managing a process by using the Dynamic Workload Console



© Copyright IBM Corporation 2017

Managing a process by using the Dynamic Workload Console

To see a process in the context of other workload, or manage steps in a more detailed way, you can monitor or edit an existing process by using the Dynamic Workload Console. To view or edit a process by using the Dynamic Workload Console, perform the following steps.

1. Highlight the process you want to publish in the Process Library.
2. Right-click the process, and select one of the following options to open the process.
 - a. Click **Advanced Options > Launch Workload Designer** to open the process in the Workload Designer.
 - b. Click **Advanced Options > Monitor from DWC** to open the Dynamic Workload Console, and start a job stream monitoring task that filters the list to match the process you selected.

Enhancing a process

Name the following objects

1. The job stream
2. The variable table
3. The jobs
4. The event rules, for event triggers

Click **Enhance** to move the process to the Workload Console

Click **Create a copy** to copy the process to the Workload Console while maintaining the existing Application Lab process

The screenshot shows the 'Enhance your process' dialog box. It has several input fields and lists. At the top, it says 'Specify the name of Dynamic Workload Console objects'. Below that is a note about job stream names. The main area contains:

- Job stream name:** SP2BOS001 (highlighted with a yellow circle labeled 1)
- Variable table name:** TX (highlighted with a yellow circle labeled 2)
- Step:** A list with four items (1, 2, 3, 4) each with a corresponding job definition name.
- Job definition name:** J09YR_UTQGS_31894, J7ANC_M0SBY_33570, J8YPR_POESY_28619, J0JST_BILKS_30025
- Job name:** J09YR_UTQGS_31894, J7ANC_M0SBY_33570, J8YPR_POESY_28619, J0JST_BILKS_30025
- Trigger:** File Created (highlighted with a yellow circle labeled 3)
- Event rule name:** BOS_FILE_TR (highlighted with a yellow circle labeled 4)

At the bottom are three buttons: Enhance, Create a copy, and Cancel. The 'Enhance' button is highlighted with a yellow circle labeled 5.

© Copyright IBM Corporation 2017

Enhancing a process

From the Application Lab, you can enhance your process to change it into definitions that you edit by using the Dynamic Workload Console. If your workload becomes more complex, or you want to integrate it with other complex workloads, you can enhance your processes. When you enhance a process, you change the process into a **job stream**, and the steps into **jobs**.

When you enhance a process, you change it in the following ways:

- You can run jobs in parallel, with dependencies between jobs.
- You can use the same jobs in many different job streams. You are not required to redefine the same script or command in more than one job.
- You can use more complex dependencies and conditions to start jobs or job streams, such as when other job streams complete successfully, or when a file is created on an agent's file system.
- You can create run cycles that start job streams respecting business calendars, working days, offsets, and exclusive days.
- You can manage and monitor jobs and job streams in views that show the dependency relationships graphically.
- You can manage and monitor jobs and job streams by using command-line interfaces.



Note: After you enhance a process, it can no longer be modified by using the Application Lab. The changes take effect immediately.

To enhance a process, perform the following steps.

1. Highlight the process you want to publish in the Process Library.
2. For the process to be published, it must be disabled. If the process is enabled, click **Disable** to disable the process. Wait for the “Process disabled successfully” message in the messages window.
3. Right-click the process, and click **Enhance**.
4. In the “Enhance your process” window, optionally, click **Learn more** to read about the benefits of enhancing a process, or **Watch a video tutorial** to see a video on YouTube.com about enhancing processes. You can also select **Do not show this again**, to automatically bypass the tutorial window when you enhance a process. Click **Continue**.
5. In the “Enhance your process” window, you must uniquely name the objects that become part of the database that you manage by using the Dynamic Workload Console. You name the following objects, if applicable:
 - The job stream
 - The variable table, if the process contained variables
 - The job definition names, and job names. If two steps have the same definition (they run the same command or script), there will be two steps that use the same job definition name, but different job names.
 - Event rules that trigger the job stream, if applicable.
6. Click **Enhance** to move the workload process to the Workload Console. Processes that are enhanced are no longer managed by the Application Lab. Optionally, click **Create a copy** to keep the process definition in the Application Lab, and a copy of it in the Dynamic Workload Console.

Exercises

- Open the Application Lab interface
- Follow the built-in tutorial
- Create a process library
- Create a process
- Add triggers, variables, and steps to the process
- Enable and run the process
- Monitor processes



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this unit.

Review questions

1. How many sublibraries can you create in an Application Lab Process Library?
2. Which users share your libraries and sublibraries?
3. How many triggers can you create for each process?
4. What types of event-based triggers can you create for a process?
5. What characteristic of an active process must you change before you edit the process?

Review answers

1. How many sublibraries can you create in an Application Lab Process Library?
You can create as many sublibraries as you need.
2. Which users share your libraries and sublibraries?
Users with the same tenant ID share libraries and sublibraries.
3. How many triggers can you create for each process?
You can create as many triggers as you need.
4. What types of event-based triggers can you create for a process?
*You can create **file created**, **file deleted**, and **file modified** triggers.*
5. What characteristic of an active process must you change before you edit the process?
You must disable a process before you can delete or edit the process.

Summary

You now should be able to perform the following tasks:

- Use the process status overview page
- Organize your work into process libraries and sublibraries
- Create a process, including steps, triggers, and variables
- Enable your process to run automatically or manually
- Enhance workload applications built by using the Application Lab
- Import scheduling definitions from cron or Windows Task Scheduler

© Copyright IBM Corporation 2017

Unit summary

Unit 3 Monitoring Workload Scheduler production

IBM Training



Monitoring Workload Scheduler production

© 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 use the Dynamic Workload Console and the `conman` command to monitor your system and the current production day plan.

Objectives

In this unit, you learn about monitoring Workload Scheduler production by using the Dynamic Workload Console and command-line interfaces. You learn how to perform the following tasks:

- Use the monitoring dashboard to see the production workload plan at a glance
- Customize your views and tasks in the Dynamic Workload Console
- Create and use monitoring tasks by using the Direct Query feature
- Create and use monitoring tasks for other scheduling objects and events
- Monitor plan objects, jobs, and job streams by using the conman interface
- Monitor the plan graphically by using the Plan View interface

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Monitoring Workload Scheduler production by using queries

Monitoring Workload Scheduler production by using queries

© Copyright IBM Corporation 2017

In this lesson, you learn how to use the Dynamic Workload Console to monitor your system and the current production day plan. By using the system status and health section of the Dynamic Workload Console, you can monitor the progress of the production plan. You use monitoring *tasks* in the console to view job streams, jobs, workstations, domains, resources, prompts, and file dependencies. You can use the Dynamic Workload Console or the command-line interface to specify plan modifications or to submit new workload into the plan.

You can create, modify, and use *tasks* for monitoring and managing the status and health of the scheduling environment. In this lesson, you learn how to create tasks to filter objects based on many criteria, such as name or status. You also learn how to use these filtered lists to retrieve specific information that is needed to manage workloads of different types.

Monitor workload

1. Select one or more engines
2. Select an object type
3. Type a query string
4. Click **Run**

QUERY STRING EXAMPLE

Workstation#JobStream.Job + filter
ALPHA#SP2@. @+STATE=WAIT,EXEC

- # Separates workstation and job stream
- . Separates job stream and job
- + Include filter parameters
- ~ Exclude filter parameters
- @ Wildcard for one or more characters
- ? Wildcard for one character

© Copyright IBM Corporation 2017

Monitor workload

A *monitoring task* is a filter that you run to search the plan for occurrences of the object instances that you select. From this filtered list, you click an item to view its properties or to run actions against it. By creating tasks that you customize, you can streamline your daily workflow and avoid unnecessary repetitive steps to monitor Workload Scheduler and perform actions. You can create or modify a monitoring task by using the Monitor Workload page.

Creating a monitoring task by using queries

To use the query page, click **System Status and Health > Monitor Workload**. Use this page to create or edit monitoring tasks for these kinds of object instances:

- Jobs
- Job streams
- Workstations
- File dependencies
- Resources
- Domains
- Prompts

You change the filter criteria by specifying the query text or selecting options from the editor interface.

To create a monitoring task, complete the fields on the page.

1. Select one or more engines on which the task runs from the **Engine** menu.
2. Select the type of object to monitor from the **Object Type** menu.
3. You can optionally choose to run the query from an alternative plan, such as a forecast or history archive. Choose an alternative plan from the **List Plans** menu.
4. In the **Query** field, type the query that contains the filter of the object type you selected. The query syntax is similar to the `conman show` commands syntax that is used in the command-line interface. If you are unfamiliar with the command-line utility syntax, click **Edit** and choose filtering criteria from the options available.



Hint: For example, to display the status of all jobs in wait or running status in the job streams with names that start with SP2 on workstation ALPHA, type the following string in the **Query** field:

`ALPHA#SP2@. @+STATE=WAIT, EXEC`

The syntax of the query string follows these conventions:

- The hash symbol (#) separates the workstation name from the job stream name.
- The at symbol (@) replaces one or more alphanumeric characters (wildcard).
- A question mark (?) replaces one alphanumeric character (wildcard).
- A period (.) separates the workstation and job stream names from the job name.
- Use the plus symbol (+) to include filter parameters.
- Use the tilde symbol (~) to exclude filter parameters.
- Separate filter options, such as states, with commas.

Click **Run** to start the task. You see the filter results on a new page. From the new page, you can edit the query or run it again. You can instead click **View as Report** to see the query in a new window that shows the query data in a format suitable for printing.

Editing a monitoring task

Click **Edit** to save the task you are creating, or to select filter criteria from a list of options rather than specifying them in the **Query** text box. Selecting options automatically refreshes the Query line with your selections. When you specify your filter criteria in the **Query** text box, the options on the user interface reflect the query that is specified.

The options that you specify for filtering depend on the type of object you want to monitor. You select options from the applicable sections to specify your filter criteria.

General filter

The general filter section contains fields for the periodic refresh rate and the name of the object. Job and job stream filters contain more options such as the status, priority, and rerun options.

Time data filter

The time data filter section, available for jobs and job streams, contains options for filtering the results by start and end times.

Dependencies filter

The dependencies filter section, available for jobs and job streams, contains options for filtering the results by other objects on which they depend. For example, you can create a filter for all jobs that are awaiting the arrival of a file on the file system.

Columns definition

You can change which columns are displayed in the result. Depending on the selection of filters for your query, you might want to display different columns. For example, if you filter for jobs that are waiting to run, the *Actual Start Time* and *Return Code* columns are blank. In this case, you might choose not to display those columns, and use *Not Satisfied Dependencies* and *Estimated Duration* instead.

In the columns definition section, select which columns to show in your direct query. Choose one or more columns you want to add to your query from the **Available Columns** list, and click **Add** to move them to the **Selected Columns** list.

To order the columns in your query, select a column in the **Selected Columns** list and click **Move Up** or **Move Down** to arrange the columns as you want.



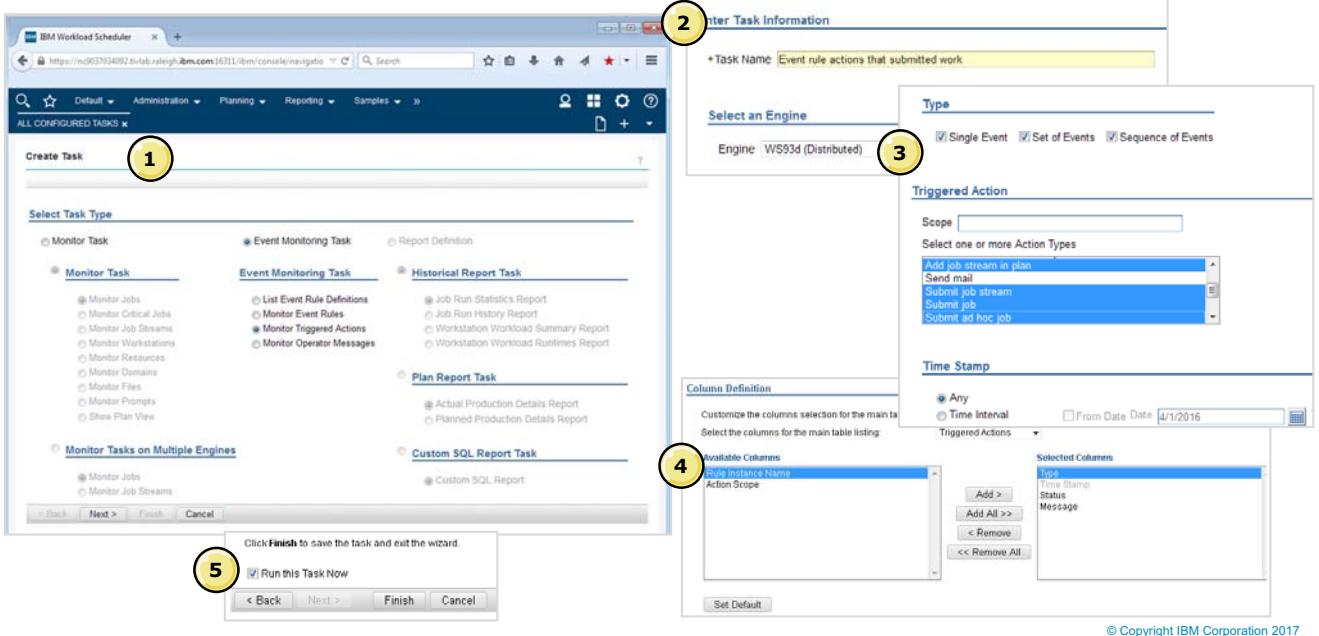
Hint: The first column on the list of **Available Columns** is fixed to the left of the resulting table. As you scroll the table horizontally, the leftmost column remains visible.

You can find descriptions of the columns in the help for the page. Click the question mark to see related help.

Saving your query

To permanently save your query, type a name for the task in the **Task Name** field, and click **Save**. To use a task you saved, click **System Status and Health > All Configured Tasks**, and choose your task from the **Task Name** column.

Creating and customizing all types of monitoring tasks



Creating and customizing all types of monitoring tasks

You can create and modify any type of monitoring task by using the methods described in this section. A few monitoring tasks that you need are not available on the Direct Query page. For example, Show Plan View and event management tasks are not available by using the Direct Query page.

Creating tasks

To create tasks, perform the following steps:

- Click **System Status and Health > All Configured Tasks**.
- On the All Configured Tasks page, click **New**. You see the **Create Task** dashboard with a selection of task types.



Hint: You can access the same dashboard by clicking **New** from the navigation bar, above the table of results in any Monitor task.

- Choose **Monitor Task**, **Event Management Task**, or **Monitor Tasks on Multiple Engines** depending on the type of task you want to create.
- Choose the type of task you want to create from the list that is highlighted.
- Click **Next**.
- Modify the values as required, clicking **Next** after each page. Click **Finish** to save the new task.

Selecting an engine

Each task is associated with a scheduling engine. By default, there is no assigned engine for each task. Instead, you are requested to select an engine each time you run the task.

Associate a task with one scheduling engine by selecting the name of the connection from the menu in the **Select an Engine** section while creating or modifying a task.



Hint: Associate each task in your Dynamic Workload Console with a scheduling engine connection to save the time that is required to select an engine every time you run the task.

You can also select an alternative plan to monitor. To use an alternative plan, click **Select**. Choose a plan from the Select Plan page, and click **OK**. For more information about other plan types, see “[Alternative plan types](#)” on page 6-60.

Sharing tasks

You can share each task that you create with other users of the Dynamic Workload Console. You share a task during the creating or customizing process.

In the **Task Sharing Options**, choose **None** to not share the task with other users. Choose **All** to share the task with any user who logs in to the Dynamic Workload Console. Choose **Selected** to select from a list of groups that are available for sharing. Click the group name to share with one group. Hold the Ctrl key to select more than one group. When the group or groups with which you want to share are highlighted, click **Add** to move your selected groups to the list of **Groups Selected for Sharing**.



Important: The Dynamic Workload Console administrator can disable task sharing. If someone shares a task with you, you cannot edit that task or remove it from your list of tasks. You can create a copy of the shared task and modify the copy.

Customizing an existing task

You can customize any task that you create, and any shared task you copy. To customize your tasks, perform the following steps:

- Click **System Status and Health > All Configured Tasks**.
- On the All Configured Tasks page, select a task by selecting the box in the leftmost column, and click **Task Properties**. The task opens and displays tabs on the left that refer to different categories of task properties.



Hint: You can access the same task window by running the task and clicking the **Task Properties** icon from the navigation bar, above the table of results.

- Click the tabs that for the properties that you want to customize.
- Modify the values and click **Save** to save the modified task.

Choosing columns to display

You change the columns in the monitoring task table by using the Columns Definition page. The default columns are already selected and included in the **Selected Columns** list.

Creating secondary queries

You run a *secondary query* when you click a hyperlink in the table of a monitoring task. For example, from the list of job streams in a Monitor Job Streams task, you run a secondary query to list the associated jobs by clicking the number in the Total jobs column.

While editing or creating a task, you can choose the columns to display in secondary queries. You can also customize these secondary queries from the Columns Definition page of any task. On the Columns Definition page of a Monitor Jobs task, you can choose the columns to display in job stream and workstation tasks. However, this column selection applies only to the lists of job streams and workstations that are obtained by drilling down from the results of the Monitor Jobs task. The column does not apply to any other generic monitoring task about job streams or workstations.

Using monitoring tasks

Job Type	Workstation (Job)	Job Stream	Predecessors...	Workstation (Job Stream)	Scheduled Time	Net Satisfied Dependencies	Priority	Job Number	Earliest Start	Actual Start	Deadline	
Successful SUCC	EXPORT_JOB_STAT	restful	AGENTO	S AGENTO	8/16/17 12:00 AM	0	10	564494306		8/16/17 4:21 PM E		
Successful SUCC	ELAB_JOB_STAT	spstorforecast	AGENTO	S AGENTO	8/16/17 12:00 AM	0	10	564494308		8/16/17 4:21 PM E		
Successful SUCC	IMPORT_JOB_STAT	restful	AGENTO	S AGENTO	8/16/17 12:00 AM	0	10	564494310		8/16/17 4:22 PM E		
Successful SUCC	JSLGA_GEOODE_35063	ECHO	Kill	ECHO	8/16/17 12:00 AM	0	10	564494309		8/16/17 4:21 PM E		
Running EXEC	JSLGA_JOYLL_35015	ECHO	Log...	Cancel	8/16/17 12:00 AM	0	10	564494312		8/16/17 4:29 PM E		
Running EXEC	JSLGA_JELLY_35115	UN	Dependencies...	Critical Path	ECHO	8/16/17 12:00 AM	0	10	564494313		8/16/17 4:29 PM E	
Running EXEC	JSLGA_KETCH_34908	UN	Release Dependencies	Confirm SUCC	ECHO	8/16/17 12:00 AM	0	10	564494314		8/16/17 4:29 PM E	
Running EXEC	JSLGA_LOWLY_35049	UN	Run...	Confirm ABEND	ECHO	8/16/17 12:00 AM	0	10	564494315		8/16/17 4:29 PM E	
Waiting HOLD	JSLGA_MOPAY_35095	UN	What if	Download Job Log	ECHO	8/16/17 12:00 AM	▲ 4	10				
Waiting HOLD	JSLGA_PIECE_34856	UN	More Actions	Properties...	ECHO	8/16/17 12:00 AM	▲ 1	10				
Waiting HOLD	JSLGA_PIXIE_34801	UNIX		Properties (New Window)	ECHO	8/16/17 12:00 AM	▲ 1	10				
Waiting HOLD	JSLGA_QUOTE_34835	UNIX		Show Run Instances	ECHO	8/16/17 12:00 AM	▲ 3	10				
Waiting HOLD	JSLGA_SEPUM_34886	UNIX		Job Run History	ECHO	8/16/17 12:00 AM	▲ 1	10				
				Show Spool List	ECHO	8/16/17 12:00 AM	▲ 1	10				
				Restart Process Chain	ECHO	8/16/17 12:00 AM	▲ 1	10				
				Percent Activity								

Using monitoring tasks

Regardless of the method that is used to create a monitoring task, you use the tasks in the same way. To open a monitoring task, select **All Configured Tasks** from the **System Status and Health** menu. From the list of available tasks, open a task by clicking the task name. Most monitoring tasks have properties in common.

Viewing your task as a report

You can run tasks as HTML-based reports. Select the task that you want to run by selecting the check box next to its name. Click **View as report**. A new browser window opens containing the results of the task filter that is displayed in report format. You can print the report by using the browser's print function.

Using breadcrumbs

A list of pages that are located horizontally above the table shows your navigation path. The page that you are currently viewing is shown in blue. You can use this list to switch between the views you have open. Every step can be expanded to view its history.

All open tasks of the same type are listed under **Active Tasks**. By clicking **Active Tasks**, you see the tasks of the same type that you started. If, for example, you open several Monitor jobs tasks, you can switch between them by selecting them from the list of active tasks. The displayed table is

automatically updated to show the results of the selected task. There are three types of navigation bar icons you can use:

- Click the **History** icon () to open a chronological view of all the pages you have open (pages that contain task results are in bold).
- Click the **Task Properties** icon () to customize the type of information that is displayed in the current task. Click the icon to open the task definition, where you can change the task's filters and columns. Changes that you make to column definitions take effect immediately; to have other changes take effect, click the **Refresh** icon.

Your changes apply only to the current task. They have no impact on other tasks of the same type.

- Click the **Close** icon () to close currently open tasks and their details views. The close icon, which is located next to **Active Tasks**, closes all the tasks of the same type that are currently open (indicated by the number in parentheses). The **Close** icon located next to a task name, closes the currently displayed task and its details views.

Table toolbar

Icons in the toolbar at the top of monitoring task tables are used to select items, set or change the refresh interval, and perform other actions:

Table 1 Monitoring task tools

Tool	Description
	Click the Refresh icon to refresh the contents of the table.
	Click the Set Auto Refresh icon to toggle the auto refresh tool.
	Pause, stop, or resume the refresh countdown by clicking the appropriate icon. When the counter reaches zero, a new query starts and the view is refreshed. If you reset the countdown, it restarts from the beginning. These icons are only available if autorefresh options are specified in the task filter definition.
	Click the Select-all icon to select all items in the table.
	Click the Deselect-all icon to clear the current selection.

Table 1 Monitoring task tools

Tool	Description
	Click Sort to enable advanced sorting on multiple columns. You can specify up to 3 columns to sort, in priority order. The data in the table is refreshed, with the order of the rows that are based on the new sort settings. Icons in the table headings indicate which columns are being used for sorting, and the direction of the sort. Press OK to save the sort setting.
	Click Search to search for items in the table. Specify the text to search and the column that must contain the text. Select the direction and click Find Next to begin the search. The first item that matches the search is highlighted and selected. Each time that you click Find Next , the following item that matches the search is highlighted and selected. Only one item at a time is selected. You can use wildcard characters by entering a question mark (?) for any single character, and asterisk (*) for zero or more characters.
	Click the Add Link icon to save the task in your browser favorites. The Add Link icon is only available in some tasks. When you click the bookmark link from your browser, you directly open the task that you bookmarked after you log in to the Dynamic Workload Console. When adding a link in Firefox, make sure that you clear the Load this bookmark in the sidebar check box if you do not want the link page to be shown in the sidebar of the window.
	In Monitor tasks on multiple engines views, click Statistics to view connection status information and statistical information about the engines that are defined in the task. A pie chart that shows the number of query results and job status is displayed for each engine on which the task successfully runs. Click the pie sections to see more details. If the task did not successfully run on one or more engines, a message informs you providing details about the errors. The statistics view is only available for Monitor tasks on multiple engines.
	You can dynamically filter the results in the table to items that contain text you input. You can select either all columns (default selection) or one single column to match the text you input. Click the down arrow to expand the list of columns to apply the filter, and select a column from the list. Enter a string in the field and click the button to show the filtered list. As a result, the table displays only the items that contain the specified string in the selected column. You can use wildcard characters by entering a question mark (?) for any single character and asterisk (*) for zero or more characters.



Hint: You can run multiple tasks at the same time and drill down into the details of various objects from menu selections on the table.

Table footer

At the bottom of the table, you can specify the number of lines to be displayed for each page by changing the **Lines per page** field. The value that you set applies to the current table.

If the number of items in the filter is greater than the lines per page settings, you change pages by clicking **Go to previous page** (<<) or **Go to next page** (>>). Jump directly to a page by typing the page number into page number field and pressing Enter.

To change the value globally, open **System Configuration > Set User Preferences**.

Table cells

You can perform several actions within table cells:

- You can temporarily sort the table by a column by clicking the arrow at the right of the column header. Clicking the carat (^) icon sorts the table by that column in ascending order. Clicking the carat (^) icon again sorts the table by that column in descending order.
- You can change the width of a column by dragging the column header border to the left or to the right.
- Double-click the border of a column header to automatically resize the column width to match the width of the contents of that column.
- Right-click an object to display the list of available actions that you can take on that object. A context-sensitive menu shows actions that you can use.

Lesson 2 Using the Workload Dashboard

Using the Workload Dashboard

© Copyright IBM Corporation 2017

In the Workload Dashboard, you see a consolidated view for monitoring workload status across one or more IBM Workload Scheduler networks. In this lesson, you learn how to use the Workload Dashboard that is part of the Dynamic Workload Console.

IBM Training

Workload Dashboard overview

Select an engine, or All engines

Jobs status

Status	Count
Blocked	~100
Canceled	~100
Completed	~100
Error	~100
Held	~100
Ready	~100
Running	~100
Suppressed by Co.	~100
Undecided	~100
Waiting	~9000

Jobs in error

Late Jobs

Min Duration

Max Duration

Plan Progress

Log Messages

Operator messages

Links to monitoring task

Each widget has a menu

© Copyright IBM Corporation 2017

Workload Dashboard overview

The workload dashboard is built by using features in the IBM Dashboard Application Services Hub, along with data sets provided by IBM Workload Scheduler. In addition to the provided Workload Dashboard, you can create your own customized dashboard by using predefined or customized widgets.

The workload dashboard contains widgets that show the results of the most commonly used queries. Each widget links to a monitoring task that contains information that relates to the widget's topic.

The widgets link to a table of results. When **All engines** is selected from the **Engines** widget, the widgets accumulate the values of the queries of all the engines that are defined in the Dynamic Workload Console, and open **Multiple Engine** monitoring tasks when you click them.

You can enhance the view of a widget in the following ways.

- To expand a widget to the full size of the Workload Console, click the menu () in the upper-left corner of the widget, and select **Maximize**.
- To restore a widget to its original size, click the menu, and select **Restore**.
- To update the data that is displayed in the widget, click the menu, and click **Refresh**.
- To read helpful information about the widget, click “?”.

The **Engines** widget shows the engines that are defined in the Dynamic Workload Console. Select an engine for the monitoring widgets to reflect, or **All engines**. Right-click an engine on the list to view properties for that engine.

The following section describes the widgets that you see on the workload dashboard by default.

Workstations

The *unavailable workstations* widget shows the number of workstations that are not available. Click to view a **Monitor Workstations** task for a single engine.

The available workstations widget shows the number of workstations that are available. Click to view a **Monitor Workstations** task for a single engine.

Prompts

The *prompts* widget shows the number of asked prompts for the selected engine. A prompt is *asked* when an operator reply is requested. Click to view a **Monitor Prompts** task for a single engine.

Job status

View the **Job status** widget to see how many jobs for the selected engine are in each status. The bars in the job status chart link to a **Monitor Jobs** task that contains list of jobs that match the status.

View the **Critical status** widget to see how many jobs are at each risk level. The bars in the critical status chart link to a **Monitor Critical Jobs** task that contains a list of jobs that match the risk level. You can run this query on a single engine; it is not supported for multiple engines.

Other job-related widgets show how many jobs meet the following criteria.

- The **Jobs in error** widget shows how many jobs ended with an error condition.
- The **Late jobs** widget shows how many jobs completed after their designated deadline.
- The **Min Duration** widget shows how many jobs completed before they reached their defined minimum duration.
- The **Max Duration** widget shows how many jobs completed after they ran longer than their defined maximum durations.

Click a job status widget to view a **Monitor Jobs** task that shows the jobs in plan that met the widget's query condition.

Plan information

The **Plan Progress** widget shows the overall progress of the production plan.

The **Log Messages** widget shows operator log messages for the selected engine. You can double-click a message to see details about the message.



Hint: To post messages into the **Log Messages** widget, create an event rule that performs the **Message logger** action.

Lesson 3 Monitoring system status and health

Monitoring system status and health

© Copyright IBM Corporation 2017

In this lesson, you learn how to use the Dynamic Workload Console to perform the following tasks:

- Monitor system status and health
- Monitor dynamic agent computers
- Monitor the scheduling environment by using the Dynamic Workload Console
- Monitor the scheduling environment by using the conman command

You can monitor information about workstations:

- Link status
- Workstation
- Agent running
- Writer running
- Start time
- Run number
- Limit
- Domain
- Type
- Agent initialized
- Fence
- Node Type
- Version
- Time zone
- Node name
- Node port
- Node information
- Behind firewall
- SSL communication
- SSL port
- Host
- Method
- Event processor running
- Event monitoring
- Monitoring configuration updated

Monitoring your scheduling environment

You monitor and manage Workload Scheduler workstations in the plan by using the Monitor Workstations pages. From Monitor Workstations pages, you can see whether workstations are started or stopped, linked, or unlinked. You can see the *limit* and *fence* of each workstation, and link directly to a predefined task of job streams or jobs that are scheduled on that workstation.

Viewing workstation status

The **Monitor Workstations** task displays properties of workstations in the production plan. Choose **System Status and Health > Monitor Workload** to open a direct query page, and choose **Workstations** from the **Object Type** menu. Click **Run** to monitor a list of workstations. You can see a list of columns and their descriptions by opening the help for the task page. Click the question mark to see help.

The **Node Type** field consists of two parts, the operating system class and the workstation type. The operating type is one of the types that are listed in the following table.

Table 2 Operating system types

Type	Description
UNIX	UNIX or Linux
WNT	Microsoft Windows
ZOS	IBM System z
IBMI	IBM System i
OTHER	A type that none of the other types describe

Workstation types are listed in the following table.

Table 3 Workstation types

Type	Description
AGENT	Dynamic agent
D-POOL	Dynamic pool
FTA	Fault Tolerant Agent
MANAGER	Domain Manager
MASTER	Master Domain Manager
POOL	Pool
REM-ENG	Remote engine
S-AGENT	Standard agent
X-AGENT	Extended agent

Viewing Dynamic Workload Broker computers

1 System status and health > Monitor Computers on Broker

2 Search

3

© Copyright IBM Corporation 2017

Viewing Dynamic Workload Broker computers

To view status of Dynamic Agents in the Dynamic Workload Broker database, choose **System Status and Health > Monitor Computers on Broker** to open a search criteria page. To view all computers, leave all fields blank and click **Search**.



Hint: You can view many details about the computer systems that run dynamic agents, such as the available memory, disk space, and CPU usage.

The Computer Search Results page displays a list of computers where the Workload Scheduler agent is installed. From the view, you can view additional information about a computer.

When you click the computer display name link in the **Computer Search Results** table, you see the Computer Details page, containing detailed information about the computer. On the page, you see the characteristics of the computer, along with its associated groups, logical resources, file systems, and networks.

To hide the information about the selected computer, click **Hide details**.

Using the SHOWCPUS command

```
[wsuser@ws94mdm0 ~]$ conman
Scheduled for (Exp) 09/12/17 (#101) on MDM0. Batchman LIVES. Limit: 11, Fence: 0, Audit Level: 1
%showcpus
CPUID          RUN NODE      LIMIT FENCE DATE    TIME     STATE      METHOD      DOMAIN
MDM0           101 *UNIX MASTER  11     0 09/12/17 11:15   I J  M EA  MASTERDM
AGENT0         101 UNIX AGENT  11     0 09/12/17 11:15   LBI J  M  MASTERDM
ALPHA          101 UNIX D-POOL 11     0 09/12/17 11:15   LBI J  M  MASTERDM
BRAVO          101 UNIX D-POOL 11     0 09/12/17 11:15   LBI J  M  MASTERDM
BROKER0        101 OTHR BROKER 11     0 09/12/17 11:15   LTI JW  M  MASTERDM
CHARLIE        101 UNIX D-POOL 11     0 09/12/17 11:15   LBI J  M  MASTERDM
ECHO            101 UNIX D-POOL 11     0 09/12/17 11:15   LBI J  M  MASTERDM
MASTERAGENTS   101 OTHR POOL   11     0 09/12/17 11:15   LBI J  M  MASTERDM
%showcpus;info
CPUID          VERSION TIME ZONE          INFO
MDM0           9.4.0.01 America/New_York Linux 3.10.0-514.26.2. #1 SM
AGENT0         9.4.0.01 America/New_York OpenStack
ALPHA          9.4.0.01 America/New_York Dynamic Pool
BRAVO          9.4.0.01 America/New_York Dynamic Pool
BROKER0        9.4.0.01 America/New_York Dynamic Workload Broker
CHARLIE        9.4.0.01 America/New_York Dynamic Pool
ECHO            9.4.0.01 America/New_York Dynamic Pool
MASTERAGENTS   9.4.0.01 America/New_York Pool
%
```

© Copyright IBM Corporation 2017

Using the SHOWCPUS command

Use the conman SHOWCPUS (SC) command to show information about workstations and links. You can use wildcards to show information about all workstations or groups of them.



Note: SHOWCPUS Syntax

```
{showcpus | sc} [[domain!]workstation]
[;info|;link]
[;offline]
{showcpus | sc} [[domain!]workstation] [;getmon]
```

Here are examples of lists that you can run:

- To list all workstations with names that start in A: SHOWCPUS A@
- To list all workstations in the CORP_DM domain: SC CORP_DM!@

When you run the conman SHOWCPUS command, you see the columns that are listed in the following table.

CPUID	The name of the workstation.
RUN	The run number of the current plan that is running on the workstation. Run numbers are used to synchronize the workstations in a Workload Scheduler network. The run number is incremented when the production plan is extended, typically once per day.
NODE	The node type and workstation type.
LIMIT	The total number of jobs that can run concurrently on this workstation. The limit can be set to any number between zero and 1,024. A limit of System indicates that no limit is placed on the number of jobs that can run concurrently on this workstation.
FENCE	The workstation fence.
DATE, TIME	The date and time the workstation was initialized with a new production plan.
METHOD	For extended agents only, the name of the access method that is specified in the workstation definition.
DOMAIN	The name of the Workload Scheduler domain to which the workstation belongs.
STATE	Status information about the workstation's links, servers, and processes. The states of each column are shown in "Workstation link information" on page A-4.

The STATE column shows the status of workstations by displaying up to nine letters to denote the status or condition of the workstation.

When you use the ;INFO flag to the SHOWCPUS command, you see different columns in the output, so that you can see operating system information such as the time zone. Column descriptions are shown in [Table 3](#) on page A-4.

When you use the ;LINK flag to the SHOWCPUS command, you see information that is related to the network configuration for the workstations. The columns that you see when you use the LINK format are listed in the [Table 4](#) on page A-4.

When you use the ;GETMON flag to the SHOWCPUS command, you see a list of event rules that are defined for the monitor that runs on each of the specified workstations. The output contains the rule name, event provider, and the time at which the rule was last generated.

Lesson 4 Monitoring plan objects

Monitoring plan objects

© Copyright IBM Corporation 2017

In this lesson, you learn about monitoring jobs, job streams, and other scheduling objects that are related to the production plan.

You learn how to perform the following tasks:

- Monitor plan objects by using the Dynamic Workload Console or `conman` command
- Monitor job streams by using the Dynamic Workload Console or `conman` command
- Monitor jobs by using the Dynamic Workload Console or `conman` command
- Perform other job-related functions by using the `conman` command
- Monitor critical jobs and the critical path

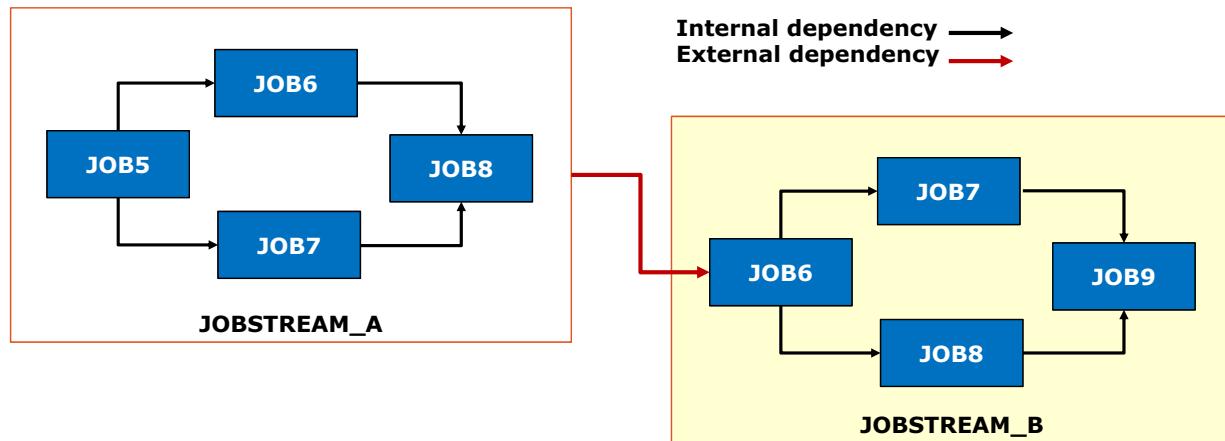
You can view and manage scheduling object instances in the plan by using the Dynamic Workload Console. These scheduling object instances are a subset of all the scheduling objects in the database.

Controlling processing by using dependencies

The correct order of workload processing is critical to business operations. Some tasks cannot start until others complete successfully. Certain tasks must be at the beginning of a sequence. Similarly, some tasks need specific system resources to run correctly. Any condition or prerequisite that must be satisfied before a job or job stream can begin processing is called a *dependency*. For example, when you are running a daily backup of the production database, you must ensure that work successfully switches to an alternative database before you begin your backup. You also must ensure that this activity starts at a set time so that it does not disrupt other critical tasks.

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

Using internal and external dependencies



Using internal and external dependencies

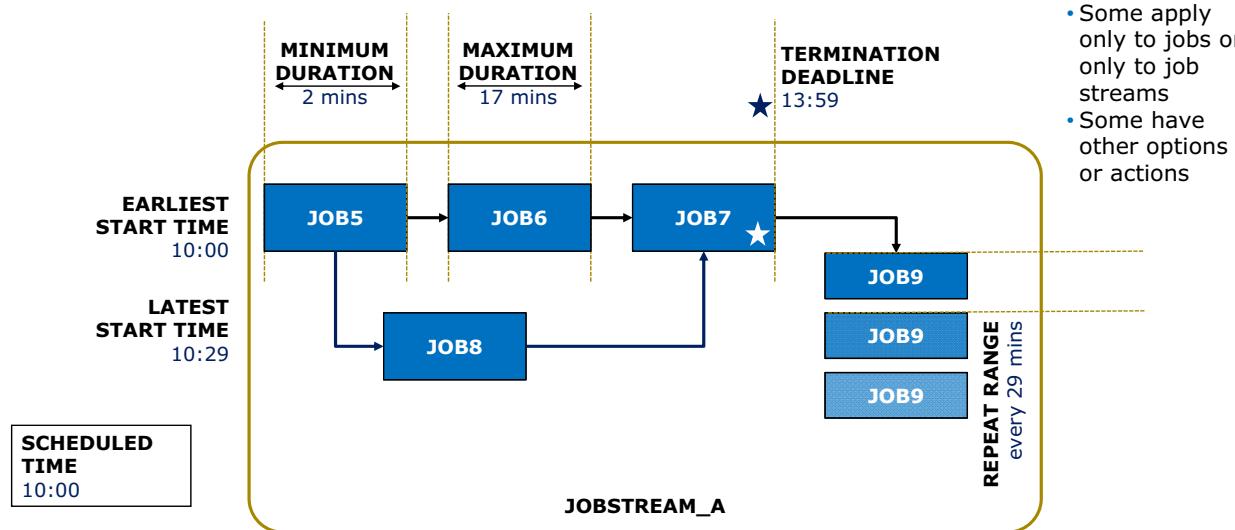
You can have dependencies between jobs, between job streams, or between jobs and job streams. They can be:

- **Internal dependencies:** Dependencies between jobs that belong to the same job stream
- **External dependencies:** Dependencies between job streams, between job streams and jobs that belong to other job streams, or between jobs that belong to different job streams

When the running order of the scheduled tasks is established, the scheduler defines predecessors and successors for the jobs or job streams. Workload Scheduler automatically manages predecessor and successor relationships. These and other dependencies establish the required processing order of the workload.

Other types of dependencies can be used to hold jobs from starting. You can use dependencies that are based on other factors such as the time of day, process concurrency, or arrival of a file on the computer file system.

Evaluating time restrictions



© Copyright IBM Corporation 2017

Evaluating time restrictions

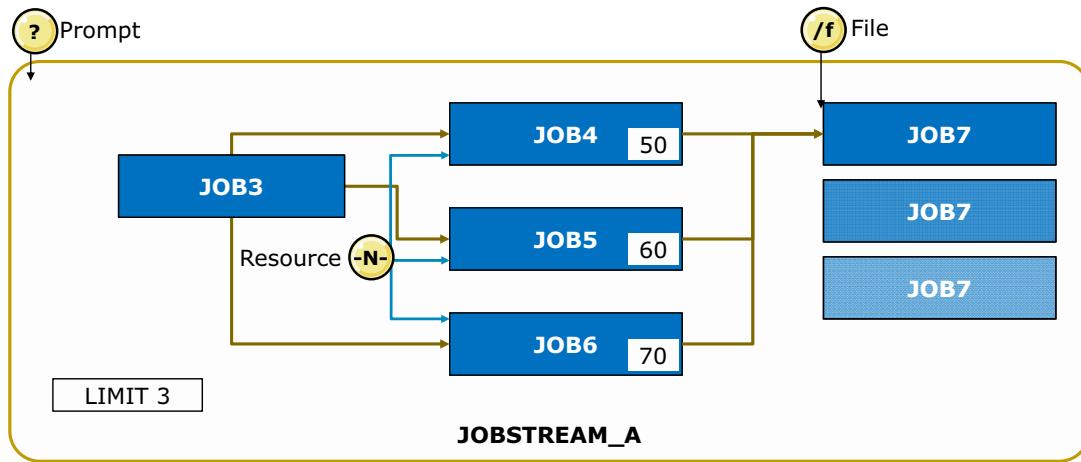
Several types of time constraints determine when jobs can begin. Time restrictions are available at the job or job stream level, and repetition rates can be set on jobs.

The types of time restrictions that are available are listed in the following table.

Earliest start time	A job or job stream cannot start before the earliest start time.
Latest start time	The latest time a job or job stream can start processing. In other words, the job or job stream must start before this time. You can also see an action that Workload Scheduler takes when the latest start time is reached before the job or job stream starts. You can use one of the following actions: Suppress : The job or job stream, and any other dependent jobs or job streams do not run when after the latest start time is reached. Suppress is the default. Continue : The job or job stream runs when all the defined conditions are met and event rules that match the latest start time are triggered. Cancel : The job or job stream is canceled. All dependencies are released so that any other jobs or job streams that depended on this one (for example, successors) can run. The side effects of canceling a job or job stream are described in "Canceling jobs and job streams" on page 4-24.
Termination deadline	The latest time by which a job or job stream must complete.

Maximum duration	Specifies the maximum amount of time that a job can be running. The scheduler can specify an action to start if the job run runs longer than the specified amount of time. Possible actions are listed here: Continue: The workload running continues (the default). Kill: The job is set to error status and ended.
Minimum duration	Specifies the minimum amount of time that a job must last. The scheduler can specify an action to start if the job run ends before the minimum duration is reached. Possible actions are as follows: Continue: The workload that is running continues. Confirm: The job is set to Confirm status. The workload requires a user confirmation to proceed. Abend: The job is set to Error status.
Scheduled time (Planned Start Time)	The scheduled time is not necessarily a time dependency. It represents the time when the job stream is positioned in the plan. The scheduled time does not prevent the job stream from processing. It uniquely identifies a job stream instance in the plan, when multiple instances of the job stream exist for a single day. For example, job streams might run twice in a 24-hour period. Time restrictions can also be combined. For example, when both an earliest start time and a latest start time for a job or job stream are defined, a time frame within which processing must start is created.
Repeat range	Jobs can be set to run at a repeating interval. A repetition rate or interval causes the job to be started repeatedly at the specified interval after all other dependencies on the job are solved.

Using other dependencies



© Copyright IBM Corporation 2017

Using other dependencies

Monitoring dependencies

You can monitor dependencies by focusing on the type of dependency, by using specific tasks. For example, from a Monitor Files task, you see a list of the files in the production plan that are used as dependencies.

You can also monitor dependencies by focusing on the jobs or job streams. For example, from a Monitor Jobs task, you see a list of dependencies that prevent a job from starting.

Using prompts

Prompts are text messages that are presented to the operator to pause job or job stream processing. Processing continues when an affirmative answer is given, either manually from the operator or automatically by an event rule action. You can use prompts as dependencies in jobs and job streams. You can also use prompts to alert an operator that a specific task was performed. In this case, an operator response is not required.



Note: Prompts are monitored in the Dynamic Workload Console by using Monitor Prompts tasks.

The three types of prompts are listed in the following table.

Global or named (predefined)	A prompt that is predefined in the database as a scheduling object. A unique name identifies the prompt, and any job or job stream can use it as a dependency. All jobs or job streams that use the same named prompt as a dependency has the dependency met by a single affirmative reply.
Local or ad hoc	A prompt that is defined within a job or job stream definition. Therefore, it does not have a name, and it is not defined as a scheduling object in the database. Other jobs or job streams cannot share the prompt as a dependency.
Recovery or abend	A special type of prompt that you define to be used when a job ends abnormally. The response to this prompt determines the outcome of the job or job stream to which the job belongs. A recovery prompt can also be associated to an action and to a special type of job that is called a recovery job.

Monitoring prompts

To manage and monitor prompts in the production plan, go to **System Status and Health > Monitor prompts**. From a Monitor Prompts task, you can see a list of prompts that matched your selection criteria. The table of results is arranged in columns by name, status, and other categories. The prompts table shows the dependency name, with a link to the dependent jobs, and the status of the prompt. Prompts can have one of the statuses that are listed in the following table.

Asked	The prompt was issued, and jobs or job streams are awaiting a reply to this prompt, but no response was received.
Not Asked	The prompt was not issued. Jobs or job streams use this prompt as a dependency, but other dependencies (such as earliest start time or file dependency) also restrict the dependent job or job stream.
Yes	The prompt was issued and a YES response was received.
No	The prompt was issued and a NO response was received.

Using resources

You can control task processing by defining and associating resources to workstations. These resources are used as dependencies for job and job stream processing for the workstation to which they are associated. A job or job stream with a resource dependency cannot start to run until the required quantity of the defined resource is available. Resources represent logical scheduling entities. You define resources to introduce pauses in the logical flow of your tasks. For example, if two or more jobs require one tape unit to be available, you can define a resource that is called **tapes** and assign it a value of 1. When you associate this resource as a dependency to one or more jobs, if both jobs request access to the resource, it is assigned to the job or job stream with the highest priority. When the job that is allocating the resource ends, the resource is released and the next job can use it. Defining resources helps you control concurrent processing.

Up to 32 jobs can use the same resource concurrently. Jobs and job streams logically bind resources. After the job or job stream completes with a status of SUCC, ABEND, or PEND, the resource is returned to the pool.

You can also define resources to avoid shortages of limited system resources. If a job on a critical system requires a fixed property to run successfully, you can associate a logical resource dependency to that job. In this way, no other job runs at the same time and causes a supply shortage.

Monitoring resources

To manage and monitor resources in the production plan, go to **System Status and Health > Monitor resources**. From a Monitor resources task, you can see a list of resources that matched your selection criteria. The table of results is arranged in columns by name, status, and other categories. The resource table shows the dependency name, and the status of the resource. The number of defined and available resources and the number of resources currently being used is shown. Resources can have one of the statuses that are listed in the following table.

Available	The resource has enough units available to service the dependent jobs or job streams.
Not Available	The resource is unavailable. Not enough units are available.
Unknown	The status of the resource is unknown.

Using files

A file can be used as a dependency for jobs and job streams. A job or job stream with a file dependency cannot start to run until the file exists with the characteristics defined in the dependency.

You use file dependencies to control job and job stream processing that is based on the existence of one or more files or directories. When you specify a file dependency, Workload Scheduler checks if the specified file or directory exists before processing begins. You can select several conditions that are associated to the file that must be true for processing to begin.

Monitoring files

To manage and monitor files that are used as dependencies in the production plan, go to **System Status and Health > Monitor files**. From a Monitor files task, you can see a list of files that matched your selection criteria. The table of results is arranged in columns by name, status, and other categories. From the table that shows the files, you can click **Jobs** to list all the jobs that have this file as a dependency or **Job Streams** to list all the job streams that have this file as a dependency.

The files table shows the dependency name, the qualifiers of the dependency, a link to the workstation, and the status of the file dependency. Files can have one of the statuses that are listed in the following table.

Not checked	Workload Scheduler did not verify that the file exists. Workload Scheduler does not check file dependencies until other dependencies, such as earliest start time, are met.
Exists	Workload Scheduler verified that the file exists. Soon after the dependency resolution, it returns into Not Checked status until next check.
Checking	Workload Scheduler is checking whether the file exists
Does not exist	The file is not available.

Using priorities and limits

Workload Scheduler has its own internal queuing mechanism, which uses levels of priority. You can assign a specific priority to a job or job stream to control its precedence and order of processing. Another level of control over processing is provided by the job fence. When you set the job fence value on a workstation, this value determines the priority level above which a job is allowed to run. Any job with a priority less than or equal to the value of the fence is not started on the workstation. Additionally, you can set limits to the maximum number of jobs that can be running simultaneously within a job stream or for a workstation. You can set a job stream or workstation limit, or both. Combining two or more of these attributes provides a method of controlling processing on given systems and to avoid resource contention or saturation.

You can set a priority for a job or job stream to determine which one starts first, when all dependencies are solved. The following rules determine the order of processing:

- The job stream with the highest priority is the first one to be started.
- Within a job stream with the highest priority, the job with the highest priority is started first.
- Resources that are used as dependencies are not allocated to the dependent object until all other dependencies are met and the job is ready to run.

Jobs on a workstation are not started if their priorities are less than or equal to the workstation's job fence value. You set a job fence to prevent low priority jobs from being started, regardless of the priorities of their job streams. When you assign a fence value, you can prevent all low-priority jobs from running even if they belong to a high-priority job stream. If you set the workstation's fence to GO, no jobs are started on that workstation, regardless of the job's priority.

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

You can set a workstation limit to define the maximum number of jobs that can run simultaneously on a workstation. If the workstation limit is set to zero, only jobs that have a priority of GO (value of 101) are started.

Monitoring scheduling objects with conman

```
% showprompts
State Message or Prompt
ASKED 1(STICKY) Reply to this prompt to start the dependent job streams.
INACT 2(EXPPROM) !Prompts can start with exclamations
YES 3(COMMENT) :Prompts can produce comments
INACT 4(BRANCH1) Selector branch one
% showresources charlie#@
CPU#Resource      Total Available   Qty UsedBy
CHARLIE#COMPUTE      1           1           No holders of this resource
% showfiles
Exists File Name
NO     MDM0#/tmp/jaxdat.dat
%
```

© Copyright IBM Corporation 2017

Monitoring scheduling objects with conman

You can use `conman SHOW` commands to display information about prompts, resources, and file dependencies in the plan. You can show information about all objects, or groups of them by using wildcards. Following are the `conman` commands that you can use to monitor scheduling objects:

- `SHOWPROMPTS (SP)`: Show status of prompts in the plan
- `SHOWRESOURCES (SR)`: Show status of resources in the plan
- `SHOWFILES (SF)`: Show status of file dependencies in the plan

Displaying the status of prompts

Use the `conman showprompts` command to see a list of prompts, and information about them. Following is the syntax of the `conman SHOWPROMPTS` command.

SHOWPROMPTS

```
{showprompts | sp} [promptname | [workstation#] msgnum] [;state[;...]]
[;keys]
[;offline]
{showprompts | sp} [promptname | [workstation#] msgnum] [;state[;...]]
[;deps[;keys | info | logon]] [;offline]
```

Displaying the status of resources

Following is the syntax of the `conman SHOWRESOURCES` command.

```
SHOWRESOURCES
{showresources | sr} [[workstation#] resourcename]
[;keys]
[;offline]
{showresources | sr} [[workstation#] resourcename]
[;deps[;keys | info | logon]]
[;offline]
```

When you use the `conman showresources` command, you see a list of resources, with the quantities in use and available and the names of any jobs or job streams that hold resources in use.

Displaying the status of file dependencies

Following is the syntax of the `conman SHOWFILES` command.

```
SHOWFILES
{showfiles | sf} [[workstation#] file]
[;state[;...]]
[;keys]
[;offline]
{showfiles | sf} [[workstation#] file]
[;state[;...]]
[;deps[;keys | info | logon]]
[;offline]
```

When you use the `conman showfiles` command, you see a list of file dependencies, and their status.

Monitoring job streams

Default columns

- Status
- Internal status
- Job stream name
- Workstation name
- Scheduled time
- Number of unsatisfied dependencies
- Total number of jobs
- Number of successful jobs
- Jobs limit
- Job stream priority

- Earliest start time
- Actual start time
- Deadline

Other columns you can add

- Latest start time
- Estimated duration
- Actual duration
- Original job stream name
- Internal identifier
- Additional information
- Dependencies

Monitoring job streams

© Copyright IBM Corporation 2017

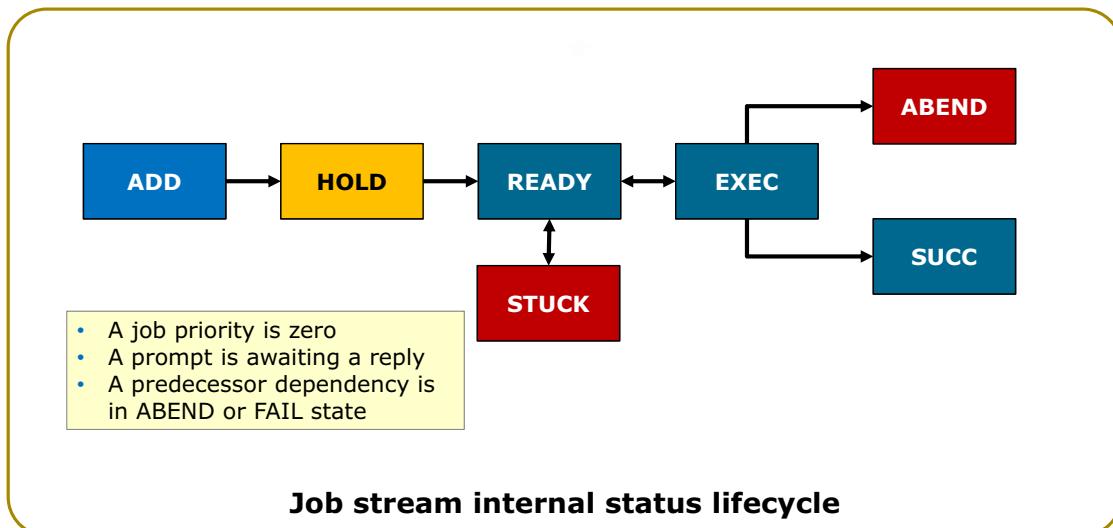
Monitoring job streams in the plan gives you the capability to see the status of workloads in progress in your production environment at a high level. You can, for example, view a task that shows all the job streams that are waiting or in error. The advantage of monitoring job streams over monitoring jobs is that you can view more of your overall workload at a glance. From the **Monitor job streams** tasks, you can quickly browse to jobs-related tasks to view or manage individual jobs.

To manage and monitor job streams in the production plan, go to **System Status and Health > All Configured Tasks**. From the All Configured Tasks page, run a **Monitor job streams** type task. You see a list of job streams that matches the task's selection criteria.



Hint: You see the times that are related to the Dynamic Workload Console time zone, which is set in the Set User Preferences page.

Job stream status



© Copyright IBM Corporation 2017

Job stream status

Your job stream monitoring tasks include the status of each job stream in your filter. [Table 4](#) lists the possible status values for **Status** and **Internal Status** columns.

Table 4 Status and internal status columns

Internal status	Status	Description
ABEND	Error	The job stream that is ended in error. This status occurs when no jobs with predecessor dependencies are waiting to run, and one or more jobs ended in error.
ADD	Waiting	The job stream is being added to the production plan.
CANCEL	Canceled	The job stream was canceled.
CANCP	Canceled	The job stream is pending cancellation. Cancellation is deferred until all of the dependencies, including an earliest start time, are resolved.
EXEC	Running	One or more jobs within the job stream is running.
EXTRN	Undefined	The job stream is in a remote Workload Scheduler network and its status is unknown.
HOLD	Waiting	The job stream is waiting for its dependencies to be resolved.
READY	Ready	The dependencies for the job stream are met, but jobs within the job stream are not running.

Table 4 Status and internal status columns (continued)

Internal status	Status	Description
STUCK	Blocked	The job stream was interrupted. No jobs can be started without operator intervention. Three possibilities exist for job streams that have this status: <ul style="list-style-type: none">• A job ended in error, and has another dependent upon its successful completion.• An operator prompt is awaiting an affirmative reply.• A job with priority zero (or manually held) is waiting to run.
SUCC	Successful	All of the jobs within the job stream completed successfully, were confirmed to be successful, or were canceled.

Job Stream Views

The table of job streams is arranged in columns by name, status, and other categories. From the table that shows the job streams, you can click one of the items that are shown in the following table.

Jobs	List the jobs included in a job stream in a new table. You can also click the hyperlink that contains the number of Total Jobs to open the list of jobs in the job stream.
Dependencies	View the dependencies of the job stream.
Release Dependencies	Release a job stream from all its dependencies.
Re-Submit	Submit the selected job stream again.
What-If	Open the What-If Analysis page to see the job stream in a Gantt view timeline.
Job Stream View	Open a graphical view of the job stream you selected. The Job Stream View , shows the job stream with all of its jobs and dependencies. The job stream that you select for graphical views must not have more than 1,000 dependencies.



Note: IBM Workload Scheduler integrates with several types of systems monitoring middleware, such as *IBM SmartCloud Monitoring*. The system administrator can select to have Workload Scheduler send events about *all* jobs and job streams (the default) or send only events about jobs and job streams that are selected for monitoring.

You can select one of the items in the following table from the **More Actions** menu in the monitor job streams task.

Predecessors	See a list of predecessors that prevent the job stream from running.
Hold	Manually hold a job stream. The job stream starts when all its dependencies are satisfied, and you choose Release from the More Actions menu.
Release	Release a job stream that is previously held. All other job stream dependencies still in effect remain intact.
Cancel	Cancel a job stream. Canceling a job automatically releases all dependencies on the job stream you cancel. Canceling job streams and the effects of doing so are covered in “Canceling jobs and job streams” on page 4-24.
Cancel Pending	Cancel the job stream after all other dependencies, including latest start time, are met. Canceling a job stream with the Pending option automatically releases all dependencies on that job stream after its own dependencies are satisfied.
Limit...	Set the job stream limit to enforce how many jobs in the job stream can run concurrently. You can also click the hyperlink in the Jobs Limit column to change the limit.
Priority...	Set the priority of the job stream. You can also click the hyperlink in the Priority column to change the priority. You can choose any value 0 - 101. You can also click Hold , High , Go , or Default to select a new priority. Hold selects priority 0, which prevents the job stream from starting (the same as selecting Hold from the More Actions menu). High selects priority 100. Go selects priority 101. Default selects the priority that was originally assigned to the job stream in the database.
Submit Job...	Submit a job from the database to run as part of the job stream you select. You learn about submitting new work in “Submitting new work” on page 4-27
Submit ad hoc job...	Submit a task to run as part of the job stream you select.
Select all Jobs for Monitoring	Set the “monitored job” flag for the jobs in the job stream you select.
Deselect all Jobs for Monitoring	Clear the “monitored job” flag for the jobs in the job stream you select.
Properties...	See detailed properties of the selected job stream. From the Properties page, you can change some of the properties of the job stream occurrence in the plan.
Properties (New Window)...	See a printable view of the selected job stream’s properties.

Recent Activity	Review the list of recent changes to the item you selected. A new window opens, where you see a list of recent changes that were applied to the object. The list includes a description of the change, the date when the change was applied, the user who applied the change, and the reason the user changed the object.
Open Documentation	This menu item is enabled only if the URL is specified in the TdwcGlobalSettings file. Click this item to access the URL, which is specified in the <code>TdwcGlobalSettings.xml</code> file, where customized documentation about the selected object is available.

Monitoring job streams with conman

% showschedules @##									
Workstation	Job Stream	SchedTime	State	Pr Start	Elapse	#	OK	Lim	Jobs Sch
ALPHA	#SM1STL_DTP	0000 09/16	SUCC	99 07:44	00:01	1	1		F
ALPHA	#SM2SEA_MNW	0000 09/16	HOLD	78 (16:00)	(0:22)	22	0		P
ALPHA	#SM3LAX_WCW	0000 09/16	HOLD	95	(0:05)	5	0		R
ALPHA	#SM3SLC_LNN	0000 09/16	HOLD	89	(0:11)	11	0	-RES1- ; input_file	O
ALPHA	#SP2ATL_YSL	0000 09/16	SUCC	25 07:53	00:21	75	75		
BRAVO	#SP2CLT_NTK	0000 09/16	READY	83 08:20	(00:06)	17	17		
BRAVO	#SP2IAH_MRL	0000 09/16	SUCC	56 14:43	00:12	44	44		
BRAVO	#SP2PIT_KLC	0000 09/16	EXEC	48 15:51	(0:52)	52	38		

Columns

- Workstation
- Job stream
- Scheduled time
- Internal status
- Job stream priority
- Start time (earliest start in parentheses)

- Elapsed time (estimated in parentheses)
- Total number of jobs
- Number of successful jobs
- Job stream limit
- Unresolved dependencies

Dependencies

- | | |
|--|---------------------------------|
| | F Follows (external job) |
| | P Prompt (number 4) |
| | R Resource (1 unit) |
| | O Opens (file) |

© Copyright IBM Corporation 2017

Monitoring job streams with conman

Use the `conman SHOWSCHEDULES (SS)` command to show information about job streams. You can show information about all job streams, or groups of them by using wildcards. Following is the syntax of the SHOWSCHEDULES command.

SHOWSCHEDULES Syntax

```
{showscheds | ss} [workstation#] jobstreamname(hhmm [ date]) [{+|~}qualifier[...]]  
    [,keys]  
    [,showid]  
    [,offline]
```

When you use the `conman showschedules` command, you see a list of job streams, and information about them. The columns in the default format are listed in the following table.

Workstation	The workstation that manages the job stream.
Job stream	The name of the job stream.
SchedTime	The time and date the job stream was scheduled in the plan.
State	The current state of the job stream.
Pr	The priority of the job stream.
(Est) Start	The start time of the job stream.

(Est) Elapsed	The duration of the job stream, in minutes. If the start or elapsed times are shown in parentheses, the time is the estimated time. If the time is not in parentheses, the time is the actual time.
Jobs #	The number of jobs in the job stream.
Jobs OK	The number of jobs in the job stream that completed successfully.
Sch lim	The limit as set in the job stream, or blank if there is no limit set.
Dependencies (not labeled)	The unresolved dependencies that are needed for the job stream to start.

Selecting job streams to view or change

You can select which job streams to monitor by using a job stream qualifier as part of your SHOWSCHEDULES command. By using the `conman` command `HELP JSSELECT`, you can see the possible job stream selections. You can select job streams that match the selector, by using the plus sign (+), or job streams that do not match the selector, by using the tilde (~). A few examples of job stream selectors are shown in the following table.

at = <i>lowtime, hightime</i>	Show job streams that have an earliest start time dependency between <i>lowtime</i> and <i>hightime</i> , inclusively.
carriedforward	Show job streams that carried forward from a previous day's production plan
finished = <i>lowtime, hightime</i>	Show job streams that finished between <i>lowtime</i> and <i>hightime</i> , inclusively.
follows = <i>jobstream</i> or follows = <i>jobstream.job</i>	Show job streams that have a dependency on the job stream or job.
started = <i>lowtime, hightime</i>	Show job streams that ran between <i>lowtime</i> and <i>hightime</i> , inclusively.
state = <i>job stream state</i>	Show job streams with the state that matches the selection. States of job streams are described in Table 4 on page 3-36.
until = <i>lowtime, hightime</i>	Show job streams with a latest start time between <i>lowtime</i> and <i>hightime</i> , inclusively.



Hint: SHOWSCHEDULES

Show all job streams in **hold** or **stuck** states:

```
% SHOWSCHEDULES @#@+STATE=HOLD,STUCK
```

Show job streams on workstation DELTA in any state other than Success:

```
% SS DELTA#@~STATE=SUCC
```

Show job streams that started between 10:00 and 11:59:

```
% SS @#@+STARTED=1000,1159
```

Using alternative listing formats for job streams

You can display different columns of information about job streams by using other format selectors on the `conman showschedules` command. A few examples of output formats are shown in the following table.

keys	Display only one column that contains the IDs of the job streams you selected. The keys format is useful for collecting a list of keys of other job streams that you can use in another <code>conman</code> command or a shell script that runs <code>conman</code> commands.
deps	Display the list of job streams that have a dependency on the job streams you selected.
showid	Display the job stream identifier, in addition to the name and dependency information. Using the showid format, you see the internal identifier that is used to manage job streams. Although you might see several job streams with the same name in the production plan, the internal identifier is unique.

Performing job stream-related activities with conman

From the `conman` prompt, you can perform several actions on job streams in the production plan, by using the `conman` commands that are shown in the following table.

ADDDEP SCHED (ads)	Add a dependency to a job stream.
ALTPRI SCHED (aps)	Change the priority of the job stream.
CANCEL SCHED (cs)	Cancel the job stream, preventing it from running. If you cancel a job stream, you release any dependency on it immediately, unless you also use the ;PEND option.
DELDEP SCHED (dds)	Delete a job stream dependency from the job stream.

LIMIT SCHED (ls)	Change the limit of the number of jobs that run concurrently within the job stream.
RELEASE SCHED (rs)	Release the job stream from all its dependencies, allowing it to start immediately.

Monitoring jobs

Default columns

- Status
- Internal status
- Job name
- Job type
- Job workstation name
- Job stream name
- Job stream workstation name
- Scheduled time
- Number of unsatisfied dependencies
- Priority
- Job number
- Earliest start time
- Actual start time
- Deadline

Other columns you can add

- Return code
- Latest start time
- Estimated duration
- Actual duration
- Run options
- Repeat frequency
- Login
- Number of dependencies
- Critical
- Maximum and minimum duration
- End time
- Deadline passed
- Late

© Copyright IBM Corporation 2017

Monitoring jobs

Monitoring jobs in the plan gives you the capability to see detailed status of workloads in progress in your production environment. You can, for example, view a task that shows all the jobs that are waiting or in error. The advantage of monitoring jobs is that you can directly see critical job status, access job output, and perform job-related actions such as rerunning jobs directly.

To manage and monitor jobs in the production plan, go to **System Status and Health > All**

Configured Tasks. From the All Configured Tasks page, run a **Monitor jobs** type task. You see a list of jobs that matches the task's selection criteria.

Job views

The table of jobs is arranged in columns by name, status, and other categories. From the table that shows the jobs, you can click an item that is listed in the following table.

Job Log...	See the output that a job created. You can also access the job log by clicking the hyperlink in the Job Number column that corresponds to the job whose output you want to see. The job log opens in a new window. From the job log window, you can print the job log or download it to your desktop.
Dependencies	View the dependencies of the job.
Release Dependencies	Release a job from all its dependencies.
Rerun...	Run the selected job again.

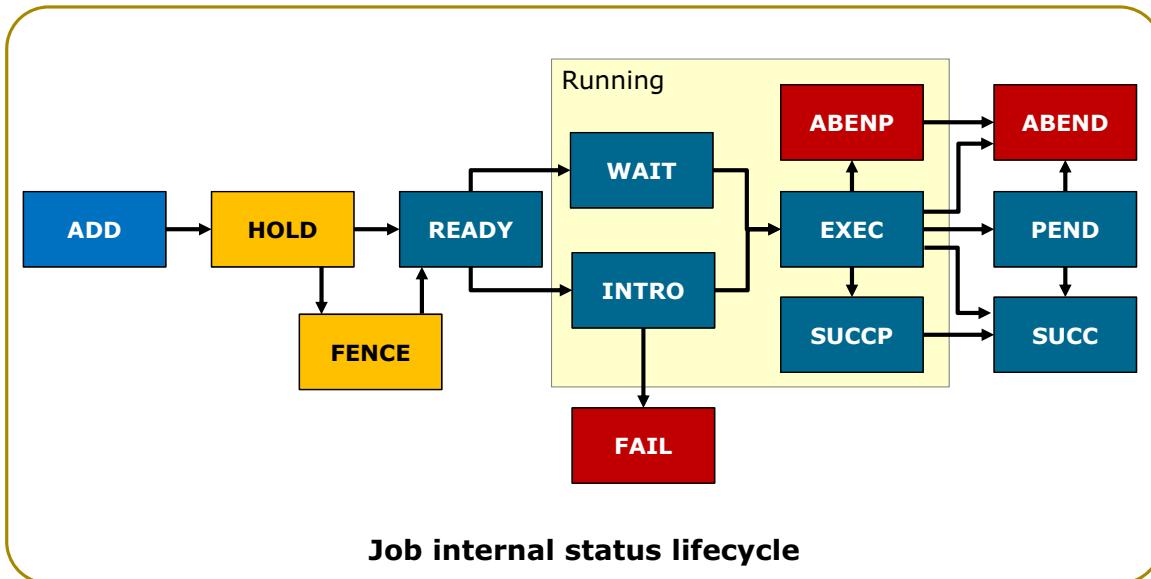
What-If	Open the What-If Analysis page to see the job in a Gantt view timeline.
Job Stream View	Open a graphical view of the job stream that contains the job you selected. The Job Stream View shows the job stream that includes the selected job, with all of its jobs and dependencies.

You can select one of the items in the following table from the **More Actions** menu in the job task.

Predecessors	See a list of predecessors that prevent the job from running.
Rerun with successors	Rerun the job that you selected, and its successors. A window opens that lists all successors of the job you selected, both internal and external, in two separate tables. In the window, you can choose whether you want to rerun all successors in the same job stream (internal successors) or all successors both in the same job stream and in any other job streams (internal and external successors).
Hold	Manually hold a job. The job stream starts when all its dependencies are satisfied, and you choose Release from the More Actions menu
Release	Release a job that is previously held. All other job dependencies still in effect remain intact.
Priority...	Set the priority of the job. You can also click the hyperlink in the Priority column to change the priority. You can choose any value 0 - 101. You can also click Hold , High , Go , or Default to select a new priority. Hold selects priority 0, which prevents the job from starting (the same as selecting Hold from the More Actions menu). High selects priority 100. Priority 100 overrides the job limit value set for the workstation and the value for any resource dependency. Priority 100 allows all the jobs with a priority higher than 0 to start. It does not override the job limit value set for the job stream or the job fence value set for the workstation. Go selects priority 101. Jobs with priority 101 are started as soon as their dependencies are satisfied. Go priority overrides the job limit value set for the workstation, and resource dependency limit. It does not override the job limit value set for the job stream or the job fence value set for the workstation. Default selects the priority that was originally assigned to the job in the database.
Cancel	Cancel a job. Canceling a job automatically releases all dependencies on the job you cancel. Canceling a job and the effects of doing so are covered in "Canceling jobs and job streams" on page 4-24.
Cancel Pending	Cancel the job after all other dependencies, including latest start time, are met. Canceling a job with the Pending option automatically releases all dependencies on that job after its own dependencies are satisfied.
Kill	Stop a job that is running. When you kill a job, it ends in error.
Critical Path	View all the jobs in the critical path of the job you selected.
Confirm SUCC	Confirm that a job ended successfully, regardless of its actual status.

Confirm ABEND	Confirm that a job ended in error, regardless of its actual status.
Download Job Log	Download the job log of the selected job.
Properties...	See detailed properties of the selected job. From the Properties page, you can change some of the properties of the job occurrence in the plan.
Properties (New Window)...	See a printable view of the selected job's properties.
Job Run History	Show a report of the selected job's run history.
Show Spool List (SAP R/3 jobs only)	See a list of spool data for the selected job.
Restart Process Chain (SAP process chains only)	Restart the process chain or rerun some of its processes.
Recent Activity	Review the list of recent changes to the item you selected. A new window opens, where you see a list of recent changes that were applied to the object. The list includes a description of the change, the date when the change was applied, the user who applied the change, and the reason the user changed the object.
Open Documentation	This menu item is enabled only if the URL is specified in the TdwcGlobalSettings file. Click this item to access the URL, which is specified in the TdwcGlobalSettings.xml file, where customized documentation about the selected object is available.

Job status



© Copyright IBM Corporation 2017

Job status

Your job stream monitoring tasks include the status of each job in your filter. [Table 5](#) lists the possible status values for **Status** and **Internal Status** columns.

Table 5 Job status and internal status

Internal status	Status	Description
ABEND	Error	The job ended with nonzero exit code, or an exit code outside the defined return code (RC) mapping.
ABENP	Running	An ABEND confirmation was received, but the job is not completed.
ADD	Waiting	The job is being submitted.
BOUND	Running	For shadow jobs, BOUND means that the shadow job matched a remote job instance in the remote plan. Shadow jobs are explained in "File Transfer and Coordination" on page 5-42.
CANCEL	Canceled	The job was canceled.
CANCP	Running	The job is pending cancellation. Cancellation is deferred until all of the dependencies, including earliest start time dependencies, are resolved.
ERROR	Undecided	For internetwork and cross-dependencies only, an error occurred while checking for the remote status.
EXEC	Running	The job is running.
FAIL	Error	Workload Scheduler cannot start the job.

Table 5 Job status and internal status (continued)

Internal status	Status	Description
FENCE	Waiting	The job's priority is below the workstation's fence.
HOLD	Waiting	The job is waiting for dependencies to be resolved.
INTRO	Running	The job is being introduced to the agent what runs it.
PEND	Running	The job that is completed, and is awaiting confirmation.
READY	Ready	The job is ready to start, and all its dependencies are resolved.
SUCC	Successful	The job ended with an exit code of zero, or an exit code within the defined return code (RC) mapping.
SUCCP	Running	A SUCC confirmation was received, but the job is not complete.
SUSP	Blocked	The job was blocked because of unfulfilled dependencies.
WAIT, WAITD	Held	The job's dependencies are resolved, but the job is waiting for connection or other system resources.

Monitoring jobs with conman

```
% showjobs charlie#sx@
```

Workstation	Job Stream	SchedTime	Job	Status	(Est) Start	Elapsed	Return	Dependencies
CHARLIE	#SX7JAX_TCB	1700 09/17	*****	HOLD	E (17:00)	(00:03)		P #4 (PROMPT1)
			J7JAX_ALIAS_00001	HOLD	51	(00:01)		L <23:00
			J7JAX_APART_00001	HOLD	61	(00:01)		R -CHARLIE#RES1-
			J7JAX_CREPT_00001	HOLD	71	(00:01)		F #4 (PROMPT1)
CHARLIE	#SX7JAX_UMJ	1700 09/17	*****	HOLD	61 (17:00)	(00:03)		D file1; <19:00
			J7JAX_PIKAS_00001	HOLD	21	(00:01)		X CHARLIE#SX7JAX_TCB (1700 09/17/14) .J7JAX_CREPT_00001
			J7JAX_SIAFU_00001	HOLD	31	(00:01)		I J7JAX_PIKAS_00001
			J7JAX_SLOGS_00001	HOLD	41	(00:01)		-CHARLIE#RES1-
								-CHARLIE#RES1-

Dependency types

Earliest start	Resource
Prompt	File
Latest start	External job
Deadline	Internal job

© Copyright IBM Corporation 2017

Monitoring jobs with conman

Use the `conman SHOWJOBS (SJ)` command to show information about jobs. You can use wildcards to show information about all jobs or groups of them. Following is the syntax of the `SHOWJOBS` command.

SHOWJOBS Syntax

```
{showjobs | sj} [workstation#] jobstreamname(hhmm[ date]) .jobname | jobnumber
[{+|~}qualifier[...]]
[;keys | info | step | logon | crit | keys retcod]
[;stdlist [;keys]]
[;short | single]
[;showid]
[;offline]
```

When you use the `conman showjobs` command, you see a list of jobs, and information about them. The columns in the default format are listed in the following table.

Workstation	The workstation that manages the job stream where the job is found.
Job stream	The name of the job stream.
SchedTime	The time and date the job stream was scheduled in the plan.
Job	The name of the job.

State	The current state of the job.
Pr	The priority of the job.
(Est) Start	The start time of the job.
(Est) Elapsed	The duration of the job, in minutes. If the start or elapsed times are shown in parentheses, the time is the estimated time. If the time is not in parentheses, the time is the actual time.
ReturnCode	The return code that the job produced, if it ran, otherwise blank.
Dependencies	The list of unresolved dependencies that are needed for the job to start.

Selecting jobs to view or change

You can select which jobs to monitor by using a job qualifier as part of your SHOWJOBS command. By using the conman command `HELP JOBSELECT`, you can see the possible job selections. You can select jobs that match the selector, by using the plus sign (+), or jobs that do not match the selector, by using the tilde (~). A few examples of job selectors are shown in the table [“conman job selection keywords”](#) on page A-8.

SHOWJOBS

Show all job in **hold** or **wait** states:

```
% SHOWJOBS @#@.S+STATE=HOLD,WAIT
```

Show jobs on workstation DELTA in any state other than **success**:

```
% SJ DELTA#@. @~STATE=SUCC
```

Show jobs that started between 10:00 and 11:59:

```
% SJ @#@+STARTED=1000,1159
```

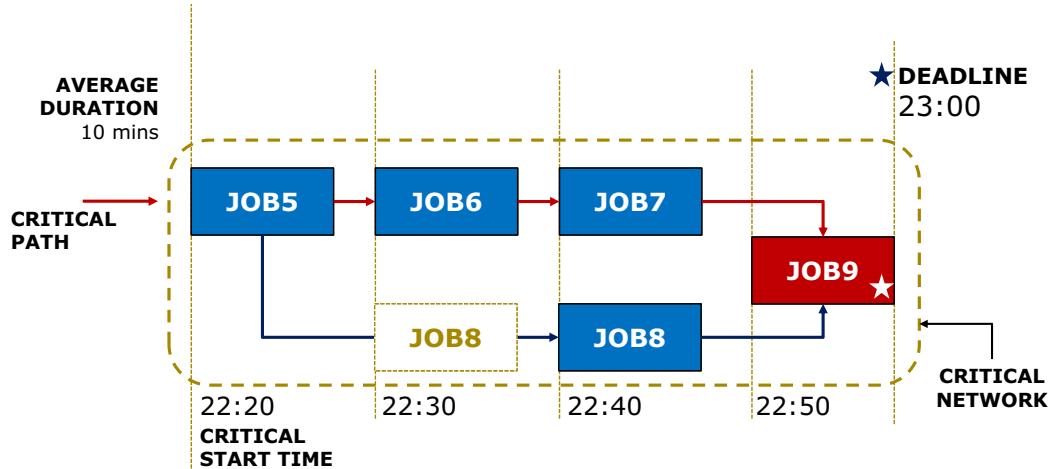
Using other listing formats for jobs

You can display different columns of information about job streams by using other format selectors on the `conman showjobs` command. The following table shows few examples of output formats:

stdlist	Display the output (job log) of the jobs selected
info	Display the script or command the jobs run.
logon	Display the system logon of the jobs selected.
props	Display detailed information about the jobs selected.

keys	Display only one column that contains the IDs of the jobs selected. The keys format is useful for collecting a list of keys of other jobs that you can use in another <code>conman</code> command or a shell script that runs <code>conman</code> commands.
deps	Display the list of jobs that have a dependency on the jobs selected.

Using workload service assurance (critical path)



© Copyright IBM Corporation 2017

Using workload service assurance (critical path)

Workload service assurance is an optional feature that provides a way to flag jobs as mission critical and process them in a timely manner. Workload service assurance is an aid to meeting defined service levels with your scheduled workloads.

Within a job stream, you can flag jobs as critical and assign them a completion deadline time. When workload service assurance is enabled, a plan monitoring task within Workload Scheduler works to calculate:

- The start times of the critical job and all the other jobs that are predecessors of the critical job. These predecessors make up the *critical network*.
- The *critical start time* of the critical job, from the job's deadline time and its average duration. The critical start time is the latest time that the critical job can start and still meet its deadline.
- The critical start time for each job in the critical network, which is calculated from each job's average duration and the critical start time of its successors.
- The estimated start and end times, from the earliest time at which the first job in the critical network might start, and the estimated duration of the other jobs in the critical network.



Note: The estimated duration of jobs is based on the statistics that are collected from previous runs of the jobs. If a job never ran before, the default value of 1 minute is used.

With this information, Workload Scheduler constantly checks the critical network to be sure that the deadline of the critical job can be met. If any change impacts the timings of the critical network, for example, if a job ends in error, Workload Scheduler recalculates the critical start times of the critical network. The critical start times are also recalculated when any job in the critical network completes to account for the actual duration of that job.

The *critical path* is the list of predecessor jobs in the critical network that most directly risk delaying the critical start time of the critical job. The critical path is dynamically updated as predecessors complete or their risk of completing late changes.

Critical path risk levels

A risk status is assigned for each critical job in the plan. The risk status is one of the following values:

- **No risk:** The job is scheduled to complete at or before its assigned deadline.
- **Potential risk:** The job is at risk of missing its deadline, and the job or one of its predecessors is assigned to the **Hot List**.
- **High risk:** The critical job misses its deadline.

The Hot List

Workload Scheduler maintains a *Hot List* of jobs that are part of the critical networks in the plan. The Hot List includes any critical network jobs that have a real or potential impact on the completion of the critical job before its deadline. Only jobs in the critical network that have no incomplete predecessors are added to the Hot List. Following are reasons why a job might be added to the Hot List:

- The job ended in error, and its critical start time is within an offset set by the scheduling administrator.
- The job is running longer than its estimated duration, within a long duration threshold set by the scheduling administrator.
- The job did not start, even if its dependencies are resolved or released. One of the following conditions is also true:
 - The critical start time is within the late offset.
 - The job is scheduled on a workstation where the limit is set to zero.
 - The job or its job stream was suppressed.
 - The job has a priority lower than its workstation's fence, or its priority is zero.

Monitoring critical jobs

Click	To see...
Critical Path	Predecessor jobs in the critical path
Hot List	Predecessor jobs that could cause the critical job to miss its deadline
Not Completed Predecessors	Predecessor jobs that have not completed successfully
All Predecessors	All predecessors in the network (including completed ones)

© Copyright IBM Corporation 2017

Monitoring critical jobs

By using the Monitor Critical Jobs views in the Dynamic Workload Console, you can access information specific to workload service assurance and critical jobs. You access the Monitor Critical Jobs tasks by selecting **System Status and Health > Monitor Workload** from the navigation bar, and selecting the **Critical Jobs** object type in the Monitor Workload window. You can also access the Monitor Critical Jobs task by clicking a risk level link on the Workload Dashboard.

In the Monitor Critical Jobs task, you can see or browse to views that contain items that are listed in the following table.

The Critical path	A list of the predecessors included in the critical path of the selected job.
The Hot List	Jobs in the predecessor network that are late, fenced, suppressed, long running, or in error. These jobs might be currently outside the critical path, but might become critical. If they do not complete successfully on time, they can prevent the critical job from completing on time.
Not Completed Predecessors	A list of the predecessors that are not in the Complete state.
All Predecessors	A list of all the predecessors of the critical jobs, regardless of their state.
Job logs	The output of jobs that already ran.

Lesson 5 Using the What-if Analysis interface

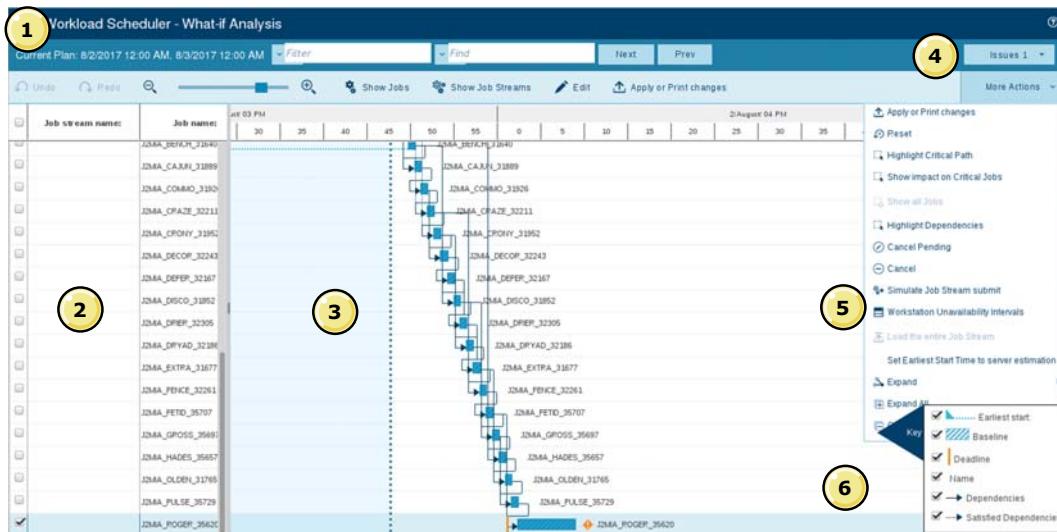
Using the What-if Analysis interface

© Copyright IBM Corporation 2017

The **What-if Analysis interface** lists jobs and job streams in the production plan visually in a Gantt chart. You can use the interface to estimate the outcome of changes to the production plan. In this lesson, you learn how to use the What-if Analysis interface to monitor the production plan. You learn how to perform the following tasks:

- Open the What-if Analysis interface with jobs or job streams in the view
- Add jobs or job streams to the view
- Filter data in the view
- Change the start time, duration, and end time of jobs or job streams to determine the impact
- Add workstation unavailability periods to determine the impact

What-if Analysis interface overview



Key areas

1. Views and filters
2. Jobs and job streams list
3. Gantt view and timeline
4. List of potential issues
5. Actions menu
6. Legend key

© Copyright IBM Corporation 2017

What-if Analysis interface overview

You start the What-if Analysis page from any job or job stream monitoring task by selecting one or more jobs or job streams in the table and clicking **What-if**. You see the **What-if Analysis** in a new browser window or tab.

The What-if page displays a list of jobs and job streams in the production plan that depends on your selection from the monitoring tasks. The page consists of the following sections.

The title, views, and filters

From the first section, you can perform the following functions.

- You see the start and end times of the production plan you are viewing.
- You can filter the view by typing part of a job, job stream, or workstation name in the **Filter** field. The view is automatically filtered to show items that match your selection.
- You can find objects in the view by typing part of a job, job stream, or workstation name in the **Find** field, and pressing Enter to find the first occurrence of your selection. Find subsequent occurrences by clicking **Next**. Click **Prev** to find the previous occurrence that matches your selection.
- Undo or redo actions that you previously performed by clicking **Undo** or **Redo**.
- Show a closer or wider view by clicking **Zoom in** or **Zoom out**, or by dragging the slider control.
- Click **Show Jobs** to add more jobs to the view. You see a new window that displays a list of jobs in the production plan that you can search and select to add to the What-if view. You can filter the list of jobs by entering **Job**, **Job Workstation**, **Job Stream**, or **Job Stream Workstation**

filters and clicking **Search**. Select one or more jobs from the list, and click **Add** to add them to the view. Additionally, you can also select **Include Predecessors**, **Until first completed**, and **Include Successors** to display the dependencies of the selected jobs in the resulting view.

- Click **Show Job Streams** to add more job streams to the view. You see a new window that displays a list of job streams in the production plan that you can search and select to add to the What-if view. You can filter the list of job streams by entering **Job Stream** or **Workstation** filters and clicking **Search**. Select one or more job streams from the list, and click **Add** to add them to the view. Additionally, you can also select **Include Predecessors** and **Include Successors** to display the dependencies of the selected job streams in the resulting view.
- Click **Edit** to change the properties of one or more jobs or job streams that you select. For example, you can set the **estimated duration**, **deadline**, and **earliest start times**, to see how the new properties alter the production plan.

Job and job streams list

From the job and job streams section, you can perform the following functions.

- Review details of the jobs and job streams by scrolling to the right, or by expanding the table. Expand the table by dragging the divider to the right.
- Sort the view by clicking the column header that shows the data to use as sorting criteria. For example, to sort the view by estimated start time, click the heading on the **Estimated start time** column. Click the heading a second time to reverse the sort order.
- Select one or more objects to edit.
- Right-click an object to select an action from the menu. Actions that you can take are described in [Table 6](#).

Gantt view

From the Gantt view section, you can perform the functions in a graphical way. For example, you can change a job's estimated duration by dragging its end to a later time. Learn more about the Gantt view in "Using the Gantt view" on page 3-60

List of potential issues

From the **Issues** menu, you can see a list of potential problems with the view. Select a link from the menu to focus the view on the issue you select. Check the Issues menu frequently to confirm that actions you perform do not create unintended problems with the plan.

Actions menu

From the **More Actions** menu, you can perform actions that are described in [Table 6](#) on page 3-58.

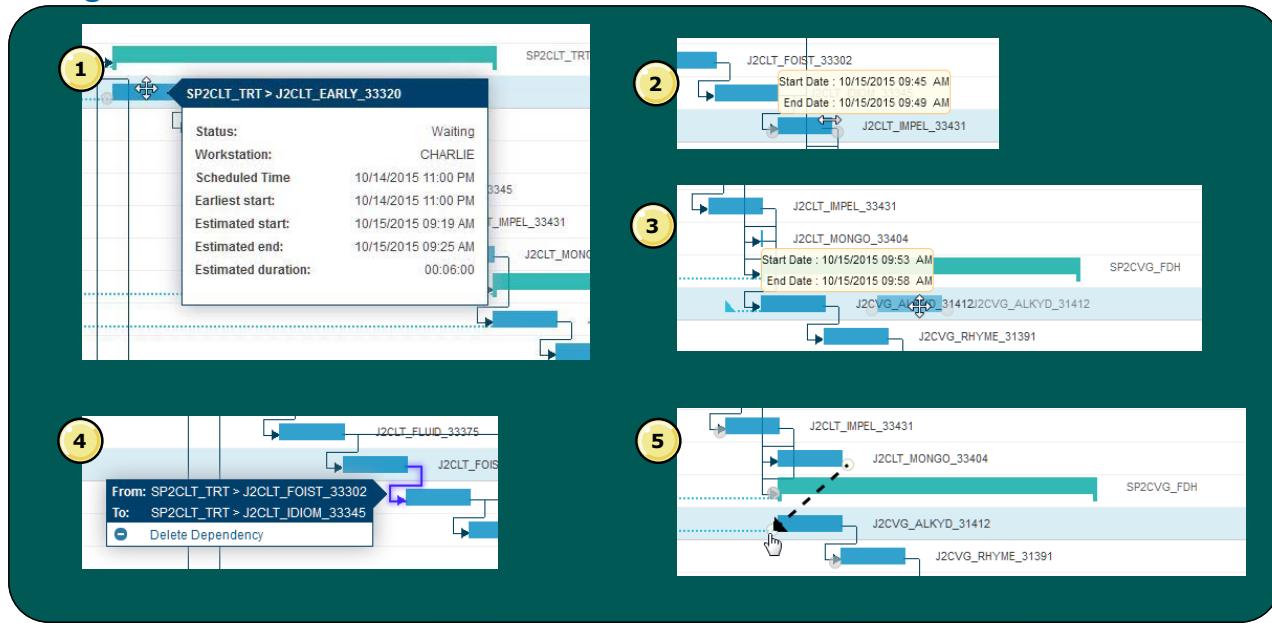
Table 6 What-if Analysis actions

Action	Result
Clear Critical Path	Clear critical path highlighting.
Reset	Reverts the view to show the items as they were when they entered the What-if interface.
Cancel Pending.	Cancel a job stream or job when all its dependencies are resolved. If the job is not started, cancellation is deferred until all the dependencies are resolved. When all the dependencies are resolved, the job is canceled, and any jobs or job streams that depend on the canceled job are released from the dependency. If the job started, the job is canceled when it completes and is moved to its final status.
Cancel	Cancel a job stream or job immediately. Even if the job did not start, all the dependencies are resolved, the job is canceled, and any jobs or job streams that depend on the canceled job are released from the dependency.
Simulate Job Stream Submit	Simulate the impact of submitting a job stream to the What-if Analysis view.
Workstation Unavailability Interval	<p>Set the time interval for when a workstation is unavailable. Select one or more workstation unavailability intervals. You can choose one interval per Workstation. To add an interval, perform the following steps.</p> <ul style="list-style-type: none"> • Select the workstation on which you want to set the time interval from the menu. • In the Start time calender, select the start time of the workstation unavailability, and click Done. • In the End time calender, select the end time of the workstation unavailability, and click Done. • Click Apply to continue, or Cancel to exit. <p>If the Job stream or Jobs loaded on the page use a single workstation, then you can set the unavailability interval for that workstation. The Workstation menu is enabled only when two or more workstations are in the job streams that are selected in the view. To remove any intervals, click Delete next to the workstation to reset.</p>
Load the Entire Job Stream	Click to load the entire job stream. Loading the job stream is useful when you selected one job to start the What-if interface.
Expand	Click to open the Expand menu to select first-level predecessors, first-level successors, all predecessors, or all successors.

Legend Key

Click **Key** to open a page on which you can find the description of the icons. You can also clear the icons that you do not want to use in the view.

Using the Gantt view



© Copyright IBM Corporation 2017

Using the Gantt view

In the Gantt view, you can perform many functions without using the menus.

1. Click an item to show information in a tooltip. In the tooltip, you see the status and estimated times for the job or job stream.
2. Drag the beginning or end of each job or job stream along the horizontal time axis to see how changing the duration changes the job or job stream status against its planned deadline. For example, if you move a job too close to its planned deadline, its status changes to indicate a potential problem. When you change the duration of a job or job stream, its dependencies are maintained and automatically recalculated.
3. Drag the whole job stream horizontally to see how changing the start time changes the job or job stream status against its planned deadline. When you move a job or job stream, its dependencies are maintained and automatically recalculated.
4. Click a dependency to highlight it. You can delete the highlighted dependency by right-clicking it and clicking **Delete Dependency** from the menu. You can also add and remove successors and predecessors for each job or job stream. To add or remove dependencies, right-click a job or job stream to display a menu and select the relevant menu item.
5. To add a dependency to a job or job stream by using a drag action, hover over the job or job stream with your mouse until the circle icons appear on either side. To make this job or job stream a predecessor, drag the right circle icon to the left side of the successor job or job stream until the red arrow turns black. To make this job or job stream a successor, drag the left circle icon to the right side of the successor job or job stream until the red arrow turns black. When you move a job or job stream in the timeline, its dependencies are modified.

Lesson 6 Monitoring the plan graphically

Monitoring the plan graphically

© Copyright IBM Corporation 2017

With the Dynamic Workload Console, you can also monitor and manage the production plan graphically. Almost all of the information and actions available in the tabular views are also available in the graphical views. Two types of graphical views are available:

- In the **Plan View**, you can view a map of all of the job streams (up to 1,000) in the production plan, and their dependencies on each other.
- In the **Job stream View**, you focus on a single job stream in detail, with its jobs and their relationships and statuses. From the view, you can also review the impact that a job stream and its jobs can have on the rest of the plan. In this view, you see, for example, which jobs are successors to an abended job.



Note: To see a job stream or job in the graphical views, you must have access to view them in the Workload Scheduler security.

The screenshot shows the 'Toolbar elements' section of the IBM Training interface. At the top, there's a blue header bar with the 'IBM Training' logo on the left and the 'IBM' logo on the right. Below the header is a toolbar titled 'Job Stream View' with several icons. A large grid below the toolbar lists various tools and their descriptions. On the right side, there's a 'Message viewer' panel showing a message about job streams.

Zoom	Find	
Zoom	Find	Export as SVG
Zoom to fit	Highlight dependencies	Export as PNG
Full screen	Show/Hide dependencies	Print
Auto layout	Show/Hide dependency status	Help on YouTube

Message viewer

© Copyright IBM Corporation 2017

Toolbar elements

The following elements are common to the Plan View and Job Stream View:

- Directional arrows represent dependency relationships.
- The job or job stream objects in the views are *color-coded* according to the status.
- The job objects have *icons* that represent the type of job.
- When you hover the mouse cursor over an object in the graphical view, a *tooltip* shows details about that object.
- When you right-click an object in the graphical view, you see a menu of **Actions**. The list of actions depends on which item is clicked and that item's status.
- Across the top of the graphical view, you see a toolbar with related action items. From left to right, the tools that are available are described in [Table 7](#).

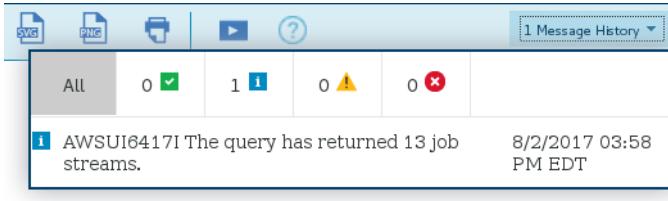
Table 7 Graphical views toolbar icons

Icon	Name	Description
	Zoom	 <p>Click Zoom in or Zoom out to see a larger or smaller view of the plan or job stream. For example, you might be monitoring a small part of it and need to zoom in to see it. Or the opposite might be true. Your plan or job stream might be too large to fit on the screen. You must zoom out to view all of it. To increase the zoom level, click Zoom in. To reduce the zoom level, click Zoom out.</p>
	Zoom to fit	Use to resize the map so that all of the objects are seen in the window.
	Toggle full screen mode	Click to expand the graphical view to fit the entire computer display.
	Auto layout	Use Auto layout to fit the objects in the graphical view to the window, and reposition objects in the graph for an optimal layout.
	Show/Hide Overview	Use Show/Hide Overview to see a high-level overview of the entire job stream or plan in the upper left corner. When the overview is shown, you can use it to quickly go to different parts of the map when zoomed in to larger views of the plan.
	Highlight Dependencies	Use Highlight Dependencies to show the arrows that depict dependencies when you click an object. Highlight Dependencies is useful when several arrows overlap in the view. Clicking a job or job stream while Highlight Dependencies is selected causes the arrows that connect the job or job stream to be highlighted in blue.
	Show/Hide Dependency (job stream view)	Use Show/Hide dependencies to hide the arrows that represent dependencies from the view, or to display the arrows when they are hidden. Hiding the arrows sometimes clarifies the view so that you can more easily read the names of the objects in the view.
	Show/Hide Dependency status (job stream view)	Use Show/Hide dependency status to hide the colors of arrows that represent dependencies from the view, or to display the arrow colors when they are hidden. The arrows that represent satisfied dependencies are green. Arrows that represent unsatisfied dependencies are red. Black arrows represent dependencies that are in an undefined status.

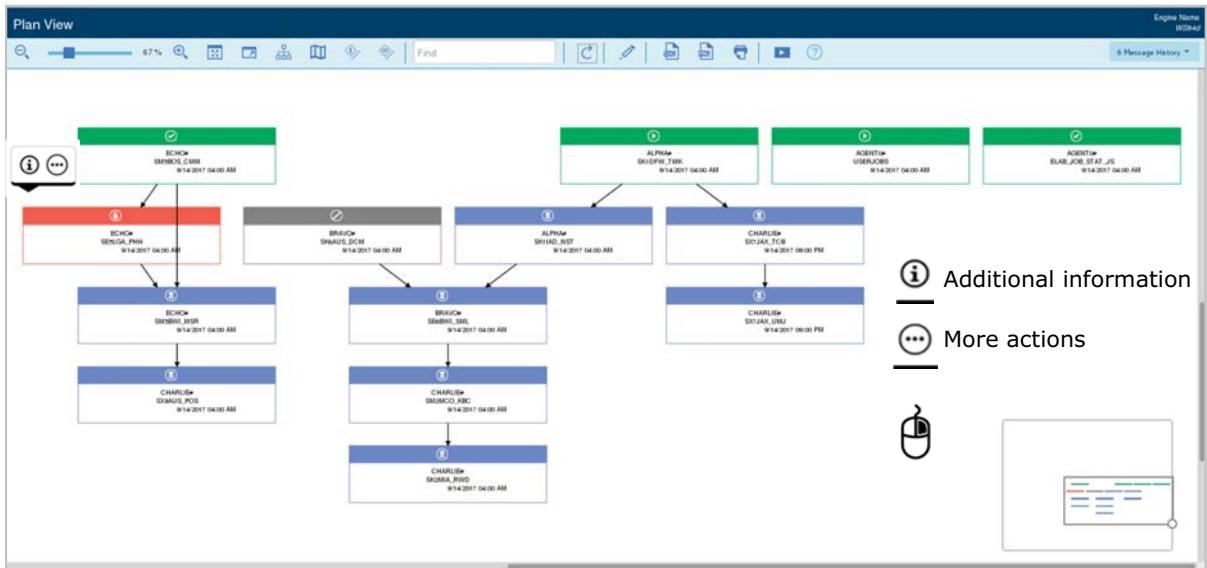
Table 7 Graphical views toolbar icons (continued)

Icon	Name	Description
	Find	 <p>Use Find to search the open view for items that have the text that you choose in the name. For example, to find an object with the name that contains <i>S_M</i>, type <i>s_m</i> in the search input, and click Find. The first object in the view is selected and highlighted in blue. Click Find again to search for the next object that contains the text that you selected. Continue to click Find. When no other items match the selected text, you see the message “Find has finished searching the view. No more matches were found.”</p>
	Refresh	Click Refresh to reload the current view. You cannot change the view while a refresh operation is running.
	Edit Filter Criteria (Plan view)	Use to edit the filter criteria for objects that appear in the Plan View.
	View in What-If (job stream view)	Use to open a What-If interface that contains the objects that are currently being viewed in the Job Stream View.
	Export to SVG	You can save the view to a file on your computer by clicking Export . When you export the view, the browser shows how to save the diagram as an SVG (Scalable Vector Graphics format) file.
	Export to PNG	You can save the view to a file on your computer by clicking Export . When you export the view, the browser shows how to save the diagram as a PNG file.
	Print	Use to send the current diagram to the printer.
	YouTube Play list	Use to open a play list that contains relevant how-to videos that are on YouTube.com.

Table 7 Graphical views toolbar icons (continued)

Icon	Name	Description
Message History		 <p>When a warning or error occurs in the graphical views, you see a message in the upper-right corner. The message is automatically dismissed after a few seconds. To review the messages that you saw since opening the graphical view, click Message History. From the Message History window, you can filter whether to see Error, Warning, or information type messages. Click Return to close the message history view and return to the graphical view.</p>
Help		 <p>Access context-sensitive help by clicking Help. You see the help in a new browser window.</p>

Using the Plan View



© Copyright IBM Corporation 2017

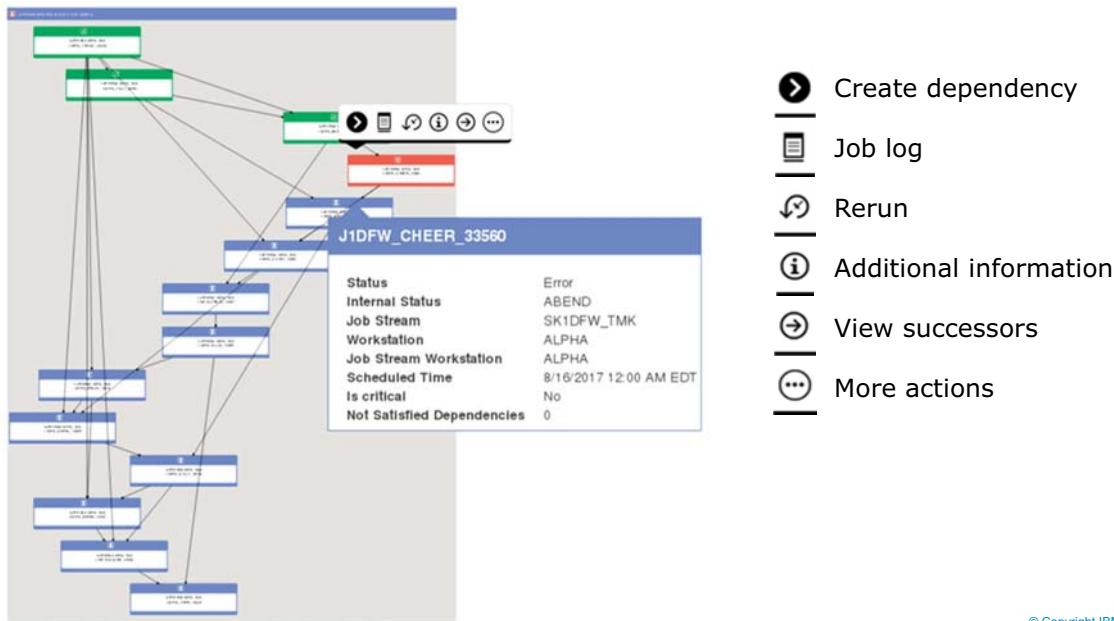
Using the Plan View

Use the **Plan View** to have an overall picture of the plan and how it is structured. You can start the Plan View from the navigation menu by selecting **System Status and Health > Show Plan View**. Plan Views are also task filters that you customize to display only parts of the full plan you want to see. For example, a Plan View task can show only job streams that run on a particular workstation or include a certain time interval.

In the Plan View, each rectangle represents one job stream instance in the plan. The arrows that connect each rectangle represent one or more external dependencies between job streams. The color of the rectangle represents the status of the job stream.

You can perform further monitoring and management functions on the job streams by right-clicking them, or by selecting the job stream, and clicking **More actions**. From the menu, you can perform actions or open new Job Stream Views or What-if Analysis on the job stream.

Using the Job Stream View



Using the Job Stream View

The Job Stream View provides a graphical representation of a job stream in the plan, with all its jobs and dependencies. From the Job Stream View, you can monitor status of jobs and change the plan. You start the Job Stream View by selecting **Job Stream View** from a job stream monitoring task or a job monitoring task. You can also start the Job Stream View by right-clicking a job stream in the Plan View and clicking **Open > Job Stream View**. When you enter the Job Stream View from a jobs monitoring task, the job that you selected before you started the Job Stream View is highlighted in the new view.

The toolbar contains items specific to Job Stream Views:

- Click **Show/Hide external dependencies** to hide dependencies on objects that are not contained in the job stream you are viewing.
- Click **Show/Hide Dependency Status** to change the color of the dependency indicators (arrows) to green or red or black, depending on whether the dependency is met, unmet, or unknown.

You can right-click many of the objects in the Job Stream View to see a menu of available actions. Depending on the status of the object that you click, items on the menu are enabled. When you right-click a job, you see monitoring and management functions that relate to jobs. In addition to several management functions already covered, you can perform these tasks:

- Click **Job Log** to see the output that the job produces.
- Click **Properties** to see details about the job instance.

- Click **Job Run History** to see a list of when the job previously ran before the current production plan.
- Select **Open > Show impact on critical jobs** to open a new window with the list of critical jobs that are successors to the selected object. From the list of critical successors, you can open the What-if interface.

You can expand the Job Stream View to include more levels of predecessors or successors.

Right-click a job or job stream object, and click **Expand > n level Predecessors...** or **Expand > n level Successors...** to refresh the view to contain more jobs or job streams in the view.



Attention: You cannot contract the Job Stream View after you expand it.

Instructor demonstration

Monitoring Workload Scheduler production demonstration



© Copyright IBM Corporation 2017

Instructor demonstration

Exercises

- Use the Workload Dashboard
- Use the Direct Query feature
- Use the What-if Analysis interface
- Use the Job Stream and Plan Views
- Create monitoring tasks



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this unit.

Review questions

1. In a job monitoring task, what jobs does the query string ALPHA#@.@@+STATE=EXEC match?
2. In a monitoring task, you use the plus symbol (+) to include filter parameters. What symbol do you use to exclude filter parameters?
3. What is a secondary query?
4. Which conman command shows the time zones of the workstations in the plan?
5. What is the difference between internal and external job dependencies?
6. What three actions can you define for a latest start time, and which is the default?
7. When a workstation's fence is set to GO, which jobs can start?
8. What possibilities exist for a job stream to have the STUCK (Blocked) status?
9. What causes a job to have the FENCE status?
10. What are a few ways you can start the Job Stream View?

Review answers

1. In a job monitoring task, what jobs does the query string ALPHA#@.@@+STATE=EXEC match?
The query matches jobs that are running on workstation ALPHA.
2. In a monitoring task, you use the plus symbol (+) to include filter parameters. What symbol do you use to exclude filter parameters?
Use the tilde symbol (~) to exclude filter parameters.
3. What is a secondary query?
You run a secondary query when you click a hyperlink in the table of a monitoring task.
4. Which conman command shows the time zones of the workstations in the plan?
*Use **showcpus;info** to see the version and time zone information for workstations.*
5. What is the difference between internal and external job dependencies?
Internal dependencies are between jobs that belong to the same job stream. External dependencies are between jobs that belong to different job streams.
6. What three actions can you define for a latest start time, and which is the default?
*You can define **suppress**, **continue**, or **cancel** actions for a latest start time. Suppress is the default.*
7. When a workstation's fence is set to GO, which jobs can start?
No jobs can start on a workstation that has its fence set to GO.
8. What possibilities exist for a job stream to have the STUCK (Blocked) status?
Three possibilities exist for job streams that have the STUCK status. A job ended in error, and has another dependent upon its successful completion. An operator prompt is awaiting an affirmative reply. A job with priority zero (or manually held) is waiting to run.
9. What causes a job to have the FENCE status?
A job has the FENCE status when its priority is lower than the workstation's fence setting.
10. What are a few ways you can start the Job Stream View?
*You start the Job Stream View by selecting Job Stream View from a job stream monitoring task or a job monitoring task. You can also start the Job Stream View by right-clicking a job stream in the Plan View and clicking **Open > Job Stream View**.*

Summary

You now should be able to perform the following tasks:

- Use the monitoring dashboard to see the production workload plan at a glance
- Customize your views and tasks in the Dynamic Workload Console
- Create and use monitoring tasks by using the Direct Query feature
- Create and use monitoring tasks for other scheduling objects and events
- Monitor plan objects, jobs, and job streams by using the command interface
- Monitor the plan graphically by using the Plan View interface

© Copyright IBM Corporation 2015

Unit summary

Unit 4 Managing Workload Scheduler production

IBM Training



Managing Workload Scheduler production

© 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 managing Workload Scheduler production that uses the Dynamic Workload Console and command-line interfaces. You learn how to perform various tasks that relate to workstations, plans, and jobs.

- Change workstations, plan objects, jobs, and job streams in the plan
- Cancel, kill, and rerun jobs in the plan
- Submit new work into the plan

You can check your progress in lab exercises.

Objectives

In this unit, you learn about managing Workload Scheduler production using the Dynamic Workload Console and command-line interfaces.

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

- Make changes to workstations, plan objects, jobs, and job streams using the command interface
- Change attributes of workstations, plan objects, jobs, and job streams using the Dynamic Workload Console
- Cancel, end, and rerun jobs in the plan
- Submit new work into the production plan

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Managing the scheduling environment

IBM Training

IBM

Managing the scheduling environment

© Copyright IBM Corporation 2017

In this lesson, you learn how to change characteristics of the scheduling environment to alter how workloads run. You learn how to start, stop and link workstations, check the health of workstations, and set computers offline.

Managing your scheduling environment

1 From the **Monitor Workstations** tasks, you can change workstation status

- Start
- Stop
- Link
- Unlink

2 You can also change

- The job limit
- The job fence
- Whether the event monitoring task is running
- Whether the agent is a domain manager

3 You can link to prefiltered views of

- Job streams
- Jobs
- Resources
- Files

© Copyright IBM Corporation 2017

Managing your scheduling environment

You manage Workload Scheduler workstations in the plan by using the **Monitor Workstations** pages. From **Monitor Workstations** pages, you can start, stop, link, and unlink workstations. You can set the *limit* and *fence* of each workstation, and link directly to a predefined task of job streams or jobs that are scheduled on that workstation.

From the **Monitor Workstations** tasks, you can perform several actions on workstations in the production plan, by selecting the check box next to the workstation, and clicking an item in the toolbar:

Start	Start all scheduling processing on a workstation.
Stop	Stop all scheduling processing on a workstation.
Link	Connect a workstation to the Workload Scheduler network.
Unlink	Disconnect a workstation from the Workload Scheduler network.

From the **More Actions** menu, some of the actions you can choose are listed in the following table.

Limit	Set the maximum number of simultaneous jobs on a workstation. You can enter values 0 - 1024, or choose System (unlimited). If you set the limit to zero, only jobs with <i>High</i> and <i>Go</i> priority in job streams in <i>Ready</i> status are started on the workstation.
Fence	View or change the job fence on a workstation. For jobs to be started, their priorities must be higher than the value you set for the workstation's fence. Enter a priority into the New Fence field to set the new fence. Click High to automatically enter a priority of 100. Click Go to automatically enter a priority of 101. For example, if you set the workstation fence to 50, only jobs with a priority of 51 or higher are started.
Jobs	List all the jobs that belong to job streams that run on the selected workstation. You see a secondary query that shows the list of jobs.
Job Streams	List all the job streams that run on a workstation.
Resources	List all the resources that are defined on a workstation.
Files	List all the file dependencies that are defined on a workstation.
Start Event Monitoring	Start events monitoring on one or more workstations.
Stop Event Monitoring	Stop events monitoring on one or more workstations.
Monitoring Configuration	Show information about the event rules active on a workstation.
Check Health Status	Verify the connectivity between the domain manager and workstations.
Properties	Display the properties of the selected workstation.

Lesson 2 Performing workstation-related activities with conman

Performing workstation-related activities with conman

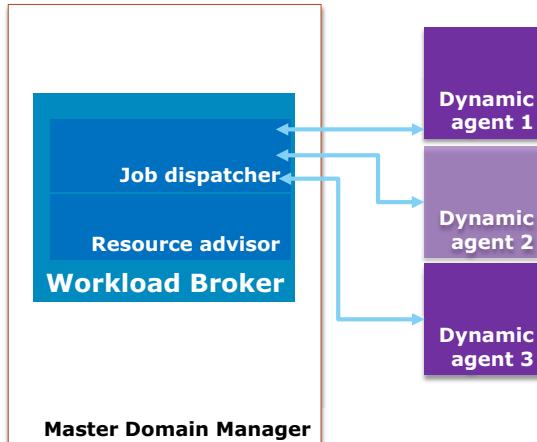
© Copyright IBM Corporation 2017

From the conman prompt, you can perform several actions on workstations in the production plan, by using the conman commands that are shown in the following table.

START	Start processing on a workstation.
STOP	Stop processing on a workstation.
SHUTDOWN (shut)	Stop the Workload Scheduler agent on the computer where the workstation is running.
LINK	Connect a workstation to the Workload Scheduler network.
UNLINK	Disconnect a workstation from the Workload Scheduler network.
LIMIT CPU (lc)	Set the maximum number of simultaneous jobs on a workstation. You can enter values from 0 through 1024 , or SYS (unlimited).
FENCE	View or change the job fence on a workstation, workstation.
STARTMON	Start events monitoring on one or more workstations.
STOPMON	Stop events monitoring on one or more workstations.
STARTEVTPROC	Start the event management processor on this workstation.

STOPEVTPROC	Stop the event management processor on this workstation.
CHECKHEALTHSTAT US (chs)	Verify the connectivity between the domain manager and the workstation.
RESETFTA	Reset a fault tolerant agent by compiling and sending a new Symphony file.

Managing Dynamic Workload Broker computers



System Status and Health > Monitor Computers on Broker

1. Select the computer you want to change
2. Select an action from the **Actions** menu
3. Click **Go**

From the Monitor Computers on Broker task, you can change computer status

- Set **offline**
- Set **online**

You can also

- Refresh the list
- **Delete** the agent from the Workload Scheduler database

© Copyright IBM Corporation 2017

Managing Dynamic Workload Broker computers

To manage Dynamic Agents in the Dynamic Workload Broker database, choose **System Status and Health > Monitor Computers on Broker** to open a search criteria page. To view all computers, leave all fields blank and click **Search**.

The Computer Search Results page displays a list of computers where the Workload Scheduler agent is installed. From the view, you can perform the following tasks on computers:

- Set computers offline so that they cannot be allocated to run jobs.
- Set offline computers back online so that they can be allocated to run jobs.
- Delete computers so that they are no longer visible when you search for computers.

When you delete a computer, it is temporarily removed from the server database. After the delete, the Workload Scheduler agent remains installed and running. Any jobs that are currently allocated and running on the computer complete. To permanently delete a computer, the Workload Scheduler agent software must be uninstalled.

- Refresh the view of the computer search results to see updated information about computers.
- View additional information about a computer.

Lesson 3 Managing scheduling objects in the plan

Managing scheduling objects in the plan

© Copyright IBM Corporation 2017

In this lesson, you learn how to change properties of the scheduling objects in the plan. By changing job stream and scheduling object properties, you can alter when workloads run, recover from errors, and in some cases, change the processing workflow.

You can cause jobs to start, or prevent them from starting. You can cause jobs to run later. You can cancel them so that they do not start and do not prevent other jobs from starting.

Adding or changing time dependencies

From the **Monitor Jobs** or **Monitor Job Streams** tasks, you can add or change time dependencies.

Go to **More actions > Properties**.
Click **Time Restrictions**

Time dependencies that you can change

Jobs and job streams

- Earliest start
 - Latest start
 - Deadline
- Jobs**
- Minimum and maximum duration
 - Repeat range

© Copyright IBM Corporation 2017

Adding or changing time dependencies

You can add or change time dependencies on a job or job stream instance in the plan. You can add or change the times that are listed in the following table.

Earliest start time	Prevent starting before the time.
Latest start time	Prevent starting after the time.
Minimum duration	Set the minimum amount of time the job should last.
Maximum duration	Set the maximum amount of time the job should last.
Deadline time	Set the time by which the job must complete.
Repeat range	How often the job repeats

To add or change time dependencies from a monitoring task, choose one of the following actions.

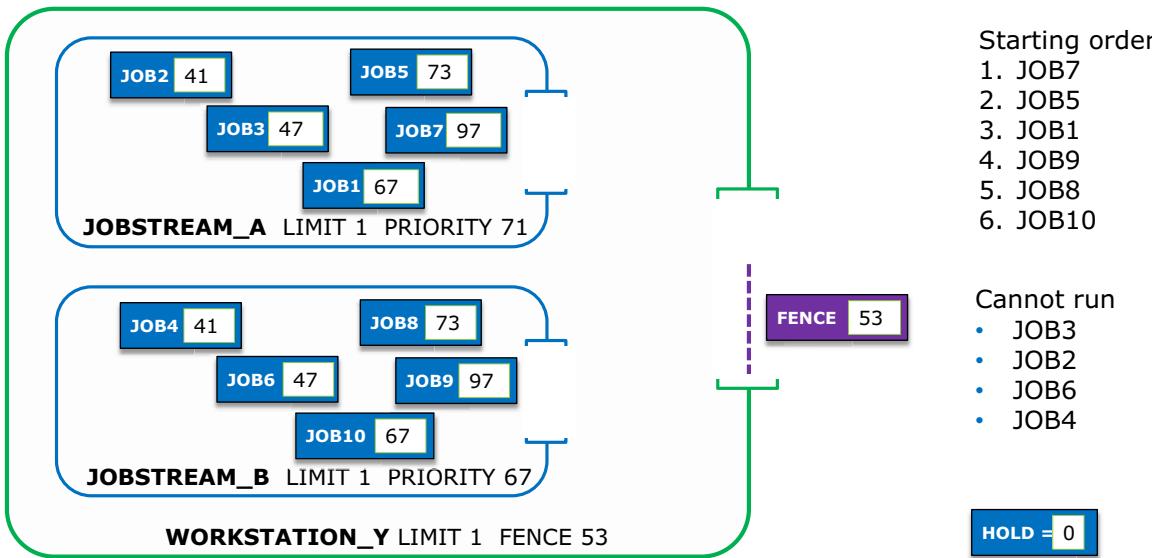
- Select an object and go to the menu **More Actions > Properties**.
- Click the name of the object to go to the **Properties** page.
- Right-click the object and click **More Actions > Properties**.

On the Properties page, expand the section that shows the type of dependency you want to change, and click the hyperlink corresponding to the time you want to change. On the Time Restrictions page, complete the fields for adding or changing the dependency, and click **OK**.



Hint: You can also use the command-line console manager, `conman add dependency` command to add or change time dependencies.

Setting job and job stream priority and limit



© Copyright IBM Corporation 2017

Setting job and job stream priority and limit

Within Workload Scheduler, you can limit how many jobs can run concurrently at each workstation and within each job stream. You can also set which jobs run first.

Changing job and job stream priorities

You can change the priority of a job or job stream to allow it to start, or prevent it from starting. The priority decides which jobs to start first, under the following conditions:

- More than one job requires the same scheduling resource as a dependency.
- The job's workstation limit is lower than the number of jobs otherwise ready to run.
- The job stream's limit is lower than the number of jobs in the job stream that are ready to run.

The priority of jobs is also used to determine whether a job can run based on the workstation fence. If a job's priority is equal to or less than its workstation's fence, the job's status changes to FENCE when it is ready to run. You can change the priority of jobs you want to exceed the workstation fence, unless the workstation fence is GO (priority 101). Conversely, you can lower the priorities of jobs you want to hold back to a new priority equal to or lower than the workstation's fence.

To change the priority of a job or job stream, perform one of the following options:

- Click the hyperlink in the **Priority** column to see the Set Priority page.
- Click the hyperlink in the **Job name** or **Job Stream** name columns to see the properties. Click the **Priority** hyperlink to see the Set Priority page.
- Right-click the job or job stream. Click **More Actions > Priority** from the menu to see the Set Priority page.

On the Set Priority page, enter the priority in the **New Priority** field. You can also click one of the buttons to set the priority:

- **Hold** selects priority zero, which prevents the job or job stream from starting, the same as selecting **Hold** from the **More Actions** menu.
- **High** selects priority 100. Priority 100 overrides the job limit value set for the workstation and the value for any resource dependency. All the jobs with a priority higher than zero start. It does not override the job limit value that is set for the job stream or the job fence value that is set for the workstation.
- **Go** selects priority 101. Job streams with priority 101 are started as soon as their dependencies are satisfied. Go priority overrides the job limit value set for the workstation and resource dependency limit. All the jobs with a priority higher than zero start. It does not override the job limit value that is set for the job stream or the job fence value that is set for the workstation.
- **Default** selects the priority that is originally assigned to the job stream in the database.

Placing jobs or job streams on hold

Placing a job or job stream on hold means setting it so that it cannot start without manual operator intervention. Placing a job on hold has the same effect as setting its priority to zero. When you place a job or job stream on hold, it cannot run until the hold is removed.

To place a job or job stream on hold, perform these steps:

1. From any job or job stream monitoring task, select the job or job stream you want to hold. Choose **More Actions > Hold** from the menu. Click **Yes** to place the job or job stream on hold.
2. From any job or job stream monitoring task, right-click an object and click **More Actions > Hold**. On the resulting page, click **Yes** to place the job or job stream on hold.
3. Use the command-line console manager command ‘altpri jobname; 0’ command.

Releasing jobs or job streams from hold

You must release a job or job stream that is on hold so that it can start.

To release a job or job stream from hold, perform these steps:

1. From any job or job stream monitoring task, select a job or job stream on hold. Choose **More Actions > Release** from the menu. Click **Yes** to release the job or job stream from hold.
2. From any job or job stream monitoring task, right-click an object and click **More Actions > Release**. On the resulting page, click **Yes** to release the job or job stream from hold.
3. Use the command-line console manager `conman 'altpri jobname'` command.

Setting concurrent jobs per job stream

To set the job stream limit, view the job stream in any **Monitor Job Streams** task or **Direct Query**. If the job stream has a limit, you can click the hyperlink in the **Jobs Limit** column to change the limit. If the job stream does not have a limit (the default), select the job stream. Choose **Limit** from the **More Actions** menu to set the new limit.

On the “job stream limit” page, enter a number in the range of 0 - 1,024 in the **New Limit** field, and click **OK**. If the job stream has a limit, you can remove it by clearing the **New Limit** field. This action sets the limit to **unlimited**.

Changing dependencies

- 1** Click **Dependencies** in a job or job stream monitoring task
- 2** Refresh, close, expand, or collapse the view; Show all dependencies, including resolved
- 3** Add or delete dependencies, or release from dependencies
- 4** Change a dependency object
 - Reply to a prompt
 - Change the available resource units

Release from a dependency

- Kept in the plan, but dismissed
- Shows again when the job is rerun

Delete a dependency

- Removed from the plan (but not the database)
- Does not show when the job is rerun

© Copyright IBM Corporation 2017

Changing dependencies

You can change the flow of jobs in job streams in the plan. You can also prevent jobs from starting, or cause jobs to start immediately. To display and set the dependencies of a job instance, in a **Monitor Jobs** task, select the job instance to view, and click **Dependencies**. You see a new page, showing the unresolved dependencies.

From the Dependencies page, you can also add new dependencies, delete dependencies, or release the job or job stream from its dependencies.

Managing prompts

Prompts provide a mechanism for Workload Scheduler to interact with the operator. Prompts are dependencies for jobs and job streams. The answer to a prompt must be yes for the dependent job or job stream to start. For example, you can use a prompt to ensure that a printer has the correct paper before running the job that prints a report can run.

A predefined prompt is defined in the database and can serve as a common dependency among many jobs or job streams. When you reply **Yes** to a predefined prompt, all the dependent jobs and job streams can start.



Hint: Event rules can reply to predefined prompts in the plan.

To reply to a prompt, you can choose any of the following options:

- Go to **System Status and Health > Monitor Prompts > All prompts in plan** to view the list of prompts in the current production plan. Select the prompt to which you want to reply. Click **Reply Yes**.
- From any job or job stream monitoring task, select an object that has a prompt dependency, and click **Dependencies** from the menu. On the Dependencies page, find the **Prompt dependencies** section. Select the prompt to which you want to reply, and click **Reply Yes**.
- From any job or job stream monitoring task, right-click an object that has a prompt dependency, and click **Dependencies**. On the Dependencies page, find the **Prompt dependencies** section. Select the prompt to which you want to reply, and click **Reply Yes**.
- Use the command-line console manager command ‘`reply promptname;yes`’.

Managing resources

Resources can represent physical or logical resources in your environment. Resources are defined in the Workload Scheduler database. You use resources as dependencies for jobs or job streams.

You can change the values of resources in the plan, setting the number of available resources to a higher number (to allow more jobs to run concurrently) or a lower number (to allow fewer jobs to run).

To change a resource in the plan, you can choose any of the following options:

- Go to **System Status and Health > Monitor Resources > All resources in plan** to view the list of resources in the current production plan. Select the resource that you want to change. Click **Change Units**. Enter the new value in the **Quantity Defined** field. Click **OK**.
- From any job or job stream monitoring task, select an object that has a resource dependency, and click **Dependencies** from the menu. On the Dependencies page, find the **Resource dependencies** section. Select the resource that you want to change, and click **Change Units**. Enter the new value in the **Quantity Defined** field. Click **OK**.
- From any job or job stream monitoring task, right-click an object that has a resource dependency, and choose **Dependencies** on the toolbar. On the Dependencies page, find the **Resource dependencies** section. Select the resource that you want to change. Click **Change Units**. Enter the new value in the **Quantity Defined** field. Click **OK**.
- Use the command-line console manager command ‘`resource resourcename,newvalue`’.

Releasing from dependencies

You can release a job or job stream from its dependencies. Releasing a job from its dependencies allows that job to start before its dependencies are naturally resolved. Releasing dependencies can cause jobs to run out of order.

To release dependencies, you can choose one of the following actions:

- To release a job from all of its dependencies, go to **System Status and Health > Monitor Jobs > All Jobs in plan (Distributed)** to view the list of jobs in the current production plan. You can also use any monitor jobs query task or a Direct Query task. Select the job that you want to change. Click **Release Dependencies**. You see a warning message: Are you sure you want to proceed? Click **Yes** to complete the action.
- To release a job or job stream from an individual dependency, from a monitoring task, select an object that has one or more dependencies. Click **Dependencies**. On the Dependencies page, find the section that shows the dependency from which you want to release. Select the dependency that you want to change, and click **Release**.
- Use the command-line console manager `conman release` command to release a job or job stream from one or more dependencies.



Note: When you release a job from its dependencies, the dependency remains in effect when the job is rerun.

Deleting dependencies

You can delete a dependency from a job or job stream instance in the plan. Deleting a job dependency allows that job to start before its dependencies are naturally resolved. Deleting dependencies can cause jobs to run out of order.

To delete dependencies, you can perform one of the following actions.

- From a monitoring task, select an object that has one or more dependencies, and click **Dependencies**. On the Dependencies page, find the section that shows the dependency from which you want to release. Select the dependency that you want to remove, and click **Delete**.
- Use the command-line console manager `conman delete dependency` command to delete one or more dependencies.



Note: When you delete a job dependency, the dependency is lost. In other words, it is no longer in effect when the job is rerun.

Adding dependencies

You can add a dependency to a job or job stream instance in the plan. Adding a job dependency prevents the job from running before the dependency is resolved. You can add predecessor, prompt, resource, and file dependencies.

To add dependencies, you can perform one of the following actions.

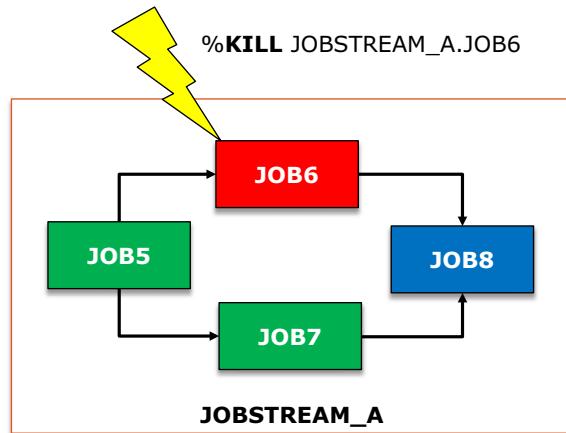
- From a monitoring task, select an object and click **Dependencies**. On the Dependencies page, expand the section that shows the type of dependency you want to add, and click **Add**. On the new page, complete the fields for adding the dependency.
- Use the command-line console manager `conman add dependency` command to add one or more dependencies.



Note: When you add a dependency, the dependency applies only to the specific occurrence in the plan. It does not change the job stream for future occurrences.

Stopping a job

- You can **kill** a job that is *Waiting* or *Running*
- The job ends in error, preventing any successor from starting



© Copyright IBM Corporation 2017

Stopping a job

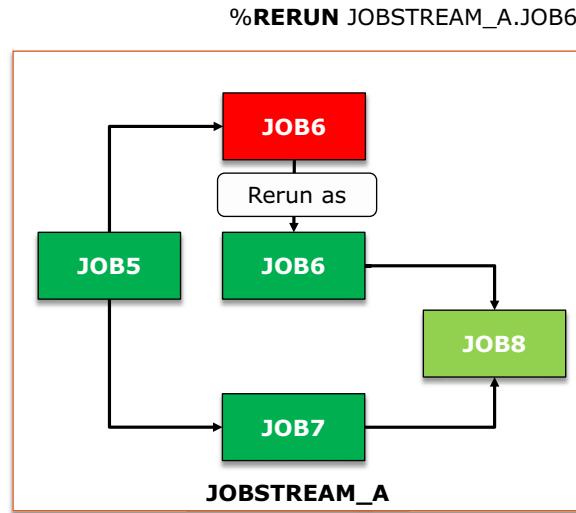
You can end a job if its status is Started or Running. Killing a job stops it running, and its status changes to **Error**.

To stop a job, you can perform one of the following actions:

- Go to a **Monitor Jobs** task to view the list of jobs in the current production plan. You can use any monitor jobs query task or a Direct Query task. Select the job that you want to end. Select **More Actions > Kill**. You see a warning message, *Are you sure you want to proceed?* Click **Yes** to complete the action.
- Go to a **Monitor Jobs** task to view the list of jobs in the current production plan. You can use any monitor jobs query task or a Direct Query task. Right-click the job that you want to stop. Click **More Actions > Kill** from the menu. You see a warning message, *Are you sure you want to proceed?* Click **Yes** to complete the action.
- Use the command-line console manager `conman kill` command to end a job.

Rerunning jobs

- You can **rerun** any job that has run before
- You can specify to use a new job definition during the rerun by using the **From** option
- The **From job** can be the same job definition, or a different one



© Copyright IBM Corporation 2017

Rerunning jobs

You can rerun any job that ends in error. You can also rerun a job that previously completed successfully. When you rerun a job, it begins again. If, after the rerun, the job completes successfully, its successors can start.

Rerunning with defaults

To rerun a job, perform one of the following tasks:

- Select the job in any **Monitor Jobs** task, and click **Rerun**.
- Right-click the job in any **Monitor Jobs** task, and click **Rerun**.

On the job rerun page, change the priority of the job if wanted, and, click **Rerun**.

You can also use the command `rerun` to rerun a job.

Rerunning with a new definition

You can change the definition of the job in the plan by using the **From Job Definition** option on the job rerun page. Rerunning with a modified or substitute definition is useful when a job in error has an incorrect definition that causes the job to end in error.

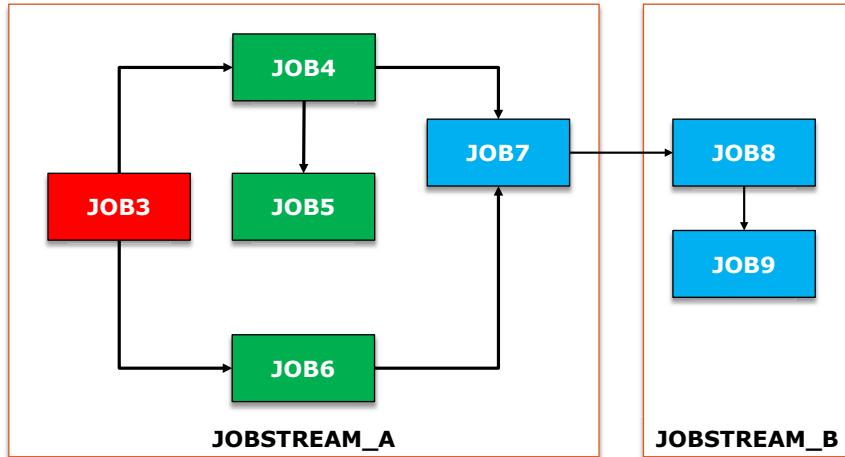
To rerun the job by using a new or modified job definition, modify the job definition in the Workload Scheduler database. Rerun the job. Select the new job definition by clicking **Search**. After finding the new definition, click **OK**.

You can also use the command `rerun` command with the `;from` option to rerun a job with a new or modified definition.

Rerunning jobs with successors

```
%RERUNSUCC JOBSTREAM_A.JOB3;ALL
```

- You can **rerun with successors** any job that has run before
- The job you rerun, and its successors, must not be running
- You can rerun all successors, or only successors in the same job stream



© Copyright IBM Corporation 2017

Rerunning jobs with successors

You can rerun a job and its successors by using one command. You can choose whether you want to rerun all successors in the same job stream (internal successors), or all successors in the same job stream and in other job streams (both internal and external successors).

When you select **More actions > Rerun with successors**, you see a list of the internal and external successors and their related status. To rerun successors in the same job stream, click **Rerun internal successors**. To rerun jobs in the same job stream and other job streams, click **Rerun all successors**.

You must have rerun access to all of the jobs that are rerun by this procedure. If you run the command, and you are not authorized to see and rerun all the successors of the failed job, the list that is displayed contains only the successors you are allowed to see. You also see an error message that there are some additional successors that you are not authorized to see or run.

The rerun with successors action is always performed on the last rerun instance of the specified job. Even if you specify the job number of an intermediate job in the rerun sequence, the action is performed on the last job in the rerun sequence.

You can rerun job successors only if they are in specific states. Successors in intermediate states, such as EXEC, WAIT, INTRO, cannot be rerun. The following states are intermediate; an error is returned and the rerun operation is not performed when any successors are in these states.

- INTRO
- EXEC
- EXTERNAL
- ABENDP/SUCCP
- READY
- PEND
- SUPPR (job stream)
- WAIT

When successors exist in HOLD, BOUND, or FENCE states, the predecessor of the job in that status is rerun, but the rerun sequence stops at the job in the HOLD, BOUND, or FENCE state.

Successors in SUPPR (job), SUCC, CANCEL, and ABEND status are rerun.

Confirming jobs

When you confirm a job, you set its completion status manually, ignoring the actual status of the job. You can confirm a job that is running completed successfully, or in error to override its actual status.

To confirm a job, do one of the following tasks:

- Select the job in any **Monitor Jobs** task. Choose **More Actions > Confirm SUCC** or **More Actions > Confirm ABEND**.
- Right-click the job in any **Monitor Jobs** task. Click **More Actions > Confirm SUCC** or **More Actions > Confirm ABEND**.

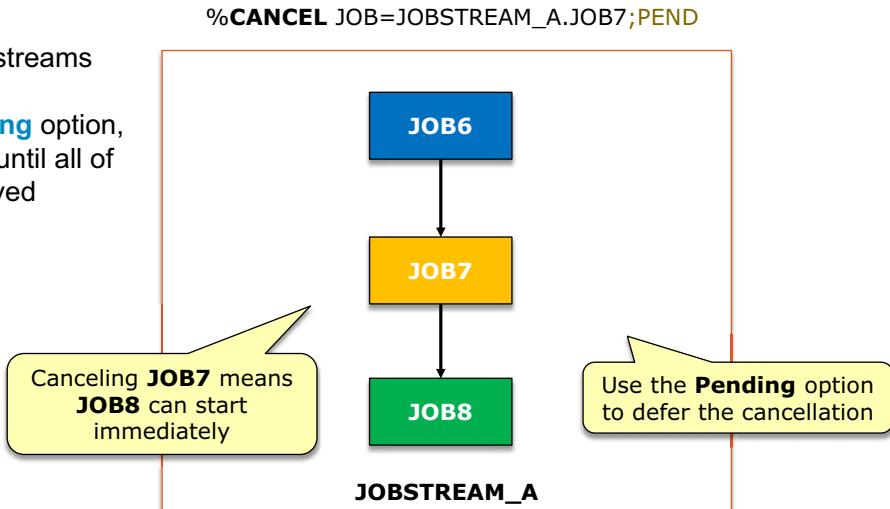
You see a warning message, *Are you sure you want to proceed?* Click **Yes** to complete the action.



Important: When you confirm a job, you change its status in the plan *and also in the recorded history* for the job. For example, if you confirm a job that is in error as SUCC, the job's history shows that the job runs successfully.

Cancelling jobs and job streams

- You can **cancel** jobs or job streams
- If you cancel with the **Pending** option, the cancellation is deferred until all of the dependencies are resolved



Cancelling jobs and job streams

You can *cancel* a job or a job stream in the plan. If you cancel a job before it runs, it does not run. If you cancel it after it starts, it continues to run. If you cancel a job that is running, and it ends in an error state, no automatic job recovery steps are attempted.

If you do not use the **Pending** option, jobs and job streams that depend on the canceled job are released immediately from the dependency. Jobs can run out of order if you cancel a job that has successors.

If you include the pending option, and the job did not start, the cancellation is deferred until all of the dependencies, including earliest start time, are resolved. When all the dependencies are resolved, the job that is canceled and any jobs or job streams that depend on the canceled job are released from the dependency. While the cancellation is deferred, you see the **Cancel Pend** notation on the job.

If you include the pending option, and the job is already running, the option is ignored and the job is canceled immediately.

You can rerun jobs that were canceled, either with or without the pending option. You can also add or delete dependencies on jobs that been canceled with the pending option.

To cancel a job, perform one of the following functions from any job monitoring task:

- Select the job in any **Monitor Jobs** task. Choose **More Actions > Cancel** or **More Actions > Cancel Pending**.
- Right-click the job in any **Monitor Jobs** task. Click **More Actions > Cancel** or **More Actions > Cancel Pending**.

You see a warning message, Are you sure you want to proceed? Click **Yes** to complete the action.

You can perform the same action in the same way for job streams, by selecting the job stream to cancel in a job stream monitoring task.

You can also cancel jobs or job streams by using the command-line `conman cancel` command.

Lesson 4 Performing job-related activities with conman

Performing job-related activities with conman

© Copyright IBM Corporation 2017

From the conman prompt, you can perform several actions on jobs in the production plan, by using the conman commands that are shown in the following table.

ADDDEP JOB (adj)	Add a dependency to a job.
ALTPRI JOB (apj)	Change the priority of the job.
CANCEL JOB (cj)	Cancel the job, preventing it from running. If you cancel a job, you release any dependency on it immediately, unless you also use the ;PEND option.
DELDEP JOB (ddj)	Delete a job dependency from the job.
KILL (k)	Kill a job that is running, ending it.
RELEASE JOB (rj)	Release the job from all its dependencies, allowing it to start immediately, if all of the job stream dependencies are met.
RERUN (rr)	Rerun a job.
RERUNSUCC	Rerun a job and its successors.

Submitting new work

Submitting predefined jobs or job streams

- **Select** the object to submit, or search for it
- Use an **alias** if needed
- Optionally choose to resolve variables from a different **variable table**
- Add **dependencies** to prevent the job stream starting immediately

If you submit a job or job stream with no dependencies, it could start *immediately*

Use an **alias** when the job or job stream exists in the plan, and you want the new occurrence to have a unique name

Submitting new work

You can submit new work into the production plan by using the Dynamic Workload Console or command-line interfaces. When you submit a job or job stream into the plan, you can cause it to run immediately as defined, or you can add dependencies to have it run later.

Submitting job streams

To submit a job stream by using the Dynamic Workload Console, choose **Administration > Submit Predefined Job Streams** to open the “Submit Job Stream into Plan” page. Search for the job stream to submit, or type the required job stream name and workstation name into the fields.



Note: When you submit a job stream, you override all run cycles that are defined in the job stream definition. For example, on Monday you submit a job stream that is scheduled to run only on Wednesdays. The job stream runs when you submit it on Monday, in addition to Wednesday.

When you are submitting a job stream, you can change the following options:

- Click **Alias** to change the name of the job stream to the alias you choose.
- Choose **Variable table** to change the variable table from which variables embedded in job definitions are expanded. Changing the variable table can be useful when a job definition is used as a template that can change based on variable information.
- Select **Scheduled time - Specify date and time** to change the time that the job stream is positioned in the plan.

You can further change more options by clicking **Properties**. These examples are some of the options that you can change:

- Limit
- Priority
- Earliest start, latest start, and deadline times
- Dependencies, that is, resources, prompts, files, and predecessors



Hint: When you submit a job stream, its run cycles are ignored. Other dependencies in the job stream definition are accepted. If a job stream definition contains predecessors dependencies on jobs or job streams not in the current production plan, they are ignored.

Addressing errors while submitting job streams

While submitting a job stream, you might see a message similar to the following error:

AWSUI0291E The job stream *jobstream_name* cannot be submitted because of the following engine error:

AWSJPL506E The job stream "*jobstream_name*" cannot be submitted because the specified scheduled time "*current_date_time*" on the master domain manager is later than the end time of the production plan "*plan_end_date_time*" on the master domain manager.

This error occurs when the production plan was not extended, or a scheduled time too far in the future was entered. To avoid this error, you can enter a scheduled time earlier than *plan_end_date_time* before submitting the job stream.

Submitting predefined jobs

To submit a job from its database definition by using the Dynamic Workload Console, choose **Administration > Submit Predefined Jobs** to open the "Submit Job into Plan" page. Search for the job to submit, or type the required job name and workstation name into the fields.

When you are submitting a job, you can change the options that are listed in the following table.

Alias	Changes the name of the job to the alias you choose.
Variable table	Changes the variable table from which variables embedded in the job definition is expanded. Changing the variable table can be useful when a job definition is used as a template that can change based on variable information.
Into	Change the name of the job stream (already in the plan) into which you submit the job stream. If you do not specify a job stream, the default job stream that is named JOBS is used. The JOBS job stream is created for you if it does not exist in the plan.
Scheduled time - Specify date and time	Change this field if the job stream specified on the Into section exists more than once in the production plan. This field is normally completed by searching for a job stream.

You can further change more options by clicking **Properties**. Here are some of the options that you can change:

- Recovery options to specify what Workload Scheduler should do automatically if the job ends in error.
- Priority
- Earliest start, latest start, and deadline times
- Minimum and maximum duration times
- Dependencies, that is, resources, prompts, files, and predecessors



Hint: If you submit a job without adding dependencies, it can run immediately.

Submitting undefined jobs

You can submit a task that is not defined as a job

1. Choose the **task type**, system **login**, and **workstation** where the task runs
2. Use an **alias** if needed
3. Choose the **job stream** that contains the job (JOBS is the default)
4. On the **Task** tab, select **Script** or **Command**, and enter the **task** to run
5. Optionally add **dependencies** to specify when the task runs

The administrator can *disable* the option to submit undefined jobs

Undefined jobs are *not logged* in the Workload Scheduler job history for reporting

© Copyright IBM Corporation 2017

Submitting undefined jobs

To submit a job from its database definition by using the Dynamic Workload Console, choose **Administration > Submit Ad Hoc Jobs** to open the “Submit Ad Hoc Job into Plan” page. Enter the required job and workstation information into the fields on the **General** tab as listed in the following table.

Task Type	The type of task to run, which can be one of Windows, UNIX, SAP, Workload Broker, or other. This selection changes the available options on the Task tab.
Login	The operating system user that runs the job on the agent.
Workstation	The name of the workstation where the job runs.

Enter the required information on the **Task** page.

Type	Choose Script or Command .
Script or command	When the task is a script, enter the file name and any options and arguments. When the task is a command, enter the name of the command for the job to run together with any options and arguments. The maximum length is 4095 characters.

When you are submitting a job, you can change the following options in other tabs on the page.

Alias	Changes the name of the job to the alias you choose.
Variable table	Changes the variable table from which variables embedded in the job definition is expanded. Changing the variable table can be useful when a job definition is used as a template that can change based on variable information.
Into	Change the name of the job stream (already in the plan) into which you submit the job stream. If you do not specify a job stream, the default job stream that is named JOBS is used. The JOBS job stream is created for you if it does not exist in the plan.
Scheduled time - Specify date and time	Change this field if the job stream specified on the Into section exists more than once in the production plan. This field is normally completed by searching for a job stream.
Recovery options	Sets what Workload Scheduler should do automatically if the job ends in error.
Priority	Selects a run priority for the job
Times	Sets the earliest start, latest start, repeat, and deadline times
Duration	Sets them minimum and maximum duration times.
Dependencies	Select resources, prompts, files, and predecessors as dependencies.

Lesson 5 Submitting new work with the command line

Submitting new work with the command line

© Copyright IBM Corporation 2017

You can also submit jobs, job streams, and system commands into the current production plan by using the command-line interface, `conman`. Within the `conman` interface, you can use the commands that are listed in the following table.

<code>submit job (sbj)</code>	Submit a job from the database.
<code>submit schedule (sbs)</code>	Submit a job stream from the database.
<code>submit docommand (sbd)</code>	Submit a system command or script as an ad hoc job.

Instructor demonstration

Managing Workload Scheduler production demonstration



© Copyright IBM Corporation 2017

Instructor demonstration

Exercises

- Change workstations, plan objects, jobs, and job streams
- Cancel, end, and rerun jobs in the plan
- Submit new work into the production plan



© Copyright IBM Corporation 2017

Exercises

Review questions

1. When you set a workstation's limit to zero, which jobs can start on that workstation?
2. When you set a workstation's fence to 50, which jobs can start on that workstation?
3. When two jobs are ready to start, but the workstation's limit is set to 1, which job starts?
4. When does a job that is placed on hold start?
5. When you *release* a job from a dependency, what happens to the dependency when you rerun the job?
6. When you *delete* a job's dependency, what happens to the dependency when you rerun the job?
7. How can you replace a job definition in the plan?
8. When you cancel a job without the pending option, what happens to the successor jobs?
9. If on Monday you submit a job stream that is scheduled to run only on Tuesdays, what occurs?

Review answers

- When you set a workstation's limit to zero, which jobs can start on that workstation?

*Only jobs with a priority of **High** or **Go** start on workstations with a limit of zero. Other jobs remain in the **ready** state.*

- When you set a workstation's fence to 50, which jobs can start on that workstation?

Only jobs with a priority of 51 or higher can run on a workstation that has its fence set to 50.

- When two jobs are ready to start, but the workstation's limit is set to 1, which job starts?

The job with the higher priority starts first.

- When does a job that is placed on hold start?

Jobs that are manually placed on hold must be manually released to start.

- When you release a job from a dependency, what happens to the dependency when you rerun the job?

When you release a job from a dependency, the dependency appears again when you rerun the job.

- When you delete a job's dependency, what happens to the dependency when you rerun the job?

When you delete a job dependency, the dependency does not show when you rerun the job.

- How can you replace a job definition in the plan?

*Rerun the job by using the **From Job Definition** option in the job rerun page.*

- When you cancel a job without the pending option, what happens to the successor jobs?

When you cancel a job without the pending option, successor jobs can run immediately.

- If on Monday you submit a job stream that is scheduled to run only on Tuesdays, what occurs?

When you submit job streams, run cycles are ignored, and the job stream is scheduled to run in today's plan.

Summary

You now should be able to perform the following tasks:

- Change workstations, plan objects, jobs, and job streams
- Cancel, end, and rerun jobs in the plan
- Submit new work into the production plan

© Copyright IBM Corporation 2017

Unit summary

Unit 5 Creating scheduling objects

IBM Training



Creating scheduling objects

© 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 define Workload Scheduler objects, such as calendars and resources, and job definitions.

Objectives

In this unit, you learn how to define Workload Scheduler objects, jobs, job streams, and event rules. You learn how to perform the following tasks:

- Open and use features of the Workload Designer
- Create scheduling object definitions
- Create job definitions of several types

© Copyright IBM Corporation 2017

Objectives

In this unit, you learn how to define Workload Scheduler objects, such as variables, jobs, prompts, resources, and calendars. You also learn about how to configure different job types, automated job recovery options, and return code mapping. These building block objects are used in subsequent lessons to build job streams and event rules.

Before you build job streams, you must define the foundation: a base of objects in a job stream. This unit contains information scheduling objects and how to define them.

Use the Dynamic Workload Console or the composer program to define and modify various scheduling objects such as variables, jobs, resources, and workstation definitions. This information is in a database that a master domain manager uses and manages.

Workload Scheduler uses database information to create instances of objects in a plan of system activity. The plan covers a production day, a period of 24 hours by default. The plan contains all of the job and job stream instances that are needed for the day and resources and workstations that are involved in running them. You can work with database objects, monitor the progress of the plan, and manage aspects of plan operations.

Lesson 1 Using the Workload Designer

Using the Workload Designer

© Copyright IBM Corporation 2017

The **Workload Designer** is the graphical user interface that you use to perform common configuration tasks. It is a component of the Dynamic Workload Console.

In this lesson, you learn how to use the Workload Designer to create and modify scheduling objects. You learn about the common tools available for creating new objects, saving objects, and finding and opening existing objects to modify. You learn how to perform the following tasks:

- Open the workload designer
- Find and open objects
- Find, create, and duplicate objects

Starting the Workload Designer

The screenshot shows the Workload Designer interface. On the left, there's a sidebar with a 'Create New' section containing icons for Job Definition, Remote SAP R/3 Job, Job Stream, Prompt, Resource, User, Calendar, Workstation Class, Variable Table, Workload Application Template, and Run Cycle Group. Below this is a note: 'Select the scheduling intent type you want to create.' In the center, there's a search bar with a placeholder 'Search for objects to use' and a 'Compute' button. A yellow box highlights the search bar. To the right of the search bar is a 'Recent Activity' list with items like 'CHARLIE#EKAUS_P01' and 'PREPOOL_EXITCODE'. Another yellow box highlights the 'Open recently edited objects' link. At the bottom of the central area are 'Edit', 'View', and 'Delete' buttons, with a red box highlighting the 'Edit' button. On the far right, a 'Search toolbar' is displayed with icons and labels for various objects: Job stream, Job definition, Prompt, Resource, User, Calendar, Workstation class, Variable table, Workload Application Template, Run cycle group, and Remote SAP R/3 job.

© Copyright IBM Corporation 2017

Starting the Workload Designer

To open the Workload Designer, go to **Administration > Manage Workload Definitions** to open the Manage Workload Definitions page. Select the engine that you use to connect to the installation of Workload Scheduler, and click **Go**. You can open multiple occurrences of the Workload Designer, in one of the following scenarios:

- As the same user, who is connected to multiple engines
- As a different user, who is connected to the same engine



Hint: The Workload Designer opens in a new browser window. Be sure that your browser allows pop-up windows for the computer that hosts the Dynamic Workload Console.

The Workload Designer has two pages that you use to create and edit objects in the Workload Scheduler database.

On the first page, you see a window with three columns.

1. The left column, **Create New**, shows a list of the types of objects you can create. For example, click **Job Stream** to create a job stream.
2. The middle column, **Search**, shows a search field and icons that represent the types of objects you can search. Choose a type of object to search by clicking its icon, and type part of the objects' name in the search field. A list of items matching the search is shown in the middle column. You can open an object in either edit mode or in view (read-only) mode. When you

open an object in edit mode, the object is locked in the database. While you have an object locked, other users can view the object only in read-only mode.

3. The right column, **Recent Activity**, shows a list of objects that were recently edited. You can edit an item on the Recent Activity list by clicking it.

Using the Workload Designer

Edit toolbar

- Save
- Close
- Edit
- Create Like (duplicate)
- Unlock
- Delete
- Reload
- Print
- Message history
- SAP connection data

The screenshot shows the Workload Designer application. On the left, the 'Working List' pane displays a tree view of objects under 'CHARLESMAH_P08'. A yellow box highlights the 'Working list' tab. On the right, the 'Object properties' pane is open for the object 'CHARLESMAH_P08', with a yellow box highlighting the 'Details' tab. The bottom of the screen shows a toolbar with various icons.

© Copyright IBM Corporation 2017

Using the Workload Designer

When you create or open an object to edit, you see the second page. You edit objects' properties on the second page. The Workload Designer page has the following three sections.

Using the Working List pane

In the *Working List*, you see the objects that you opened. From the list, you can select one or more objects, and perform several functions. The basic functions are listed in the following table.

Create Like	Duplicate the selected objects.
Edit	Lock and open the selected objects for editing.
Unlock	Unlock the selected objects to edit them.
Delete	Remove the selected objects from the database.

You can add items to the *Working List* in one of the following ways.

1. In the **Search** field, click the icon of the type of object that you want to open, complete the search criteria, and click **Search**. Wildcard characters are supported in the search. An asterisk matches one or more characters, and a question mark matches exactly one character.

The search returns a maximum of 250 items. You can further filter the searches by clicking **Advanced** and filling the added search fields before searching.

Open one or more items by selecting the items and clicking **Edit**, **View**, or **Add**. You can also open all of the items that match your search by clicking **Select all**. You can open the selected items in one of three ways:

- Click **Edit** to open the items for editing.
 - Click **View** to open the items in read-only mode.
 - Click **Add** to add the selected items to a job stream that is already open for editing.
2. Click **Create New**, and select the type of object to add.

Using the Details pane

Use the *Details* pane to edit objects from the database. Each icon in the toolbar represents an object function that you can use. You see a tooltip, which contains the name of the function when you hover the mouse pointer over the icon.

Duplicating objects with Create Like

The *Create Like* function, the tool at the top of the details pane, duplicates the object that is being edited. The *Create Like* function is available for every object that you can edit with the Workload Designer. *Create Like* is especially useful when you create large numbers of similar objects in a short time. The duplicate object has a unique, ascending number that is appended to the name, but inherits all other settings from the parent object.

Comparing and restoring previous versions of objects

The screenshot shows the IBM Workload Scheduler interface. On the left, there's a navigation bar with tabs: General, Start condition, Scheduling options, Time restrictions, Dependency resolution, and Versions. The Versions tab is selected. Below it, a table lists 'Versions of: CHARLIE#JBAUS_POS' with columns: Date, User, Category, Ticket number, and Reason. Two rows are shown: one for '9/14/17 4:38 PM EDT Current version' and another for '9/14/17 3:39 PM EDT'. To the right, a separate window titled 'IBM Workload Scheduler - Compare versions' displays two side-by-side tables: 'Revised' and 'Previous'. Both tables show scheduling objects and their dependencies, such as 'CHARLIE#JBAUS_POS' and 'CHARLIE#JBAUS_FIORD_00001'.

Comparing versions

1. Open the **Versions** tab
2. Select two versions of the object
3. Click **Compare**

Restoring a previous version

1. Open the **Versions** tab
2. Select a previous version of the object
3. Click **Restore**

© Copyright IBM Corporation 2017

Comparing and restoring previous versions of objects

Workload Scheduler maintains all versions of your scheduling objects in the database, and you can easily access them from the Dynamic Workload Console. Workload Scheduler administrators can enforce a policy by which each user who changes an object must provide a justification (reason) for the change. The audit trail contains the name of the user who performed the change, the time, and date when the change occurred, the reason why the change was implemented, and the details of the change for every modified item.

When auditing and justification are enforced, for each scheduling object, you can view the history of changes, and compare two different versions side by side. You can also restore a previous version, replacing the current version. You work with an object's versions in the following ways.

- To view the list of versions of an object, open the object in the Workload Designer, and click the **Versions** tab.
- To compare two versions of an object, open the **Versions** tab, and select two versions of the object. Click **Compare** to open a new browser window that displays a side-by-side comparison of the versions that you selected.
- To a previous version of an object, open the Versions tab, and select the version of the object to restore. Click **Restore**. You see a message that confirms your request to start the restore process. Click **Continue with the restore process**. When you restore a previous version, you must review the object, and save the object by clicking **Save**.

Lesson 2 Creating scheduling objects

Creating scheduling objects

© Copyright IBM Corporation 2017

In this lesson, you learn how to create the scheduling objects that job streams can reference. You learn how to perform the following tasks:

- Create Windows users
- Create variables and variable tables
- Create prompts and resources
- Create calendars, run cycles, and run cycle groups
- Create job definitions

Creating Windows users

Workstation	Name of a workstation representing a computer running Windows
User name	Local (User), Windows domain user (Domain\User), or User Principal Name (user@domain)
Password	The user's password on the computer

- Jobs running on computers running Microsoft Windows require an associated user name and password
- Create the user definition in the Workload Scheduler database to provide the password
- Passwords are encrypted before they are stored in the database
- Passwords can be changed temporarily in the plan

© Copyright IBM Corporation 2017

Creating Windows users

When running jobs on Windows workstations, the user name in the **Login** field must have a matching user definition. The definitions furnish the passwords that are required for Workload Scheduler to start jobs on Windows. If the password changes on the Windows workstation, you must manually update Workload Scheduler with the new password.

The `comman ALTPASS` command specifies an alternative password for the current production day plan only. Update the database definition in composer so that future production plans contain this change.



Hint: You can also alter Windows users' passwords from the Dynamic Workload Console.

This function applies to Microsoft Windows only and does not apply to other operating systems such as AIX, Linux, Solaris, and HP/UX.

Create Windows users in the Workload Designer by selecting **Create New > User** in the Workload Designer. The Windows user name can contain up to 47 characters. Windows user names are case-sensitive within Workload Scheduler. The case of the user definition must match the login in the job definitions.



Important: The operating system user must have the **Logon as a batch job** access privilege on the workstation where Workload Scheduler starts jobs. If the name is not unique, it is evaluated in this order: local user, domain user, trusted domain user.

Lesson 3 Creating user definitions by using the composer command

Creating user definitions by using the composer command

© Copyright IBM Corporation 2017

You can also create a user definition in the database by using the command-line interface. To create a user definition, run the `composer` command. Enter `new user` at the dash (-) prompt. You edit a template that contains three users by using standard text editor commands. You can also edit an existing variable table by using the `composer modify user=user_name` command.



Hint: Editing a user definition in composer

```
USERNAME FTA1#DOMAIN1\USER1  
PASSWORD "password"  
END
```

```
USERNAME FTA1#USER2  
PASSWORD "password"  
END
```

```
USERNAME USER3  
PASSWORD "password"  
END
```

Lesson 4 Creating variables and variable tables

Creating variables and variable tables

JOB J7SAT_DREP_12211
SCRIPTNAME "^\\$SCRPATH^/comp_rpt.sh ^START^ ^END^ ^STOREID^"

Variable table U1PATHS
SCRPATH "/users/reporter/bin"

Variable table Q_DATES
START "2017-04-01"
END "2017-06-30"

Variable table STOREINFO
STOREID '0291-5029'
STOREST 'TX'
STORECN 'US'

Variable table L1PATHS
SCRPATH "/home/reporter/bin"

Variable table Y_DATES
START "2017-01-01"
END "2017-12-31"

Variable table RPTINFO
STOREID '9999-0001'
SCRPATH "/tmp/scripts"
START "2017-01-01"
END "2017-06-30"

- **U1PATHS:** Assigned to UNIX workstations
- **L1PATHS:** Assigned to Linux workstations
- **Q_DATES:** Assigned to quarterly run cycles
- **Y_DATES:** Assigned to year-end run cycles
- **STOREINFO:** Assigned to job streams
- **RPTINFO:** Manually selected

© Copyright IBM Corporation 2017

You can use *variables* to define a value that is used in many scheduling objects. Using variables simplifies configuration and maintenance, and increases portability. You can use variables as substitutes for strings in file names, prompt text, login names, command strings, and command arguments within job definitions.

Variables are stored in *variable tables* in the Workload Scheduler database. You can create as many variable tables as required. The default variable table, named MAIN_TABLE, is created when you install Workload Scheduler. You can also change the default variable table to another variable table you create. You can assign variable tables to run cycles, job streams, and workstations, or manually when you submit jobs or job streams.

The same variable name can exist in multiple tables, but contain different values in each table. When you have multiple tables, you can specify different replacements under various circumstances. For example, job streams might use alternative expansions of the same variable, depending on when the job stream is selected to run.

Variable tables can be useful in job definitions when the definition is used as a template for a job that belongs to more than one job stream. You can use variable tables when multiple jobs in the same job stream contain the same set of values. Variables can also be useful when a job stream is

submitted as a result of an event rule, which can carry data from the environment that triggered the event.

You can use the same variable name in different tables to represent different values. You can also assign a variable table to every run cycle, run cycle group, job stream, and workstation. To evaluate a variable, Workload Scheduler reads its value from the variable tables in the following order and stops when the variable is resolved:

- Table that is specified during a manual submission
- Table that is specified in the run cycle or run cycle group definition
- Table that is specified in the job stream definition
- Table that is specified in the workstation definition
- The default variable table

If the planner does not find variables in any of the tables that are defined, it uses the values from the default variable table, or MAIN_TABLE. Each variable in a job definition might be evaluated from a different table.

Create variable tables specific to each workload type, and reference them in your job streams and run cycles. This method makes your workload definitions more portable and easier to use and port.

Creating a variable table

In the Workload Designer Working List pane, click **Create New > Variable Table** to create a variable table. On the General page, enter a name and description for your table.

The **Default Variable Table** check box indicates that Workload Scheduler now uses this variable table as the default table. A default table, MAIN_TABLE, is installed with the product.



Important: Variables are resolved from the default table last. If you change the default variable table, make sure that the new default variable table contains all of the variables and values that were present before. Otherwise, jobs might enter the plan without their variables resolved.

Adding variables to a variable table

To add a variable, click the **Variables** tab. On the variables page, click the plus sign (+), and enter the name of the variable and the value of your variable into the appropriate fields. The maximum length for variable names is 16 characters, and the maximum length for variable values is 72 characters. Click **Save** to save the table.

Using a variable

To use variables in the Dynamic Workload Console, click **Add variable** and search for your variable name by clicking **Search**. Select your variable from the list and click **OK**.

Lesson 5 Creating variables and variable tables by using composer

Creating variables and variable tables by using composer

© Copyright IBM Corporation 2017

To create a variable table by using the command-line interface, run the `composer` command. Enter `new vartable` at the dash (-) prompt. You edit a template that contains two variable tables by using standard text editor commands. You can also edit an existing variable table by using the `composer modify vartable table_name` command.

Editing a variable table in composer

```
VARTABLE TABLE1  
MEMBERS  
TEMP_DIR "/tmp/temp1"  
DB2_PATH "/home/db2inst1/sqllib"  
END
```

```
VARTABLE TABLE2  
ISDEFAULT  
MEMBERS  
TEMP_DIR "/tmp/temp2"  
DB2_PATH "/home/db2inst2/sqllib"  
END
```

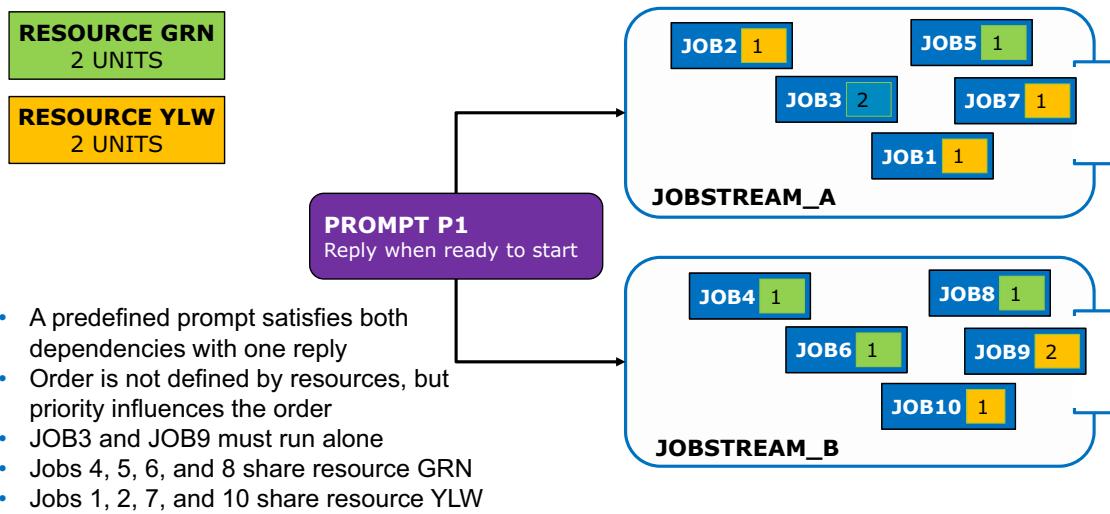
You can also edit variables by using the `composer` command in variable format.

Editing variables by using composer

```
$PARMS  
TABLE1.MY_PATH "/home/user"  
TABLE2.MY_PATH "/home/twsuser"  
DEFAULT_PATH "/usr/dev/tmp"
```

Lesson 6 Creating prompts and resources

Creating prompts and resources



© Copyright IBM Corporation 2017

Prompts provide a mechanism where Workload Scheduler can interact with the operator. Use prompts as dependencies for jobs and job streams. The reply to a prompt must be **Yes** for the dependent job or job stream to start. For example, you can issue a prompt to ensure that a printer is online before the job that prints a report runs.

A predefined prompt is defined in the Workload Scheduler database. Any job or job stream can use a predefined prompt.



Hint: You can use event rule actions to reply to prompts in the plan.

Create prompts in the Workload Designer by clicking **Create New > Prompt**. The prompt name must start with a letter and can contain alphanumeric characters, dashes, and underscores. The name can be up to eight characters long.

Creating resources

A *resource* is a counter that represents a quantity that is needed to run a job, such as tape drives, databases, or printers. Workload Scheduler defines resources as named quantities in the database. You can use resources as dependencies for jobs and job streams that run on the same workstation where the resource is defined to prevent too many jobs from starting concurrently.

Resources are allocated to jobs and job streams when they start running and are released when they complete, successfully or not. For example, you can define a resource that is called **Tapes** with quantity **2**. You define jobs that require two tape drives as a dependency. Jobs with this dependency cannot run concurrently because each time a job runs, the Tapes resource is in use.

You can alter the number of resources available in the plan from the Dynamic Workload Plan or the command-line interface.

Create resources in the Workload Designer by selecting **Create New > Resource**. The resource name must start with a letter and can contain alphanumeric characters, dashes, and underscores. The name can be up to eight characters long. The **Quantity** field can be 0 - 1024.

Lesson 7 Creating prompts and resources by using composer

Creating prompts and resources by using composer

© Copyright IBM Corporation 2017

You can also create prompt definitions by using the `composer` command. To create a prompt definition by using the command-line interface, run the `composer` command. Enter `new prompt` at the dash (-) prompt. You edit a template that contains example prompt definitions by using standard text editor commands.

Creating prompts by using composer

```
$PROMPTS  
PRMT1 "ready for job4? (y/n)"  
PRMT2 ":job4 started"  
PRMT3 "!Continue?"
```

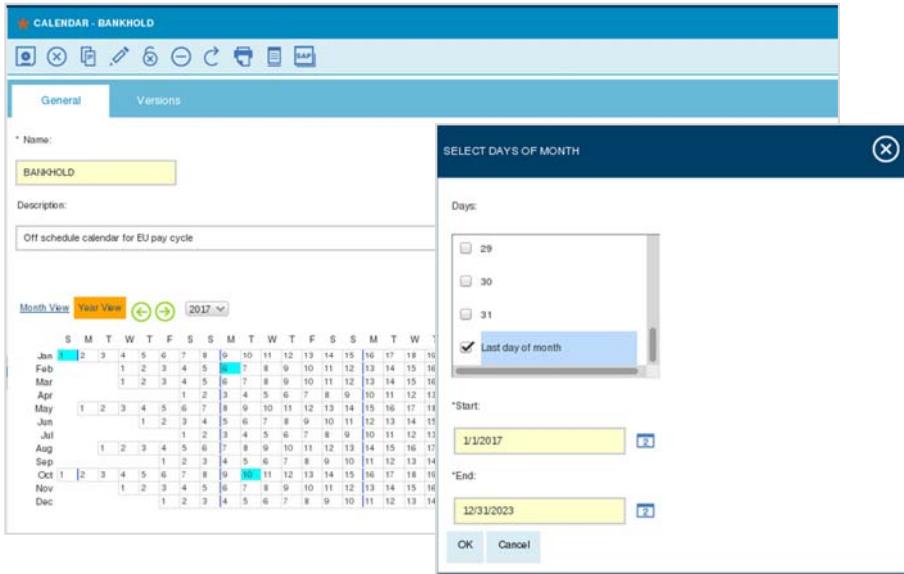
You can also create resource definitions by using the `composer` command. To create a resource definition by using the command-line interface, run the `composer` command. Enter `new resource` at the dash (-) prompt. You edit a template that contains an example resource by using standard text editor commands.

Creating resources by using composer

```
$RESOURCES  
FTA1#RES1 10 "Sample Resource"
```

Lesson 8 Creating calendars

Creating calendars



Use calendars to specify

- Nonworking days (regional holidays)
- Offset origins such as month end for *n*th workday run cycles (5th work day of the month)
- Irregular payroll, accounting, or human resources business cycles

A calendar is a list of scheduling dates that are defined in the Workload Scheduler database. A job stream that has a calendar run cycle that is assigned to it can run only on the days that are specified in the calendar. Not all run cycle options require calendars. Calendars are only necessary when you specify custom sequences of dates that require calculations that are based on holidays, or *free days*, or irregular business planning cycles. A *free day* is the opposite of a workday.

You can create as many calendars as necessary to meet your scheduling requirements. For example, you can define a calendar that is named PAYDAYS with a list of pay dates. You can also define a calendar that is named MONTHEND with a list of month-ending dates. You can create a HOLIDAYS calendar with the dates of your company holidays and nonworkdays for defining run cycles for your job streams.

Having one or more Free Days calendars extends the role of the Holidays calendar. With Free Days calendars, you can choose which calendar defines workdays for a job stream. If you decide to use your own freedays calendar, the redefined meaning of workdays is limited to that particular job stream or command. If you do not specify a freedays calendar, the default HOLIDAYS calendar is used, and workdays retain their traditional meaning.

Defining calendars

Create calendars in the Workload Designer by selecting **Create New > Calendar**. Enter a name for your calendar, and describe its function and operating range in the **Description** field.

Select **Month View** or **Year View** and click the dates to select. You must select at least one date before you can save the calendar.

To select the same date each month, click **Select days of month** to open a secondary window. Next, choose dates in each month over the calendar effective range. When you use this function, specify a start and end date for the selection.

Lesson 9 Creating calendars by using command-line interfaces

Creating calendars by using command-line interfaces

© Copyright IBM Corporation 2017

You can also create calendars by using the `composer` command or the `makecal` utility.

Creating calendars by using the composer command

To create a calendar by using the command-line interface, run the `composer` command. Enter new calendar at the dash (-) prompt. You edit a template that contains an example resource by using standard text editor commands.

Creating a calendar by using `composer`

```
$CALENDARS
```

```
CAL1
```

```
 "Sample Calendar"
```

```
01/31/2018 02/28/2018 03/31/2018
```

Creating calendars by using the makecal utility

To create a calendar by using the makecal utility, run `makecal` followed by the arguments that specify which dates you want the calendar to contain. The `makecal` command creates a calendar in the Workload Scheduler database that contains the dates you specify. The following table shows common options to the `makecal` command.

Table 1 Using the makecal utility

Argument	Description
<code>-c name</code>	Specifies a name for the calendar. The name can contain up to eight alphanumeric characters and must start with a letter. Do not use the names of weekdays, or Workload Scheduler keywords for the calendar names. If you do not use the <code>-c</code> option, the calendar is named <code>Chmm</code> , where <code>hhmm</code> is the current hour and minute.
<code>-d n</code>	Adds the <i>n</i> th day of every month to the calendar.
<code>-e</code>	Adds the last day of every month to the calendar.
<code>-f 1</code>	Creates a fiscal month-end calendar containing the last day of each fiscal month.
<code>-f 2</code>	Specify one of the following formats:
<code>-f 3</code>	<ol style="list-style-type: none"> 1. 4-4-5 week format 2. 4-5-4 week format 3. 5-4-4 week format. <p>The <code>-f</code> option also requires the <code>-s</code> option to set the first date in the calendar</p>
<code>-i n</code>	Specifies how many dates to insert into the calendar. For example, to create a calendar that contains 24 dates, use <code>-i 24</code> .
<code>-l</code>	Adds the last workday of every month to the calendar. For this argument to work properly, the production plan (Symphony file) and the holidays calendar must already exist. The new calendar this option creates also include the last workday of the month that precedes the date of creation of the calendar.
<code>-p n</code>	Adds the workday before the <i>n</i> th day of every month to the calendar. For this argument to work properly, the production plan (Symphony file) and the holidays calendar must already exist.
<code>-r n</code>	Adds every <i>n</i> th day to the calendar. This argument requires the <code>-s</code> argument to set the first (starting) date in the calendar.
<code>-s date</code>	Specifies the starting date for the <code>-f</code> and <code>-r</code> arguments. The date must be enclosed in quotation marks, and must be valid and unambiguous, for example, use JAN 10 2005, not 1/10/05.
<code>-w n</code>	Adds the workday after the <i>n</i> th date of each month to the calendar. For this argument to work properly, the production plan (Symphony file) and the holidays calendar must already exist.

Table 1 Using the makecal utility (continued)

Argument	Description
-x	Sends the calendar output to stdout instead of adding it to the database.
-freedays calendar	Specifies the name of a non-working days calendar to replace the holidays calendar to evaluate workdays. This keyword affects the processing of <code>makecal</code> with options -l , -p , and -w .

For example, to make a two-year long month-end calendar with the last day of every month selected, run the following command:

```
makecal -c MONTHEND -e -i 24
```

To make a calendar with 30 days that starts on May 30, 2017, and has every third day selected, run the following command:

```
makecal -r 3 -s "30 MAY 2017" -i 30
```

Lesson 10 Creating and using run cycles

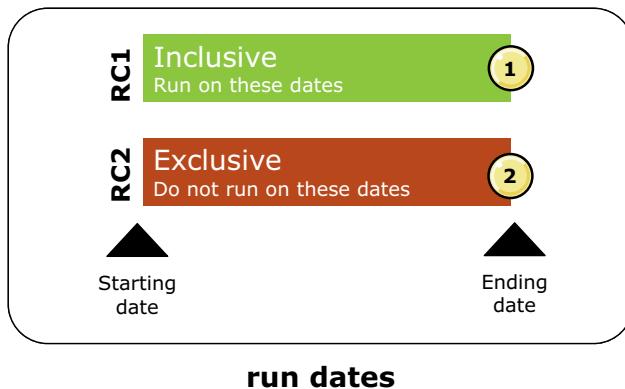
Creating and using run cycles

© Copyright IBM Corporation 2017

Run cycles specify when a job stream in the database is selected to run in the plan. Run cycles are part of job streams and run cycle groups. Most job streams use a daily (every day) run cycle. However, combinations of run cycles include and exclude various days and dates. For example, you can create a run cycle for every day except Fridays and the last day of each month.

You can specify run cycles with actual dates, days of the week, or calendars. A calendar is a set of specific dates. You can create as many calendars as required to meet your scheduling requirements.

Inclusive and exclusive run cycles



1. Inclusive run cycles are **combined**
2. Exclusive run cycles are **combined**, and then **excluded** from the included run dates

© Copyright IBM Corporation 2017

Inclusive and exclusive run cycles

Inclusive run cycles include the dates that are selected. *Exclusive* run cycles select dates that are excluded from the run dates.

Exclusive run cycles identify negative occurrences - the dates when a job stream would normally be scheduled, but should not run. The sum of the exclusive run cycles is subtracted from the inclusive ones. A negative occurrence always cancels any matching positive occurrences. You can specify a negative occurrence only if the positive equivalent exists. The dates and time restrictions of exclusive run cycles must exactly match occurrences in the inclusive run cycles for the cancellation to occur.

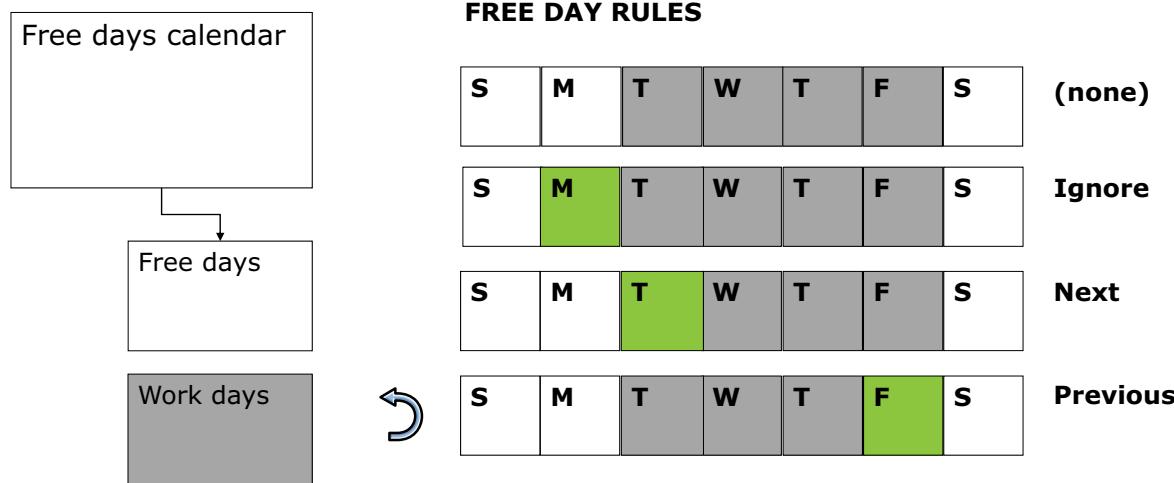
Starting and ending dates

Each run cycle can have an optional starting date or an ending date, or both. The starting date determines when the run cycle begins. The ending date determines when the run cycle ends.



Note: If you specify an ending date, that date is not included in the run cycle. For example, you create an inclusive daily run cycle, selecting 31 December 2018 as its ending date. In this case, the run cycle includes 30 December 2018, but not 31 December.

Free days and workdays



© Copyright IBM Corporation 2017

Free days and workdays

Use the **Non-working days calendar** option to specify the name of a calendar listing nonworkdays for your company. This calendar can include holidays and weekends, that is, any days when you do not want the job stream to run.

Use the calendar of nonworking days during the definition of the run cycle for the job stream. The calendar is associated with the rule for nonworking days, where you tell Workload Scheduler how to handle job streams on free days. The default Free Days calendar is named **HOLIDAYS**.

Saturday and Sunday are normally considered free days. To consider Saturday a working day, clear the **Saturday as a non-working day** selection. To consider Sunday a working day, clear the **Sunday as a non-working day** selection.

Rules for nonworking days

You use *Free day rules* to alter the selection of dates in a run cycle as it concerns holidays and free days. When a date in a run cycle falls on a free day with no rule applied, the date is included regardless. You can choose to alter the selection in one of three ways, as listed in the following table.

Ignore	Do not include the date.
Next	Include the next work day after the selected nonworking day.
Previous	Include the nearest working day before the selected nonworking day.

Run cycle rule types

Rule types	
Simple	April 1, 2018
Daily	Every day, week days, or work days
Weekly	Mondays, Wednesday, and Fridays
Monthly by day	5th day of every month
Monthly by week day	Last Friday of every month
Yearly	Once a year, on December 25th
Calendar	(or offset) On payroll days

© Copyright IBM Corporation 2017

Run cycle rule types

You can specify the following types of run cycle rules.

Simple

Use the **Simple** run cycle for running a job stream a few times on dates that do not have a repeatable pattern. If you select six dates, the job stream runs only on those six dates. You can select a month or year view and change months and years to find the dates that you need.

Use this type of run cycle to exclude dates from a job stream that runs every day. For example, you can use this type of run cycle to take the job stream out of production during a maintenance window.

After specifying a run cycle, check the dates on a calendar in the **Run Cycles Preview** tab at the top of the Details pane. Click the **Details** tab to resume your configuration.

Daily

The **Daily** run cycle is typically the most commonly used run cycle. **Everyday** or **Workdays** are the most frequently selected day types. You can choose a daily cycle to run every other day or every three days.

Weekly

The **Weekly** run cycle specifies particular days of the week. Use this cycle to specify whether to run every week, or every second week, third week, and so on. Job streams that run once a week or

once every other week (like many bimonthly paydays), are common. For example, use a weekly run cycle to run a job stream every Monday.

Monthly

Use the **Monthly by month day** run cycle to run schedule-based repeating days of the month. An example is a pay day run cycle that selects the 15th and last day of every month. An advantage of this approach is that, unlike a calendar, you do not manually update this run cycle every year.

Use the **Monthly by week day** run cycle to run the job stream on a particular day of each month. An example is running a job stream on the first Monday and the third Wednesday of every month.

Yearly

The **Yearly** run cycle runs a job stream once a year, or less frequently, if specified in the **Run every selected number of years** field. The date this run cycle starts is specified on the run cycle **General** tab, in the **Starting date** field.

Calendar

With the **Calendar** run cycle, you use a previously defined calendar. The job stream runs on dates that are included in the calendar definition or does not run on those dates, if the rule is part of an exclusive run cycle.

In the **Offset** area, you can specify an offset from the dates that are defined in the calendar. You can choose to offset by **Day** (every day of the week), **Weekdays** (every day except Saturday and Sunday), or **Workdays** (every day of the week, except for Saturdays, Sundays, and dates that are marked in the holidays or freedays calendar).

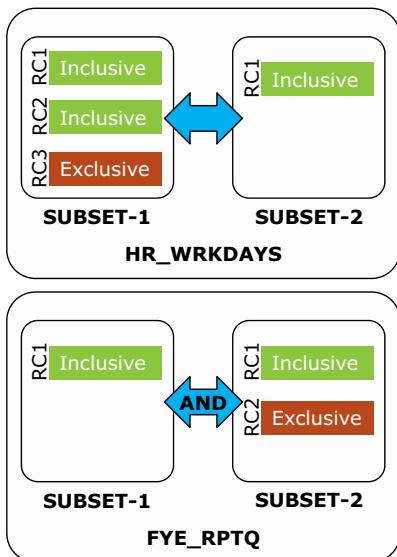
For example, if a calendar date is 15-Dec-2017, an offset of **-2 days** results in an effective date of 13-Dec-2017. For another example, if the calendar contains 22-Dec-2017 (a Friday), and the holidays calendar contains 25-Dec-2017, an offset of **+1 work days** results in an effective date of 26-Dec-2017.

Time restrictions

Each run cycle can have time restrictions. Earliest start, latest start, and deadline times determine what time any job stream by using the run cycle can run.

You can define that the earliest start time is not a time dependency by clearing the **Use time as a dependency** check box. This option is useful to place the job streams at an exact reference time to position the job stream before or after other job streams in the plan.

Creating run cycle groups



- Use the same run cycle group in many job streams
- Exclusive run cycles only exclude dates in the same subset
- Combine subsets using logical OR (union, the default) or logical AND (intersection)
- When you use logical AND, the dates must be selected in both subsets to be in the resulting run dates

© Copyright IBM Corporation 2017

Creating run cycle groups

Run cycle groups are database objects that contain one or more run cycles. A run cycle group is a list of run cycles that are combined together to produce a set of run dates. Job streams reference run cycle groups to determine when the job stream is selected in the plan. The same run cycle group can be used in any number of job streams. You define common scheduling rules in a run cycle group. You share those rules with any job stream that requires them.



Note: The result of a run cycle group is always an inclusive set of dates; it cannot be exclusive. However, you can exclude the result of a run cycle group in your job stream definition.

Using subsets

A *run cycle group* contains run cycles that are organized into subsets. Each subset contains one or more run cycles, and each run cycle group must contain at least one subset. The subsets are always in a logical OR relationship with each other. You can use any of the run cycle types that are described in [“Run cycle rule types”](#) on page 5-31 in a subset.

Enhancing exclusive run cycles with subsets

Exclusive run cycles apply only to the run cycles within the same subset. That means that you can group subsets so that exclusive, or negative dates, do not affect the positive dates in inclusive run cycles. For example, you want to run a job stream every day of the month, except the last day of

the month. However, you also want to run the same job stream on 31 December 31, which is the last day of a month.

Subsets in a group

Subset 1

- Run cycle 1 Every day, inclusive
- Run cycle 2 Last day of every month, exclusive

Subset 2

- Run cycle 3 Yearly, on 31 December, inclusive

Run cycle 2 cancels the last day of every month in subset 1, but run cycle 3 generates 31 December as a separate date. Because subsets have a logical OR relationship, 31 December is selected, but other month end dates are excluded.

Using logical AND relationships between run cycles

Run cycles in the same subset by default have a logical OR relationship with each other. For example, if the first run cycle selects all Mondays, and the second run cycle selects the fifth day of each month, the job stream selection includes both Mondays and the fifth day of the month.

You can change the relationship of run cycles in the same subset to a logical AND. For example, if the first run cycle selects all Mondays, and the second run cycle selects the fifth day of each month, the job stream selection includes only dates when Monday is the fifth day of the month.

To use an AND relationship between run cycles, at least two run cycles must be selected as **Logical And** and the result must be a positive (inclusive) set of dates.

Calculating relationships between run cycles

All of the run cycles that contain a logical AND are calculated first. The other run cycles are calculated against the result.

Creating a run cycle group

In the Workload Designer, click **Create New > Run Cycle Group** to create a run cycle group. On the General page, enter a name and description for your run cycle group. You can also enter a description in the **Description** field.

From the **Variable Table** field, you select a variable table from which jobs resolve variables when run cycles in this run cycle group select the job streams that reference the table.

To add a subset to the run cycle group, right-click the group name on the Details page, and select **Add Subset**. You can change the name of the subset by replacing the default name in the **Subset Id** field.

To add a run cycle to a subset, click **Add Run Cycles**, or right-click the subset name in the Details and click **Add Run Cycle**.

To create each run cycle, you must complete required fields on the **General** and **Rule** tabs before clicking **Save**.

Previewing run cycle groups

To see the result of the run cycles in the run cycle group, click the **Run Cycle Preview** tab. In the run cycle preview, you can filter the display to show only one run cycle, or the cumulation of all run cycles, by selecting an option in the **Display** menu.

Each day on the calendar in the preview is highlighted in a different color that depends on its selection or lack of selection by one or more run cycles. Click the **Legend** link to see a description of all the color and notation options.

Lesson 11 Creating run cycle groups by using composer

Creating run cycle groups by using composer

© Copyright IBM Corporation 2017

To create a variable table by using the command-line interface, run the `composer` command. Enter `new runcyclegroup` at the dash (-) prompt. You edit a template that contains two run cycle groups by using standard text editor commands. You can also edit an existing run cycle group by using the `composer modify runcyclegroup rcg_name` command.



Hint: You can shorten `runcyclegroup` to `rcg` in `composer`.

Editing a run cycle group in composer

```
RUNCYCLEGROUP RCG1
  ON RUNCYCLE RULE1 "FREQ=DAILY;"
END

# Sample RunCycle Group

RUNCYCLEGROUP RCG2
DESCRIPTION "Sample RunCycle Group"
VARTABLE TABLE1
FREEDAYS CAL1 -SA -SU
ON RUNCYCLE RUN_CYCLE1 VALIDFROM 03/31/2018 VALIDTO 04/12/2018 DESCRIPTION
"Inclusive Run Cycle" VARTABLE TABLE1 "FREQ=DAILY;" FDIGNORE
(AT 0700 +2 DAYS UNTIL 0900 +2 DAYS ONUNTIL SUPPR DEADLINE 1000 +2 DAYS)

EXCEPT RUNCYCLE RUN_CYCLE2 VALIDFROM 03/31/2018 VALIDTO 04/12/2018 DESCRIPTION
"Exclusive Run Cycle" CAL1 FDPREV SUBSET SUBSET_A AND
(AT 0700 +2 DAYS)

EXCEPT RUNCYCLE RUN_CYCLE3 VALIDFROM 03/31/2018 VALIDTO 04/12/2018 DESCRIPTION
"Exclusive Run Cycle" 04/01/2018 FDNEXT SUBSET SUBSET_A AND
(SCHEDTIME 0700 +2 DAYS)
SCHEDTIME 0700 TZ Europe/Berlin +2 DAYS UNTIL 0900 TZ Europe/Berlin +2 DAYS
ONUNTIL CONT DEADLINE 1000 TZ Europe/Berlin +2 DAYS
END
```

Lesson 12 Defining jobs

IBM Training



Defining jobs

© Copyright IBM Corporation 2017

In this lesson, you learn how to create the scheduling objects that job streams can reference. In this lesson, you learn how to use the Workload Designer to create job definitions. You learn how to perform the following tasks:

- Create job definitions for Windows and UNIX tasks.
- Create job definitions for file transfer and database tasks.
- Define conditions that can determine conditional branching logic.
- Define recovery options and actions for jobs that do not complete successfully.

Creating job definitions



Sample job types

- Native (Windows, UNIX, Remote) system commands or scripts
- File transfer and coordination (FTP, Shadow)
- Database and integrations (MSSQL, SQL queries, or updates)
- Java (EJB)
- Web services (WSDL, RESTful)

Use variables in job definitions

- **^VARIABLE^** resolved during submission
- **\${VARIABLE}** resolved at run time (dynamic agents only)
- Path names
- Environment variables
- Logon names
- Other fields in specific types (file name in File Transfer jobs, SQL SELECT clause in Database jobs)

© Copyright IBM Corporation 2017

Creating job definitions

A *job* is the basic unit of work that specifies what action to start on a Workload Scheduler agent in the network. Jobs definitions are independent of job streams and other scheduling objects.

Workload Scheduler tracks the history of jobs so you can report on them when necessary.

Jobs do not have scheduling properties such as start times, predecessors, or run cycles. They are assigned those properties as part of their association with job streams. You can use the same job in many different job streams.

Workload Scheduler supports *multiple* types of jobs that run in many environments and operating systems. Many job types perform specific IT functions, such as file transfer or database queries. You can also define your own job types by authoring job type plug-ins or downloading them from IBM's Integrated Service Management library.

It is helpful to categorize the various job types in different ways. One way is to categorize them by where each type of job can run.

Exploring different job types

Jobs types with advanced options can run only on dynamic agents, pools, or dynamic pools. They cannot run on fault-tolerant agents. Jobs types that can run on fault-tolerant agents and dynamic agents are Native (UNIX and Windows) and ERP Extended Agent jobs.

Native executable jobs have many more options than native jobs. In an executable job, you can define environment settings, environment variables, and input and output streams. You can also embed a shell script directly in the job definition.



Note: Only Native Windows, Native UNIX, and Native Other (extended agent) are jobs that run on fault-tolerant agents.

Job types that are available in Workload Scheduler

- Windows
- UNIX
- Executable (Windows or UNIX)
- IBM I
- z/OS
- Remote command
- Shadow
- File transfer
- Java and Java EE
- Database
- Web Services (WSDL)
- Workload Broker
- Provisioning
- Shadow Distributed, Shadow z/OS
- Hadoop Distributed File System
- Centralized Agent Update
- Job Management
- Job Stream Submission
- Variable Table
- JSR 352 Java Batch
- MQTT
- RESTful

© Copyright IBM Corporation 2017

Job types that are available in Workload Scheduler

Several new job types were added to Workload Scheduler with version 9.3 and 9.4. A summary of new job types is listed in [Table 2](#).

Table 2 Job types newly available in version 9.4.

Job type	Description
Centralized agent update	You can define a job to update one or more (up to 20) Workload Scheduler agents with new versions or fix packs. You can later schedule the <i>centralized agent update</i> job to run in context with other jobs in the workflow.
IBM WebSphere MQ	You can define, schedule, and monitor jobs to manage WebSphere MQ message queues. You can manage mixed processes, where multiple background workflows synchronize their data in real-time using message queues.
RESTful web services	You can schedule jobs that add, download, delete, and modify resources or data on RESTful Web services by using HTTP methods in any available content type such as JSON, XML, and XHTML.
SAP BusinessObjects	You can define, run, and monitor your SAP BusinessObjects Business Intelligence reports in the Workload Scheduler environment.
Netezza Performance Server	You can define, schedule, and monitor Workload Scheduler jobs that perform Netezza Performance Server bulk data load operations. You can condition the load of data upon successful execution of preliminary manipulation or delay data loading until the updates are included. The support for Netezza is integrated with the <i>Database</i> job type.

Table 2 Job types newly available in version 9.4.

Job type	Description
IBM Sterling Connect:Direct	An IBM Sterling Connect:Direct job runs IBM Sterling Connect:Direct programs to transfer one or more files from a primary node to a secondary node. You can automate the entire file transfer process, increasing the success of any subsequent processing like decryption, renaming, parsing, and retransmission of those files.
Salesforce	IBM Workload Scheduler integrates with Salesforce to automate, monitor, and control workflows that contain batch APEX jobs. You gain greater control of your Salesforce jobs with both calendar-based and event-based workload automation and have a single point of control to handle exceptions and automate.
Apache Hadoop	<p>IBM Workload Scheduler provides three plug-ins to simplify and automate your Apache Hadoop batch processing:</p> <ul style="list-style-type: none"> Define, schedule, monitor, and control the execution of Hadoop MapReduce procedures by using the plug-in for <i>Hadoop Mapreduce</i>. Access the Hadoop Distributed file system from any computer, and work on files and directories by using the plug-in for <i>Hadoop Distributed File System</i>. A plug-in for <i>Apache Hive Database</i> that provides you with an SQL-like interface to Hadoop.
IBM BigInsights	<p>With the job plug-in for BigInsights for Hadoop you can:</p> <ul style="list-style-type: none"> Monitor and control workflows that contain BigInsights workbooks and applications. Fully automate BigInsights process execution with calendar and event-based scheduling, and a single point of control to handle exceptions, and automate recovery processes.

Some of the many available job types that are installed with Workload Scheduler are listed in the following table.

Native	
Windows	Run general commands and scripts on Microsoft Windows agents.
UNIX	Run scripts and commands on UNIX and Linux agents.
Executable	Run tasks on Windows or UNIX environments. In the Executable job type, you can specify how to modify the runtime environment of the job. You can also embed shell scripts and commands directly into the job definition.
Remote Command	Run tasks on a computer that is not running a Workload Scheduler agent.
File Transfer and Coordination	

Shadow	Map the status of a job that runs in a different Workload Scheduler network to a job in the current network. Shadow jobs are useful for using jobs in another network as dependencies.
File Transfer	Send or retrieve a single file to or from a remote file server.
Database and Integrations	
Java	Run Java jobs in a Java container.
Database	Run Microsoft SQL jobs or SQL queries and updates.
Web Services	Call web service description operations.

File transfer and coordination jobs can run only on dynamic agents.

Defining native jobs

Native jobs are the most common types of jobs in Workload Scheduler. They are useful for most kinds of tasks that involve calling a local operating system command or running a shell script directly on the system where the agent runs.

To create a native UNIX job, click **Create New > Job Definition**, and click **UNIX** in the **Native** section. To create a native Windows job, click **Create New > Job Definition**, and click **Windows** in the **Native** section. On the **General** tab, complete the fields that are listed in the following table.

Name	The name of the job must start with a letter and be 40 characters or fewer.
Workstation	The workstation where the job runs. You can click the Search icon to find the workstation name from the list of available workstations in the database.
Login	The name of the operating system user that starts the job. You must specify a user that can log on to the workstation where the job runs. For jobs that run on Windows workstations, you must also define the user in the database. The login name can contain up to 47 characters. If the name contains special characters, you must enclose the name in quotation marks ("").
Description (optional)	Include a description of the job.
Successful Output Conditions	You have better control over job stream execution if you specify multiple return codes to define a successful job. In the Successful Output Conditions section, you can create a logical expression that specifies a success return code range. <i>Successful output conditions</i> , if met, force the job to end with a success status.

Other Output Conditions	Other output conditions, if met, do not change the completion status of the job. You use the condition name to qualify conditional dependencies on the job within job streams.
--------------------------------	--

Condition name

Enter the unique name of the output condition, that can be used when defining conditional dependencies in job streams. You cannot use condition names which Workload Scheduler predefines, such as **SUCC**, **ABEND**, or **DONE**, for example.

Condition value

Define return code conditions that are Boolean expressions to determine the return codes (RC) that are required to consider a job successful or set the output condition. This expression can contain a combination of comparison operators (=, <>, !=, >, <, >=, <=) and logical operators (AND, OR, NOT).

For example, to define a return code of 3, 5, 6, 7, 8, or 9 as successful, create the following line:

```
RC = 3 OR (RC >= 5 AND RC < 10)
```

Be careful to add spaces around the operators and use the parentheses. If a job starts and quits abnormally, it has a status of ABEND. If a job does not start at all, it failed, and no return code was evaluated.



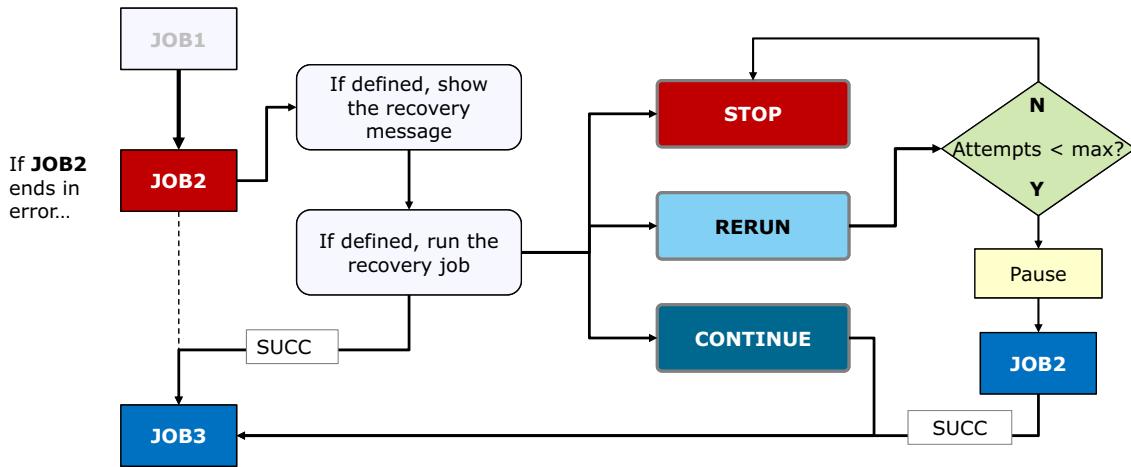
Hint: For security purposes, the login field should be a user who has only privileges necessary to run the job. Do not use **root**, **Administrator**, or the Workload Scheduler **install** users as general batch job stream login users.

On the Task tab, select a **Command** (operating system) or a **Script** (any type). Enter the script or command into the text box and any arguments. The text box can contain up to 4095 characters. You can use one or more variables in all or part of the text box.

When specifying a Windows job, include file extensions. You can use Universal Naming Convention (UNC) names. For Windows jobs, the Interactive check box indicates that the job runs interactively on the Windows desktop. On the **Task** tab, complete the fields that are listed in the following table.

Command or Script	The file name and any options and arguments.
Add variable...	Click when you want to use a variable in an existing variable table to define part of your task.

Job recovery options



© Copyright IBM Corporation 2017

Job recovery options

Use the **Recovery options** tab to specify the behavior of the system if the job that you configured does not end successfully. Use the recovery action to stop the job stream, continue, or start again.

The operator sees the message that is displayed as a recovery prompt and must click **Reply Yes** to start the job. The job specifies a recovery job that can run in place of the original job if the original job completes unsuccessfully.

The *message* is a local prompt that is displayed if the original job completes unsuccessfully. Message text can contain up to 64 characters and can use variables, for example, a pager number in a variable can differ by shift. The operator must respond to the recovery prompt before Workload Scheduler proceeds to other recovery actions or options.

A recovery job is run in place of the original job if it completes unsuccessfully. The recovery job must already be defined in the Workload Scheduler database.

If the recovery job completes unsuccessfully, processing stops unless a *Continue* action is specified. Recovery jobs run only once for each stopped instance of the parent job. Job recovery options are ignored when a job is run as a recovery job.

The recovery job runs on the workstation. The *Workstation* field is completed automatically when you select a recovery job. The workstation of the recovery job must be in the same domain as the workstation of the parent job.

When you select the **Rerun** action, you can specify the number of attempts that Workload Scheduler should retry the job before it ends successfully. You can also specify that Workload Scheduler waits a number of minutes before it starts the job again.

When the job runs on a pool or dynamic pool workstation, you can specify that recovery actions run on the same agent as the original job.

Defining native executable jobs

Native executable jobs have many more options than native jobs. In an executable job, you can define environment settings, environment variables, and input and output streams. You can also embed a shell script directly in the job definition.

To create a native executable job, click **Create New > Job Definition**, and click **Executable** in the **Native** section.

Completing the General tab

On the **General** tab, complete the fields that are listed in the following table.

Name	The name of the job must start with a letter and be 40 characters or fewer.
Workstation	The workstation where the job runs. You can click the Search icon to find the workstation name from the list of available workstations in the database.
Description (optional)	Include a description of the job.
Variable resolution at run time (optional)	Select this option if variables should be resolved at run time. For more information, see “Using variables in job definitions” on page 5-55.

Successful Output Conditions	You have better control over job stream execution if you specify multiple return codes to define a successful job. In the Successful Output Conditions section, you can create a logical expression that specifies a success return code range. <i>Successful output conditions</i> , if met, force the job to end with a success status.
-------------------------------------	--

Condition name

Enter the unique name of the output condition, that can be used when defining conditional dependencies in job streams. You cannot use condition names which Workload Scheduler predefines, such as **SUCC**, **ABEND**, or **DONE**, for example.

Condition value

Define return code conditions that are Boolean expressions to determine the return codes (RC) that are required to consider a job successful. This expression can contain a combination of comparison operators (=, <>, !=, >, <, >=, <=) and logical operators (AND, OR, NOT).

For example, to define a return code of 3, 5, 6, 7, 8, or 9 as successful, create the following line:

RC = 3 OR (RC >= 5 AND RC < 10)

Ensure that you put the spaces around the operators and use the parentheses. If a job starts and quits abnormally, it has a status of ABEND. If a job does not start at all, it failed, and no return code was evaluated.

Expanded return code mappings

With advanced job types, you can use expanded return code mappings, as explained in "[Using expanded return code mappings](#)" on page 5-48

Other Output Conditions	Other output conditions, if met, do not change the completion status of the job. You use the condition name to qualify conditional dependencies on the job within job streams.
--------------------------------	--

Condition name

The unique name of the output condition, that can be used when defining conditional dependencies in job streams. You cannot use condition names which Workload Scheduler predefines, such as **SUCC**, **ABEND**, or **DONE**, for example.

Condition value

Define return code conditions that are Boolean expressions to determine the return codes (RC) that set the output condition. This expression can contain a combination of comparison operators (=, <>, !=, >, <, >=, <=) and logical operators (AND, OR, NOT).

Expanded return code mappings

With advanced job types, you can use expanded return code mappings, as explained in [“Using expanded return code mappings”](#) on page 5-48

Using expanded return code mappings

In the **Return code mapping expression** field, in addition to return code mappings explained on [page 5-47](#), you can also specify to have the job complete successfully when variables other than the return code evaluate as true.

You can set the success condition by analyzing the output that the job creates. Check the contents of the **this.stdlist** variable. For example, to set the job as successful when the job output contains the word “Complete”, use the following return code mapping expression.

```
contains(${this.stdlist}, "Complete")
```

You can analyze the job properties by examining the properties of the job, or another job in the same job stream. Properties that you can reference are listed in the documentation at the following location.

http://www.ibm.com/support/knowledgecenter/SSRULV_9.3.0/com.ibm.tivoli.itws.doc_9.3/distr/src_ref/awsrgpassjobprop.htm?cp=SSGSPN_9.3.0&lang=en

The functions that you can use for return code mapping expressions are **XPath (XML Path Language)** functions. Some examples of functions you can use are in the following list.

- String comparisons (starts-with, contains, string-length, and so on)
- String manipulations (substring, substring-before, translate, and so on)
- Boolean operators (not, true, false, and so on)
- Numeric comparisons (=, !=, >, <, and so on)
- Numeric functions and operations (sum, floor, ceiling, round, and so on)

Completing the Task tab

On the **Task** tab, select a **Command** or **Inline script**.

If you choose **Command**, perform these steps:

1. Enter the script or command into the field. You can use one or more variables in all or part of the field. When specifying a Windows job in the Command field, include file extensions. You can use Universal Naming Convention (UNC) names.
2. Do not enter any arguments in the command field. You add arguments in extra fields that follow.

If you choose **Inline script**, perform these steps:

1. Enter the entire script into the **Script body** field. The script must be 16,383 characters or fewer.
2. In your script, you can include one or more variables from variable tables that you create, if you want. For more information about variables in job definitions, see [“Using variables in job definitions”](#) on page 5-55.
3. If the script runs on Windows operating systems, type the file name extension for the script when it runs on the agent. Do not insert a period in the extension.

Add any arguments that you want to pass to the script or command. Click the plus (+) symbol to specify the name and value of one or more arguments. You can use variables for all or part of any arguments you want to pass to the command or script.

Completing the Affinity tab (optional)

Jobs that are defined to run on dynamic pool workstations can run on different computers, depending on how the workstation is defined. To have the job that you define run on the same computer as another job, enter the job name of another job on the same workstation.

Completing the Recovery Options tab (optional)

The recovery options for executable type jobs are the same as the options for native UNIX and native Windows job types. For more information, see [“Job recovery options”](#) on page 5-45.

Completing the Credentials tab

By default, executable type jobs that are run as the user under which the agent is installed on the computer. If you want the job to run as a different user, complete the fields on the **Credentials** tab.

- Specify the **Group** to be used when running the command. Use this field if the user who must run the script or command belongs to a user group defined on the operating system.
- Specify a **User** who is defined on the target system. The command runs by using this user name. Use this field to run the program or script with a user account that is different from the user account under which the agent runs.

Specify the **Password** to be used when running the command. Use this field if a password is required for running the program or script. Click the ellipsis to display other password type options. Select one of the options that are listed in the following table.

Password	Use the password value that is entered in the Password field.
User	Resolve the password at run time by using the password value that is defined for User Name in the Workload Scheduler database. You can also specify the user and the related password of another workstation if the user is defined in the database.
Agent User	Resolve the password at run time with the password value defined for <i>User Name</i> locally on the dynamic agent that runs the job (or on any agent of a pool or dynamic pool that can run the job) with the <code>param</code> command.
Variable	<p>Resolve the variable at run time with the value defined for the variable that you enter in the field by using the <code> \${variable_name}</code> notation. You must define the variable either locally on the agent, by using the <code>param</code> command, or in the Workload Scheduler database, for example:</p> <p>If a variable is defined locally on the agent, enter the field as <code> \${agent:file_section.password.secret}</code></p> <p>If a variable is defined in the database, enter it as <code> \${password:workstation#user}</code>. You can specify the password of the remote user of a different workstation (if it was defined in the database) by entering the following string in the adjacent field: <code> \${password:other_workstation#user}</code></p> <p>Variables are resolved both when you generate a plan and when you submit a job or a job stream. While defining jobs, the variables are not resolved and cannot be used in lists or for test connections.</p>



Important: User definitions lack referential integrity. If a user definition in the credentials section is changed or deleted, no warning or error message is returned until the job runs.

Completing the Environment tab (optional)

On the **Environment** tab, you can specify the standard input, standard output, and standard error files for the command or script. The fields on this tab can contain one or more variables. You can complete the fields that are listed in the following table.

Working directory	Specify the working directory where the job should run. If you do not specify a working directory, on UNIX systems, the job runs in the \$HOME directory of the login user. On Windows systems, the job runs in the dynamic agent's working directory.
Standard input	The file the job should use as input, if required. Enter a relative or absolute path name to the file. You can also use the output of another job in the same job stream as the input to the job you define. To use another job's output as the input stream, enter the following syntax. \${job:JOBNAME.stduri}
Standard output	The file where the job should copy its output, if required. Enter a relative or absolute path name to the file.
Standard error	The file where the job should copy its error messages, if required. Enter a relative or absolute path name to the file.
Interactive	For Windows jobs, the Interactive check box indicates that the job runs interactively on the Windows desktop.
Environment variables	In this section, specify the name and value of any environment variables the job should set before running the command or inline script. Click the plus (+) symbol to enter a new environment variable. The name and value fields can contain variables from your variable tables, and Workload Scheduler variables that the scheduler defines.

Defining file transfer jobs

File transfer jobs have options for moving a file from one computer to another. In a file transfer job, you define which file to transfer and in which direction, whether to upload from the agent or download to the agent. You can transfer a file to or from the agent on which the job runs by using one of several available transfer protocols.

To create a file transfer job, click **Create New > Job Definition**, and click **File Transfer** in the **File Transfer and Coordination** section.

Completing the General tab

On the **General** tab, complete these fields:

Name	The name of the job must start with a letter and be 40 characters or fewer.
Workstation	The workstation where the job runs. You can click the Search icon to find the workstation name from the list of available workstations in the database.
Return code mapping expression (optional)	Create a logical expression that specifies a success return code range. You can use expressions that are described in “Using expanded return code mappings” on page 5-48.
Description (optional)	Include a description of the job.
Variable resolution at run time (optional)	Select this option if variables should be resolved at run time. For more information, see “Using variables in job definitions” on page 5-55.

Completing the File Transfer tab

Follow these steps to complete the fields on the **File Transfer** tab:

1. On the **File Transfer** tab, select the transfer type.
 - Select **Download** to transfer the file from the server you specify in the **Server** field to the computer that is running the Workload Scheduler agent. You can also optionally specify the permissions to apply to the file on the agent when the download completes. Specify the permissions in **Octal Notation**, for example, 0644, to allow read and write permissions to the owner, and read permissions to group members and other users.
 - Select **Upload** to transfer the file from the computer that is running the Workload Scheduler agent to the server you specify in the **Server** field. You can also select **Delete source files after transfer** to automatically remove the files that the agent transferred.
2. In the **Server** field, enter the DNS name or IP address of the remote server. All or part of the **Server** field can be a variable.
3. Enter the path to the remote file in the **Remote file** field. When you choose **Download**, the path is the source file. When you choose **Upload**, the path is the destination file. All or part of the **Remote file** field can be a variable.
You can use asterisks or question marks as wildcards when downloading files. If you want to maintain the same file names, start the Remote file path with two backslashes (\\\) or end the part with a forward slash (/).
4. Enter the path to the local file in the **Local file** field. When uploading, the path is the source file. When downloading, the path is the target destination. All or part of the **Local file** field can be a variable.
You can use asterisks or question marks as wildcards when uploading files.
5. Choose the file transfer protocol from the options in the following table.

FTP	File Transfer Protocol (FTP) is the standard Internet file transfer protocol.
FTPS	Secure File Transfer Protocol (FTPS) is an extended file transfer protocol. It includes <i>implicit</i> SSL/TLS encrypted FTP that works similarly to HTTPS. This protocol is considered deprecated, but still widely used. TLS protocol Version 1 is supported. The SSL session reuse configuration is not supported. If you specify this protocol, only the user and password authentication is supported.
FTPES	Explicit Secure File Transfer Protocol (FTPES) is another extended file transfer protocol. It provides <i>explicit</i> SSL/TLS encrypted FTP. The connection starts as plain FTP over port 21, but switches to an SSL/TLS connection. This protocol is newer than FTPS. If you specify this protocol, only the user and password authentication is supported.
SSH	This network protocol provides file access, file transfer, and file management functions over any data stream. When transferring ASCII files, the local and remote code pages are identified automatically. SSH protocol is used when you run the <code>scp</code> command from a UNIX or Linux environment, or a program such as WinSCP on Microsoft Windows.
WINDOWS	This selection is the Microsoft file sharing protocol. When you specify this protocol, use the Samba syntax to specify the path of the remote file, for example, <code>//fileserv/shared1/xmit/data.csv</code> . The folder that contains the remote file must be shared on the remote server.
Auto	Automatically detect between SSH and Windows protocols. The agent tries the SSH protocol first; then, it tries the Windows protocol.

6. Complete the **User name** and **Password** fields for the remote credentials. You learned about these fields in [“Completing the Credentials tab”](#) on page 5-50.
7. Complete the **User name** and **Password** fields for the local credentials.
8. Optionally, provide certificate information if you use encrypted transfer protocols.

Completing the Transfer Option tab (optional)

On the **Transfer Options** tab, you can specify further file transfer options that you might to correctly connect to the remote server and convert the file as it transfers. You can complete the fields that are listed in the following table.

Transfer Mode	Specify that the file must be transferred in binary or text mode.
Code page conversion	Specify the conversion from local to remote code pages during the transfer. The default timeout is 60 seconds.

Connection mode	Specify that the file transfer use only Active Mode (the server establishes the connection to the client, the default) or Passive Mode (the client establishes the connection with the server).
Port range	Specify the FTP PORT range to use. This option might be required when transferring files past highly restrictive firewalls. You can specify a minimum port value, maximum port value, or both.

Defining database jobs

Database jobs run database queries or stored procedures. In a database job, you define which database server to connect, and the SQL or procedure to run.

To create a database job, click **Create New > Job Definition**, and click **Database** in the **Database and Integrations** section.

Completing the General tab

On the **General** tab, complete the fields that are listed in the following table.

Name	The name of the job must start with a letter and be 40 characters or fewer.
Workstation	The workstation where the job runs. You can click the Search icon to find the workstation name from the list of available workstations in the database.
Return code mapping expression	Create a logical expression that specifies a success return code range.
Description (optional)	Include a description of the job.
Variable resolution at run time (optional)	Select this option if variables should be resolved at run time. For more information, see "Using variables in job definitions" on page 5-55.

Completing the Database tab

Follow these steps to complete the fields on the **Database** tab.

1. On the **Database** tab, select the database management system by choosing **Predefined** or **Custom**.
2. If you chose **Predefined**, complete the following fields:
 - a. Select the **DBMS** option the job should use. Valid selections are **Microsoft SQL Server**, **Oracle DBMS**, **IBM DB2**, **IBM Netezza**, **BigSQL**, or **Hive**.
 - b. Enter the name of the database to which the job connects in the **Database name** field.
 - c. Enter the host name or IP address of the database server in the **Server** field.

- d. Enter the database port number in the **Port number** field.
3. If you chose **Custom** for the **Database management system** option, you can specify values to connect to custom databases with integration not provided with the IBM driver. Complete these two fields.
- a. The **JDBC driver class name** field contains the Java name of the JDBC driver class, for example:
`com.mysql.jdbc.Driver`
 - b. The **JDBC connection string** is a connection string that contains the host name, port number, and database name, for example:
`jdbc:mysql://mysqldbhost:3306/mysql`
4. If required, enter the path name to the database Java drivers in the **JDBC jar class path**.
5. Complete the **User name** and **Password** fields for the database credentials. Review the instructions about how to complete these fields in "[Completing the Transfer Option tab \(optional\)](#)" on page 5-53.
6. Click **Test connection** to verify that the fields are entered correctly.

Completing the SQL tab

On the **SQL** tab, you specify which database commands or procedures to run. All or parts of the fields on the **SQL** tab can be variables. You can choose **Standard SQL** or **Stored Procedure**. If you choose **Standard SQL**, type or paste your SQL statements into the **SQL** field. The exact syntax of the SQL that you use depends on the type of database management system that you configure on the **Database** tab.

If you chose **Stored Procedure**, enter the name of the DB2, Oracle, MSSQL, or Netezza Performance Server procedure for the job to run. You must also enter procedure parameters for the stored procedure to call. Click **Lookup Parameters** to request a list of valid parameters for the procedure from which you can select. You can also add parameters manually by clicking the green plus (+) sign and filling the **Name** and **Value** fields.

Using variables in job definitions

You can use variables that you create in variable tables in your job definitions.

Using variables that are resolved when the plan is extended

Variables are resolved by the plan extension process or when the job or job stream is submitted, either by a user in the Dynamic Workload Console or conman command line, or by an event rule.

To specify a variable that is resolved during plan extension, you must enter the name of the variable, and enclose it in caret (^) characters, for example, ^FILENAME^.

In Native UNIX and Native Windows job definitions, and job definitions that apply to fault tolerant agents, only plan-generated variables are resolved. Fields that can contain variables have a **Variables** option that you can choose to retrieve a list of variables in the database.

Using variables that are resolved when the job runs

In advanced job types, you can specify the variable in two ways:

1. Variables that are resolved when the plan is extended, or when the job is submitted, as previously described.
2. Variables that are resolved when the job runs. Use this option if you want the variables to be resolved just before the job runs, rather than at the beginning of the production day.

To use variables that are resolved when the job runs, you must perform the following steps:

1. Define the variables in a variable table other than the default variable table.
2. On the **General** tab, select the **Variable resolution at runtime** option.
3. Specify the variable in the job definition field by using this notation: `${variable_name}`.

Using automation utility jobs

Variable table

Add a variable to a variable table, or update an existing variable

Job stream submission

Restart the same job stream, or submit a new one

Centralized agent update

Apply software updates to Workload Scheduler agents

Job management

- Rerun the job
- Rerun the job and all its successor jobs
- Rerun the job and its successor jobs in the same job stream
- Release the job from a manual hold
- Release the job from its dependencies
- Cancel the job
- Cancel the job with the pending option
- Place the job on hold
- Terminate (Kill) the job
- Confirm the job as ABEND
- Confirm the job as SUCC

Using automation utility jobs

Automation utility jobs provide ways to automate Workload Scheduler functions such as submitting a job stream or canceling a job within your schedule workflows. Four types of automation utility jobs are available.

Centralized agent update

Use this type of job to schedule software updates on the Workload Scheduler agents.

Job Management

Use this type of job to perform actions on jobs that run in the Workload Scheduler environment. You can start the following actions in a Job Management type of job:

- Rerun the job
- Rerun the job and all its successor jobs
- Rerun the job and its successor jobs in the same job stream
- Release the job from a manual hold
- Release the job from its dependencies
- Cancel the job
- Cancel the job with the pending option
- Place the job on Hold
- Terminate (Kill) the job
- Confirm the job as ABEND
- Confirm the job as SUCC

Job Stream Submission

Use this type of job to submit a job stream to run as part of the plan. You can specify which job stream to run, or submit the same job stream where the Job Stream Submission type of job run.

Variable Table

Use this type of job to add a variable to a variable table, or change the value of a variable that exists in the variable table.

Lesson 13 Creating job definitions by using composer

Creating job definitions by using composer

© Copyright IBM Corporation 2017

To create a job definition by using the command-line interface, run the `composer` command. Enter `new job` at the dash (-) prompt. You edit a template that contains an example of each job type by using standard text editor commands. You can also edit an existing job definition by using the `composer modify job workstation#jobname` command.

Editing a job definition in composer

\$JOBS

FTA1#JOBDEF1

DOCOMMAND "dir"
STREAMLOGON Administrator
TASKTYPE WINDOWS
RECOVERY STOP

FTA1#JOBDEF2

SCRIPTNAME "myscript.sh"
STREAMLOGON root
DESCRIPTION "Sample Job Definition"
TASKTYPE UNIX
INTERACTIVE
RCCONDSUCC "RC>=0"
RECOVERY CONTINUE
AFTER FTA1#JOBDEF1
ABENDPROMPT "Do you want to continue ?"

AGENT1#JOBDEF3

TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
<jsdl:application name="executable">
<jsdle:executable interactive="false">
<jsdle:script>dir</jsdle:script>
</jsdle:executable>
</jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Sample Job Definition"
RCCONDSUCC "RC>=0"

REMENG1#Z_SHADOW_JOB

TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:zshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/zshadow">
<jsdl:application name="zShadowJob">
<zshadow:ZShadowJob>

```

<zshadow:JobStream>JS1</zshadow:JobStream>
<zshadow:JobNumber>1</zshadow:JobNumber>
<zshadow:matching>
<zshadow:previous/>
</zshadow:matching>
</zshadow:ZShadowJob>
</jsdl:application>
</jsdl:jobDefinition>

DESCRIPTION "Sample Shadow Job Definition for ZOS environment"

REMENG2#D_SHADOW_JOB
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:dshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow">
<jsdl:application name="distributedShadowJob">
<dshadow:DistributedShadowJob>
<dshadow:JobStream>JS1</dshadow:JobStream>
<dshadow:Workstation>FTA1</dshadow:Workstation>
<dshadow:Job>JOB1</dshadow:Job>
<dshadow:matching>
<dshadow:previous/>
</dshadow:matching>
</dshadow:DistributedShadowJob>
</jsdl:application>
</jsdl:jobDefinition>

DESCRIPTION "Sample Shadow Job Definition for distributed environment"

ZOSAGENT#JCLDEF
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdljcl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdljcl">
<jsdl:application name="jcl">
<jsdljcl:jcl>
<jsdljcl:JCLParameters>
<jsdljcl:jcl>
<jsdljcl:byRefOrByDef>
<jsdljcl:byDefinition>
<jsdljcl:jclDefinition>//NORMAL JOB , 'TWS JOB', CLASS=A,MSGCLASS=A,
// MSGLEVEL=(1,1)
//*
//STEP1 EXEC PGM=IEFBR14</jsdljcl:jclDefinition>

```

```
</jsdl:jcl:byDefinition>
</jsdl:jcl:byRefOrByDef>
</jsdl:jcl:jcl>
</jsdl:jcl:JCLParameters>
<jsdl:jcl:JOBParameters>
<jsdl:jcl:jobStreamName>${tws.jobstream.name}</jsdl:jcl:jobStreamName>
<jsdl:jcl:inputArrival>${tws.job.ia}</jsdl:jcl:inputArrival>
</jsdl:jcl:JOBParameters>
</jsdl:jcl:jcl>
</jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Sample JCL Job Definition"
```

Instructor demonstration

Creating scheduling objects demonstration



© Copyright IBM Corporation 2017

Instructor demonstration

The instructor demonstrates how to start the Workload Designer and create scheduling objects.

Exercises

- Open and use the Workload Designer
- Create scheduling objects
- Create jobs



© Copyright IBM Corporation 2017

Exercises

Perform the exercises for this unit.

Review questions

1. Which function do you use to duplicate a scheduling object definition?
2. How many versions of a scheduling object can you compare in one viewing?
3. For which operating systems must you create user objects to run jobs?
4. What is the name of the default variable table?
5. In how many tables can the same variable name exist?
6. In the order of variable tables from which variables are resolved, which variable table is second-to-last, before the default variable table?
7. Which type of run cycle prevents a job stream from starting on a set of days?
8. When you use the **Freeday Next** rule, which day is included after a nonworking day?
9. In a job definition, when is a variable that is notated with carets (^) evaluated?
10. In a job definition, when is a variable that is notated with \${} evaluated?
11. When does a recovery job run?
12. When you specify the rerun option, how many times does the job automatically rerun?

Review answers

1. Which function do you use to duplicate a scheduling object definition?

Use the create like function to duplicate a scheduling object.

2. How many versions of a scheduling object can you compare in one viewing?

You can compare two versions of an object.

3. For which operating systems must you create user objects to run jobs?

You must create user objects to run jobs on Windows computers.

4. What is the name of the default variable table?

The default variable table is MAIN_TABLE.

5. In how many tables can the same variable name exist?

Variables can exist in as many variable tables as you need.

6. In the order of variable tables from which variables are resolved, which variable table is second-to-last, before the default variable table?

The variable table that is specified in the workstation definition is evaluated before the default variable table.

7. Which type of run cycle prevents a job stream from starting on a set of days?

Exclusive run cycles identify the dates when a job stream would normally be scheduled, but should not run.

8. When you use the **Freeday Next** rule, which day is included after a nonworking day?

The Freeday Next rule specifies to include the next work day after the selected nonworking day.

9. In a job definition, when is a variable that is notated with carets (^) evaluated?

Variables that are notated with carets are evaluated and resolved during job submission.

10. In a job definition, when is a variable that is notated with \${} evaluated?

Variables that are notated with \${} are evaluated and resolved at run time, only on dynamic agents.

11. When does a recovery job run?

Recovery jobs run in place of the original job if it completes unsuccessfully.

12. When you specify the rerun option, how many times does the job automatically rerun?

The job reruns once, unless you specify a retry count.

Summary

You now should be able to perform the following tasks:

- Open and use the features of the Workload Designer
- Create scheduling object definitions
- Create job definitions of several types

© Copyright IBM Corporation 2017

Unit summary

Unit 6 Scheduling, forecasting, and migrating workloads

IBM Training



Scheduling, forecasting, and migrating workloads

© 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 define Workload Scheduler job streams and event rules. You learn how to create various definitions and rules. You also learn how to forecast future work and migrate workload definitions by using workload application templates.

Objectives

In this unit, you learn how to define Workload Scheduler job streams, application templates, and event rules. You learn how to perform the following tasks:

- Create job streams to schedule work in advance
- Create event rules to provide workload automation
- Forecast future work
- Export and import scheduling definitions

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Creating job streams

Creating job streams

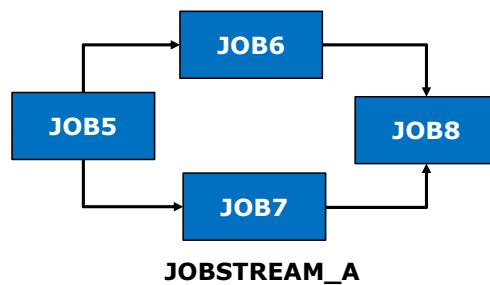
© Copyright IBM Corporation 2017

Job streams, also known as *schedules*, are central to the management of batch job execution. During the daily planning process, only job streams are evaluated for their eligibility to be included in the selected planning period. The other scheduling objects are included in the plan only if they relate to one of the job streams included the planning cycle.

In this lesson, you learn how to create a job stream by learning its overall structure, followed by detailed information about adding dependency objects such as prompts, resources, and jobs. You learn to set dependency property values and job stream attributes.

Planning job streams

- Workstation
- General details
- Scheduling options
- Time restrictions
- Start conditions
- Run cycles
- Job stream dependencies
- Jobs and their dependencies



Job stream considerations

- All jobs in a job stream are scheduled to start on the same run dates
- No limit on the number of jobs within a job stream
There can also be zero jobs in a job stream
- No limit on the number of job streams that can be scheduled
- Planning the job stream flow in advance increases productivity and reduces error

© Copyright IBM Corporation 2017

Planning job streams

The job stream is like a parent with child jobs. Jobs are grouped into streams typically because they are all related and run as a unit, such as a series of server maintenance jobs. The job stream uses a run cycle to specify the dates and times to run the job stream. The job stream also has dependencies that can be tied to a parent job stream, child jobs, or both.

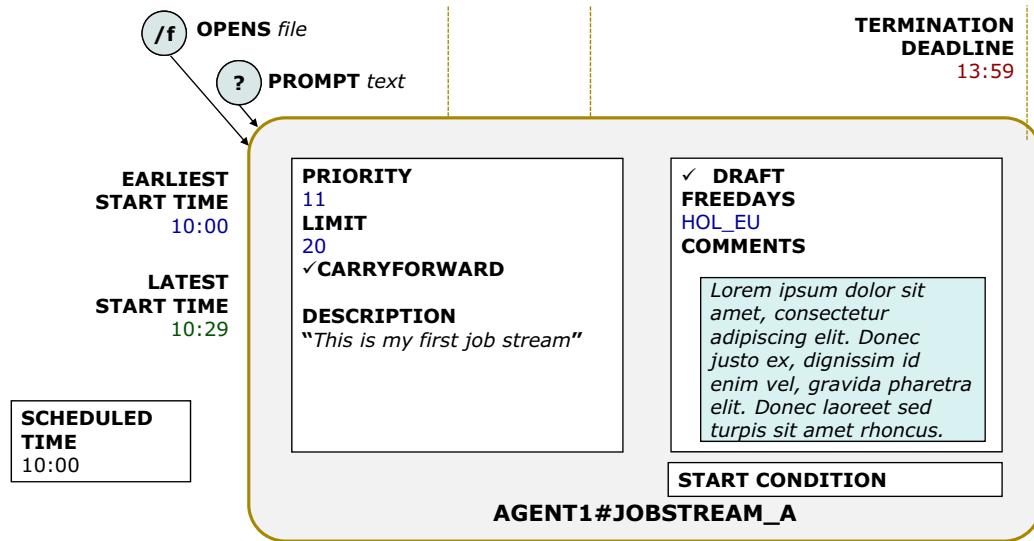
When planning your workflow, keep in mind the characteristics of both Workload Scheduler and workflow management.

- Consider putting jobs with interworkstation dependencies in the same job stream.
- Workload Scheduler does not run jobs on different days in the same job stream. For example, a job stream cannot have jobs that run every Wednesday and other jobs that run every third Thursday.
- Use separate job streams to follow your company functional organization. For example, have job streams for the Accounting Department and different job streams for the Personnel Department.
- There is no limit to the number of jobs you can manage in each job stream.

As an operator or a scheduler, your goal is to maximize system usage by finishing as many jobs as possible in the least amount of time. Constructing a processing flow diagram helps you to achieve this goal and provides the following benefits:

- Planning job stream flow increases productivity and reduces error messages.
- A diagram makes a visual reference for processing flow and aids in documenting job streams.
- By using a diagram, you see the data that is needed and the questions to ask before implementing the job streams.

Building the job stream



© Copyright IBM Corporation 2017

Building the job stream

Creating and editing job streams can seem daunting and difficult to learn. The many options that are spread throughout the editor screens can be confusing. Concentrate on the basics before attempting advanced features.

You create job streams by using the graphical *Workload Designer* in the Dynamic Workload Console or with the `composer` command-line utility.

Follow these steps to create a new job stream:

1. Start the Workload Designer in the Dynamic Workload Console portfolio by opening **Administration > Manage Workload Definitions**.
2. Select your engine and click **Go**.
3. In the Workload Designer, click **Create New** and select **Job Stream**.

The upper part of the data area contains a **Details** tab, which summarizes the job stream, jobs, and dependencies in an expandable list. Behind it is the **Graphical View** tab, where you can edit the job stream dependencies graphically. On the **Run Cycles Preview** tab, you see the days that the job stream runs based on the run cycles that are added.

You can use the graphical view in Workload Designer to view and edit job streams, and to add and modify dependencies to jobs and job streams. To access the graphical view, define or recall a job stream to the Workload Designer data area, and click the **Graphical View** tab at the top. The job stream is displayed as a box, with its jobs inside of it. Dependencies are displayed as smaller boxes along the periphery of the job stream.

The top menu bar of the graphical view provides tools for zooming, panning, and adding and removing dependencies. Hover help is provided for each of the tools on the menu bar. Hovering over objects in the diagram also displays additional information about the object.

The lower part of the data area displays the job stream properties and has four tabs: **General**, **Scheduling options**, **Time restrictions**, and **Dependency resolution**.

Step 1: Completing the general details

On the General page, populate the following fields. Mandatory fields are highlighted in yellow on the GUI.

- Job stream **Name** field follows your organizational naming convention and represents the purpose of the job stream, such as SALES_RPT_BOM_1 or ACC_RCV_Daily_5. Naming your job stream is important to operations, administration, and management because naming identifies the purpose of the job stream while it is in production.
- The **Workstation** resolves the job stream dependencies during production and resolves time zones, resources, and external predecessor dependencies. The jobs in the stream do not have to run on the same workstation that controls the stream, but it is helpful for ease of management and monitoring.
- The **Description** field displays a short summary of the basic details of the job stream.
- You can use the larger **Comments** field for detailed information such as ownership, modification history, and special notes for the job stream.

There are a few other fields on the General page you might want to complete:

- If you are creating a job stream that will not run until some date in the future, complete the **Valid from** field. Date versioning of job streams is described in [“Creating future versions of a job stream”](#) on page 6-38.
- If you are creating a job stream that you want to save, but do not want to run in production, check the **Draft** option. When Draft is selected, the job stream cannot be selected to run, or submitted to run.
- Optionally, enter the name of a variable table to use for resolving variables in the jobs you select for this job stream. Remember that you can specify variable tables also in the workstation and run cycle definitions.

Step 2: Completing the scheduling options

The **Scheduling options** tab contains the following options:

- The **Carry forward** check box. Carry forward marks this job stream eligible to run in the next plan if not completed in the current production day.
- The value in the **Limit** field restricts how many concurrent jobs can run in this job stream.
- The **Priority** of the job stream can be 0–101:
 - **Go** (101) overrides the workstation Fence and Limit settings.
 - **High** (100) is the highest normal priority.
 - **0** (Hold) prevents the job stream or job from starting.
 - **10** is the default value.
- In the **nonworking days calendar** section, you can refine how the job stream is scheduled around holidays and other nonworking days.
- In the **actions** section, select the rule to apply when repeating job streams overlap. You learn about parallel scheduling options in [“Using parallel instance actions”](#) on page 6-35.

Using Carry forward

Job streams already carried forward retain the **Carry forward** option. Job streams with the carry forward option can be carried forward again. The number of days a job stream is carried forward is set by specifying a **Latest Start Time plus x number of days**.



Note: Workload Scheduler administrators can set carry forward behavior with a system option. Depending on the setting of the option, the **Carry forward** check box might be ignored.

The default is to carry forward all incomplete job streams, regardless of whether the check box is selected. Contact your Workload Scheduler administrator to determine which carry forward option is used in your environment.

Using limits and priorities

The Limit setting restricts the number of jobs that can run concurrently in this job stream. Workload Scheduler has 102 priority levels. Assigning priorities to your jobs and job streams provides added control over precedence and order of execution. Workload Scheduler uses priorities to determine the job run order if there is competition for resources or a job limit in effect. If resources are available and limits are sufficiently high, Workload Scheduler does not consider priority. All jobs with satisfied dependencies start.

Using the nonworking days calendar

Use the **Nonworking days calendar** section (also called FREE DAYS) to specify how the job stream handles holidays. The **Use default** option means that the job stream does not run on days that are specified in the (default) HOLIDAYS calendar. Use other options to specify an alternative holiday calendar name, and specify whether Saturday and Sunday are nonworking days.

Job streams use actual dates, days of the week, or calendars. A calendar is a set of specific dates. You can create as many calendars as needed to meet your scheduling requirement.

Step 3: Completing the time restrictions

Use the **Time restrictions** tab to specify time constraints for the job stream. Time restrictions are optional. In this section, you learn about the fields on the **Time restrictions** tab. You learn about how to set better time restrictions in [“Setting time and date restrictions”](#) on page 6-24. By specifying the times on this tab, you can define a window in which jobs in the job stream run. If you have difficulty correctly specifying the time, click the **clock** icon beside the time field. Insert a time from the selection list and use it as a sample.



Note: The **Time zone** field specifies the time zone in which the job stream runs. All time dependency values that are specified are based in the time zone that is being used. When the time zone is not specified, the default time zone is the time zone of the workstation where the job stream starts.

These fields set the times that the job stream should run:

- **Earliest start** specifies the start time for the job stream, if all dependencies are met. Select the **Use as time dependency** check box to hold the job stream from starting until the time you enter. **Delay for days** is the number of days to offset the start time from the first day you select the job stream in the plan.
- **Latest start** specifies the time after which execution is not attempted. Use the **Action** menu to select what happens if a job or job stream does not start by the latest start time.
- The **Deadline** time is the time after which a running job or job stream is considered late. If a job exceeds the deadline, it is still started or continues to run, but it is marked late in the plan.

Step 4: Adding other job stream dependencies

After defining the job stream, add dependencies by right-clicking the stream name and clicking **Add Dependencies**.

Adding resource dependencies

Use resource dependencies at the job stream level or job level, but not at both levels. Resources have a maximum limit of 1024 units. Only 32 jobs and job streams can use these units at any one time. Job 33 must wait on HOLD.

You can define jobs and job streams on multiple workstations. A job or job stream that is running on a workstation can use resources that are managed on another workstation.

Adding prompt dependencies

A prompt dependency can be on an ad hoc prompt or a predefined prompt.

- An **ad hoc** prompt is specified when the job stream is defined or after a job is added to the job stream. The prompt requires a reply every time.
- A **predefined** prompt is a database object that retains its status in a production day. If a predefined prompt is answered at the start of a day, the prompt remembers its state for all successive jobs and job streams.



Hint: You can include one or more Workload Scheduler variables as part or all the text string of the prompt. To use a variable, place its name between carets (^).

To create ad hoc prompts, right-click the job stream name at the top of the data area and click **Add Dependencies**. Select **Ad-hoc Prompt**. Ad hoc prompts require an answer each time a job stream or job uses them.

To create predefined prompt dependencies, right-click the job stream name at the top of the data area and click **Add Dependencies**. Select **Predefined Prompt**. Unlike ad hoc prompts, the predefined prompts must exist in the database.

After the operator answers *Yes* to a predefined prompt, it retains its state for the remainder of the production day, unless an operator answers the same prompt with a *No* reply. Other jobs or streams that use that prompt also have their dependencies that are met by a *Yes* reply.

Adding file dependencies

To create file dependencies, right-click the job stream name at the top of the Details page and click **Add Dependencies > File**. You must type the full path to the file and specify the workstation where the file is located. The combination of file name and qualifier cannot exceed 148 characters, and the base file name cannot exceed 28 characters.

Use the **Qualifiers** field to specify the test that is applied to the file. On UNIX, the valid qualifiers are the same as UNIX `test` command conditions. On Windows, the valid qualifiers are shown in the following table.

<code>-d %p</code>	True if the file exists and is a directory
<code>-e %p</code>	True if the file exists
<code>-f %p</code>	True if the file exists and is a regular file
<code>-r %p</code>	True if the file exists and is readable
<code>-s %p</code>	True if the file exists and its size is greater than 0
<code>-w %p</code>	True if the file exists and is writable

On UNIX and Windows, these qualifiers perform in the following ways:

- The expression `%p` inserts the file name.
- Entering `notempty` is the same as entering `-s %p`.
- If no qualifier is specified, the default is `-f %p`.
- The Condition Unary negation operator (!) tests that the condition is false. For example, use the following expression to specify that the file does not exist.

`! -f %p`

You can use the string-matching and integer comparison capabilities of the UNIX `test` command to further customize the file (OPENS) dependency. Use this customization to test for conditions beyond simple file existence. You can find examples of expressions to use in [“File dependency qualifiers”](#) on page A-9.



Hint: File dependencies use the most system resources of any dependency type. Use them cautiously. Also, consider adding other dependencies in addition to file dependencies. For example, if you add an earliest start time, the system will not start checking for the file until after the start time expires.

Step 5: Adding job stream predecessor dependencies

Use a *job stream predecessor dependency* to set the current job stream to start only after the completion of another job stream in the database. To create job stream predecessor dependencies, right-click the job stream name at the top of the data area and click **Add Dependencies > Job Stream**. The predecessor job stream must exist in the database. Click **Search** in the Add window to list all job streams already defined.

Use the *job predecessor dependency* to set the current job stream to start after any other job in the plan completes. To create job predecessor dependencies, right-click the job stream name at the top of the data area and click **Add Dependencies > Job**.

Use a *Join dependency* to gather a set of predecessors as a single dependency. You learn more about join dependencies in [“Using conditional dependencies”](#) on page 6-19.

Step 6: Optionally adding a start condition

IBM Training



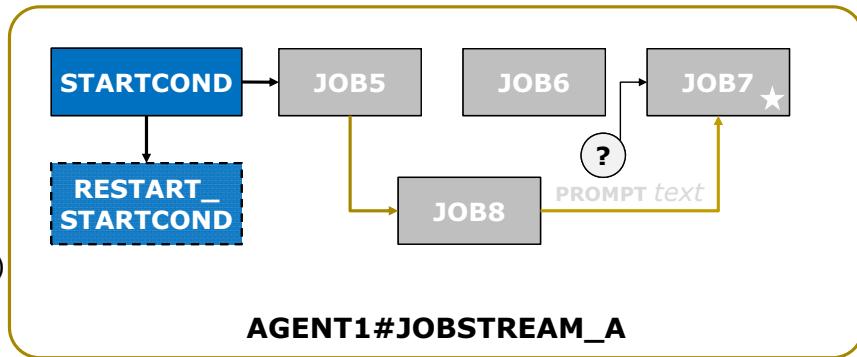
Adding a start condition

Start condition types

- File created
- File modified
- Job condition met

Start condition options

- Start once (do not restart)
- Start condition alias



© Copyright IBM Corporation 2017

You can define a start condition to precede the jobs you define in a job stream, and repeat the job stream each time the condition is met. When you define a start condition, Workload Scheduler automatically generates a job (or inserts a job that you define) that runs before the other jobs you add to the job stream.

You can select one of the following types of start conditions for the job stream.

- File created
- File modified
- Job condition met

Saving your work

When creating complex definitions, periodically click **save** to preserve your work. An asterisk (*) beside the job stream name in the **Working List** and **Data** area indicates that you have unsaved changes in the definition.

You can work simultaneously on definitions of scheduling objects, jobs, and job streams in the Workload Designer. The **Working List** displays all definitions that are open. Click a name in the list to bring the name to the foreground in the Details pane. Save often because you might have unsaved changes in objects in the background.

Lesson 2 Applying jobs to job streams

IBM Training



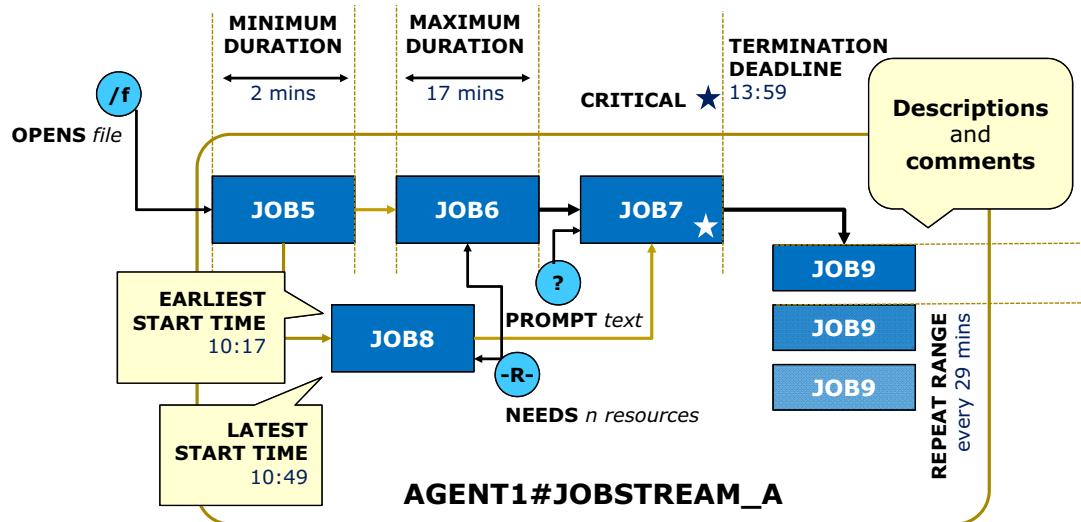
Applying jobs to job streams

© Copyright IBM Corporation 2017

In this lesson, you learn how to add jobs to job streams. You learn how to perform the following tasks:

- Add jobs to a job stream
- Set dependencies and properties of jobs within a job stream
- Selecting jobs with no operation
- Use advanced statistical analysis to report job durations
- Using conditional dependencies to customize workflows

Applying jobs to a job stream



© Copyright IBM Corporation 2017

Applying jobs to a job stream

In the next steps, you add jobs and assign dependencies to the jobs. You already know about some dependencies, for example, adding prompts to job streams and jobs. However, a few dependencies are specific only to jobs. There are several ways to add jobs to job streams. You can add the same job to the job stream one or more times.

Adding jobs by using the Action menu

To add jobs to the job stream, perform the following steps:

1. With the job stream open in the **Details** pane, click the **Select an Action** menu and click **Add Jobs**.
2. In the “Add to Selected - Job Definition” window, enter a job definition filter and click **Search** to display a list of jobs. Wildcards are acceptable.
3. Select the jobs and click **Add**.

Adding jobs by using the Search field

To add jobs to the job stream, perform the following steps:

1. With the job stream open in the **Details** (or Graphical View) pane, click the **Search** field, and click the **Job Definition** icon.
2. Enter a job definition filter to display a list of jobs. Wildcards are acceptable.

3. Select the jobs and click **Add**.
4. Alternately, select the jobs. Using the mouse, drag the list to the job stream pane, and drop them on the job stream name.

Adding jobs by using the Working List pane

If you already have jobs or other objects (such as resources) open in the Working List, you can add them to any open job stream. To add jobs to the job stream, perform the following steps:

1. Select the jobs that you want to add in the Working List.
2. Right-click the job stream name in the Details pane, or open the **Select an Action** menu.
3. Click **Add Objects Selected in Working List**. The jobs are added.

Setting job properties

You can change the dependencies and properties that determine when jobs in a job stream runs.

Click a job in the job stream, and in the bottom of the Details pane, configure these job details:

- On the **General** tab, enter text into the **Description** and **Comments** fields. These fields are related to the job stream definition and do not change the job definition. If you added the same job definition more than once, each job that uses the same job definition must have a different name. Enter a unique name within the job stream in the **Name** field. You learn about the **No operation** setting in [“Using no operation jobs”](#) on page 6-18.
- On the **Scheduling Options** tab, the **Critical** check box identifies this job to workload service assurance. Workload service assurance monitors the predecessors and acts upon the stream if the job is at risk of missing its deadline. When setting the *Critical* flag, also assign a deadline on the Time Restrictions page. When enabled, **Requires confirmation** places the job in *PEND* state on completion until confirmed by the operator. Select a priority for the job in the **Priority** section.
- On the **Time Restrictions** tab, enter any time restrictions that apply only to the job that you are modifying. You learn more about time restrictions in [“Setting time and date restrictions”](#) on page 6-24.

Adding and removing job dependencies

The order that you see jobs in the job stream window is the order in which jobs were added. The appearance does not affect the order that the jobs run. With no dependencies or limits, all the jobs you add run at the same time. You determine the order when you assign dependencies. However, to avoid confusion, you can do multiple additions so that the jobs listing reflects the execution order.

The most common predecessor dependency for a job is another job in the same stream, but predecessor jobs are not necessarily in the same job stream. A job can have as its predecessor a job in an external job stream or an entire external job stream.



Hint: Be careful when adding dependencies to avoid creating circular dependencies. The Workload Designer checks for circular dependencies within a job stream but cannot detect circular dependencies that involve external jobs, which are in some cases valid, given the correct properties. Jobs can enter the plan and become deadlocked because dependencies cannot be satisfied. You can avoid this state by drawing your jobs and job stream dependencies in a processing diagram.

You can work with dependencies from either the **Details** tab or the **Graphical View** tab.

Adding dependencies from the Details view

To add a dependency to a job from the Details view, by using the menus do the following steps.

1. Right-click the job and click **Add Dependencies**.
2. Select the type of dependency to add from the menu (**Job in the same Job Stream** is the most common).
3. Complete the search filter and click **Search**.
4. Select the objects that you want to use dependencies, and click **Add**.

The objects that you selected are added to the job stream as dependencies.

You can also add dependencies on objects other than jobs by dragging them from the Quick Open pane to the object that depends on it.

Adding dependencies from the Graphical View

To add a dependency to a job from the Graphical View, by using the menus, do the following steps:

1. Right-click the job or job stream and click **Add Dependencies** from the menu or select the object and choose an option from the **Select an Action** menu.
2. From the menu, select the type of dependency you want to add.
3. From the Search page, search for the required object and add it to the job or job stream.

To add a dependency to a job from the Graphical View, do the following steps:

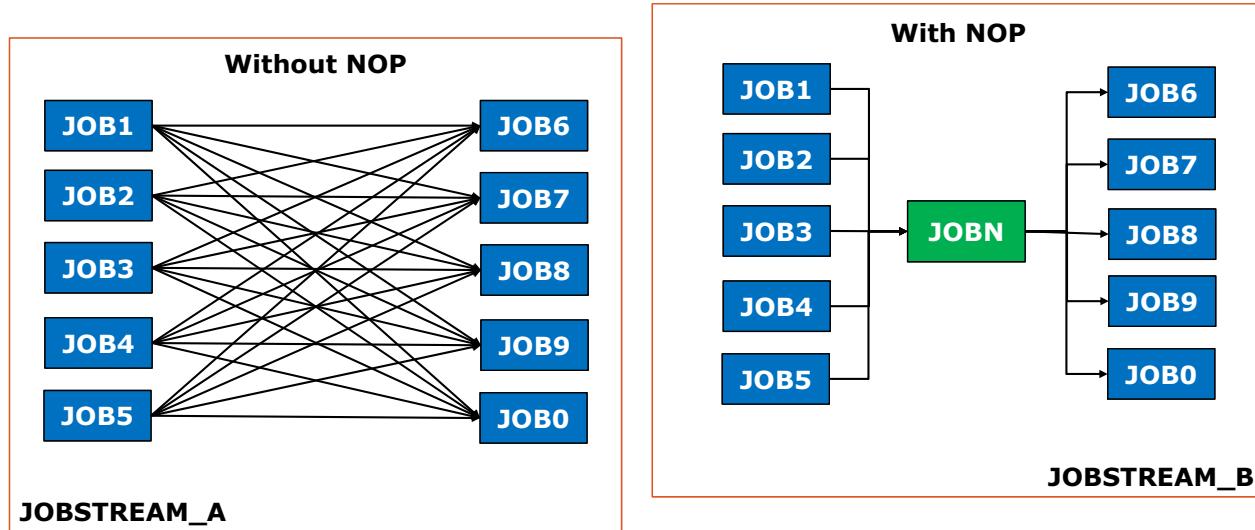
1. Click the job that is the predecessor.
2. Select the **Create Dependency** icon (>).
3. Draw a line from the dependency to the job or job stream that must depend on it. You can use this tool to create dependencies only between objects that are already displayed in the view.

Removing a dependency

You can remove a dependency from the **Details view** tab or the **Graphical View** tab:

- From the **Details** view, click the **Remove** icon at the right end of the dependency row.
- From the **Graphical View**, select the dependency by clicking the node, if it is an external dependency, or by right-clicking the arrow. When the arrow that you want to remove is highlighted, click the **Remove** icon on the toolbar or from the menu. If an object has multiple dependencies, click it again until the arrow you want to remove is highlighted.

Using no operation jobs



Using no operation jobs

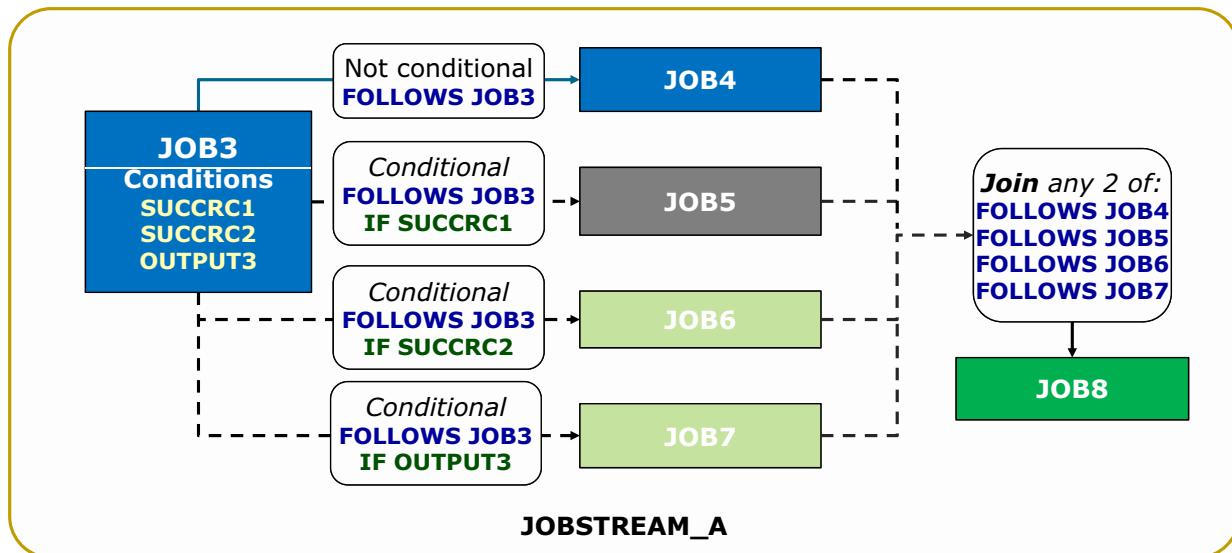
You can set a job within a job stream to have no operation (**NOP** keyword). When you set the no operation flag, the job is included in the job stream as part of the production plan, but placed in *Cancel Pending* status. Jobs in cancel pending status do not run. However, their dependencies are released and their successors are run. Using the no operation flag has the following benefits.

- You can simplify complicated dependency maps and reduce the number of dependency definitions by using a no operation job.
- You can predefine a job to not run in production, without removing it from the job stream.

To use the *no operation* flag, complete the following steps.

1. In the **Details** pane, click the job that you want to set to *no operation* so that it is highlighted.
2. In the **Properties** pane, click the **General** tab.
3. Select **No operation**.

Using conditional dependencies



© Copyright IBM Corporation 2017

Using conditional dependencies

You can create conditional dependencies, which specify that one job depends on another only when specific conditions are met. By using different sets of jobs that match known possible conditions, you can map alternative processing branches to the output conditions, return code conditions, or completion statuses of a job. You can also join dependencies and conditional dependencies into groups, such that a job can start when one (or a number you define) of the possible conditions is met (also known as an OR condition.)

Adding conditional dependencies

To create a conditional dependency, you first add jobs to the job stream, and add dependencies between the jobs. You then add conditions to the dependencies you create. In the Workload Designer, you can configure a dependency to use conditions from the **Details** or **Graphical View** tabs. To configure a dependency to use conditions, complete the following steps.

1. Select the dependency that you want to customize, by choosing it with the **Select** tool in the **Graphical View** or highlighting the **Job Dependency** row in the **Details** view.
2. In the **properties** pane, select **Conditional Dependency**.

3. Choose the option that best satisfies the condition you want to satisfy from the following list.
 - **Job started:** Select this option so that the job starts when the predecessor starts. Both jobs run at the same time, and the completion status of the predecessor job is not used.
 - **Successor job runs if the predecessor job or job stream completes with any of these statuses:** Select this option, and choose one or more status options to start the job when any of the selected statuses is reached.
 - **Successor job runs if any of these conditions are met:** Select this option, and select one or more of the conditions that are listed, to start the job when any of the selected conditions is satisfied. The conditions are created in the job definition for the predecessor job.

Adding join dependencies

To create a join dependency, you first add jobs to the job stream. You then complete the following steps to define the join.

1. In the **Details** view, right-click the successor job, and click **Add Dependencies > Join Dependencies** from the menu. You can instead right-click the successor job in the **Graphical View**, and click **Add Dependencies > Join Dependencies** from the menu.
2. In the **Properties** pane, enter a name for the join dependency in the **Name** field.
3. In the **Details** tab, right-click the Join Dependencies row, and click **Add Dependencies > Job in the same Job Stream** from the menu. You can instead choose **Job Stream** or **Job in Another Job Stream**.
4. You see a Search window where you can search and select jobs to use as dependencies, and click **OK**.
5. Customize the dependencies that were added in the previous step as needed. The dependencies can optionally be conditional.
6. Click the row that contains the join dependency rule's name.
7. In the **Properties** pane, choose the number of dependencies that must be satisfied to consider the join satisfied.

Analyzing job durations

	Actual	Estimated	Confidence interval	Actual	Estimated smooth 10	Estimated smooth 50
Monday	JOB1 5:00	5:00	1:00	JOB2 15:11	15:11	15:11
Tuesday	JOB1 4:58	4:59	1:30	JOB2 15:23	15:12	15:17
Wednesday	JOB1 5:01	5:00	0:40	JOB2 16:29	15:20	15:53
Thursday	JOB1 4:59	5:00	1:40	JOB2 16:17	15:26	16:05
Friday	JOB1 5:01	5:00	1:10	JOB2 17:11	15:36	16:38
Global	JOB1 7:20	6:30	2:30			

- A shorter confidence interval means the duration is more predictable
- A higher smoothing factor weights the estimated duration to the most recent iteration

© Copyright IBM Corporation 2017

Analyzing job durations

During the daily planning cycle, two jobs collect the completed day's job data, update the job history and statistics information, and provide text-based reports about the previous day's results. In this lesson, you learn how statistics about job run times and durations are collected and analyzed.

The UPDATESTATS job runs the `logman` program to retrieve job data from the archived Symphony file, computes the average job durations, and retains the information in the Workload Scheduler database. The `statsHistory` global option determines how many job instances are preserved in the job history.

The `logman` program provides the estimated duration of each job run, and its related confidence interval. The estimated duration of a job run is based on the average of its preceding runs. The duration is calculated by analyzing five different time series: `GLOBAL`, `WEEK_DAY`, `MONTH_DAY`, `MONTH_DAY_REVERSE`, and `RUN_CYCLE`.

By default, sudden changes in the jobs' run times are not immediately reflected in the average. To respond more quickly to such changes, you can configure the `smooth` option to weight the average in favor of the most recent job run. You use the `-smooth` option to set a weighting factor that is the percentage of the current job run in relationship to the existing average. For example, the `logman -smooth 50` command causes `logman` to use a weighting factor of 50 percent for the most recent runs of the job, and 50 percent for the existing average. The `logman -smooth 100` command causes the most recent run of the job to override the existing average. The default value for the `-smooth` option is 10. The `logmanSmoothPolicy` global option determines the smoothing factor that `logman` uses during the production planning process.

A powerful statistical tool for the prediction of estimated job durations - in addition to the one provided by the `logman` command - is available by installing the subset of **IBM SPSS Statistics** software that is packaged with IBM Workload Scheduler. SPSS uses a sophisticated algorithm on the previous 90 days' history to forecast the estimated durations for the next five days.

The `logman` program provides accurate estimates when the workload is subject to regular, periodical shifts. The advanced statistical tool is ideal when more complex patterns are present. For example, the tool can be useful to see beyond the accepted impacts of already known cyclic events, and understand what apparently hidden conflicts can affect the duration of a job. It can be effective to measure and forecast the durations of jobs along a critical path that occasionally does not meet its deadline.

You can override the estimated duration of a job in a job stream from the Workload Designer. For example, for a new job, the estimated duration is set to **00:00:00**. You can set the estimated duration to an initial value that is more realistic, based on your experience. You can also set the estimated duration based on the periods in which the job runs. For example, based on experience, you know that the job runs longer on Fridays than on other days of the week, so you set the estimated duration for Fridays.

The predictive analysis process associates a 95% confidence factor to the estimated duration of each job, and provides a confidence interval. The confidence interval is an estimated range of values, which is likely to include the job duration with 95% confidence. For example, if the estimated duration of a job is 5 minutes (5:00), with a confidence interval of 7 seconds (00:00:07), you can be 95% certain that the actual duration of the job is between 4 minutes, 53 seconds (7 seconds less than the estimated duration), and 5 minutes, 7 seconds (7 seconds longer than the estimated duration).

The confidence interval is set to 0 for every estimated duration that you override. While statistical data is collected, `logman` replaces the estimated duration that you set according to the described algorithm. When a job's confidence interval is shorter, the duration is more predictable. As illustrated in the example, [“Analyzing job durations,”](#) on page 6-21, JOB1 has a weekly trend that is stable on Wednesdays. However, historical duration data for different month days or run cycles made the *global average duration* less predictable. The lower global predictability means that on another period, JOB1's duration varies significantly from the average.

You see the estimated durations and associated confidence intervals in the output of the `conman showjobs;props` command and in the following Dynamic Workload Console views:

- Job properties
- Monitor Workload
- Critical path
- What-if Analysis

Selecting jobs to use advanced statistics

The SPSS Statistics tool records and analyzes only jobs that you select within job streams. To select jobs to be analyzed, complete the following steps.

1. In the workload designer, edit the job stream that contains the job you want to select.
2. In the **Details** pane, select a job to use advanced analysis.
3. In the **Properties** pane, click the **Duration forecast** tab.
4. Select **Use Advanced analytics**.
5. Save the job stream by clicking **Save**. You see the message, **The object (job stream) has been modified**.
6. Optionally, click **Override Estimations** to open the Override Estimations window. You can set estimated durations, in **hh:mm:ss** format, in one or more of the following groups.
 - In the **All Instances** group, enter an estimated duration to apply to all of the other estimations in the Override Estimations window. Click **Apply to All** to set the durations.
 - In the **Run Cycle** group, enter the estimated durations for the job when it is selected to run by a run cycle on the list.
 - In the **Monthly** group, enter the estimated durations for the job when it runs on one or more of the days of the month.
 - In the **Weekly** group, enter the estimated durations for the job when it runs on one or more of the days of the week.
 - Click **Save** to store the estimated durations, or **Discard Changes** to close the window without saving the changes.

Lesson 3 Setting time and date restrictions

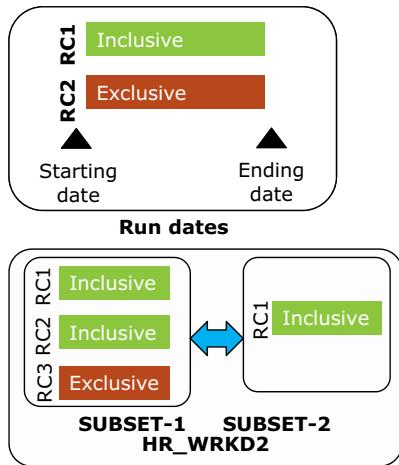
Setting time and date restrictions

© Copyright IBM Corporation 2017

In this lesson, you learn how to set date and time restrictions on your jobs, run cycles, and job streams. You learn how to perform the following tasks:

- Apply run cycles to job streams
- Determine which time restriction applies to a job stream
- Set time restrictions
- Apply time restrictions
- Repeat work at regular intervals

Selecting run dates



- Use **run cycles** to specify which **production days** contain your job stream
- Add as many run cycles and **run cycle groups** as you need to schedule your job streams
- Run cycles are **not required** if the job stream is event driven

© Copyright IBM Corporation 2017

Selecting run dates

Run cycles specify when a job stream in the database is selected to run in the plan. You can create run cycles as part of the job stream, or create run cycles in run cycle groups separately from the job stream. You can also combine run cycles internal to the job stream with run cycle groups.

To create a run cycle for your job stream, right-click the job stream name at the top of the **Details** pane and click **Add Run Cycle**. On the **General** tab of the Run Cycle Properties window, you can change the name to something descriptive, or leave it at its system-assigned value. You can place additional information about this run cycle in the **Description** field.

The **Starting** and **Ending** fields specify a range of dates over which this job stream is valid. The **Variable table** field specifies the table to use to evaluate variables in this run cycle.

Use the **Rule for non-working days** menu to select how to handle nonworking days. The **Inclusive** and **Exclusive** options indicate whether the job stream runs (Inclusive) or does not run (Exclusive) on the days specified.



Hint: You might need to associate multiple run cycles to a job stream to satisfy complicated scheduling requirements.

After specifying a run cycle, check the dates on a calendar on the **Run Cycles Preview** tab at the top of the Details pane. Click the **Details** tab to resume your configuration.

Selecting run cycle rules

When you complete the **General** tab, click the **Rule** tab to select the type of run cycle and specify the particulars.

In addition to the type of run cycle rules that are described in [“Creating and using run cycles”](#) on page 5-28, you can use an existing run cycle group as the basis for one or more of the job stream’s run cycle selections. In the **Repeat schedule** menu, select **Run Cycle Group**. Specify which run cycle group to use in the **Run Cycle Group** field.

You can also schedule a job stream on an offset from each resulting day in a run cycle group. For example, to schedule a job stream to run two before each day in a run cycle group, select the minus sign (-) from the offset menu, **2** for the **Number** field, and **Days** from the **Type** field.

Adding time restrictions to run cycles

You can also add time restrictions to each run cycle. Adding a time restriction might add job stream occurrences to the production plan. You learn more about time restrictions in [“Setting time and date restrictions”](#) on page 6-24.

Types of time restrictions

Jobs and job streams

EARLIEST START TIME 10:17	Job or job stream must not start <i>before</i> this time	LATEST START TIME 10:49	Job or job stream must not start <i>after</i> this time
REPEAT RANGE every 29 mins		SUPPRESS CONTINUE CANCEL	
	Repeat the job or job stream (use with earliest and latest start times, and parallel running options)	TERMINATION DEADLINE 13:59	Action to perform if not started on time

Jobs only

MAXIMUM DURATION 17 mins, 105%	Job must not last longer than this time	MINIMUM DURATION 2 mins	Job must last longer than this time
CONTINUE KILL	Action to perform if maximum exceeded	CONTINUE CONFIRM ABEND	Action to perform if minimum not reached

© Copyright IBM Corporation 2017

Types of time restrictions

Time restrictions are optional. By specifying the time restrictions, you can define a window in which jobs in a job stream run. If you have difficulty correctly specifying the time, click the **clock** icon beside the time field. Insert a time from the selection list and use it as a sample.



Note: The **Time zone** field specifies the time zone in which the job or job stream runs. All time dependency values that are specified are based in the time zone that is being used. When the time zone is not specified, the default time zone is the time zone of the workstation where the job stream starts.

Several types of time constraints determine when jobs can begin, and when jobs must complete. Time restrictions are available at the job or job stream level.

The available types of time restrictions are shown in the following table.

Table 1 Types of time restrictions

Earliest start time	A job or job stream cannot start before the earliest start time (AT keyword). Earliest start specifies the start time for the job stream, if all dependencies are met. Select the Use as time dependency check box to hold the job stream from starting until the time you enter. Delay for days is the number of days to offset the start time from the first day you select the job stream in the plan.
Scheduled time (Planned Start Time)	For job streams only, <i>scheduled time</i> is not a time dependency. It represents the time when the job stream is positioned in the plan. Specifying a scheduled time does not prevent the job stream from processing. It uniquely identifies a job stream instance in the plan, when multiple instances of the job stream exist for a single day (job streams that run more than once in a 24-hour period.). Time restrictions can also be combined. For example, when both an earliest start time and a latest start time for a job or job stream are defined, a time frame within which processing must start is created.
Latest start time	The latest time a job or job stream can start processing. (UNTIL keyword). In other words, the job or job stream must start before this time. You can also define an action to take when the latest start time is reached before the job or job stream starts. One of the following actions can be used (ONUNTIL): When you select Suppress , the job or job stream and any other dependent jobs or job streams do not run after the latest start time is reached (the default). When you select Continue , the job or job stream runs when all the defined conditions are met, even after the latest start time, and event rules that match the latest start time are triggered. By choosing the Cancel option, the job or job stream is canceled if it does not start before the latest start time. All dependencies are released so that any other jobs or job streams that depended on this one (for example, successors) can run.
Apply to each job (job streams only)	When you apply a latest time to a job stream, the definition applies effectively only to the first jobs in the job stream. Any jobs in the job stream that do not start before the latest start time are suppressed, even if the first job in the workflow starts. Use the apply to each job setting so that the latest start time that is defined for the job stream applies to all of the jobs.
Termination deadline	The latest time by which a job or job stream must complete (DEADLINE keyword). If a job exceeds the deadline, it is still started or continues to run, but it is marked late in the plan. You can also specify that Workload Scheduler should terminate a job that is running when its deadline expires. When you choose the Kill option, the job is set to error status and ended after the deadline is reached.

Table 1 Types of time restrictions

Repeat range	<p>The repeat range (EVERY keyword) specifies that the job or job stream should start repeatedly at the specified interval after all other dependencies are satisfied.</p> <p>A repeated job starts at the regular interval specified by the repeat range, or immediately after its previous instance completes, if the duration of the previous instances is longer than the repeat range. The <code>bm late every</code> option setting defines if a repeating instance could be skipped.</p> <p>Repeat every for job streams specify that the job stream is scheduled repeatedly at the interval of hours and minutes you select. At plan generation time, a number of instances of the job streams that apply this run cycle are created in the plan. The instances begin at the defined <i>Earliest start</i> time, or at the start of the processing day if no start time is defined. If you use a repeat interval, you must complete the Repeat end time field with the time to end scheduling instances.</p>
Maximum duration	<p>For jobs only, this option specifies the maximum amount of time that a job can be running. The value can be expressed in absolute time in hours and minutes or as a percentage of the estimated duration of the job. You can also specify an action that must be started if the job run runs longer than the specified amount of time. Event rules that match the maximum duration specification are also triggered. The possible action options are listed:</p> <p>The Continue option allows the workload to continue running (the default). When you choose the Kill option, the job is set to error status and ended after the maximum duration is reached.</p>
Minimum duration	<p>For jobs only, this option specifies the minimum amount of time that a job must last. The value can be expressed in absolute time in hours and minutes or as a percentage of the estimated duration of the job. You can also specify an action that must be started if the job ends before the minimum duration is reached. Event rules that match the maximum duration specification are also triggered. Possible actions are as follows:</p> <p>If the Continue option is selected, the workload running continues.</p> <p>When the Confirm option is selected, the job is set to <i>Confirm</i> status, and the workload requires a user confirmation to proceed.</p> <p>The Abend option sets the job to <i>Error</i> status if it completes before the minimum duration is met.</p>

Using time restrictions to add job stream instances

You can create extra run instances of a job stream in the plan by adding run cycles with new start times. For example, if you want to run the same set of jobs at 10:00 AM and again at 10:00 PM, add the jobs to a single job stream that contains two run cycles. In the first run cycle, enter an **Earliest start** time 10:00 AM. In the second run cycle, enter an **Earliest start** time 10:00 PM. On days that

you select for both run cycles, two instances of the job stream are planned. Select **Use as time dependency** only if you want the job stream not to start before the earliest start times.

You can also apply negative criteria, or exclusive run cycles to selected instances by matching the earliest start times. For example, you want the jobs to run at 10:00 AM every day, but you want the jobs to run at 10:00 PM every day unless it is the last day of the month. In this case, create a third run cycle that follows these criteria:

- Set the run cycle to be an exclusive run cycle.
- In the rules, select the last day of the month.
- In the **Time restrictions**, match the **Earliest start time** to the 10:00 PM run cycle.

Use the **Run Cycle Preview** tab to check your run cycle selections. When you click a date in the **Year view** or **Month view**, you see a window with the total number of run instances for that day, and a list of instances based on the run cycles in the job stream.

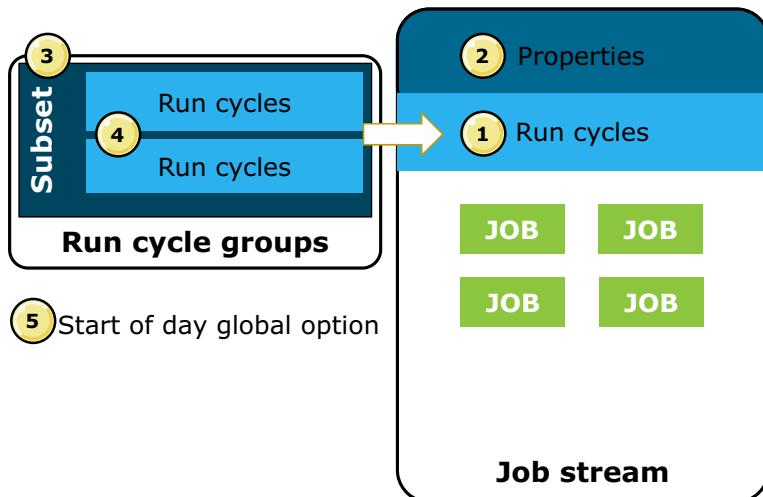
Following the time restriction hierarchy

Time restriction hierarchy

1. Run cycles in the job stream
2. Job stream
3. Run cycle group subset
4. Run cycle group
5. Start of day setting

Time restrictions on jobs

- The latest earliest start time
- The earliest latest start time



© Copyright IBM Corporation 2017

Following the time restriction hierarchy

Time restrictions on the jobs within a job stream do not affect how the job stream instances are selected during the planning process. The time restrictions on jobs are included in all instances of the job stream.

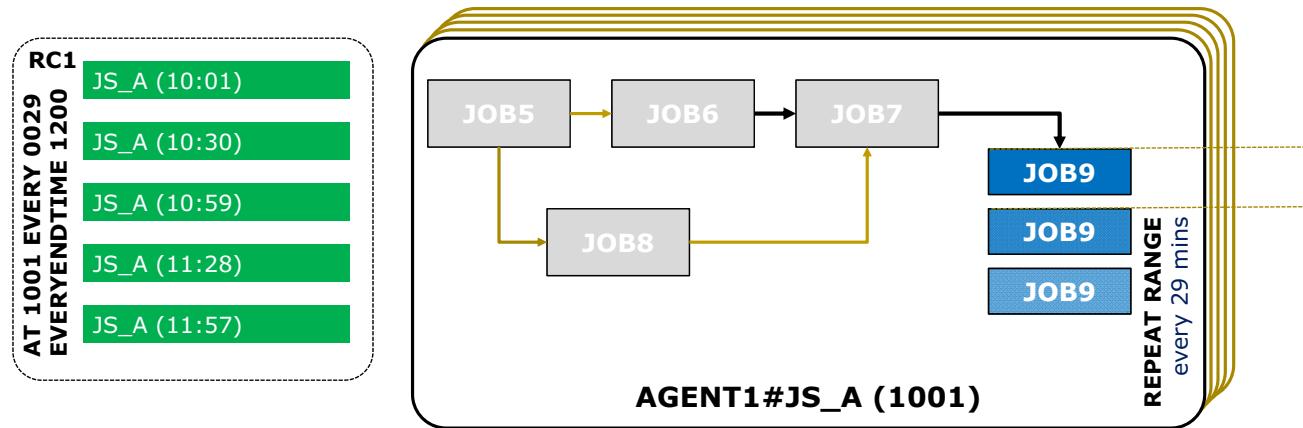
For job streams, you set time restrictions in one or more places in the job stream or run cycle group definitions. These settings can cause some confusion as to which time setting is effective for planning the production day. The following hierarchy shows the order that Workload Scheduler evaluates the time restrictions:

1. Run cycle properties that are defined in the job stream
2. Job stream properties
3. Run cycle group subset properties
4. Run cycle group properties
5. Start of day

Time restrictions in the job stream definition override and take precedence over any other time restrictions that are defined in the run cycles or run cycle groups that are associated to the job stream. Additionally,

- If you define a time restriction in the job stream properties, time restrictions in run cycle groups are ignored.
- If you define different *earliest start time* restrictions in different run cycles, job stream instances are added.
- If there are no time restrictions in the job stream nor in the run cycle group, the start of day time is used to generate the start times, depending on the system options set. In this case, the job stream starts as soon as all its dependencies are satisfied.

Repeating work



- Scheduled by run cycles
- Submitted by event rules
- Repeated by **every** option

The start time of repeating job instances are affected by the duration and **bm late every** setting

© Copyright IBM Corporation 2017

Repeating work

In Workload Scheduler, there are several ways to repeat a task. Use the method that fits your purpose without using too much system resources or flooding your workload plan with too much information.

There are at least four different ways to schedule repetitive work in Workload Scheduler. You can choose the one that best serves your requirements for flexibility, performance, and reliability.

1. Use the repeat job function (**every** keyword).

Within the job stream, you can choose the repeat range option for any job you want to repeat. The job can repeat at any rate, in hours and minutes, even longer than 24 hours. Workload Scheduler automatically starts repeating the job as soon as all of its dependencies are resolved. You can also specify a start time for the job to start repeating and an end time (latest start time). The repeat job function has several advantages:

- The repeat time is the easiest option to set up. You can do it in just a few clicks.
- Repeating jobs are easy to monitor. They show in the list of jobs with the rest of your workload.
- You can easily rerun a repeating job in the case of an intermittent failure.

Using the repeat job function also has disadvantages:

- If you do not specify a latest start time, the repeating job might keep running for days.
- A Workload Scheduler agent uses the resources necessary to run each instance.
- Repeated jobs increase the size of the current day's production plan, slightly degrading performance throughout the day. Each repeated instance is logged in to the Workload

Scheduler job history database, which uses space and reporting resources. If you schedule many repeating jobs, be sure to size the Workload Scheduler tables in your database to accommodate the history of those jobs.

2. Use the **repeat every** function in run cycles that are applied the job stream, either within run cycle groups that are assigned to the job stream, or in run cycles that are defined as part of the job stream. Using **repeat every** in run cycles has advantages:
 - You can repeat a set of jobs (a whole job stream), not just one job.
 - Monitoring job streams is easy because the job stream occurrences are part of the plan at the beginning.
 - You don't need a latest start time for the jobs unless you want one because you schedule an exact number of occurrences of the job stream.

This method has the drawbacks that Workload Scheduler enters all the job streams instances (with the same name) into the plan whether they are needed or not.

3. Inject an *automation utility job* into the job stream to continuously rerun it.

You can add a *job stream submission* job, or a *job management* job to the job stream. By using an automation utility job, you can control more precisely when the job or job stream should repeat.

4. Use a *start condition* in the job stream. When you use a start condition, the job stream is automatically run again when the condition is met. You can also use other dependencies, such as a latest start time, to control when start conditions are effective.
5. Trigger the work with an event rule. By triggering your repeating work with an event rule, you use the Workload Scheduler event rule engine to insert new jobs or job streams into the plan only when there is new work to perform.

You create an event rule that detects, for example, a file that arrives on a Workload Scheduler agent computer and submits the job or job stream when the file is detected. As part of the event rule, you can also include variables that contain data about the event that triggers the job submission. For example, each time a file is detected, the event rule passes the name of the new file in the definition of the job stream it submits. Every job in the job stream receives the same file name.

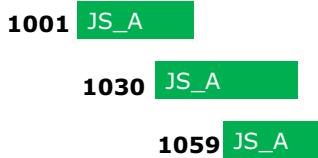
Event rule submission of job streams is fast and efficient. You can customize the event plug-in to detect many kinds of data, not just files, and other jobs. Workload Scheduler doesn't have to run any jobs if there is no data to process, and there are no extra entries in the job history database for jobs that do not process data.

However, event rules are difficult to monitor, especially when there is no data to process. It's not easy to produce alerts when no data arrives.

Using parallel instance actions

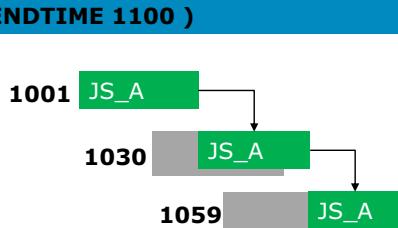
Run the new instance in parallel

(AT 1001 EVERY 0029 EVERYENDTIME 1100)



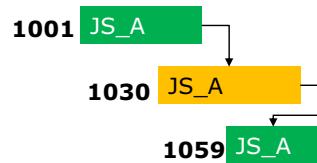
ONOVERLAP PARALLEL
(default)

Queue the new instance



ONOVERLAP ENQUEUE

Do not start the new instance



ONOVERLAP DONOTSTART

© Copyright IBM Corporation 2017

Using parallel instance actions

In the **Actions** section, you can specify how job streams that repeat through the day are configured to run when another instance of the same job stream is still running. You can choose from three options.

- **Run the new instance in parallel** (the default): If another instance of this job stream is scheduled, start the new instance. Two (or more) instances of the job stream can run in parallel.
- **Queue the new instance**: If another instance of the same job stream is scheduled, an *external dependency* is applied. The dependency queues the instances to run after previous instances complete successfully.
- **Do not start the new instance**: If another instance of the same job stream is scheduled, a *Latest Start Time* is applied to prevent the instances from overlapping.

Lesson 4 Using advanced scheduling options

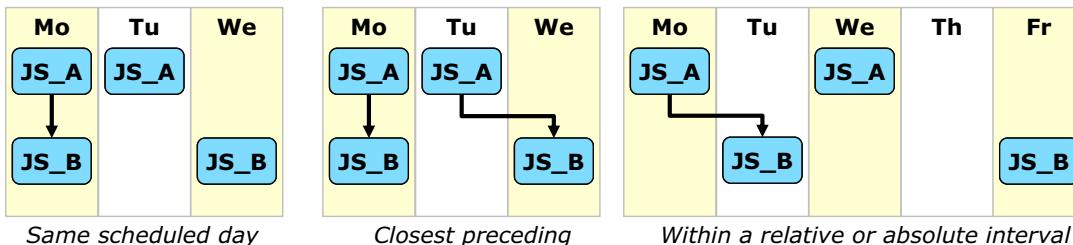
Using advanced scheduling options

© Copyright IBM Corporation 2017

In this lesson, you learn how to use advanced scheduling options that allow you to create workflows that closely match business objectives.

Using resolution criteria

Match external dependencies by using resolution criteria



© Copyright IBM Corporation 2017

Using resolution criteria

In your job stream definitions, you can alter how an external dependency is matched to a specific job stream or job instance in the plan. Because the plan might have multiple instances of the same job or job stream, you can identify the instance that resolves the external follows dependency. You can also specify that a job stream depends on another job stream (or the same one) from previous days. Choose one of the resolution criteria that are listed in the following table.

Same scheduled day	The job stream or job instance that resolves the dependency is the first one scheduled to start on the same day as the instance that includes the dependency. This selection is the default.
Closest preceding	The job stream or job instance that resolves the dependency is the closest preceding the instance that includes the dependency. In other words, the job stream instance immediately before, regardless of which day, the instance that includes the dependency. This option is useful when you want today's instance of a job stream not to start before yesterday's instance of the same job stream finishes successfully.
Within a relative interval	The job stream or job instance that resolves the dependency is the closest one within a time interval you choose. The interval represents the number of hours and minutes before or after the job stream that includes the dependency.
Within an absolute interval	The job stream or job instance that resolves the dependency is the closest one in a time interval you choose. The time interval is fixed, that is, based on the time.

Creating future versions of a job stream

Create future versions of a job stream by using validity dates



© Copyright IBM Corporation 2017

Creating future versions of a job stream

The **Valid from** field in the job stream properties defines that the job stream can be considered for inclusion in the plan only after the date specified. For example, if you select January 1, 2016, as a **Valid from** date, the job stream cannot be planned or submitted manually into the plan until 1 January 2016.

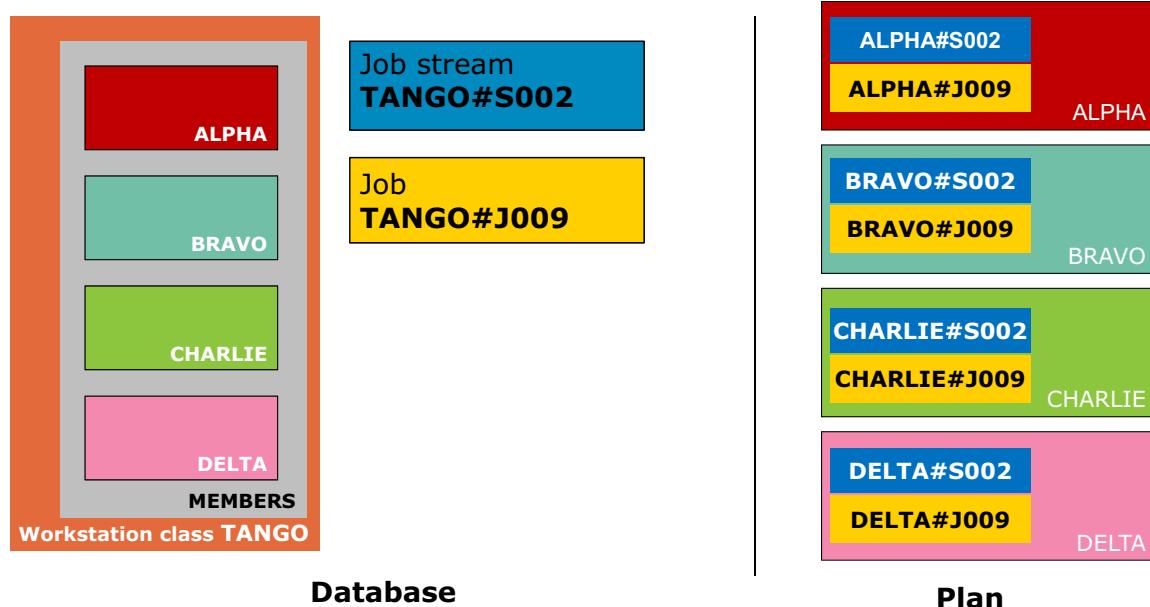
Using the **Valid from** field, you can create a job stream in advance of its requirement to be scheduled. You can start creating job and job stream definitions for a new application that becomes active on a future date.

By default, a job stream remains in effect indefinitely. You can create future versions of the same job stream in advance of proposed scheduling changes. For example, you want to add several jobs to the job stream that run starting next month. To complete this task, follow these steps:

1. In the Workload Designer, open the job stream to which you add the jobs.
2. Create a duplicate of the job stream by clicking **Create Like**. A new job stream is created. The new job stream adds a **1** to the end of the job stream name.
3. Change the job stream name to be the same as the original.
4. Enter the future date in the **Valid from** field, and optionally save the job stream.
5. Add jobs to the job stream, and make any other changes you require, including dependencies, run cycles, and so forth.
6. Save the job stream.

Now there are two versions of the same job stream in the database. After you save the future version, you can check the original version to confirm that it has a date set in the **Valid to** field.

Using workstation classes



© Copyright IBM Corporation 2017

Using workstation classes

A workstation class is a group of workstations with similar characteristics. You can create as many workstation classes as you need. A workstation can be a member of many workstation classes, and a workstation class can contain as many workstations as needed.

Create a workstation class to duplicate work to all of the workstations in the class. When you create a scheduling object for a workstation class, the object is replicated to each member workstation in the class when the plan is generated. This method makes it easier to duplicate the same scheduled workload to many workstations at the same time.



Note: Workstation classes do not appear in the plan. For example, a job stream you create for a workstation class is replicated to each member workstation in the plan, but the name of the workstation class is not included in the plan.

Lesson 5 Creating job stream definitions by using composer

Creating job stream definitions by using composer

© Copyright IBM Corporation 2017

To create a job stream by using the command-line interface, run the `composer` command. Enter `new jobstream` at the dash (-) prompt. You edit a template that contains an example of job streams that use standard text editor commands. You can also edit an existing job stream definition by using the `composer modify js=workstation#jobstreamname` command.

Using composer to create job stream definitions

- Use **composer new jobstream** to create a job stream starting with a template
- You can modify one or many job streams in one session
- If you change the name or workstation, saving the file creates a new job stream
- Use the **composer rename** command to rename a job stream

```
-modify js=CHARLIE#SX7JAX_UMJ valid from 11/01/2017
AWSBIA291I Total objects: 1

SCHEDULE CHARLIE#SX7JAX_UMJ VALIDFROM 11/01/2017
ON RUNCYCLE RGWORKDAYS $RCG DAILY_WORKDAYS
AT 1700 UNTIL 1900 DEADLINE 2300
OPENS TWS92MDM0#/tmp/file1.dat"
PROMPT PROMPT1
PRIORITY 61
:
CHARLIE#J7JAX_PIKAS_00001
PRIORITY 21
NEEDS 1 CHARLIE#RES1
FOLLOWS CHARLIE#SX7JAX_TCB.J7JAX_CREPT_00001

CHARLIE#J7JAX_SIAFU_00001
PRIORITY 31
NEEDS 1 CHARLIE#RES1
FOLLOWS CHARLIE#SX7JAX_UMJ.J7JAX_PIKAS_00001

CHARLIE#J7JAX_SLOGS_00001
PRIORITY 41
NEEDS 1 CHARLIE#RES1
END
```

© Copyright IBM Corporation 2017

Using composer to create job stream definitions

Here is an example of editing a job stream definition in composer:

```

SCHEDULE FTA1#JS1
ON EVERYDAY
:
FTA1#JOBDEF1
END

# Sample Job Stream

SCHEDULE FTA1#JS2 VALIDFROM 03/31/2008 TIMEZONE Europe/Berlin

DESCRIPTION "Sample Job Stream"
DRAFT
VARTABLE TABLE1
FREEDAYS CAL1 -SA -SU

ON RUNCYCLE RUN_CYCLE1 VALIDFROM 03/31/2008 VALIDTO 04/12/2008 DESCRIPTION
"Inclusive Run Cycle" VARTABLE TABLE1 "FREQ=DAILY;" FDIGNORE
(AT 0700 +2 DAYS UNTIL 0900 +2 DAYS ONUNTIL SUPPR DEADLINE 1000 +2 DAYS)

EXCEPT RUNCYCLE RUN_CYCLE2 VALIDFROM 03/31/2008 VALIDTO 04/12/2008 DESCRIPTION
"Exclusive Run Cycle" CAL1 FDPREV
(AT 0700 +2 DAYS)

EXCEPT RUNCYCLE RUN_CYCLE3 VALIDFROM 03/31/2008 VALIDTO 04/12/2008 DESCRIPTION
"Exclusive Run Cycle" 04/01/2008 FDNEXT
(SCEDTIME 0700 +2 DAYS)

SCEDTIME 0700 TZ Europe/Berlin +2 DAYS UNTIL 0900 TZ Europe/Berlin +2 DAYS
ONUNTIL CONT DEADLINE 1000 TZ Europe/Berlin +2 DAYS
CARRYFORWARD
MATCHING RELATIVE FROM -0200 TO 0300
FOLLOWS FTA1#JS1.@

KEYSCHED
LIMIT 10
PRIORITY HI

NEEDS 4 FTA1#RES1
OPENS FTA1#"myfile.txt"
PROMPT PRMT1
PROMPT "!Continue?"
PROMPT ":js2 started"
PROMPT "ready for js2? (y/n)"
```

:

```
# Sample Job 1
FTA1#JOBDEF1 AS JOB1
    AT 0700 TZ Europe/Berlin +2 DAYS UNTIL 0900 TZ Europe/Berlin +2 DAYS ONUNTIL
    CANC DEADLINE 1000 TZ Europe/Berlin +2 DAYS
    EVERY 0100
    CONFIRMED
    PRIORITY GO
    KEYJOB
    CRITICAL
    PROMPT PRMT1
    OPENS FTA1#"myfile.txt"
    FOLLOWS FTA1#JS1.@ PREVIOUS

# Sample Job 2
FTA1#JOBDEF2 AS JOB2
    PRIORITY 15
    FOLLOWS JOB1 SAMEDAY

END
```



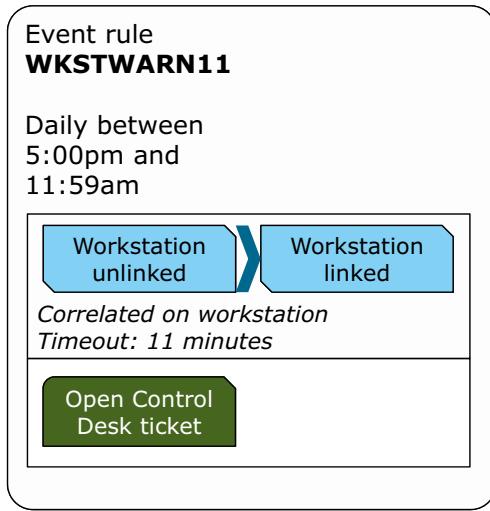
Lesson 6 Creating and using event rules

Creating and using event rules

© Copyright IBM Corporation 2017

Event-driven workload automation is a Workload Scheduler feature that starts predefined actions in response to events. In this lesson, you learn the concepts of event management, and how to use this feature in the Dynamic Workload Console.

Event rule basics



Event trigger

Internal events

- Job or job stream status change
- Workstation change
- Event queues filling

External events

- File created, deleted, modified
- Log file entry matches

Action

Timeout action

Perform new work

- Submit jobs, job streams
- Reply to prompts
- Run commands

Forward information

- Send alerts
- Open trouble tickets
- Log messages
- Forward events to external monitoring

© Copyright IBM Corporation 2017

Event rule basics

An *event rule* encapsulates the monitoring, correlation, and triggering of actions in the Workload Scheduler environment. Event rules consist of *availability*, *events*, and *actions*.

When you define an event rule, you specify its name and its *availability*. An event rule can be in draft mode, making it unavailable. You can also specify that an event rule is available only during certain hours of the day.

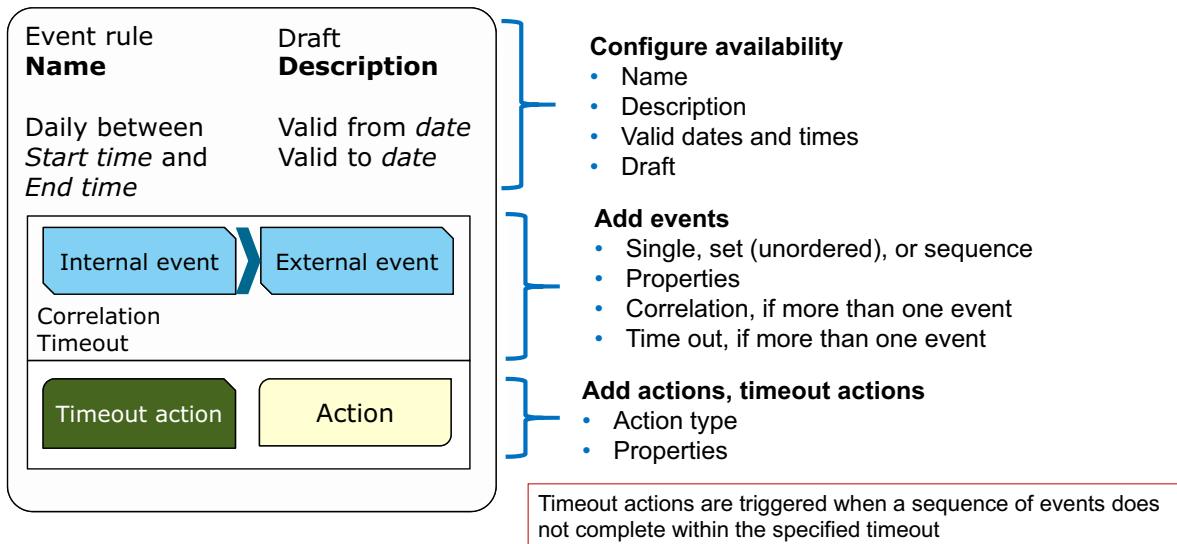
An *event* (sometimes called an *event trigger*) represents a set of circumstances that match criteria you select. Events can be related to Workload Scheduler scheduling objects, changes to file systems and log files, and custom events sent by external applications.

The events that Workload Scheduler can detect for action triggers can be internal or external:

- *Internal events* involve Workload Scheduler internal application status and changes in the status of Workload Scheduler objects. Events of this category can be job or job stream status changes, critical jobs, or job streams that become late or canceled, and workstation status changes.
- *External events* not directly involving Workload Scheduler might nonetheless affect workload submission. Events of this category can be messages that are written in log files, events sent by third-party applications, or a file that is being created, updated, or deleted.

When the Workload Scheduler detects an event or a sequence of events, an *action* or series of actions can be started to respond to the event chain. As with events, you can specify various actions, including notification, submitting commands and jobs, and custom-defined commands.

Defining the event rule object



© Copyright IBM Corporation 2017

Defining the event rule object

You define an event rule in the Dynamic Workload Console to specify triggering events, correlating conditions, and corresponding actions. You can also specify validity dates, a daily time interval of activity, and a common time zone for all the time restrictions that are set.

Configuring availability

To configure event rules in the Dynamic Workload Console, go to **Administration > Workload Design > Create Event Rules**. In the **General Information** section, you specify the name of the rule, the valid dates for the rule, and the valid daily times. For example, you can create a rule to be valid only from 1 September through 31 October, and only from 5 p.m. through 8 a.m.

Adding events

In the **Events** view, you can specify the event to monitor. You can define multiple events that must occur in a specific sequence and specify common values to correlate in the sequence. To define the parameters for a particular event, you select the event and open the **Properties** view.

When you select an event type, you see the details about that event type in the **Properties** view. Different events have different types of properties. You add properties to more precisely specify conditions that must be satisfied to trigger an action.

Specifying multiple events

You can also specify multiple events to occur in a specific sequence, and within a specific time window, for the action to be triggered.

An *event set* is used if you want a response action to start if all the specified events occur, regardless of the order, within the defined time period. Use an *event sequence* if you want a response action to start if all the specified events occur *in a specific order* within the defined time period.

If you selected a *set* or *sequence* of two or more events, you can optionally specify a *timeout period*. When you set the timeout option, the specified period is calculated starting from the first event that occurs. If all the remaining events occur within the specified timeout, the *response actions* are started. If the timeout period expires before all the specified events take place, the defined *timeout actions* start.

Correlating multiple events

Within a rule, two or more events can be correlated through *correlation* attributes such as a common workstation or job. The correlation attributes provide a way to direct the rule to create a separate rule or copy of itself for each group of events that share common characteristics. Typically, each active rule has one copy that is running on the event processing server. However, sometimes the same rule is needed for different groups of events, which are often related to different groups of resources. Using one or more correlation attributes is a method for directing a rule to create a separate rule copy for each group of events with common characteristics.



Hint: When monitoring for a sequence of events with names that match wildcards, you must be sure all of the events in the sequence are monitoring the same object. Without correlation, each event in the sequence might monitor *any* event that matches the wildcard specification. This action would cause a *false positive trigger*, incorrectly starting the actions that are defined in the event rule.

Specifying actions

In the Actions view, you can specify what is done if one or more events occur. Use the **Timeout actions** tab to specify the action that is taken after a specified period of inactivity.

To add an action, follow these steps:

1. Expand the action category for the type of action you want to add.
2. Click the link that relates to the action to trigger. For example, to have the event rule trigger a job stream to be submitted, click **Tivoli Workload Scheduler actions** to expand the action type. Click **Submit job stream** to add a submit job stream action to the list of actions in the event rule.

3. When you define an action, you can specify the details of the action in the Properties view. For example, for an email notification action, you can specify the address, subject line, and the body contents of the message.

As with events, a wide variety of actions are available. Notification actions can notify users of events through email, Tivoli Enterprise Console events, SmartCloud Control Desk, and message logs. Operational actions initiate Workload Scheduler plan actions such as submitting or a job or replying to a prompt. Custom actions initiate operating system commands on the computer that runs the event processor.

Configuring multiple actions

You can specify that an event rule triggers multiple actions. Add each new action by repeating the same steps.

When you specify more than one action in an event rule, each action is run in the order it was defined. However, remember these points:

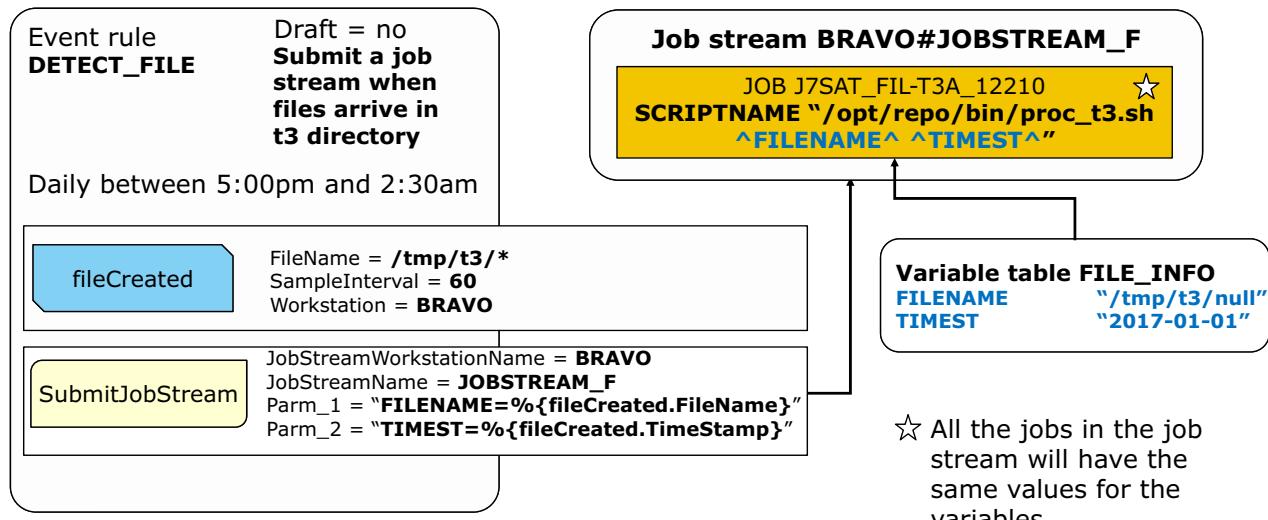
- The order that the actions run is not guaranteed. The second action might start running before the first action is complete.
- There is no dependency relationship between actions. That means, if the first action ends in error, the other actions that are defined in the event rule still run.
- No data is passed between the actions.

To have multiple actions start in order, with dependency relationships, create a job for each action, add the jobs to a job stream, including dependencies. Specify that the event rule submits the job stream.



Hint: Use the **Message logger action** to produce an entry in the database that you can monitor from the Dynamic Workload Console. Messages that are created by the message logger action can be monitored by clicking **System Status and Health > Monitor Operator Messages**.

Passing data from the event to the action



© Copyright IBM Corporation 2017

Passing data from the event to the action

Each event in an event rule carries data about the environment at the time the event was detected. This data can be passed to the action by using *event variables*. Each type of variable has different kinds of data that can be passed to the actions you configure in the event rule.

For example, you require an email notification when a job ends in error. To show which job ended in the **Subject** field, in the Properties section, click **Variables** and choose the job name from the list of available variables. The variable name is included, enclosed in braces, in the Subject line.

You can also use the event variables to populate jobs within job streams the event rule submits. To enable this activity, follow these steps:

1. Create a variable table to hold the variables and their default values. While this step is not strictly required, it makes managing the data easier and streamlines troubleshooting later.
2. Create job definitions with the task that contains the variables that you configured in Step 1. You can create as many jobs as necessary to complete the work flow. All of the jobs can use all or some of the variables in the variable table.
3. Create a job stream to manage the dependencies and ordering of the jobs. Specify the variable table that you created in Step 1 as the job stream variable table. You are not required to create any run cycles for the job stream because event rules submit the job stream as necessary.
4. Add the jobs that you created in Step 2 to the job stream, and specify any dependencies to order the job runs.

5. Create an event rule to submit the job stream.
 - Choose any events that you require to be monitored before submitting the job stream, for example, **File monitor > File created**. Fill the required properties for the event.
 - Choose the action **Tivoli Workload Scheduler actions > Submit job stream**. Complete the properties to configure which job stream to submit.
 - For each event variable you want to pass, choose **Select a property to add > Custom parameter n**.
 - In the **Custom parameter n** field, enter the name of the variable from the variable table, followed by an equal sign, for example, `FILENAME=`. Click **Variable** to see a list of variables that are provided by the event. Select the event variable to use. You see its name in the **Custom parameter n** field in braces after the variable name; for example, `FILENAME=%{fileCrtEvt1.filename}`.
 - After you add all of the required parameters, save and close the event rule.
6. Wait for the event rule to become active.

Troubleshooting event rules

	Event not triggered	Event triggered
Key points to remember		
<ul style="list-style-type: none"> • Event rules can detect only when something changes • Event rules cannot detect if something remains the same as it was before • Event rules do not detect things that happened before they were deployed 	File exists Workstation unlinked » Workstation still unlinked File created at 9:57 » Event rule deployed at 10:01	File does not exist » File exists Workstation unlinked » Workstation linked Event rule deployed at 9:57 » File created at 10:01
The event rule is available		
<ul style="list-style-type: none"> • Does the rule have start times or dates? 		
The event rule is deployed		
<ul style="list-style-type: none"> • Determine if the event rule is active 		Monitor triggered actions to see the result <ul style="list-style-type: none"> • Properties show the results of the action • See how variables, if any, were substituted

© Copyright IBM Corporation 2017

Troubleshooting event rules

When managing event rules, two main areas of troubleshooting prevail. One is matching the event rule definition to actual conditions. The other is matching the action definition to the wanted action's performance.

Matching the event rule to actual conditions

When you create or modify an event rule, it is not immediately effective. The Workload Scheduler event processor deploys the event monitors to the appropriate agents after the rule is stored in the database. You can see the event rule status in the Dynamic Workload Console by viewing **Administration > Manage Event Rules > All Event Rule Definitions**. By viewing the **Internal Status** column, you can determine whether the event rule is active.

The event processor can detect only when something changes. It cannot detect a preexisting condition. Check to determine whether the event occurred outside the event rule's availability period. For example, if the event rule monitors for a file to be created, but the file exists before the event rule becomes effective, the event rule is not triggered.

Each event in a set or sequence must be different in order for the event rule to be satisfied. For example, in an event sequence of **Job Status Changed - Running > Job Status Changed - Running**, the second event in the sequence would never be detected because the event processor can detect only changes.

To see what event rules Workload Scheduler managed, go to **System Status and Health > Monitor Event Rules > All Event Rules**. You can see in the resulting list which events were processed.

Matching the action definition to the wanted action's performance

When an event rule is triggered, it starts the actions that are defined in the rule. Sometimes the actions are not correctly run. Usually this failure is a result of misspelled object or variable names.

To see what event rules Workload Scheduler managed, go to **System Status and Health > Monitor Triggered Actions > All Triggered Actions**. You can see in the resulting list which event actions were run. Click the item in the **Type** column to see the details about the action that is performed by the event rule. On the Properties page, you can see how variables were replaced, which action was attempted, and the result.

Creating and modifying event rules by using composer

- Use **composer new event** to create event rules starting with a template
- You can modify one or many event rules in one session
- Event rules are exposed in XML format
- Use **planman deploy** to compile and send new event rule monitoring configurations to the agents

```
[wsuser@ws94mdm0 ~] $ composer
IBM Workload Scheduler(UNIX) /COMPOSER 9.4.0.01 (20170626)
- modify event=RULE1
AWSBIA291I Total objects: 1
AWSBIA288I Total objects updated: 0
```

Creating and modifying event rules by using composer

© Copyright IBM Corporation 2017

To create an event rule that uses the command-line interface, run the `composer` command. Enter `new eventrule` at the dash (-) prompt. You edit a template that contains an event rule that has several events and actions, by using standard text editor commands. You can also edit an existing run cycle group by using the `composer modify eventrule=event_rule_name` command. Event rule definitions are saved in XML format.



Hint: You can shorten `eventrule` to `erule` in `composer`.

Here is an example of editing an event rule in `composer`:

```

<?xml version="1.0"?>

<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
               xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules
               http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules/EventRules.xsd">

    <eventRule name="RULE1" ruleType="filter" isDraft="yes">

        <description>Sample Event Rule</description>
        <timeZone>Europe/London</timeZone>
        <validity from="2008-02-05" to="2008-02-29"/>
        <activeTime start="06:00:00" end="17:00:00" />

        <eventCondition name="fileCrtEvt1" eventProvider="FileMonitor"
                        eventType="FileCreated">
            <filteringPredicate>
                <attributeFilter name="FileName" operator="eq">
                    <value>/myfile.txt</value>
                </attributeFilter>
                <attributeFilter name="SampleInterval" operator="eq">
                    <value>60</value>
                </attributeFilter>
                <attributeFilter name="Workstation" operator="eq">
                    <value>FTA1</value>
                </attributeFilter>
            </filteringPredicate>
        </eventCondition>

        <action actionProvider="MailSender" actionType="SendMail"
               responseType="onDetection">
            <parameter name="To">
                <value>recipient@mycompany.com</value>
            </parameter>
            <parameter name="Subject">
                <value>File %{fileCrtEvt1.FileName} was created</value>
            </parameter>
        </action>
    </eventRule>

    <eventRule name="RULE2" ruleType="sequence" isDraft="yes">

```

```

<description>Sample Event Rule</description>
<timeZone>Europe/London</timeZone>
<validity from="2008-02-26" to="2008-03-20"/>
<activeTime start="06:00:00" end="17:00:00" />

<eventCondition name="jobLateEvt1" eventProvider="TWSObjectsMonitor"
eventType="JobLate">
    <filteringPredicate>
        <attributeFilter name="JobStreamWorkstation" operator="eq">
            <value>FTA1</value>
        </attributeFilter>
        <attributeFilter name="JobStreamName" operator="eq">
            <value>JS1</value>
        </attributeFilter>
        <attributeFilter name="JobName" operator="eq">
            <value>JOB1</value>
        </attributeFilter>
    </filteringPredicate>
</eventCondition>

<eventCondition name="jobMaxDurationExceededEvt1"
eventProvider="TWSObjectsMonitor" eventType="JobMaxDurationExceeded">
    <filteringPredicate>
        <attributeFilter name="JobStreamWorkstation" operator="eq">
            <value>FTA1</value>
        </attributeFilter>
        <attributeFilter name="JobStreamName" operator="eq">
            <value>JS1</value>
        </attributeFilter>
        <attributeFilter name="JobName" operator="eq">
            <value>JOB1</value>
        </attributeFilter>
    </filteringPredicate>
</eventCondition>

<eventCondition name="jobMinDurationNotReachedEvt1"
eventProvider="TWSObjectsMonitor" eventType="JobMinDurationNotReached">
    <filteringPredicate>
        <attributeFilter name="JobStreamWorkstation" operator="eq">
            <value>FTA1</value>
        </attributeFilter>
        <attributeFilter name="JobStreamName" operator="eq">
            <value>JS1</value>
        </attributeFilter>

```

```

<attributeFilter name="JobName" operator="eq">
    <value>JOB1</value>
</attributeFilter>
</filteringPredicate>
</eventCondition>

<eventCondition name="jobPromoEvt1" eventProvider="TWSObjectsMonitor"
eventType="JobPromoted">
    <filteringPredicate>
        <attributeFilter name="JobStreamWorkstation" operator="eq">
            <value>FTA1</value>
        </attributeFilter>
        <attributeFilter name="JobStreamName" operator="eq">
            <value>JS1</value>
        </attributeFilter>
        <attributeFilter name="JobName" operator="eq">
            <value>JOB1</value>
        </attributeFilter>
    </filteringPredicate>
</eventCondition>

<eventCondition name="jobRiskLvlChgEvt1" eventProvider="TWSObjectsMonitor"
eventType="JobRiskLevelChanged">
    <filteringPredicate>
        <attributeFilter name="JobStreamWorkstation" operator="eq">
            <value>FTA1</value>
        </attributeFilter>
        <attributeFilter name="JobStreamName" operator="eq">
            <value>JS1</value>
        </attributeFilter>
        <attributeFilter name="JobName" operator="eq">
            <value>JOB1</value>
        </attributeFilter>
    </filteringPredicate>
</eventCondition>

<eventCondition name="wksStatChgEvt2" eventProvider="TWSObjectsMonitor"
eventType="WorkstationStatusChanged">
    <filteringPredicate>
        <attributeFilter name="Workstation" operator="eq">
            <value>FTA1</value>
        </attributeFilter>
    </filteringPredicate>
</eventCondition>

```

```
<action actionProvider="MessageLogger" actionType="MSGLOG"
responseType="onDetection">
    <parameter name="ObjectKey">

<value>%{wksStatChgEvt2.Workstation}#{jobLateEvt1.JobStreamName} .#{jobLateEvt1
.JibName}</value>
    </parameter>
    <parameter name="Severity">
        <value>Warning</value>
    </parameter>
    <parameter name="Message">
        <value>Job #{jobLateEvt1.JibName} is late</value>
    </parameter>
</action>

</eventRule>

</eventRuleSet>
```

Lesson 7 Forecasting and previewing future dates

Forecasting and previewing future dates

© Copyright IBM Corporation 2017

You can check to see that job streams you create are planned for specific dates. In addition to the production plan, which you actively monitor, Workload Scheduler maintains an archive of old production plans from previous days. You can also create *trial plans* and *forecast plans* to project what the production plan would be at a different time. In this lesson, you learn about alternative plan types and how to create and view them. You also learn about viewing the Workload Scheduler preproduction plan.

Alternate plan types

Create a...	To see...	In order to answer questions like...
Forecast plan	What the production plan would look like for a specific time interval	What job streams will be selected on 1 January 2018?
Trial plan	What the current production plan would look like if it was extended	If today's plan is extended by 12 hours, will include my new job stream?
Archived plan	What a completed production plan from a previous date looked like at the end of the day	What was the status of my jobs on 24 December 2016?

View alternate plans in one of the following ways:

- Set the alternate plan temporarily as the default plan to monitor
- Associate a monitoring task to an alternate plan
- Associate the alternate plan to an engine connection

© Copyright IBM Corporation 2017

Alternative plan types

You can view archived production plans, and create trial and forecast plans from the Dynamic Workload Console.

Forecast plans

You can create a *forecast plan* to see a projection of what the production plan would be for a specific time interval. For example, if you want to know what the plan would be for next weekend, you can generate a forecast plan that covers those days.

You can view forecast plans just as you view the production plan. In a forecast plan, you see all the job streams, jobs, scheduling objects and dependencies that are normally part of a production plan. Even start times can be predicted in a forecast plan if workload service assurance is enabled in Workload Scheduler. Workload Scheduler calculates the average run for each job based on all previous runs, and uses the average durations to compute predicted start times.

Updates being processed in the current production plan are not considered in forecast plans. When you view a forecast plan, you see that all the job streams are in one of the following states:

- **HOLD** if they depend on other job streams or if their start time is later than the plan start time.
- **READY** if they have no dependencies and their start time elapsed.

Trial plans

A *trial plan* is a projection of what the production plan would be if it covered a longer time. For example, if today's production plan ends at 5:59, but you want to know what the plan would be if it ended at midnight, you can generate a trial plan.

You typically create a trial plan to get an idea of future impacts of scheduling changes on the scheduling environment. If there is a valid production plan, the start time option is gray and unavailable. By default, the trial plan start date is the same as the production plan end date.

Using this type of plan you can, for example, see how the current production evolves based on the job and job stream dependencies that are defined in the production plan, if available, or in the preproduction plan. Trial plans are based on the information that is either in the production or in the preproduction plan. If neither is available, a trial plan cannot be created.

Archived plans

An *archived plan* is a copy of an old production plan that ran in the Workload Scheduler environment. Using this type of plan you can, for example, see the results of running a past production plan. You can see the results of all of the jobs that ran in the archive time interval, and which jobs and job streams did not start before the plan ended.

The difference between using an archived plan and a forecast plan that covers the same time interval is that an archived plan shows how the real production jobs and job streams were processed. A forecast plan shows how the production was planned for that time.

It is sometimes useful to create a forecast of a past date to compare with the actual archived production for the same day.

Creating trial and forecast plans

The procedure for creating trial and forecast plans is nearly the same. To create a trial plan, from the navigation bar, click **Planning > Create Trial Plan**. To create a forecast plan, from the navigation bar, click **Planning > Create Forecast Plan**.

On the Generate Plan page that is shown, perform these steps:

1. Select the engine that you use to generate the plan.
2. Enter a name for the plan (required).
3. For forecast plans only, select a date and time for the start of the plan. For trial plans, the start time is predefined as the end of the current production plan.
4. Define a plan end date by performing *either* of these steps:
 - Select the **Date and time** option and choosing a specific date and time
 - Select the **Duration** option and entering the number of days, hours, and minutes that the plan should cover

5. Select the appropriate time zone for the plan.
6. Click **Generate Plan**. You see a message that confirms the creation of the plan or a message that the plan is being generated in the background.
7. Click **OK**.
8. You can close the Generate Plan page or create another plan by using the same steps.

Viewing alternative plans

You can choose which plan to use for monitoring. The default is the active production plan, but you can select to monitor an archive, trial, or forecast plan instead. There are several ways to use an alternative plan for monitoring.

Applying a direct query to an alternative plan

When you monitor plan objects by using the Direct Query interface, you can select to monitor them in an alternative plan by selecting the plan from the **List Plans** menu before you run or edit the query.

Associating a plan to a monitoring task

Monitoring task definitions include the plan they reference. You can create or modify monitoring tasks so that they reference an archived, forecast, or trial plan instead of the active production plan. For more information about how to create and modify monitoring tasks, see [“Using monitoring tasks”](#) on page 3-10.

Defining a temporary plan

You can make a temporary plan selection by changing the active plan to an alternative plan. This plan is used as the active plan for the current session only. All tasks and engines that are set to use the *active plan* use the plan that you select. This plan selection is valid unless a different plan is defined as the active plan in the monitoring task configuration.

To temporarily set the active plan to an alternative plan, follow these steps to define a filter, list, and select an alternative plan from the available plans:

1. From menu, click **Planning > Manage Available Plans**.
2. On the List Available Plans page:
 - Under **Select an Engine**, select the engine where you want to list the plans.
 - Under **Select Plan Type**, select the type of plan you want to list. Selections are mutually exclusive. By default, all available plans are listed.
 - Under **Select Plan Name**, optionally, specify the name of the file that contains the plan that you want to search. You can use wildcard characters.
 - Click **Display Plans List** to see a list of available plans.
3. From the **List Available Plans** displayed list, select a plan and click **Set as active** to set it as the active plan. The plan that you select is the active plan for monitoring until you log off or until you return to this list and choose **Restore default plan**.
4. Click **Close View** to return to the Manage Available Plans page.

For the rest of the current session, when you monitor any plan objects that use tasks that select the *active plan*, you are viewing the alternative plan that you selected.

Associating a plan to an engine

If none of the above settings are specified, tasks are run against the plan that is associated to the engine. By default, this plan and the active plan are the same. You can create an engine definition that references a specific alternative plan.



Hint: Creating an engine to monitor an alternative plan is useful because you can create monitoring tasks to monitor multiple engines. You might create a monitoring task to compare the results of two or more different plans in the same window by using this method.

To associate a plan to an engine, perform the following steps:

1. From the menu, click **System Configuration > Manage Engines**.
2. On the Manage Engines page, perform *either* of these steps:
 - Select an engine and click **Connection Properties** to modify an existing engine
 - Click **New** to create an engine definition
3. Under the **Plan** section, click **Select** to view a list of the available plans.
4. Choose a plan from the list and click **OK**.
5. Click **OK** to save the engine connection properties.

Any monitoring task that you create or modify that references this engine monitors the plan that you selected.



Note: This setting is overridden by the plan selection that is made in the monitoring task definition or, if not available, by the selection that is made on the List Available Plans page.

Using the preproduction plan

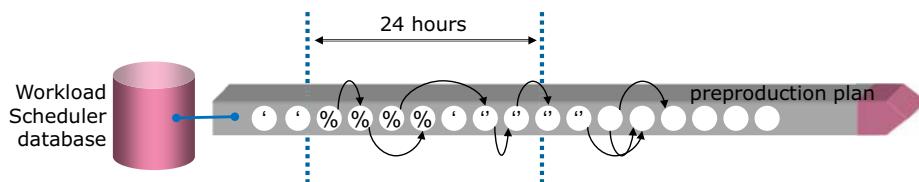
You cannot edit the preproduction plan directly, but you can edit job streams and reload the preproduction plan view

The **preproduction plan** includes the following elements:

- Job stream instances planned to run
- External dependencies
- Scheduled times

The preproduction plan does not include the following elements:

- Jobs
- Other dependencies
- Run times
- Status



© Copyright IBM Corporation 2017

Using the preproduction plan

The preproduction plan is used to identify the job stream instances and the job stream dependencies for a specified period in advance. This method improves performance when generating the production plan by preparing a high-level schedule of the predicted production workload.

The preproduction plan contains the following elements:

- The job stream instances that are scheduled to run during the time interval of the plan.
- The external follows dependencies that exist between the job streams and jobs included in different job streams.

Workload Scheduler automatically generates, expands, and updates the preproduction plan. You cannot modify it manually.

All users are allowed to view the preproduction plan. However, the preproduction plan content depends on the permissions you have on job streams. You can see only the job streams that you are allowed to see.

Viewing the preproduction plan

You can view the preproduction plan in the Dynamic Workload Console to get an overview of job streams and their external dependencies for certain dates, or all dates.

From the Dynamic Workload Console, you can view the preproduction plan graphically. When you view the preproduction plan, you see only the job streams with the time they are scheduled to start and their external dependencies. All the remaining information about the job streams and the other

scheduling objects (calendars, prompts, domains, workstations, resources, files, and users) are not included.

To open the preproduction plan view, perform these steps:

1. From the **Planning** menu, select **View Preproduction Plan**. On the displayed page, select the engine whose preproduction plan you want to view.
2. Optionally, specify a filter to reduce the number of job streams that are shown in the view. Only the job streams containing the string that you entered as a filter are displayed in the preproduction plan view. By default, all job streams are shown. You can change your filtering criteria directly from the preproduction plan graphical view page.
3. Specify the start and end dates to view only a portion of your preproduction plan. If you do not specify any date, the whole plan is shown. Optionally, you can organize the view by grouping the job streams by scheduled date.

In the preproduction plan view, you see the job streams included in the plan along with their job stream dependencies. Each box represents a job stream. Right-click a job stream to open the job stream definition within Workload Designer.



Hint: You can modify a job stream from the Workload Designer and reload the updated preproduction plan view.

Lesson 8 Exporting and importing scheduling definitions

IBM Training



Exporting and importing scheduling definitions

© Copyright IBM Corporation 2017

In this lesson, you learn how to migrate workload definitions by using workload application templates, and import scheduling definitions from UNIX crontabs and Windows scheduler tasks.

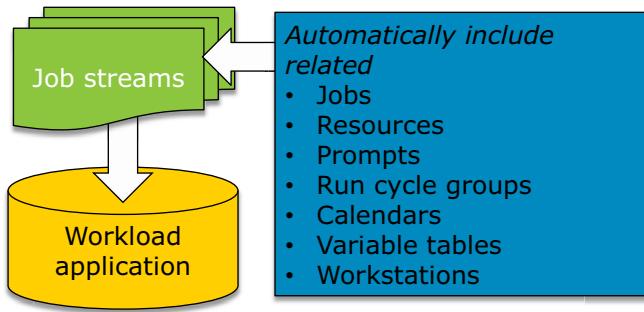
IBM Training



Using workload application templates

Useful for

- Migrating applications from test to production
- Making backups of critical application definitions
- Sharing business process with other Workload Scheduler users



© Copyright IBM Corporation 2017

Using workload application templates

A *workload application template* is a database object that contains one or more job streams and all of their related jobs and internal or external dependencies such as files, resources, and prompts. You use the template to transfer a set of workload definitions from one Workload Scheduler environment to another. Additionally, you can map the names of objects in the template to the names of similar objects in the environment into which you import the template. For example, you can export job streams from workstation ALPHA in one environment and import them to workstation GAMMA in another environment. From the Workload Designer, you can create the template of a workload that can be imported and run in another environment. You can create a workload application template that contains one or more job streams. You can export the workload application template to deploy and run it in another environment.

To use workload application templates, in general, you perform these steps.

1. From the Workload Designer, create the workload application template, and add job streams to it.
2. Save the workload application.
3. Export the application from your environment, saving the compressed file to your local system.
4. Check and optionally edit the mapping properties file.
5. Transfer the files to the new environment (they must be decompressed on the target environment).

6. Import the template from the Dynamic Workload Console or by using the `wappman` command.

The export process produces a compressed file package that contains three files as described in [“Managing workload applications”](#) on page 6-73.

Considerations for using workload application templates

Contents

- Definitions
- Source Environment
- Mapping

Tip: Use variables in your job definitions, and Windows users definitions, so that you can easily adapt them to different environments by just changing the variable table

Tip: Maintain the mapping file separately; you use the same mappings frequently

Tip: Use specific variable tables related to your business application rather than the default variable table. In this way, your variables do not conflict with the default variables on the target system

© Copyright IBM Corporation 2017

Considerations for using workload application templates

Before you create workload application templates, there are some practices that you should consider to make the objects you move more portable.

Maintaining the mapping files

Separately maintain the mapping files that you use frequently. Build a mapping file that represents the changes from one environment to another. You can reuse the same mapping often, without editing the file in the application template each time.

Creating scheduling objects more effectively

There are several considerations to take when building your scheduling objects and definitions. By following these recommendations, your definitions are more portable and easier to migrate to different environments.

Job definitions

Jobs that refer to elements that depend on the environment or topology in which they run, such as web service jobs, file transfer jobs, and database jobs to name a few, should use variables when specifying these elements such as credentials, paths, and port numbers. Variables can be managed in the mapping file so that the correct values can be assigned to the variable on the target system.

Workstation names

When jobs and job streams are extracted from the workload application during the export process, the names of the workstations are extracted as they are found in the source environment.

Meaningful names or a standardized naming convention can simplify the mapping process.

Users

User definitions are also extracted as they are found in the source environment. If the same user is not present in both source and target environment, variables should be used to specify the user.

Job stream variable table

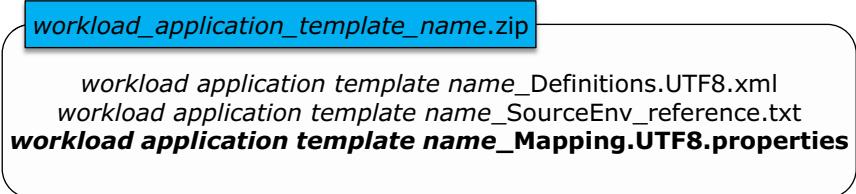
Variables representing objects in the workload application should be added to a specific variable table related to the job stream in the workload application. This method enables the customization of the job stream to reflect the target environment through the mapping file. Avoid associating the default variable table to a job stream. The default variable exists on the target environment. As a result, the import process fails because a variable table with the same name exists.

Run cycle variable table

Variables representing objects in the workload application should be added to a specific variable table related to the run cycle in the workload application. This method enables the customization of the run cycle to reflect the target environment through the mapping file. Avoid associating the default variable table to a run cycle. The default variable exists on the target environment. The import process fails because a variable table with the same name exists.

Creating and managing workload application templates

-  1. Select which job streams to package
 2. Export the workload application
 3. Upload the package file
 4. Edit the mappings
 5. Import the workload application



command line

```
wappman -import template_name_Definitions.UTF8.xml
template_name_Mapping.UTF8.properties
```

© Copyright IBM Corporation 2017

Creating and managing workload application templates

From the Workload Designer, you can create the template of a workload that can be imported and run in another environment. You can create a workload application template that contains one or more job streams. You can export the workload application template to deploy and run it in another environment.

Creating a workload application template

To create a workload application template, perform the following steps:

1. From the navigation toolbar, click **Administration > Manage Workload Definitions**. Specify the engine connection. Click **Go** to open the Workload Designer.
2. In the Workload Designer, select **Create New > Workload Application Template**. You see the workload application template and its properties in the **Details** view.

In the properties pane, specify the attributes for the workload application template that you are creating, as shown in the following table.

Name (required)	The name of the workload application template. The maximum length is 80 characters.
-----------------	---

Description	Text to help workload application users understand the purpose and characteristics of the workload application. The maximum length is 120 characters.
Provider	Specify the creator of the workload application template. It can be useful to inform workload application users about who created and provided it. The maximum length is 120 characters.

3. From the **Details** view, right-click the workload application template and click **Add Job Stream** to add job streams to it.
4. From the lookup window, select the job streams that you want to add. The corresponding dependencies for the job streams you select are automatically also added to the workload application template.
5. Click **Save** to store the workload application template in the database.

Exporting a workload application template

To export the workload application template, right-click the workload application template and click **Export** to produce a compressed file, named `workload_app_export.zip`. Save the compressed file locally.

Managing workload applications

When you export the workload application, the compressed file contains three files:

- ***workload application template name_Definitions.UTF8.xml***

The file is an XML file that contains the definitions of all the exported objects. These definitions are imported into the database on the target environment. The objects in the definition file can remain as they are or you can rename them by editing the mapping file. If an object does not have a definition in the definition file, for example, a workstation, at import time, a corresponding object is not created in the target environment. The expectation is that such object is already present in the target environment. Therefore, for these types of objects, you must map them in the mapping file.

- ***workload application template name_Mapping.UTF8.properties***

This file is the mapping file that is used to replace the names of the objects in the source environment with the names that these objects have in the target environment. The objects that are created in the target environment can have the same names as the objects in the source environment, or you can specify a different name in this file.

- ***workload application template name_SourceEnv_reference.txt***

This file contains reference information with the definitions of the workstations that are used in the workload application template and other information that can be useful to correctly map the source environment into the target environment to allow the workload application to run.

Resolving the mapping file

When you export a workload application, a mapping file is produced with the names of all the objects that are referenced by the job streams in the application.

The mapping file contains a list of these elements that job streams in the workload application use:

- Calendars
- Run cycle groups
- Resources
- Prompts
- Job streams
- Jobs
- Workstations
- Referenced job streams
- Referenced jobs

Some of the elements depend on the topology of the environment in which it is used. These elements must be customized to reflect the target environment. The definitions file can contain a complete object definition or, in some cases, only a name or reference to the object that is extracted. You must map the references to objects that exist in the target environment.

For example, you create a workload application that contains a job stream, which depends on another job stream that is not part of the workload application. The definitions file from the export contains the full definition of the job stream that is part of the workload application, and references to the job stream that is not part of the workload application. You must check and possibly edit the mapping file so that the references are to valid job streams that exist in the target environment.

You can view and edit the mapping file with a text editor. It is organized in sections and contains comments to assist you in assigning the correct values to the elements. You can also change the mappings from the Workload Console while you import the application template, as described in the next section.

Importing the workload application

Import the contents of the compressed file into the target environment where the workload application is deployed. The required objects are created in the target environment.

To import a workload application template, follow these steps.

1. From the Dynamic Workload Console, select **Administration > Import Workload Application**.
2. On the Import Workload Application page, from the **Engine** menu, select the engine connection to use for the import process.
3. Click **Choose file** to select a zip file on your computer to upload.

4. Click **Upload**. You see an information window that confirms the file was successfully uploaded. Click **OK**.
5. For each type of object to import, you can change the name of the object in the template to a different name in the target environment. For example, you can update a run cycle group from **DAILY_TEST** in the template, to **DAILY_PROD** in the production environment. You can also use the **Find and Replace** function to rename the objects to import changing text in the names. For example, you can change all occurrences of “abc” to “xyz”. The find and replace function is not case-sensitive.
6. Click **Import**. You may need to confirm that some of the objects in the template overwrite existing objects in the database. Click **Yes** to confirm the import and overwrite existing objects, or **No** to skip the import.
7. You see a message confirming the import process. Click **OK**.

Updating workload application templates

You can update a workload application after it is created. There are two ways in which the workload application can be updated or modified.

- **Modify the template in the source environment:** An updated version of the template can be deployed again into the target environment. Any objects already present in the Workload Scheduler database of the target environment are replaced with the updated versions. Objects that do not exist in the target environment are created. Objects are deleted from the target environment if the object definition was removed from the updated workload application. The same mapping file that is used to originally deploy the workload application can be used to update it, customizing any new objects that are being deployed with the update.
- **Modify the instance in the target environment:** After a workload application is deployed, you can add a job to a job stream, modify a job definition, or remove a job or job stream. However, these changes are not maintained if the workload application is updated with a revised workload application template.

Lesson 9 Using workload application templates from the command line

Using workload application templates from the command line

© Copyright IBM Corporation 2017

When you download the workload application template to your computer, you can view and edit the mapping file with a text editor. It is organized in sections and contains comments to assist you in assigning the correct values to the elements.

You can import the workload application from workstation ALPHA on the source environment to workstation GAMMA on the target environment by making one change to the mapping file.

```
#-----  
#Workstation names  
#Replace the value with the name of a workstation that already exists in the  
target environment.  
#Refer to the ALPHA_MIGRATE_SourceEnv_reference.txt file for details about the  
workstation.  
#  
#This workstation is of type Dynamic Pool  
WORKSTATION_ALPHA=GAMMA
```

For each object listed in the mapping file, either assign the name of an existing object in the target environment, or rename it with the name you want the object to have in the target environment.

Importing the workload application

To use the workload application in a new environment, import the workload application template after optionally modifying the mapping file to reflect the destination environment.

You start the import operation by running the `wappman` command, passing the mapping file and the definition file as arguments. The `wappman` command can be used to import, replace, list, display, and delete a workload application.

From the command line, submit the following command, indicating the file names of both the definitions file and the customized mapping file:

```
wappman -import definition_xml_file mapping_properties_file
```

Replacing a workload application

You can also replace an existing application template with the following command:

```
wappman -replace definition_xml_file mapping_properties_file
```

An updated version of the workload application template is imported into the target environment. The objects that were initially created in the database are updated if they are still present in the database, created if they are no longer present, and deleted if they are no longer present in the updated workload application template. You can use the same mapping file for the replace operation that you use for the original import operation.

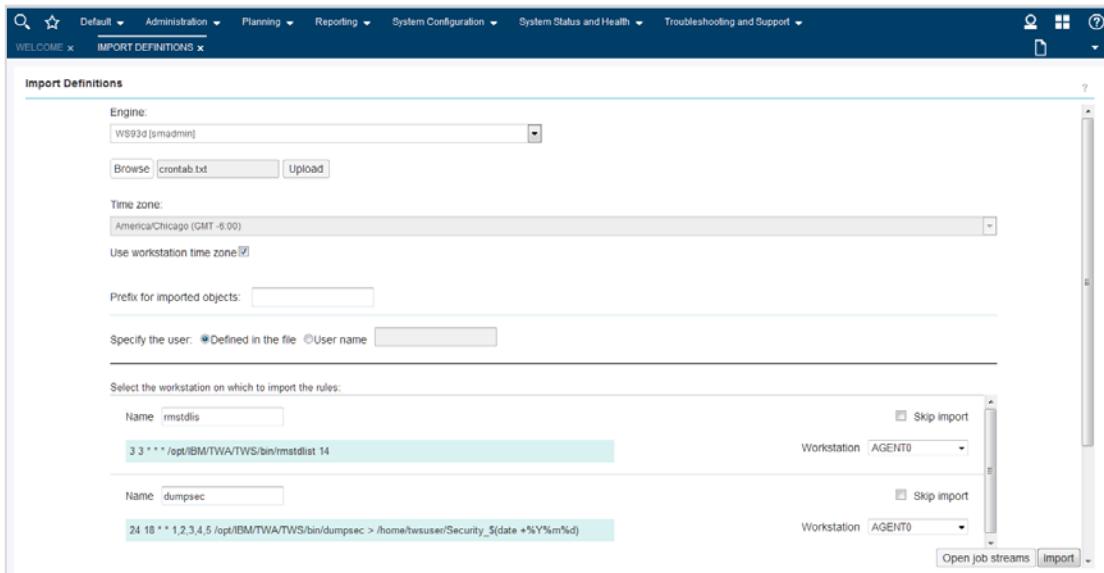
Deleting a workload application

You can delete a workload application and all of its objects, with the following command:

```
wappman -delete workload_application_name
```

The `wappman -delete` command removes the specified workload application and all the objects that were added to the environment when the workload application was originally deployed.

Importing crontab and Windows task scheduler definitions



© Copyright IBM Corporation 2017

Importing crontab and Windows task scheduler definitions

You can import definitions for the UNIX and Linux **cron** scheduler into Workload Scheduler by saving the definitions in a text file, and by using the import feature of the Dynamic Workload Console. You can also import definitions for the **Windows Task Scheduler** by using the same steps.

To import a crontab or Windows task scheduler definition, complete the following steps.

1. Select **Import Definitions** from the **Administration** menu.
2. Select the engine that you want to use from the **Engine** menu.
3. Click **Browse** and find the file on your computer that contains the definitions to import.
4. Click **Upload** to transfer the file to the Workload Scheduler server.
5. You see the tasks that are included in the file you transferred. Complete the following fields.

Table 2 Import processes fields

Field name	Description
Import for user	Specify the operating system user that runs the tasks on the agent you select.
Name	Enter the name that the task is given when the import is complete.
Do not import	Select Do not import if you do not want to import the task.
Agent	Select which Workload Scheduler agent runs the tasks.

6. Click **Import**. You see a message that confirms a successful import, or messages that show syntax or other errors.

To change the properties of the imported tasks in the Workload Designer, click **View job streams**.

Instructor demonstration

- Creating job streams
- Setting time restrictions
- Creating event rules
- Creating forecast plans
- Exporting and importing scheduling definitions



© Copyright IBM Corporation 2017

Instructor demonstration

Exercises

- Create job streams
- Create event rules
- Create forecast plans
- Create workload application templates



© Copyright IBM Corporation 2017

Exercises

Review questions

1. Can you schedule jobs in the same job stream to run on different days?
2. How many job streams can be scheduled per day?
3. How many jobs can there be in a job stream?
4. How do you create a job stream that you do not want to schedule or submit?
5. How do you repeat a job stream each time a job condition is met?
6. When you set a job to **No operation**, what status does the job acquire in the plan?
7. How do you set job dependencies based on return codes or output conditions of the preceding job?
8. What is the confidence factor of the estimated duration of each job?
9. How do you create a job stream that contains different jobs at some date in the future?
10. Which time dependency prevents a job or job stream from starting after the specified time?
11. If you specify a time restriction in both a run cycle and the job stream, which restriction is effective?
12. If two occurrences of a job stream overlap, what options are available to prevent them starting at the same time?
13. How many workstation classes that you create are shown in the production plan?
14. When you specify multiple actions in an event rule, in what order do they start?
15. Which Workload Scheduler object contains one or more job streams and all of their related jobs and internal or external dependencies such as files, resources, and prompts?

Review answers

1. Can you schedule jobs in the same job stream to run on different days?

All jobs in a job stream are scheduled to start on the same run dates. Workload Scheduler does not run jobs on different days in the same job stream. For example, a job stream cannot have jobs that run every Wednesday and other jobs that run every third Thursday.

2. How many job streams can be scheduled per day?

There is no limit on the number of job streams that can be scheduled.

3. How many jobs can there be in a job stream?

There is no limit on the number of jobs within a job stream. There can also be zero jobs in a job stream.

4. How do you create a job stream that you do not want to schedule or submit?

If you are creating a job stream that you want to save, but do not want to run in production, check the Draft option. When Draft is selected, the job stream cannot be selected to run, or submitted to run.

5. How do you repeat a job stream each time a job condition is met?

You can define a start condition to precede the jobs you define in a job stream, and repeat the job stream each time the condition is met.

6. When you set a job to **No operation**, what status does the job acquire in the plan?

When you set the no operation flag, the job is included in the job stream as part of the production plan, but placed in Cancel Pending status.

7. How do you set job dependencies based on return codes or output conditions of the preceding job?

You can create conditional dependencies, which specify that one job depends on another only when specific conditions are met.

8. What is the confidence factor of the estimated duration of each job?

The predictive analysis process associates a 95% confidence factor to the estimated duration of each job, and provides a confidence interval. The confidence interval is an estimated range of values, which is likely to include the job duration with 95% confidence.

9. How do you create a job stream that contains different jobs at some date in the future?

*If you are creating a version of a job stream that will not run until some date in the future, complete the **Valid from** field.*

10. Which time dependency prevents a job or job stream from starting after the specified time?

*The **Latest start time** is the latest time a job or job stream can start processing. In other words, the job or job stream must start before this time.*

11. If you specify a time restriction in both a run cycle and the job stream, which restriction is effective?

Time restrictions in the job stream definition override and take precedence over any other time restrictions that are defined in the run cycles or run cycle groups that are associated to the job stream.

12. If two occurrences of a job stream overlap, what options are available to prevent them starting at the same time?

You can choose to queue the new instance, or not start (skip) the new instance.

13. How many workstation classes that you create are shown in the production plan?

None. Workstation classes do not appear in the plan.

14. When you specify multiple actions in an event rule, in what order do they start?

Event rule actions start in the order in which you define them.

15. Which Workload Scheduler object contains one or more job streams and all of their related jobs and internal or external dependencies such as files, resources, and prompts?

You use the Workload Application Templates to consolidate and transfer a set of workload definitions from one Workload Scheduler environment to another.

Summary

You now should be able to perform the following tasks:

- Create job streams to schedule jobs
- Create event rules to provide workload automation
- Forecast future work
- Export and import scheduling definitions

© Copyright IBM Corporation 2017

Unit summary

Unit 7 Reporting with Workload Scheduler

IBM Training



Reporting with 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 create and run Workload Scheduler reports.

- Create historical reports from the Dynamic Workload Console
- Create production reports from the Dynamic Workload Console
- Create reports with the text-based report utilities

Reference: *IBM Workload Scheduler User's Guide and Reference*

Objectives

In this unit, you learn how to create and run Workload Scheduler reports. You learn how to perform the following tasks:

- Create historical reports by using the Dynamic Workload Console
- Create production reports by using the Dynamic Workload Console
- Create statistical and historical reports by using the Common Reporting tool
- Use the batch reporting utility to run reports without using the Dynamic Workload Console
- Use the text-based report utilities to create reports

© Copyright IBM Corporation 2017

Objectives

Lesson 1 Using reporting features

Using reporting features in the Dynamic Workload Console

© Copyright IBM Corporation 2017

IBM Workload Scheduler maintains a history of the jobs that run as part of the production plan in the database. Using the history and other data in the database, you can create and customize reports. In this lesson, you learn how to create and run Workload Scheduler reports.

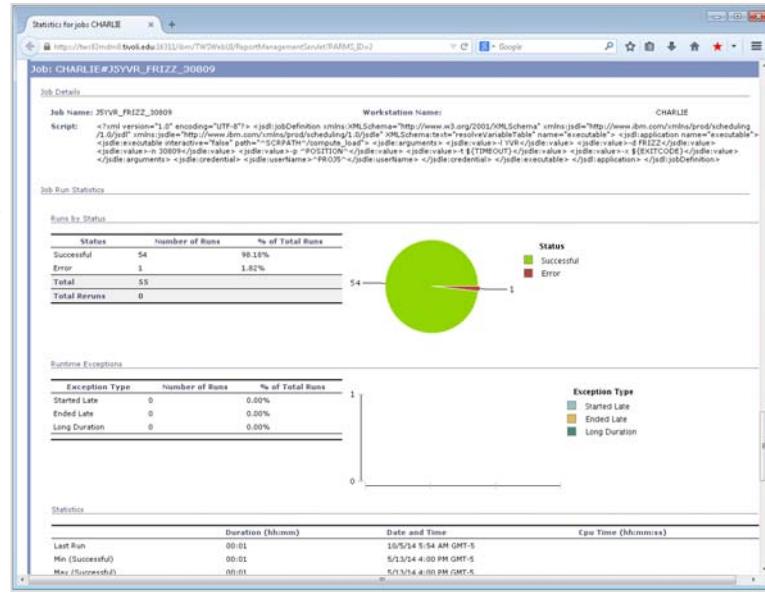
With reporting features in Workload Scheduler, you can get answers to workload-related questions such as these examples:

- Which jobs have a greater than 20 percent error rate?
- During which times are workstations the least busy?
- What days did your report job run the longest?
- In what job streams is your job used?

Within Workload Scheduler, there are several ways to report on the data available:

- Create report tasks in the Dynamic Workload Console
- Use the Common Reporting interface to run existing reports, or create custom reports
- Run the reporting command line against report templates you create or modify
- Run utility commands that show different kinds of data
- Query the database directly by using SQL or other reporting tools

Creating reports by using the Dynamic Workload Console



© Copyright IBM Corporation 2017

Creating reports by using the Dynamic Workload Console

You can create report tasks to customize and generate Workload Scheduler reports. You can view, print, and save reports in different kinds of output. Report tasks are very much like monitoring tasks that you already created.

You can create two types of reports from the Dynamic Workload Console. *Historical reports* collect data from the Workload Scheduler database and format it in different ways, depending on the options you choose. *Production reports* present the production, forecast, or archive plans as either CSV or XML files you view in an external application such as Microsoft Excel or Microsoft Project.

For historical reports, you see the following sections:

- The **Header** contains the report title, description, engine name, engine type, creation time, type, and the total number of the result set extracted.
- A **Table of contents** provides a set of hyper-links to each section and subsection further in the report.
- The **Formatted data** selection contains the data from the history. The report output can be in one of these formats:
 - Table format, which is organized in rows and columns, which are delivered in a CSV or HTML file.
 - Graphical format (HTML), depending on the report type, and on the information you choose, you can have data that is displayed in pie charts, bar charts, line charts, or tables.



Important: Reports that display in HTML format open in a pop-up browser window. Configure your browser to allow pop-up windows from the Dynamic Workload Console server.

Choosing a historical report type

Following are the types of historical reports that you can create tasks to run.

Job run history

In a *job run history report*, you request the historical job run data from the time interval you select. In a job run history report, you see a row of content for every job *occurrence* in the history tables that matches the criteria you select. In other words, this report contains more data for jobs that run more often. Use a job run history report to show detail about jobs that ended in error or were late. You can also see which jobs missed their deadline, or ran too long (duration). You choose which details are on the report, such as rerun indicators for reruns, times, or return codes. Job run history reports are available in table format.

Job run statistics

You request job run statistics in a *job run statistics report*. In a job run statistics report, you see one row for each job *defined* that matches your criteria. In other words, you see only one row for each job no matter how often it runs. Use a job run statistic report to see success, error rates; minimum, maximum, and average durations; late and long duration statistics. This type of report is available in graphic and table formats, and it can be HTML or CSV.

You can select the statistical characteristics of the jobs in the report. You can use the following characteristics:

- Percent of job runs that ended successfully
- Percent of job runs that ended in error
- Percent of job runs that started late
- Percent of job runs that ended late
- Percent of runs that ran longer than the average duration
- How many times the job ran
- How many times the job was rerun

You can combine the characteristics to show better analysis. For example, a simple selection of **% error at least 20** might produce a report that is unfair to new business applications. A new job that ran twice but ended in error once has a 50% error rate. By also selecting **at least 10 total runs**, your report shows only jobs with a higher maturity.

Workstation workload summary

You can use a *workstation workload summary report* to show the workload on workstations. You see the workload in terms of number of jobs that ran on the workstations. These reports help you with capacity planning adjustments (for modeling workloads and tuning workstations). You can select to show how many jobs ran on a selected workstation within an arbitrary date range, or since the last three days, for example. The workstation workload summary report is available in graphic and table formats, and they can be delivered as HTML or CSV. You can choose to show aggregate data for workstations, and group by day, hour, or production day.

Workstation workload runtimes

Use a *workstation workload runtime report* to show job run times and duration on workstations. These types of reports help you make capacity planning adjustments (for modeling workloads and tuning workstations). You see a timeline, which depicts when jobs ran on each workstation. Workstation workload runtime reports are available in graphic and table formats, and can be HTML or CSV files.

Custom SQL

In a *custom SQL report*, you create reports that best fit your business needs by writing or importing SQL queries into the report definition. You can write an SQL query or import SQL scripts. Custom SQL reports are available in table format only, and can be HTML or CSV. The table views and columns for each view are described in the *IBM Workload Scheduler Database Views* guide.

Choosing a production report type

Following are the types of production reports that you can create tasks to run.

Planned production details

You can run a *planned production details report* to extract information about planned production plans into either an XML or a CSV format to be used with Microsoft Project and Microsoft Excel. With this type of report, users who do not know the Workload Scheduler can access plan information in a familiar format. This report is available in CSV or XML format only.

Actual production details

Similar to the planned production details report, use the *actual production details report* to extract current plan information into either an XML or a CSV format to be used with Microsoft Project and Microsoft Excel.

Creating a report task

To create a report task by using the Dynamic Workload Console, follow these steps:

1. From the navigation bar on the left, select **Reporting > Workload Reporting > Manage Workload Reports** to open the **Manage Workload Reports** view.
2. Click **New** to open the “Create Task” page.
3. Select the type of report you want to create. Click **Next**. A wizard guides you through the next steps.
4. Complete the “Enter task information” page with the following information for the report task that you create and click **Next**:
 - The name of the task
 - The engine connection the task uses
 - Other users with whom you want to share the task
5. Enter the title and description for your report in the “Report header” page. Choose whether to include the selection criteria and table of contents on the report. Click **Next**.
6. On the Filter Criteria page, choose the options to select the job run information that appears on the report. Each report type has different forms of selection criteria. For example, in a Custom SQL report task, the filter criteria is an SQL query statement. Click **Next**.
7. Customize the Report Output Content page to meet your requirements. You can choose whether to display charts or tables, select which data to include in the report, and the output type. Click **Next**.

Lesson 2 Using Common Reporting with Workload Scheduler

IBM Training

IBM

Using Common Reporting with Workload Scheduler

© Copyright IBM Corporation 2017

In this lesson, you learn how to use Common Reporting (formerly Tivoli Common Reporting) to build reports with Workload Scheduler data.

Types of Workload Scheduler reports

Job history and statistics

- Job run history
- Job run statistics charts
- Job run statistics table

These reports mimic the BIRT-based reports found in the Dynamic Workload Console in **Reporting > Historical Reports**

Statistical analysis

- Job duration estimation error
- Job duration standard deviation

These reports use data that was processed by the **advanced statistics tool**

Types of Workload Scheduler reports

© Copyright IBM Corporation 2017

Following are the types of historical reports that you can create tasks to run.

Job run history

The *job run history report* is similar to job run history reports you create by using Dynamic Workload Console reporting feature.

Job run statistics

The *job run statistics report* is similar to the job run statistics report you create by using the Dynamic Workload Console reporting feature.

Job duration estimation error

The job duration estimation error report shows the categories (low, medium, and high) of average estimation error. With the report, you see jobs with low accuracy rates or frequent errors. From the report, you can drill down to display the jobs within the accuracy threshold. You can drill down again to see charts that help you identify jobs with high estimation errors.

Job duration standard deviation

The job duration standard deviation report shows variances in job duration categorized into high, medium, and low variation. From the report, you can drill down to display the jobs in each category in chart format. This report helps you find jobs that have varying durations.

Starting the Common Reporting interface

Dynamic Workload Console: **Reporting > Common Reporting**

Name	Modified	Actions
Analysis_Job_Duration_Estimation_Error	December 16, 2015 11:00:06 PM	More...
Analysis_Job_Duration_Standard_Deviation	April 22, 2015 6:24:25 AM	More...
Common Reporting	February 12, 2013 1:24:04 AM	More...
TWA	May 5, 2015 12:51:04 PM	More...

Starting the Common Reporting interface

To start the Common Reporting interface, log in to the Dynamic Workload Console, and select **Reporting > Common Reporting**. You see the list of folders that contain predefined reports.

To run a report, click the folder name to see the list of available reports, and click the report you want to run.

Customizing the report by filtering data

WELCOME x COMMON REPORTING x

Filter Criteria -Analysis job duration standard deviation

Job Properties

Job Name:	%
Workstation Name:	%
Ignore all time duration less than	(s)

Job Stream Properties

Job Stream Name:	%SLA
Workstation Name:	%

Thresholds

Enter thresholds from low to average	* 10
Enter thresholds from average to high	* 60

Job Execution Interval

<input checked="" type="radio"/> Relative Interval <input type="radio"/> Date Range	Last 365 days
--	---------------

Buttons: Cancel < Back Next > Finish

© Copyright IBM Corporation 2017

Customizing the report by filtering data

When you run a predefined report, you can customize the output by filtering the data that appears on the report. To filter the data, enter the criteria on the Filter Criteria page before clicking **Finish**.

Lesson 3 Using the batch reporting utility

Using the batch reporting utility

© Copyright IBM Corporation 2017

Using the command-line batch reporting utility, you can run reports on a scheduled basis or as needed. The batch reporting utility produces reports from the command line (shell) in HTML or CSV format. The reporting utility can also send the result by email.

To run batch reports, you create properties files that are related to the report you want to run. Several templates are provided for different types of reports. You run the `reportcli` command and specify the properties file to use, which output location stores the result, and the name of the report output. You can also override values in the properties file by passing key values on the command line.

Editing the properties file

1. Choose one of the following templates

- **ad.properties**: Audit details report
- **ag.properties**: General audit report
- **jrh.properties**: Job run history report
- **jrs.properties**: Job run statistics report
- **sql.properties**: Custom SQL report
- **sql.properties.ORACLE**: Custom SQL for Oracle RDBMS
- **wwr.properties**: Workstation workload runtime report
- **wws.properties**: Workstation workload summary

2. Make a copy or edit the template

3. Run the **reportcli** command

© Copyright IBM Corporation 2017

Editing the properties file

The templates that are provided with the batch reporting facility are listed in the following table. You can edit the files directly, or copy them to use as customized report templates.

ad.properties	Audit details report
ag.properties	General audit report
jrh.properties	Job run history report
jrs.properties	Job run statistics report
sql.properties	Custom SQL report
sql.properties.ORACLE	Custom SQL for Oracle RDBMS
wwr.properties	Workstation Workload Runtime Report
wws.properties	Workstation Workload Summary

Inside the properties files, you find parameters that you can change to build the report that you want to produce. The properties templates contain comments that document the purpose of each parameter. The categories of properties correspond to the pages that you edit when you create a reporting task in Dynamic Workload Console:

- Enter title and description for your report in the Report Header section. Choose whether to include the selection criteria and table of contents on the report.
- In the Filter Criteria section, specify the options to select what job run information to include in the report. Each report type has different forms of selection criteria. For example, in a custom SQL report task, the filter criteria is an SQL query statement.

- In the Report Output Format and Content section, you can choose whether to display charts or tables, select which data to include in the report, and the output type.

Each property in the file has a key name, starting with PARAM. You can set the values of the keys that do not change often, and override the others by specifying new values on the reportcli command line.



Note: Example job run history properties file

```
#####
# REPORT HEADER SECTION
#####
### Report Title
# A meaningful title for the report.
PARAM_ReportTitle=Job Run History
### Report Description
# Some free text to describe the report behavior or content.
PARAM_ReportDescription=this is a description
### Append report selection criteria to the Report Header
# Set to true if you want to append the report selection criteria
# to the report header.
# Possible values: true and false.
PARAM_AppendSelectionCriteria=false
### Include Table of Contents
# Set to true if you want to include a table of contents after the header.
# Possible values: true and false.
PARAM_Toc=false
### Report type
# This report is JOB_RUN_HISTORY type. Do not change this value.
REPORT_TYPE=JOB_RUN_HISTORY
#####
# FILTER CRITERIA SECTION
#####
# Use this section to specify the filter to include job runs in the report,
# based on: status, delay indicators, and run time.
===== Job Filters =====
### Job Name
# The name of the job. If an alias exists for a job, this alias is the job name
# used in this field.
# Wildcard characters are supported. Blank is equivalent to wildcard (*).
# By default all jobs are selected.
PARAM_FILTER_JobName=*
```

Running the report command line

```
./reportcli.sh -p /home/twsuser/history.properties -o /home/wsuser -r
jrMEhist -k PARAM_FILTER_WorkstationJob=ALPHA -k
PARAM_FILTER_RelativeInterval=1M
com.ibm.net.SocketKeepAliveParameters
com.ibm.net.SocketKeepAliveParameters
AWSBRC105I Report total build time: 21 seconds.
AWSBRC106I Report available on: http://ws94mdm0/reports/jrMEhist.html
```

© Copyright IBM Corporation 2017

Running the report command line

You use the reporting command line, `reportcli`, at the system shell or in a job you define in Workload Scheduler. The batch reporting interface must already be installed and configured for you to use. The setup includes properties for sending reports by email and publishing reports on a web server.

To build a report, run the `reportcli` command, passing arguments to specify the report name, output directory, which properties template to use, and optionally any key parameters you want to override. The syntax of the `reportcli` command is as follows:

```
reportcli.sh -p report_name.properties -o output_directory -r report_name -k
key=value
```

- The `report_name.properties` file is the properties file that you customize.
- The `output_directory` folder is the location of the HTML or CSV file where the results are stored.
- The text `report_name` is the base name of the file that contains the report output.
- The `key` name is one of the PARAM names in the properties file that you choose to override.
- The `value` text is the new value that you give to the PARAM key from the properties file.

The following example shows running the report command line, overriding the template to create a report of all jobs that ran on workstation ALPHA for the past month.

Running the batch reporting command line:

```
./reportcli.sh -p /home/twsuser/history.properties -o /home/twsuser -r jrMEhist  
-k PARAM_FILTER_WorkstationJob=ALPHA -k PARAM_FILTER_RelativeInterval=1M  
com.ibm.net.SocketKeepAliveParameters  
com.ibm.net.SocketKeepAliveParameters  
AWSBRC105I Report total build time: 21 seconds.  
AWSBRC106I Report available on: http://tws92mdm0/reports/jrMEhist.html
```

Lesson 4 Using the preformatted text-based report utilities

Using the preformatted text-based report utilities

© Copyright IBM Corporation 2017

In this lesson, you learn how to use the reporting utilities that produce preformatted text-based output.

Report utility commands provide information by using the data in the Workload Scheduler database and production plans. You run report commands locally on the Workload Scheduler master domain manager.

Running report commands

Preformatted reports	Command	Report
	rep1	Job details listing
	rep2	Prompt listing
	rep3	Calendar listing
	rep4a	Parameter (variables) listing
	rep4b	Resources listing
	rep7	Job history listing (detailed)
	rep8	Job histogram (time lime)
	rep11	Planned production schedule (calendar)
	xref	Cross-reference report
	reptr	<pre>-pre -summary Planned production summary -pre -detail Planned production detail -post -summary Actual production summary -post -detail Actual production detail</pre>

© Copyright IBM Corporation 2017

Running report commands

The commands that you can use are listed in the following table.

Report 01 rep1	Job Details Listing shows information similar to the job run statistics reports in preformatted text output.
Report 02 rep2	Prompt Listing shows the prompt definitions in the database.
Report 03 rep3	Calendar Listing shows the calendar definitions in the database.
Report 04A rep4a	Parameter Listing shows the variables in the default variable table and their values.
Report 04B rep4b	Resource Listing shows the resource definitions in the database.
Report 07 rep7	Job History Listing shows the details of every job run available in the database.
Report 08 rep8	Job Histogram shows a timeline with the list of jobs that ran in that time.
Report 09A reptr -pre -summary	Planned Production Summary shows a list of the job streams that are planned to run in the next production day.
Report 09B reptr -pre -detail	Planned Production Detail shows a detailed list of jobs and job streams that are planned to run in the next production day.

Report 10A	Actual Production Summary shows a list of job streams that were included in the archived production plan and their status.
reptr -post -summary schedlog/Myyyymmddhhmm	
Report 10B	Actual Production Detail shows a detailed list of all the jobs and job streams that were included in the archived production plan, and their status.
reptr -post -detail schedlog/Myyyymmddhhmm	
Report 11	Planned Production Schedule shows a calendar-based printout of job streams that run in the next production cycle.
rep11	
Report 12	Cross- reference Report produces a list of objects with references to other objects. For example, the prompts section shows a list of prompts with the jobs and job streams that depend on each prompt.
xref	

rep1

```

Job: ALPHA      #J1ATL_GROOM_33965          Description: Compute ATL service
applied 33965 matrix on GROOM data
JCL File      : executable
Logon         :                               Creator: twsuser
Recovery Job   :
Recovery Type  : STOP
Recovery Prompt :
Composer Autodoc : Yes
Total Runs    : 38 - 38 Successful,     0 Aborted

Elapsed (mins)      CPU (secs)
Total             38           0
Normal            1
Last Run          1           0 (On 20140822 at 11:49)
Maximum           1           0 (On 20140430)
Minimum           1           0 (On 20140430)

```

Change the output of the report commands by changing the MAESTRO_OUTPUT_STYLE environment variables before running the command.

MAESTRO_OUTPUT_STYLE=LONG sets the report to contain full length (long) fields for object names. If the variable is set to anything other than LONG, long names are truncated to 7 characters and a plus sign, for example: A123456+.

You should use a fixed font size to obtain the correct format of the reports outputs.

Running extract commands

Command	Data extracted
jbxtract	Tab-delimited job details listing and statistics
prxtract	Fixed-length prompt listing
caxtract	Calendar listing
paxtract	Parameter (variables) listing from the default variable table
rextract	Resources listing
r11xtr	Job stream listing with number of jobs and days of the week
xrxtrct	Cross reference data <ul style="list-style-type: none"> xdep_job Jobs and job streams that depend on each job xdep_sched Jobs and job streams that depend on a job stream xfile Jobs and job streams that depend on files xjob Job streams in which jobs appear xprompt Jobs and job streams that depend on prompts xresource Jobs and job streams that depend on resources xsched Job stream information xwhen Information about when job streams will run (run cycle to job stream reference)

© Copyright IBM Corporation 2017

Running extract commands

In addition to report utilities, data extraction programs are provided to generate some of the reports. You can use the programs that are listed in the following table to extract data from the Workload Scheduler database for reporting or importing data into other external programs. Examples of each program and output formats are provided in the *Workload Scheduler User's Guide and Reference*.

jbxtract	Produces a tab-delimited file that contains detailed information about every job in the database, including definition and statistical information.
prxtract	Produces a fixed-length list of prompts in the database and their values.
caxtract	Produces a fixed-length list of calendars in the database with their descriptions. The dates in the calendar are not included.
paxtract	Produces fixed-length output of the contents of the default variable table.
rextract	Produces a fixed-length list of resources in the database and their values.

r11xtr	Produces a fixed-length list of job streams with the number of jobs and day of week.
xrxtrct	Produces 8 fixed-length files that contain cross-reference data from the database: <ul style="list-style-type: none">• Jobs and job streams that depend on each job• Jobs and job streams that depend on a job stream• Jobs and job streams that depend on files• Job streams in which jobs are shown• Jobs and job streams that depend on prompts• Jobs and job streams that depend on resources• Job stream information• Information about when job streams run (run cycle to job stream reference)

Instructor demonstration

Create reports by using the Dynamic Workload Console reporting feature and the Common Reporting tool.



© Copyright IBM Corporation 2017

Instructor demonstration

Exercises

- Create reports
- Create text extracts



© Copyright IBM Corporation 2017

Exercises

Review questions

1. What are some of the ways you can report the data that is maintained by Workload Scheduler?
2. Which type of historical report shows one row of data for each job defined that matches the criteria?
3. Which template for the reporting command line shows the job run history report?

Review answers

1. What are some of the ways you can report the data that is maintained by Workload Scheduler?

Within Workload Scheduler, there are several ways to report on the data available. You can create report tasks, use the Common Reporting interface, run the reporting command line, run utility commands, or query the database directly.

2. Which type of historical report shows one row of data for each job defined that matches the criteria?

In the job statistics report, you see one row for each job defined that matches your criteria. In other words, you see only one row for each job no matter how often it runs.

3. Which template for the reporting command line shows the job run history report?

The jrh.properties file is a template for job run history reports.

Summary

You now should be able to perform the following tasks:

- Create historical reports by using the Dynamic Workload Console
- Create production reports by using the Dynamic Workload Console
- Create historical and statistical reports by using the Common Reporting tool
- Use the batch reporting utility to run reports without using the Dynamic Workload Console
- Use the text-based report utilities to create reports and data extractions

© Copyright IBM Corporation 2015

Unit summary

Appendix A Reference material

In this appendix, you can find detailed descriptions of information you can view when using the Dynamic Workload Console or command-line interfaces.

Monitor workstation task columns

The default columns for *monitor workstations* tasks are listed in the following table.

Table 1 Monitor workstations task columns

Link Status	A workstation can be either LINKED or UNLINKED. When a workstation is UNLINKED, it cannot communicate status information to the master domain manager. You see a yellow lightning bolt when the link to the master domain manager is active. This field is not applicable for Extended Agent workstations.
Workstation	The name of the workstation.
Agent Running	Yes if the process responsible for starting jobs and monitoring the workload processing is running on the workstation. If the workstation is of type remote engine, this flag indicates that the communication between the remote engine workstation and the remote engine is enabled.
Writer Running	Yes if the agent is able to send messages from a fault-tolerant agent to its domain manager. The column is blank when the writer is not active, the agent is not a fault-tolerant agent, or the status is unknown.
Start Time	The date and time when the agent process was started on the workstation.
Run Number	The run number of the current plan that is running on the workstation. Workload scheduler synchronizes workstations in the network by using run numbers. The run number is incremented when the production plan is extended, typically once per day.
Limit	The total number of jobs that can run concurrently on this workstation. The limit can be set to any number 0 - 1,024. A limit of <i>System</i> indicates that no limit is placed on the number of jobs that can run concurrently on this workstation.
Domain	The name of the Workload Scheduler domain to which the workstation belongs.
Type	The type of workstation.
Agent initialized	Yes if the agent has access to the most current production plan.
Fence	The current fence of the workstation.

Table 1 Monitor workstations task columns (continued)

Node Type	The type of agent that is running at the workstation, or workstation type (such as a pool).
Version	The version of Workload Scheduler software that is running on the workstation.
Time zone	The time zone that is configured for the workstation.
Node name	The node name that is configured for the workstation (can be blank for some types).
Node port	The node port number for the workstation configuration (can be blank for some types).
Node information	Information about where the workstation runs, such as the type of operating system.
Behind firewall	Yes if the workstation definition has the BEHINDFIREWALL option set, otherwise No .
SSL communication	Disabled if SSL communication is not used, otherwise blank.
SSL port	The port number that is used for SSL-based communication with the agent.
Host	The name of the workstation that hosts the agent (used for Extended agent and Dynamic agent types).
Method	For extended agent workstations, the access method that is used.
Event processor running	Yes if the event processor is running on the workstation, otherwise No . Normally, only the Master Domain Manager runs the event processor.
Event monitoring	Active if event monitoring is running on the workstation. Does not apply (always shows Not Active) for pools, dynamic pools, or extended agents.
Monitoring configuration updated	Yes if event rule monitoring configurations are deployed to the agent. Does not apply (always shows No) for pools, dynamic pools, or extended agents.

Workstation state column flags

The `conman SHOWCPUS STATE` column shows the status of workstations by displaying up to 9 letters to denote the status or condition of the workstation.

Table 2 Workstation state flags

Column	Flag	Status indication
1	L	The workstation is linked to its domain manager. Dynamic agents or remote engines link to the workload broker server. Workstations with type pool or dynamic pool are registered with the workload broker, and are linked.
1	F	Workload Scheduler is configured in Enhanced Switch Fault Tolerance mode, and the workstation is linked to its domain manager, and all backup domain managers.
2	T	The fault-tolerant agent is directly linked to the domain manager from where you run the command.
2	H	The workstation is linked through its host. This flag is valid for workstations with node type S-AGENT.
2	X	The workstation is linked as an extended agent (type X-AGENT).
2	B	The workstation communicates through the workload broker server.
3	I	If the workstation is a MASTER, MANAGER, FTA, S-AGENT, or X-AGENT, this flag indicates that the workstation completed its initialization. If the workstation is a dynamic agent, pool or dynamic pool, this flag indicates that the agent is correctly initialized. If the workstation is a remote engine type, this flag indicates that the communication between the remote engine workstation and the remote engine is correctly initialized.
4		The fourth column is blank.
5	J	If the workstation is a MASTER, MANAGER, FTA, S-AGENT, or X-AGENT, this flag indicates that the jobman program is running. If the workstation is a dynamic agent, this flag indicates that the JobManager program is running. Because dynamic pool workstations do not include monitoring, for this type of workstation, the J character is always shown. If the workstation is a 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 a remote engine, this flag indicates that the ping command to the remote engine is successful.
6	W	The writer process is active on the workstation. This flag is valid for MASTER, MANAGER, FTA, and S-AGENT workstations.
6	X	For X-AGENT type workstations only. If the workstation that is running <code>conman</code> is the extended agent's host, the state of the extended agent is LXI JX.

Table 2 Workstation state flags (continued)

Column	Flag	Status indication
6	H	For X-AGENT type workstations only. If the workstation that is running <code>conman</code> is not the extended agent's host, the state of the extended agent is LHI JH
7		The seventh column is blank.
8	M	The monman process is running. The monman process is used to trigger event rules on the workstation.
9	D	The workstation has an up-to-date configuration package for event rule monitoring.
10	E	The event processor service is running on this workstation.
10	e	The event processor service is installed on this workstation, but not running.
11	A	The WebSphere Application Server was started on this workstation.
11	R	The WebSphere Application Server is restarting.

When you use the `;INFO` flag to the `conman SHOWCPUS` command, you see different columns in the output, as shown in the following table.

Table 3 Workstation information

CPUID	The name of the workstation.
VERSION	The version of Workload Scheduler that is installed on the computer where the workstation runs.
TIME ZONE	The time zone that is defined for the workstation.
INFO	Information about the workstation. For MASTER, MANAGER, FTA, and AGENT, and S-AGENT workstations, this column shows which operating system is running on the computer where the agent is installed.

When you use the `;LINK` flag to the `conman SHOWCPUS` command, you see the columns that are listed in the following table.

Table 4 Workstation link information

CPUID	The name of the workstation.
HOST	The name of the workstation that acts as the host to a standard agent or extended agent. For domain managers and fault-tolerant agents, it is the same as the CPUID. For standard agent and broker workstations, it is the name of the domain manager. For extended agents, it is the name of the host workstation, as specified in the workstation definition.
ADDR	The TCP port that is defined for the workstation.

Table 4 Workstation link information

NODE	The node name that is defined for the workstation. It is blank for pools and dynamic pools.
FLAGS	Up to five columns, each with a character that displays characteristics of the workstation's properties: <ul style="list-style-type: none">• A: Auto linking is defined and turned on for the workstation.• F: Full status mode is activated for the workstation.• s: The mailman server ID for the workstation. This flag can be one letter or one number.• T: The link is defined as TCP/IP.

When you use the ;GETMON flag to the SHOWCPUS command, you see a list of event rules that are defined for the monitor that runs on each of the specified workstations. The output contains the rule name, event provider, and the time at which the rule was last generated.

Monitor job streams task columns

The columns that you see in a default monitor job streams task are listed in the following table.

Table 5 Monitor job streams task columns

Status	The status of the job stream.
Internal Status	The internal status of the job stream. Status and internal status are discussed in “Job stream status” on page 3-36.
Job Stream	The job stream name. The name is a link to the job stream properties.
Workstation	The name of the workstation on which the job is started. The name is a link to the secondary query of workstations that are related to the job stream you selected.
Scheduled Time	<p>The time when the job stream is positioned in the plan. Scheduled time is not a time dependency. It represents the time when the job stream is positioned in the plan. The scheduled time does not prevent the job stream from processing. It uniquely identifies a job stream instance in the plan, when multiple instances of the job stream exist for a single day. For example, job streams might run twice in a 24-hour period.)</p> <p>Time restrictions can also be combined. For example, when both an earliest start time and a latest start time for a job or job stream are defined, a time frame within which processing must start is created.</p>
Not Satisfied Dependencies	The number of dependencies for this job stream that are not satisfied. The number links to a secondary query that shows all of the dependencies for the selected job stream.
Total Jobs	The number of jobs in the job stream. The number is a link to a secondary query that shows the jobs in the selected job stream.
Successful Jobs	The number of jobs in the job stream that ended successfully. The number is a link to a secondary query that shows the jobs in the selected job stream.
Jobs Limit	The number of jobs in the job stream that can run concurrently. Possible values are 0 - 1,024. If you specify a job limit of 0, no jobs within the job stream are started. The default is no limit.
Priority	The priority that is assigned to the job stream. The priority level determines which job stream starts first after all dependencies are solved. The default value is 10. The number is a link to a secondary page, where you can enter a new priority value.
Earliest Start	The earliest date and time the job or job stream can start (also known as the AT time).
Actual Start	The time the job or job stream started, if it already started.
Deadline	The time by which the job or job stream must complete, if configured in the job stream definition.

Table 5 Monitor job streams task columns

Latest Start Time	The latest time the job stream can start (also known as the UNTIL time).
Estimated Duration	The expected duration time of the job stream.
Actual Duration	The actual duration time of the job stream, which is shown after the job stream completes.
Original Job Stream	The original name of the job stream, for job streams that are submitted with an alias name.
Internal Identifier	The unique alphanumeric job stream identifier that Workload Scheduler automatically generates and assigns to the job stream.
Information	Additional information about the job stream.
Monitored	Selected if IBM Business Systems Manager monitors the job stream, and every job that it contains.
Job Stream Dependencies	The number of predecessor job streams for the job stream.
Job Dependencies	The number of predecessor jobs for the job stream.
Prompt Dependencies	The number of prompt dependencies for the job stream.
Latest Start Time	The latest time the job stream can start (also known as the UNTIL time).
File Dependencies	The number of file dependencies for the job stream.
Resource Dependencies	The number of resource dependencies for the job stream.

Console manager job selection keywords

These are the keywords that you can use to select jobs for viewing by using the `conman showjobs` command.

Table 6 conman job selection keywords

at = <i>lowtime</i>, <i>hightime</i>	Show jobs that have an earliest start time dependency between <i>lowtime</i> and <i>hightime</i> , inclusively.
critical	Show jobs that were flagged as critical in the job stream definition.
critnet	Show jobs that are critical or jobs that are predecessors of critical jobs.
deadline = <i>lowtime</i>, <i>hightime</i>	Show jobs that must complete between <i>lowtime</i> and <i>hightime</i> , inclusively, if they have the deadline dependency set in the job stream.
every = <i>lowrate</i>, <i>highrate</i>	Show jobs that are defined with a repetition rate between <i>lowrate</i> and <i>highrate</i> , inclusively.
finished = <i>lowtime</i>, <i>hightime</i>	Show jobs that finished between <i>lowtime</i> and <i>hightime</i> , inclusively.
follows = <i>jobstream</i> or follows = <i>jobstream.job</i>	Show jobs that have a dependency on the job stream or job.
logon = <i>username</i>	Show jobs that run on the computer as <i>username</i> .
opens = <i>filename</i>	Show jobs that are waiting on <i>filename</i> as a dependency.
started = <i>lowtime</i>, <i>hightime</i>	Show jobs that ran between <i>lowtime</i> and <i>hightime</i> , inclusively.
state = <i>job state</i>	Show jobs with the current state that matches the selection. States of jobs are described in “Job status” on page 3-47.
until = <i>lowtime</i>, <i>hightime</i>	Show jobs with a latest start time between <i>lowtime</i> and <i>hightime</i> , inclusively.

File dependency qualifiers

Use the string-matching capability of the UNIX `test` command to customize the file (OPENS) dependency. Use this customization to test for conditions beyond simple file existence. In the following expression examples, the `%p` is replaced with the contents of the **Filename** field, which is typically a fully qualified file name:

The file conditions that you can use are listed in the following table.

<code>-r %p</code>	The file exists and is readable.
<code>-w %p</code>	The file exists and is writable.
<code>-x %p</code>	The file exists and is an executable file.
<code>-f %p</code>	The file exists and is a regular file.
<code>-d %p</code>	The file exists and is a directory.
<code>-c %p</code>	The file exists and is a character special file.
<code>-b %p</code>	The file exists and is a block special file.
<code>-p %p</code>	The file exists and is a named pipe (FIFO).
<code>-u %p</code>	The file exists and its set-user-ID bit is set.
<code>-g %p</code>	The file exists and its set-group-ID bit is set.
<code>-k %p</code>	The file exists and its sticky bit is set.
<code>-s %p</code>	The file exists and has a size greater than 0.
<code>-h %p</code>	The file exists and is a symbolic link.

The string comparison conditions that you can use are listed in the following table.

<code>-z string</code>	The length of the specified string is 0.
<code>-n string</code>	The length of the specified string is not 0.
<code>s1 = s2</code>	The strings s1 and s2 are identical.
<code>s1 != s2</code>	The strings s1 and s2 are not identical.
<code>s1</code>	The string s1 is not empty (null string).

The integer number comparisons are listed in the following table.

<code>n1 -eq n2</code>	n1 is equal to n2.
<code>n1 -ge n2</code>	n1 is greater than or equal to n2.
<code>n1 -gt n2</code>	n1 is greater than n2.
<code>n1 -le n2</code>	n1 is less than or equal to n2.

<i>n1 -lt n2</i>	<i>n1</i> is less than <i>n2</i> .
<i>n1 -ne n2</i>	<i>n1</i> is not equal to <i>n2</i> .

You can combine comparison forms by using terms that are listed in the following table.

<i>(conditions)</i>	Parentheses are used for grouping.
!	Condition Unary negation operator. True if condition is false.
<i>cond1 -a cond2</i>	Binary AND operator. True if both conditions are true.
<i>cond1 -o cond2</i>	Binary OR operator. True if either condition is true.

When checking for many files, use the *-a* option. See some examples in the following table.

Table 7 File dependency qualifier examples

To check that...	Use the file name and qualifier...
File c:\users\fred\datafiles\file88 is available for read access.	File name: c:\users\fred\datafiles\file88 Qualifier: -r %p
Three directories, /users/john, /users/mary, and /users/roger, exist.	File name: /users Qualifier: -d %p/john -a -d %p/mary -a -d %p/roger
Spool has created its x pipe.	File name: /var/lib/spool/x Qualifier: -p %p
The file /tmp/text.txt does not exist.	File name: /temp/text.txt Qualifier: ! -f %p
The file D:\work\john\execit1 exists and is not empty.	File name: D:\work\john\execit1 Qualifier: notempty
Check that file C:\tech\checker\startf exists with a size greater than 0 and is writable.	File name: C:\tech\checker\startf Qualifier: -s %p -a -w %p



ibm.com/training

Authorized
IBM | Training