IBM

IBM® Workload Scheduler

Dynamic Workload Console User's Guide

IBM® Workload Scheduler

Version 10.2.5

Note

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

This edition applies to version 10, release 2, modification level 5 of IBM® Workload Scheduler (program number 5698-T09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Note....ii
.... iii
List of Figures.... vii
About this publication....viii
What is new in this release....viii
Accessibility viii
Technical training.... viii
Support information.... viii
Chapter 1. Getting Started.... 9
Navigating the Dynamic Workload Console....10
Convention for specific platform information....11
Naming conventions for scheduling objects.... 11
Creating and managing engine connections.... 13
Event management configuration....13
Installing and configuring the Dynamic Workload Console....15
Chapter 2. Running IBM Workload Scheduler from a mobile device.... 17
Chapter 3. Managing users and repositories.... 19
Managing user settings....19
Moving configurations and settings definitions from one database to a different database.... 19
Changing DB2 User....21
Chapter 4. Configuring High Availability....22
Chapter 5. Customizing your console....23
Personalizing interfaces with custom images....23
Customizing roles.... 24
Customizing your portfolio....24
Customizing your global settings.... 25
Customize video URLs.... 28
Managing session timeout.... 28
Override graphical view limits.... 29
Plan View in new window.... 29
Plan View auto refresh interval.... 30
Disable and customize NewsFeed function..... 30
Disable and customize the creation of predefined tasks....32
Add customized URL to job and job streams..... 33
User registry....35
z/OS http connections.... 36
Limit the number of objects retrieved by queries....36
Limit task and engine sharing.... 38
Entries in Workload Designer search windows.... 38
Show all dependencies.... 39
Auditing mobile app activity.... 39
Modifying the number of archived plans displayed in the Dynamic Workload Console.... 40
Show or hide predecessors from What-if Analysis Gantt view....41
TdwcGlobalSettings.xml sample....41
Integrating AIDA in the Dynamic Workload Console....45
Integrating AIDA after an environment update.... 47
Chapter 6. IBM Workload Scheduler Concepts.... 50
Scheduling environment....50
Workstation....50
Domain.... 54
Scheduling objects....57
Job....58
Job stream....59
Folder....59
Workload application.... 59
Period.... 60
Calendar.... 62
Run cycle....62
Run cycle group....64
Operator instructions.... 69
Parameter....69
Dependencies.... 69
User.... 83
Workstation class....84
Variable table....84
Production process....85
Database.... 86
Plans....86
Preproduction plan....89
Engine connections.... 90
Event management.... 91
Reports....93
Workload service assurance.... 96
Processing and monitoring critical jobs..... 98
Planning critical jobs.... 101
IBM Workload Scheduler for SAP....102
Chapter 7. Creating and editing items.... 104
Designing your scheduling environment.... 104
Graphical Designer overview 106
Designing your workload....110
Managing calendar definitions....112
Managing credentials definitions....112
Managing folder definitions....113
Managing job definitions.... 114

Managing job stream definitions..... 117
Managing variable table definitions.....119
Managing workstation definitions.....120
Creating an event rule.....121
Editing event rules.....122
Chapter 8. Managing Workload Security..... 125
Managing access control list.....126
Managing security domains.....127
Managing security roles 129
Authenticating the command line client using API Keys.....130
Service API Key with no group associated.....132
Managing folders.....132
Actions on security objects..... 134
Attributes for object types..... 139
Specifying object attribute values..... 140
Chapter 9. Changing user password in the plan..... 145
Chapter 10. Monitoring your environment..... 147
Display a graphical plan view..... 147
Graphical views in the plan..... 149
Graphical View - modelling.....156
Analyzing the impact of changes on your environment.....159
Workload Dashboard..... 161
Widgets and Datasources..... 163
Creating a customized dashboard for monitoring.....164
Exporting and importing a dashboard..... 166
Monitoring your Scheduling Environment..... 167
Creating a task to Monitor Workstations.....168
Creating a task to Monitor Domains.....169
Monitoring your workload..... 170
Orchestration Monitor overview.....171
Monitoring scenario..... 172
Configuring the Federator to mirror data on a database..... 173
Mirroring the z/OS current plan to enable the Orchestration Monitor.....175
Monitoring your items in the plan..... 178
Monitoring event rules.....180
Controlling job and job stream processing..... 185
Chapter 11. Working with Plans..... 191
Selecting the working plan.....191
Generating Trial and Forecast Plans..... 192
Display a graphical preproduction plan.....193
Chapter 12. Submitting Workload on Request in Production..... 195
Submitting ad hoc jobs..... 195
Submitting predefined jobs..... 196
Submitting predefined job streams..... 196
Setting properties for ad hoc jobs and predefined jobs and job streams.....197
Chapter 13. Keeping track of changes..... 199
Auditing justification and reporting..... 200
Checking version information.....201
Auditing justification and reporting- a business scenario..... 202
Streamline release management - a business scenario..... 202
Version control - a business scenario..... 203
Chapter 14. Reporting.....205
Predefined Reports..... 205
Creating a task to generate a Job Run Statistics report.....205
Creating a task to generate a Job Run History report.....206
Creating a task to generate a Workstation Workload Summary report..... 207
Creating a task to generate a Workstation Workload Runtimes report.....208
Creating a task to Create Plan Reports..... 208
Creating a task to Create Custom SQL Reports.....210
Personalized Reports..... 210
Manage personalized reports with BIRT.....211
Chapter 15. Scenarios..... 212
Using workload service assurance to monitor z/OS critical jobs.....212
Monitoring jobs running on multiple engines..... 214
Chapter 16. Troubleshooting the Dynamic Workload Console.....218
Chapter 17. Reference..... 219
Users and groups..... 219
Type of communication based on SSL communication options.....221
Status description and mapping for distributed jobs..... 222
Status description and mapping for z/OS jobs.....225
Status description and mapping for distributed job streams.....227
Status description and mapping for z/OS job streams.....229
Workstation type.....231
Regular expressions and SQL reports.....236
Regular Expressions.....237
SQL report examples..... 242
Event rule.....245
Action properties.....245
Event properties..... 246
Chapter 18. Glossary..... 247
Notices..... cclxi

Index..... 265

List of Figures

Figure 1: Single-domain network.....55

Figure 2: Multiple-domain network.....56

Figure 3: Example of a condition dependency definition.... 79

Figure 4: Example of a condition dependency at run time.....80

Figure 5: Auto-recovery job stream with step level dependency.....80

Figure 6: Example of recovery job with condition dependencies.....81

Figure 7: Cross dependencies.....83

Figure 8: Critical path.....99

Figure 9: Mirroring data onto the database.....176

Figure 10: Monitoring data from the Orchestration Monitor.....177

Figure 11: Data flow to perform actions on the monitored objects.....178

About this publication

IBM Workload Scheduler simplifies systems management across distributed environments by integrating systems management functions. IBM Workload Scheduler plans, automates, and controls the processing of your enterprise's entire production workload. The *IBM Dynamic Workload Console User's Guide* provides detailed information about how to configure and use the Dynamic Workload Console to manage your IBM Workload Scheduler environment.

What is new in this release

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

For information about the APARs that this release addresses, the Dynamic Workload Console Release Notes at Dynamic Workload Console Release Notes.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For detailed information, see the appendix about accessibility in the *IBM Workload Scheduler User's Guide and Reference*.

Technical training

Cloud & Smarter Infrastructure provides technical training.

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

Support information

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

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

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

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

Chapter 1. Getting Started

Information about the Dynamic Workload Console installation and configuration.

For more information about this installation, see the *IBM Workload Scheduler: Planning and Installation*.

To configure the Dynamic Workload Console, see the section about configuring the Dynamic Workload Console in the *IBM Workload Scheduler: Administration Guide* to find information about:

- Launching in context with the Dynamic Workload Console
- Configuring access to the Dynamic Workload Console
- Configuring Dynamic Workload Console to use Single Sign-On
- Configuring the use of Lightweight Third-Party Authentication
- Configuring Dynamic Workload Console to use SSL
- Customizing your global settings
- Configuring Dynamic Workload Console to view reports

You can access the Dynamic Workload Console from any computer in your environment using a web browser through either the secure HTTPS or HTTP protocol.

Note: If you are changing the settings related to the language in Mozilla Firefox, to ensure that the Dynamic Workload Console output is displayed correctly:

1. Clean the cache.
2. Close and open the browser again.

The first and main actions that you perform when you connect to the Dynamic Workload Console are:

Creating a connection to a IBM Workload Scheduler engine

You specify the details (such as IP address, user name, and password) to access a IBM Workload Scheduler engine, and, optionally, a database to operate with objects defined in plans or stored in the database.

Important: To ensure compatibility, the Dynamic Workload Console version installed must always be equal to or greater than the version of any engine it connects to.

From the Dynamic Workload Console, you can access the current plan, a trial plan, a forecast plan, or an archived plan for the distributed environment or the current plan for the z/OS® environment.

You might want to access the database to perform actions against objects stored in it or to generate reports showing historical or statistical data.

In addition, working both on the database and on plans, you can create and run *event rules* to define and trigger actions that you want to run in response to events occurring on IBM Workload Scheduler nodes.

Defining a scheduling environment

You define your IBM Workload Scheduler network. You create workstation definitions in the database to represent the physical machines or computer systems on which your workload is scheduled to run. A IBM Workload Scheduler network is made up of the workstations where job and job stream processing occurs. When you design your network, you assign roles to these workstations to suit your specific business requirements. You can design your network with multiple domains, to divide control of a large network into smaller manageable groups. A typical IBM Workload Scheduler network consists of a workstation acting as the master domain manager and at least one domain.

Defining scheduling objects in the database

You define your workload, which consists of jobs that are concatenated in job streams. Then, you specify the calendars and run cycles according to which job streams must run. You can also define dependencies to condition the workload processing. All these definitions can be done within the Workload Designer.

Creating tasks to manage IBM Workload Scheduler objects in the plan

You specify some filtering criteria to query a list of scheduling objects whose attributes satisfy the criteria you specified. Starting from this list, you can navigate and modify the content of the plan, switch between objects, open more lists, and access other plans or other IBM Workload Scheduler environments.

Related information

Creating and managing engine connections on page 13

Designing your workload on page 110

Monitoring your Workload on page 170

Navigating the Dynamic Workload Console

An overview to the Dynamic Workload Console.

For an interactive overview of the product and its features, you can view several demo scenarios, available (in English only) on the Workload Automation YouTube channel.

After you have installed and configured the Dynamic Workload Console, you can start defining and scheduling your workload. Log in by connecting to:

```
https://<your_ip_address>:<https_port>/console/login.jsp
```

You can access the Dynamic Workload Console from any computer in your environment using a web browser through the secure HTTPS protocol.

To have a quick and rapid overview of the portal and of its use, after logging in, the Welcome page for the Dynamic Workload Console is displayed in the console window. This window has a navigation menu across the top, organized in categories.

Each category drops down to display a number of options that when clicked, display a page in the work area on the left. Each page displays with a title in its tabbed window in the work area.

Several products can be integrated in this portal and their related entries are listed together with those belonging to the Dynamic Workload Console in the navigation bar displayed at the top of the page.

The navigation bar at the top of the page is your entry point to the Dynamic Workload Console.

Related information

Scenarios on page 212

Convention for specific platform information

Icons to identify the information related only to specific platforms.

This publication uses the following icons to identify the information related only to specific platforms:

Distributed

The information applies only to IBM® Workload Scheduler running in a distributed environment.

z/OS

The information applies only to IBM® Workload Scheduler running in a z/OS environment.

All information that is not marked by an icon applies to all the supported environments.

Naming conventions for scheduling objects

The Dynamic Workload Console allows you to manage and control IBM® Workload Scheduler production for z/OS and distributed environments.

There are some differences in the processing and behavior between the IBM® Workload Scheduler products for z/OS and distributed environments. When there are differences, the descriptions and related actions of scheduling objects are explained for both environments.

Table 1: Naming convention for scheduling objects on page 11 lists the objects and object names typical of the IBM® Workload Scheduler environment where they are defined.

Table 1. Naming convention for scheduling objects

Object description	Object name in a distributed environment	Object name in a z/OS environment
An ordered list of activities in plan for the current production period. The production plan contains information about the processes to run,	Production Plan	Current Plan

Table 1. Naming convention for scheduling objects (continued)

Object description	Object name in a distributed environment	Object name in a z/OS environment
on which workstation, and what dependencies must be satisfied before each process is launched. The production plan is automatically created and managed by the product and requires no user intervention. The production plan is generated daily at 05:00 CDT time.		
A unit of work that is part of an application or a job stream and that is processed at a workstation.	Job	Operation. An operation can contain a list of steps to run.
A list of jobs that run as a unit to accomplish a task (such as calculating payroll), together with times, priorities, and other dependencies that determine the order in which the jobs run.	Job stream	Application
A run of a job stream or an application scheduled in the plan.	Instance	Occurrence
A type of application description related to run cycle, calendar information, or job descriptions common to all applications defined as members of the group.	N/A	Application Group
A physical or logical asset where job processing occurs.	Workstation. It is qualified according to its position in the topology of the scheduling network and on its ability to interact with the information contained in the current plan.	Workstation. It is qualified according to the type of job processing it does in computer workstation, general workstation, print workstation.
IBM® Workload Scheduler database	A customized set of tables in a relational database containing definitions for all scheduling objects, network topology, variables, and job processing statistics.	A collection of six sets of data, acting as a flat database, that contain information about calendars, periods, workstation descriptions, JCL variable

Table 1. Naming convention for scheduling objects (continued)

Object description	Object name in a distributed environment	Object name in a z/OS environment
		tables, application descriptions, and operator instructions.

Related information

Workstation on page 50

Job on page 58

Job stream on page 59

Production process on page 85

Creating and managing engine connections

How you can create, modify, or delete engine connections.

About this task

To create, modify, or delete an engine connection, perform the following steps.

Note: You can modify or delete only engine connections that you have created.

Note: Only Administration roles can manage engine connections.

1. From the navigation toolbar, click **Administration** > **Manage Engines**.
2. From the displayed panel you can create, edit, delete, or share an engine connection, and test the connection to the remote server where IBM Workload Scheduler is installed. You can order the list of engine connections displayed in this panel by using sorting criteria that you select with the buttons at the top left corner of the table.

Related information

Scheduling objects on page 57

Engine connections on page 90

Event management configuration

Authorizations needed to use event management.

You can use the event management feature both from the IBM Workload Scheduler command line interface and from the Dynamic Workload Console.

You need the following authorizations to perform event management operations from the Dynamic Workload Console:

On Dynamic Workload Console

The user ID you use to log in to Dynamic Workload Console must be defined as user in the **Manage Roles** section and must be defined within one of the following groups:

Table 2. Event Management Authorizations

Groups	Event management operations you can perform
Operator	List and manage Event Rule Instances, Log Messages, and Triggered Actions.
Developer	Create, list, and manage Event Rules.

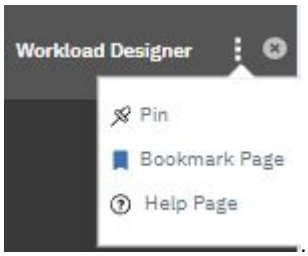
Note: Dynamic Workload Console users belonging to the **Administrator** group can perform all operations available in the web-based user interface.

How to get help

A new contextual help has been created to better guide you through the Dynamic Workload Console interface and its fields.

Note: If you are using Microsoft Edge, the help tool is available only on version 44.19041 or later (Chromium-based versions).

To open and start using the help tool, from the Overflow menu click Help Page



.

The help pane opens on the left side of the interface. To expand it, drag the right margin or select the detach icon to view the help pane in a separate window.

You can also change the font size of the topics by selecting the font size icon .

The home page shows the main topics related to the page you are navigating and three useful links that redirect you to the What's new page, to our YouTube channel, or our Community page.

If you cannot find the information you are looking for in the main topics, you can enter a keyword in the search bar and find the information you need.

After you have performed a search, go back to previous searches by clicking the back icon

or go back to the home page by clicking the home icon .

Leaving the help opened and moving from a field to another to configure them, the help automatically updates itself and shows you the information about the selected field.

On IBM Workload Scheduler

The IBM Workload Scheduler user credentials defined in the engine connection must belong to a IBM Workload Scheduler user authorized to perform event management operations in the IBM Workload Scheduler security file.

You need the *create* permission set for the *rule* object. You also need the *use* permission on the objects (job, job stream, and so on) that you want to use as events.

For more information about how to define and manage user authorizations in the security file, see *IBM Workload Scheduler: Administration Guide*.

Related information

Event management on page 91

Installing and configuring the Dynamic Workload Console

For more information about this installation, see the section about the typical installation scenario in *Planning and Installation Guide* or Planning and Installation

To configure the Dynamic Workload Console, see the *Administration Guide* to find information about:

- Launching in context with the Dynamic Workload Console
- Configuring access to the Dynamic Workload Console
- Configuring Dynamic Workload Console to use Single Sign-On
- Configuring Dynamic Workload Console to use SSL
- Customizing your global settings
- Configuring High Availability for Dynamic Workload Console
- Configuring Dynamic Workload Console to view reports

For more information about z/OS connector instances configuration by using WebSphere Application Server tools, see *IBM Z Workload Scheduler: Planning and Installation> IBM Z Workload Scheduler connector > Installing, Upgrading and Uninstalling IBM Z Workload Scheduler connector on distributed systems*.

Chapter 2. Running IBM Workload Scheduler from a mobile device

Use your mobile device to easily and quickly interact with your IBM Workload Scheduler environment.

The IT market is moving towards mobile devices, which help you perform a large number of tasks, such as manage your sales workforce, read your email, check your accounting system, or attend a web conference. Applications designed for mobile devices must be intuitive and user-friendly while remaining robust and reliable, and providing instant access to business and client data wherever they are.

You can interact with IBM Workload Scheduler by using the Self-Service Catalog and Self-Service Dashboards applications.

Note: To use an engine connection from a mobile device, you must enable the checkbox in the Engine Connection Properties page, and configure the Dynamic Workload Console to use the Single Sign-On. For more information, see the section about configuring the Dynamic Workload Console to use Single Sign-On in the *Administration Guide*.

Self-Service Catalog

The scheduler or application designer creates job streams in the Dynamic Workload Console and marks them as services, so that they are available for managing from the Self-Service Catalog interface. Services correspond to IBM Workload Scheduler job streams, which you can submit from your mobile, even if you do not have any experience with IBM Workload Scheduler.

Launch the Self-Service Catalog from your mobile device by connecting to the following URL:

```
https://host_name:port_number/console/ssc
```

where *host_name* and *port_number* are the host name and port number of the Dynamic Workload Console you are connecting to.

Mobile User is the minimum role required to access Self-Service Catalog. Users with this role can view services to which they are authorized and submit service requests. Associate at least one entity to this role to allow other roles access to the Self-Service Catalog.

Self-Service Dashboards

By defining filter criteria to be applied to your jobs and workstations, you can view dashboards and drill down to more detailed information about the jobs and workstations that match the criteria. You can also perform recovery actions on the jobs and workstations.

Launch the Self-Service Dashboards app from your mobile device by connecting to the following URL:

```
https://host_name:port_number/dwc/addOns/devices/ssmanagement/ssmanagement.jsp
```

where *host_name* and *port_number* are the host name and port number of the Dynamic Workload Console you are connecting to.

Mobile User is the minimum role required to access Self-Service Catalog. Users with this role can view services to which they are authorized and submit service requests. Associate at least one entity to this role to allow other roles access to the Self-Service Catalog.

You can open the applications also from the Single Entry Point page. For more information, see IBM Workload Scheduler user interfaces.

To open this home page on your mobile device, access the following URL:

```
https://host_name:port_number/dwc/mobile.jsp
```

where *host_name* and *port_number* are the host name and port number of the Dynamic Workload Console you are connecting to.

You can also open the applications from the welcome page of the Dynamic Workload Console.

For more information, see the section about product user interfaces in *IBM Workload Automation: Overview*.

Note: If you are using a Dynamic Workload Console at the latest fix pack level connected to a back-level master domain manager, you must access the previous version of the Self-Service Catalog.

Related information

Workload Dashboard on page 161

Chapter 3. Managing users and repositories

How to configure, change, and share your settings repository and change the DB2 user. How to manage the user settings.

The user settings such as user preferences, saved tasks, and engine connections, which by default are stored in a settings repository that is a local XML file, must be exported and stored in a settings repository on a DB2 database. Using a database as your repository, all your existing user settings relating to current Dynamic Workload Console are saved in the database, and all the operations involving user settings are run using the settings in this repository.

Managing user settings

How to export the user settings and import them into a new Dynamic Workload Console

Before you begin

To perform this task you need to have Administrator role.

About this task

User settings such as user preferences, saved tasks, and engine connections are stored in the settings repository, which by default is a local file. However, you can decide to have your settings repository on a database for all Dynamic Workload Console operations that involve user settings.

You can export the content of your settings repository as an XML file, optionally modify it, and then import it into the same or another instance of Dynamic Workload Console.

This is particularly useful for migration purposes or if you want to modify the same settings in multiple Dynamic Workload Console instances.

To export the settings and import them into a new Dynamic Workload Console, perform the following procedure.

Note: Import and export operations are performed from and to the currently-selected repository.

Note: The Import/Export operation does not apply to the Custom Board.

1. From the navigation toolbar, click **Administration** > **Manage Settings**.
2. In the Manage Settings panel, click **Export Settings** and save the XML file to a directory of your choice.
3. Optionally, edit the file using an XML editor and save it.
4. Log in to the Dynamic Workload Console where you want to import the settings and open the Manage Settings panel.
5. Click **Import Settings** and browse to the XML file containing the settings you want to import.
6. Log out from the Dynamic Workload Console and log in again.

Moving configurations and settings definitions from one database to a different database

About this task

You can use the procedure described below to to move Dynamic Workload Console configurations and settings definitions from one database to a different database.

1. From the navigation toolbar, click **Administration** > **Manage Settings**.
2. In the Manage Settings panel, click **Export Settings** and save the XML file to a directory of your choice.
3. Optionally, edit the file using an XML editor and save it.
4. Optionally export your custom boards, as described in Exporting and importing a dashboard on page 166.
5. Stop the Dynamic Workload Console as described in Application server - starting and stopping.
6. Browse to the `datasource_<db_vendor>.xml` file located in one of the following paths:

   On UNIX operating systems

   DWC_home`/usr/servers/dwcServer/configDropins/templates`

   On Windows operating systems

   DWC_home`\usr\servers\dwcServer\configDropins\templates`

7. Copy the `datasource_<db_vendor>.xml` to the path for your operating system:

   On UNIX operating systems

   DWC_DATA_dir`/usr/servers/dwcServer/configDropins/overrides`

   On Windows operating systems

   DWC_home`\usr\servers\dwcServer\configDropins\overrides`

8. Configure the `datasource_<db_vendor>.xml` file based on the specifics of your environment.
9. Only if you are moving from an Oracle database to a different database, browse to the following files:
   - *DWC_DATA_dir*`>/usr/servers/dwcServer/apps/DWC.ear/DWCRest.war/META-INF/orm.xml`
   - *DWC_DATA_dir*`/usr/servers/dwcServer/apps/DWC.ear/Reporting.war/META-INF/orm.xml`
   - *DWC_DATA_dir*`/usr/servers/dwcServer/apps/DWC.ear/TWSWebUI.war/META-INF/orm.xml`

   and replace the contents of the **<schema>** tag with **<schema>TDWC</schema>**.
10. Only if you are moving to an Oracle database, browse to the following files:
    - *DWC_DATA_dir*`>/usr/servers/dwcServer/apps/DWC.ear/DWCRest.war/META-INF/orm.xml`
    - *DWC_DATA_dir*`/usr/servers/dwcServer/apps/DWC.ear/Reporting.war/META-INF/orm.xml`
    - *DWC_DATA_dir*`/usr/servers/dwcServer/apps/DWC.ear/TWSWebUI.war/META-INF/orm.xml`

    and replace the contents of the **<schema>** tag with the name of the database user.
11. Restart the Dynamic Workload Console as described in Application server - starting and stopping.
12. Click **Import Settings** and browse to the XML file containing the settings you want to import.
13. Import custom boards, as described in Exporting and importing a dashboard on page 166.

Results

You have now exported your configurations and settings definitions from one database to a different database.

Changing the Dynamic Workload Console user of DB repository

How to change the Dynamic Workload Console user that updates the settings repository on DB2.

Before you begin

To perform this task you need to have the Administrator role.

You must have switched the Dynamic Workload Console settings repository from a local file to a database repository, as described in Moving configurations and settings definitions from one database to a different database on page 19.

About this task

Only users with database administrator rights are authorized to initialize the Dynamic Workload Console related tables on the database.

If you want the Dynamic Workload Console to access the database repository with a user without database administrator privileges you must follow these steps:

1. Create a new DB2 user and grant this user with SELECT, INSERT, UPDATE, DELETE rights on all the following tables, belonging to `TDWC` schema:

   ```
   TDWC_EngineConnection
   TDWC_QueryTask
   TDWC_ReportTask
   TDWC_MEQueryTask
   TDWC_Credential
   TDWC_ConfigurationProperty
   TDWC_Preferenceable
   ```

   The above are the default permissions. However, if you need to restrict your policy, you can give the following permissions to the new DB2user:

   ```
   revoke connect,bindadd, createtab, implicit_schema on database from public;
   revoke use of tablespace USERSPACE1     from public;

   grant use of tablespace userspace1 to  user twsdb2;
   grant createtab on database to user  twsdb2;
   grant implicit_schema on database to user twsdb2;
   ```

2. Change Dynamic Workload Console user accessing DB2
   a. From the navigation toolbar, click **Administration** > **Manage Settings**.
   b. In the **Database Settings** section, specify the credentials of the newly created user that must to connect to the database.

   Note: As a result of this user switch, theDynamic Workload Console without database administrator privileges will no longer be authorized to initialize the database in the Manage Settings panel.

Chapter 4. Configuring High Availability

How to configure, change, and share your settings repository.

Performance can be highly improved by configuring multiple Dynamic Workload Console instances in a High Availability configuration, so as to have multiple console instances working at the same time and with the same repository settings.

The Dynamic Workload Console is set to be always in High Availability and a front-end Network Dispatcher must be set up to handle and distribute all incoming session requests.

If you use a Dynamic Workload Console in High Availability configuration, when you connect to a Dynamic Workload Console you are not actually connecting to a specific console but to a load balancer that dispatches and redirects the connections among the nodes in the configuration. Therefore, for example, if a node fails, new user sessions are directed to other active nodes in the configuration.

To implement this kind of configuration, the Administrator must ensure that the `datasource.xml`, located in the following path `opt/wa/DWC/DWC_DATA/usr/servers/dwcServer/configDropins/overrides`, has the same configuration on every Dynamic Workload Console in the cluster.

Chapter 5. Customizing your console

You can customize your console to have just the tasks that you want to access.

When you log in to the Dynamic Workload Console, a welcome page containing quick start information is displayed with links to additional information. A horizontal navigation bar across the top contains categories with different tasks.

Personalizing interfaces with custom images

You can customize the Dynamic Workload Console and the Self-Service Catalog login pages with your company images.

About this task

By default, when you log into the Dynamic Workload Console or the Self-Service Catalog, IBM® Workload Scheduler logo and icon images are displayed. Administrators can personalize the Dynamic Workload Console by uploading logo and icon images for the login page and the header. When you upload a logo, the same image is also displayed on the Self-Service Catalog login page. The icon image is also used as favicon.

You can upload a logo and an icon directly from the Dynamic Workload Console; you can set both a logo and an icon image, or you can set just one of them. If you do not set an image, or if you restore image settings, the default IBM® Workload Scheduler images are used.

In the table below you can find the maximum file size for each type of image, and the recommended dimensions that guarantee a better image quality.

Image properties	Logo	Icon
Maximum file size	50 KB	25 KB
Recommended image size	1600x960 px	800x480 px

The supported file extensions are PNG, JPG, JPEG, and SVG.

If you want to restore image settings, select the **remove option (x)** next to the uploaded image.

To upload your custom icon and logo, follow the procedure described below:

1. From the Dynamic Workload Console toolbar, click **Administration** and then **Private Label Settings**.
2. In **Upload icon**, click **Upload** to select the image that you want to place on the Dynamic Workload Console header.
3. Optionally, you can adjust the picture by using the crop and zoom tools.
4. Click **Save** and close the panel.
5. In **Upload logo**, click **Upload** to select the image that you want to place on the login pages of the Dynamic Workload Console and of the Self-Service Catalog.
6. Optionally, you can adjust the picture by using the crop and zoom tools.
7. Click **Save** and close the panel.

Results

You have now customized the Dynamic Workload Console header with your brand icon, and the Dynamic Workload Console and Self-Service Catalog login pages with your company logo.

Customizing roles

Following this scenario you can easily create your own customized roles in few steps.

About this task

Create a customized role for the Dynamic Workload Console

1. from the top panel click **Administration > Security > Manage Roles**
2. Click **Add new**, insert name, description and give the access to the pages desidered.
3. Click **Ok**
   From the same tab you can edit, delete or duplicate the new role.

Add the new role to an existing entity

1. From the **Manage Roles** panel, select the entity corresponding to the new role created by clicking on the number under the column **Entity**.
2. click **Add** and type the name of the entity and specify whether it is a user or a group
   Ensure you already have set the entity in the security file as described in the Configuring Authentication section of the Administration Guide. the default security file is `authentication_config.xml`.

Customizing your portfolio

How to customize your portfolio.

About this task

Many different elements can be customized in the navigation toolbar across the top and on the left side of you Dynamic Workload Console.

You can create a list of your favorite pages, including only the tasks you use most often.

The **Favorites** icon , lets you access your daily tasks.

To add a task to Favorites, simply click the icon when opening the drop-down menu. To remove an item from your list of favorites, click the icon .

What to do next

You can also define which pages must be automatically launched when logging into the Dynamic Workload Console by pinning them on the left bar. When the page is opened, click the Overflow menu and pin it.

Customizing your startup page

How to customize the startup page

About this task

In the Dynamic Workload Console, you can define the list of pages that are launched, on the left bar, every time you log in to the console. To pin a page on the left bar at every login, click the Overflow menu and then pin it when the page is displayed. Alternatively from the **Manage Roles** section, users with role Administrator, can add or remove the pinned pages for each role.

Create and customize your boards

About this task

In the Dynamic Workload Console, users can create and customize a **Board** not only with widgets to monitor the scheduling environment but also with web contents. For example, you can add a widget in the dashboard with your company's website and other external contents. see Creating a customized dashboard for monitoring on page 164 to have your own dashboard.

Customizing your global settings

How to customize global settings.

About this task

To customize the behavior of the Dynamic Workload Console, you can optionally configure some advanced settings. These settings are specified in a customizable file named `TdwcGlobalSettings.xml.template`.

By default, the customizable file is copied into the following path after you install the Dynamic Workload Console:

On Windows operating systems:

DWC_home`\usr\servers\dwcServer\registry\TdwcGlobalSettings.xml.template`

On UNIX and Linux operating systems:

DWC_home`/usr/servers/dwcServer/registry/TdwcGlobalSettings.xml.template`

If you have Administrator privileges, you can modify the file to replace default values with customized ones and enable commented sections. To enable commented sections, remove the tags that enclose the section. You then save the file locally with the name `TdwcGlobalSettings.xml`.

You can add and modify some customizable information, such as:

- The URLs that link to videos in the Dynamic Workload Console. For example, you can link to a company intranet server to view help videos rather than to a public video site.
- The maximum number of objects to be shown in the graphical views.
- The setting to display the plan view in a new window.
- The auto refresh interval for the **Show Plan View** graphical view.

- The creation of predefined tasks.
- The URLs where you can store customized documentation about your jobs or job streams to associate customized documentation to them.
- The current user registry in use.
- The timeout to read and write information on a IBM® Z Workload Scheduler engine.
- The maximum number of objects to be retrieved with a query, the maximum number of rows to display in a table, and the maximum number of direct queries to maintain in history.
- Allowing or preventing users from sharing tasks and engine connections.
- The display of all dependencies, both satisfied and unsatisfied.
- The use of audit files to track activities in the Self-Service Catalog and Self-Service Dashboards mobile applications.
- Displaying or hiding all predecessors from the What-if Analysis Gantt view.

This file is accessed at each login, and all configurations specified in the file are immediately applied, except for the **precannedTaskCreation** property. This property is read only when a user logs in for the first time and is then used whenever this user logs in again.

You can use any text or XML editor to edit this file, but ensure that you save it is as a valid XML file.

The file is organized into sections that group similar properties. An explanation of each section is available in the file. For more information, see TdwcGlobalSettings.xml sample on page 41.

Sections can also be repeated multiple times in the same file and applied differently to different user roles. To apply a section only to the users belonging to a role, the section must be included within the tags `<settings role="user_role">` and `</settings>`, where:

<user_role>

The user for which the enclosed configuration must be applied. The default value is all users, unless otherwise specified.

Only one **settings** section can be specified for each role. If a user has more than one role, the settings associated to the higher role are used.

To edit the file, proceed as follows:

1. Stop WebSphere Application Server Liberty Base using the following command:

   UNIX™

   Stop the application server

   ```
   ./stopAppServer.sh  [-direct]
   ```

   Windows™

   Stop the application server

   ```
   stopAppServer.bat [-direct
                [-wlpHome <installation_directory>]
                [-options <parameters>]]
   ```

as described in the section about starting and stopping WebSphere Application Server Liberty Base in *Administration Guide*.

2. Log is as root or Administrator to the Dynamic Workload Console.
3. Browse to

   On Windows operating systems:

   DWC_home\usr\servers\dwcServer\registry\TdwcGlobalSettings.xml.template

   On UNIX and Linux operating systems:

   DWC_home/usr/servers/dwcServer/configDropins/templates/TdwcGlobalSettings.xml.template

4. Edit the file as necessary, rename it to `TdwcGlobalSettings.xml` and save it.
5. Start WebSphere Application Server Liberty Base using the following command:

   UNIX™

   Start the application server

   ```
   ./startAppServer.sh  [-direct]
   ```

   Windows™

   Start the application server

   ```
   startAppServer.bat [-direct]
   ```

   as described in the section about starting and stopping WebSphere Application Server Liberty Base in *Administration Guide*.

Example:

```
<?xml version"1.0"?>
<tdwc>
.
.
<settings>
<graphViews>
<property name="planViewNewWindow" value="true"/>
</graphViews>
</settings>

<settings  role="TWSWEBUIOperator">
<graphViews>
<property name="planViewNewWindow" value="false"/>
</graphViews>
</settings>
.
.
</tdwc>
```

To view the complete syntax for the file, see TdwcGlobalSettings.xml sample on page 41.

Customize video URLs

This section shows how you should customize your URLs that link video content in the Dynamic Workload Console so that you can link to a company intranet server to view help videos rather than a public video site.

The `_baseURL` prefix will be added to all your video URLs . If you do not specify a link for your video the default setting will automatically be used.

```
<?xml version"1.0"?>
<tdwc>
.
.
<settings>
-<videoGallery>
<property name="_baseURL" value=""></property>
<property name="depLoop" value=""></property>
<property name="highlightRelDep" value=""></property>
<property name="viewDepPrompt" value=""></property>
<property name="usingImpactView" value=""></property>
<property name="createUseTasks" value=""></property>
<property name="weAddRemoveFile" value=""></property>
<property name="weCreateDeps" value=""></property>
<property name="weAddJob" value=""></property>
<property name="weHighlightDeps" value=""></property>
<property name="weCreateJCL" value=""></property>
</videoGallery>
```

Managing session timeout

The Dynamic Workload Console might encounter memory leak issues in a high user count scenario if users remain connected indefinitely without logging out.

The `DWC.timeout.expiration` parameter in the `ssl_config.xml` file can be set to properly manage session timeouts. The value to assign to the parameter is not fixed, and it depends on a number of factors, like the size of the environment and the number of Dynamic Workload Console users, and might require some fine tuning.

Note: The `DWC.timeout.expiration` value must always be smaller than the `ltpa.keys` expiration value visible in the `ssl_config.xml` file.

Example

In the following example:

```
<jndiEntry jndiName="DWC.timeout.expiration" value="32400000"/>
```

the value of the `DWC.timeout.expiration` parameter is set in milliseconds, while the `ltpa.keys` expiration value is set in minutes:

```
<ltpa keysPassword="{xor}Ozo5PiozKw=="
keysFileName="${server.config.dir}/resources/security/ltpa.keys"
expiration="600"/>
<webAppSecurity ssoUseDomainFromURL="false"
logoutOnHttpSessionExpire="true"/>
```

```
<httpSession invalidationTimeout="30m"
invalidateOnUnauthorizedSessionRequestException="true"
accessOnTimeout = "false"/>
```

To make the `DWC.timeout.expiration` parameter effective, the Java WebSphere Application Server Liberty must be restarted.

Override graphical view limits

This section contains the configuration parameters that apply to the graphical views in the plan, such as the maximum number of objects shown in each view.

planViewMaxJobstreams

The maximum number of job streams displayed in the Plan View. Default value is **1000**. Values greater than **1000** are not supported.

preProdPlanViewMaxJobstreams

The maximum number of job streams displayed in the preproduction plan view. Default value is **1000**. Values greater than **1000** are not supported.

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
<graphViews>
<property name="planViewMaxJobstreams" value="1000"></property>
<property name="preProdPlanViewMaxJobstreams" value="1000"></property>
</graphViews>
 </settings>
.
.
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Plan View in new window

This section is used to prevent Internet Explorer 7 from freezing while using the Plan View. To solve the problem, set value to **true**.

planViewNewWindow

Set it to **true** if you want the plan view to be displayed in a new window each time it is launched. Default value is **false**.

```
<?xml version"1.0"?>
<tdwc>
.
.
```

```
 <settings>
<graphViews>
<property name="planViewNewWindow" value="true"/>
</graphViews>
.
.
 </settings>
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Plan View auto refresh interval

Use this section to change the default setting of the auto refresh interval for the Show Plan View graphical view for all users. By default, the auto refresh interval is `300 seconds` (five minutes).

PlanViewAutorefresh

The graphical representation of the Plan View is automatically refresh every 300 seconds by default. To change this setting, edit the value assigned to the **DefaultTime** property. The minimum value you can set is 30 seconds. Any value specified below this value is reset to 30 seconds. You must restart the Dynamic Workload Console application server after modifying this value.

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
<PlanViewAutorefresh>
<property name="DefaultTime" value="300"/>
</PlanViewAutorefresh>
.
.
 </settings>
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Disable and customize NewsFeed function

This section contains the configuration details to be constantly up-to-date with product information.

FeedURL

Contains the URL from which you receive news and updates. Default value is:https://community.ibm.com/community/user/legacy

FeedType

A string that identifies the format of update information. Default value is **JSONP**.

PollInterval

The interval in seconds between two checks for updates. Default value is **600**.

PollInitialDelay

An initial delay in seconds before the first attempt to read the news feeds. After the initial load, the poll interval is used. Default value is **120**.

NewsFeed

Property used to add further customized news feeds. Specify the format and address of the file that contains the customized communication. Supported formats are RSS 2.0 and ATOM 1.0. You must write the communication in ATOM 1.0 or RSS 2.0 format and store this file in the an HTTP server complying with the *same origin policy*. For browser security reasons, this policy permits to access information only on server using the same protocol, hostname and port number as the one to which you are connected. Optionally, if you want to store your customized feed on an external server, you must configure an HTTP reverse proxy server mapping the external server address.

```
<property name="NewsFeed" type="RSS"
value="http://DWC_hostname:portnumber.com/news.rss" />
```

Note: To specify multiple feeds, you must specify multiple **NewsFeed** properties.

NewsFeedCategory

The name of the customized information. It can be used to identify informational, warning or alert messages, for example. The path to an image can also be added to better identify the information with an icon.

To add more category images, specify a list of properties named **NewsFeedCategory**, for example:

```
<property name="NewsFeedCategory" value="my company info"
icon="http://www.my.company.com/info.png" />
<property name="NewsFeedCategory" value="my company alert"
icon="http://www.my.company.com/alert.png" />
```

If no customized feed is specified, the default feed is used, which retrieves the latest product information from official support sites. To disable any notification, comment the entire section. To disable only external notifications about product information updates, assign an empty string as value to the `FeedURL` property of `JSONP` feed like:

```
  <property name="FeedURL" type="JSONP" value="" />
```

Example:

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
<NewsFeed>
<property name="NewsFeed" type="RSS"
value="http://www.DWC_hostname:portnumber.com/my_rss.xml" />
<property name="NewsFeed" type="ATOM"
value="http://www.DWC_hostname:portnumber.com/my_atom.xml" />
```

```
<property name="PollInterval" value="600" />
<property name="PollInitialDelay" value="1" />

<property name="FeedURL" type="JSONP" value="" />

<property name="NewsFeedCategory"
value="my company info" icon="http://www.DWC_hostname:portnumber.com
/info.png" />
<property name="NewsFeedCategory"
value="my company alert" icon="http://www.DWC_hostname:portnumber.com
/alert.png" />

</NewsFeed>
 </settings>
.
.
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Disable and customize the creation of predefined tasks

This section defines the environment for which predefined tasks are created.

precannedTaskCreation

Some predefined tasks are created by default and are available when you log in to the console. There is a predefined Monitor task for every object, for both z/OS® and distributed engines. Default value is **all**. To change this setting, use one of the following values:

all

All predefined tasks are created. This is the default.

distributed

Only predefined tasks for distributed engines are created

zos

Only predefined tasks for z/OS engines are created

none

No predefined task is created.

```
<?xml version"1.0"?>
<tdwc>
.
.
   <settings>
     <application>
      <property name="precannedTaskCreation" value="all"/>
     </application>
   </settings>
.
```

```
.
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Add customized URL to job and job streams

This section contains URLs where you can store customized documentation about your jobs or job streams. By default, this setting is not specified. If you want to associate customized documentation to a job or job stream, use this setting to specify the external address where this information is located.

If you want to specify a URL where customized documentation for a job and job stream is stored, uncomment the section lines, specify the required URL, and optionally assign a name to the UI label by specifying a value for the customActionLabel property. By default this name is **Open Documentation**. This label is then displayed in the **More Actions** menus in Monitor Jobs and Monitor Job Streams tasks, as well as in the graphical views of the plan (in the object's tooltips, context menus and properties). In this example, selecting **Open Documentation** accesses the relevant documentation making it possible to open the documentation while monitoring your job or job stream in the plan.

To implement this setting, assign values to the following keywords:

customActionLabel

The name of the action displayed in menus, object properties, and tooltips to access customized documentation about your jobs or job streams. By default this name is "Open Documentation" unless you customize the name with this keyword.

jobUrlTemplate

The address of your job documentation. No default value available.

jobstreamUrlTemplate

The address of your job stream documentation. No default value available.

Consider the following example:

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
  <twsObjectDoc>
    <property name="jobstreamUrlTemplate"
      value="http://www.yourhost.com/tws/docs/${js_encoded_folder_path}${js_name_w}"/>
    <property name="jobUrlTemplate"
      value="http://www.yourhost.com/docs/jobs/${job_name_w}"/>
    <property name="customActionLabel" value="Your Custom Label Name"/>
  </twsObjectDoc>
 </settings>
.
.
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

These properties must be valid URLs, containing one or more of the variables listed in the table below.

If you use any of the following special characters in the URL, you must write them as follows:

Table 3. Syntax for special characters

Special characters	Write them as...
quote (")	\"
apostrophe (')	'
ampersand (&)	&
less than (<)	&lt
greater than (>)	&gt
backslash (\)	\\

Multiple variables can be included in a URL and must be specified using the following syntax: `${variable}`:

Table 4. Variables used in the URL definition

Name	Object	Description
job_number_w	Job z/OS®	The number of the job
job_wkst_w	Job	The name of the workstation on which the job runs and the folder where it is stored, if any.
job_jsname_w	Job	The name of the job stream that contains the job and the folder where it is stored, if any.
job_jswkst_w	Job	The name of the job stream that contains the job and the folder where it is stored, if any.
job_actualarrival_w	Job z/OS®	The actual start time of the job (date format: YYYY-MM-DDThh:mm:ss)
job_actualend_w	Job z/OS®	When the job actually completed (date format: YYYY-MM-DDThh:mm:ss)
job_starttime_w	Job	The start time of the job (date format: YYYY-MM-DDThh:mm:ss)
job_id_w	Job	The ID of the job
job_returncode_w	Job	The return code of the job
js_name_w	Job stream	The name of the job stream that contains the job
js_wkst_w	Job stream	The name of the job stream that contains the job and the folder where it is stored, if any.

Table 4. Variables used in the URL definition (continued)

Name	Object	Description
js_id_w	Job stream	The job stream ID
js_latest_start_w	Job stream	The latest time at which a job stream can start (date format: YYYY-MM-DDThh:mm:ss)
engine_name_w	Engine	The name of the engine connection
engine_host_w	Engine	The hostname of the engine connection
engine_port_w	Engine	The port number of the engine connection
engine_plan_w	Engine	The ID of selected plan
engine_serv_w	Engine	The remote server name of the engine connection

User registry

Use this section to configure some properties related to the User Registry in use.

groupIdMap

The property groupIdMap is related to the groups of User Registry, and can be modified to map and display the specified value of each group. By default the common name of the group is displayed.

importSettingsMaxFileSize

The property importSettingsMaxFileSize is related to the "Manage settings" > "Import Settings" functionality and defines the max file size of the uploaded TDWCSettings.xml. KB is the unit of measure, and by default, it is set to 102400 KB (100 MB). If you need to upload a property file bigger than 100MB, you can increase this value, but for security purposes, it is strongly suggested to revert the file size back to the default value once the import has been performed.

Examples:

```
<?xml version"1.0"?>
<tdwc>
.
.
<settings>
<security>
<property name="groupIdMap" value="cn"></property>
<property name="importSettingsMaxFileSize" value="102400"></property>
</security>
</settings>
.
.
</tdwc>
```

Therefore, if you need to change the default value `"cn"` to "`racfid`", you can define this property as follows:

```
<property name="groupIdMap" value="racfid"></property>
```

See the section about the TdwcGlobalSettings.xml sample in *Administration Guide* to view the complete syntax for the file.

or see the section about user settings in *IBM Workload Automation: Dynamic Workload Console User's Guide* to manage Dynamic Workload Console settings.

For more information about how to customize global settings, see Customizing your global settings on page 25.

z/OS http connections

Use this section to configure the timeout to read and write information on IBM® Z® Workload Scheduler engine. When you connect to the IBM® Z® Workload Scheduler engine to retrieve a list of defined objects, you receive an error message if the list is not returned within the timeout period. The value is expressed in milliseconds.

Example:

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
<http>
<property name="zosHttpTimeout" value="90000" />
</http>
.
.
 </settings>
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Limit the number of objects retrieved by queries

Use this section to configure: the number of results displayed for Monitor tasks, the maximum number of rows to display on each page, and the number of direct queries to maintain in history.

If you want to limit the number of results produced by your queries, you can specify the maximum number of items that must be retrieved using the monitorMaxObjectsPM property.

Note: monitorMaxObjectsPM property only limits the number of results for archived plans queries. The property does not affect current plan queries.

For Multiple engine tasks, this limit is applied to each engine included in the query. Therefore, if you specify a limit of 500 results and, for example, you run a Monitor jobs on multiple engine task on three engines, the results produced by your query will be no more than 500 *for each engine*, for a maximum of 1500 rows.

Note: This setting does not apply to Monitor critical jobs tasks.

To set the maximum number of rows to display in a table view, configure the maxRowsToDisplay property.

To set the maximum number of direct queries to maintain in history, configure the maxHistoryCount property. These queries are available from the pull-down for the Query field on the Monitor Workload page.

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
  <monitor>
    <property name="monitorMaxObjectsPM" value="2000"></property>
  </monitor>

  <ph rev="v92"><monitor>
    <property name="maxRowsToDisplay" value="25"></property>
  </monitor>

  <monitor>
    <property name="maxHistoryCount" value="100"></property>
  </monitor>
</ph>
 </settings>

<settings>
        <search>
                  <property name="search_max_limit" value="1500"></property>
        </search>
    </settings>
.
.
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Related information

Job on page 58

Managing job definitions on page 114

Adding a job to a job stream

Job stream on page 59

Managing job stream definitions on page 117

Workload service assurance on page 96

Using workload service assurance to monitor z/OS critical jobs on page 212

Monitoring jobs running on multiple engines on page 214

Limit task and engine sharing

Use this section to prevent users from sharing tasks and engines.

By default there is no limit to task and engine sharing and all users are authorized to share their tasks and engine connections. If you want to change this behavior, preventing users from sharing tasks and engines, set this property to **true**.

The property default value is **false**, set it to **true** to enable the limit:

limitShareTask

Set to true to prevent users from sharing tasks.

limitShareEngine

Set to true to prevent users from sharing engine connections.

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
  <security>
   <property name="limitShareTask"     value="false" />
   <property name="limitShareEngine"   value="false" />
  </security>
 </settings>
.
.
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Entries in Workload Designer search windows

This section contains the configuration parameters that apply to the search views of Workload Designer.

search_max_limit

This optional parameter sets the maximum number of entries displayed in Workload Designer search windows. The default value is 250. It is recommended not to use values greater than 250.

Example:

```
<?xml version"1.0"?>
<tdwc>
.
.
<settings>
```

```
 <search>
  <property name="search_max_limit" value="250"></property>
 </search>
</settings>
.
.
</tdwc>
```

For more information about how to customize global settings, see Customizing your global settings on page 25.

Show all dependencies

This section defines whether to show all dependencies displayed, regardless of their being satisfied or not.

ShowDependencies

When you open the dependencies panel from Monitor jobs and Monitor job streams task results, by default only **Not Satisfied** dependencies are shown. Uncomment this section and leave the value set to "**true**" to have all dependencies displayed, regardless of their being satisfied or not. Possible values are:

true

All dependencies displayed, regardless of their being satisfied or not.

false

Only not satisfied dependencies are displayed.

```
<?xml version"1.0"?>
<tdwc>


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

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Auditing mobile app activity

This section defines whether to track activities performed in the Self-Service Dashboards application in an auditing log file.

For information about the name and location of the log file, see the logs and traces section in the *Troubleshooting Guide*.

SSAuditing

This value is set to "**true**" by default so that operations performed in the Self-Service Dashboards application are written to a log file. The log file contains information such as creation, modification and deletion dates, the operations performed in the mobile apps, and the user performing the operations. Possible values are:

true

Operations performed in the Self-Service Dashboards application are tracked in an auditing log file.

false

Operations performed in the Self-Service Dashboards application are not tracked in an auditing log file.

SSAuditingLogSize

The maximum size of a log file in KB. When a log file reaches the maximum size, the system rolls that log file over and creates a new file. By default, the maximum size of a log file is 100 KB.

SSAuditingLogFiles

The default number of log files to create. When this number is met and the latest log file reaches its maximum size, the system deletes the oldest log file and rolls the latest file over and creates a new file.

```
<?xml version"1.0"?>
<tdwc>
.
.
 <settings>
 <SSCMAuditing>
            <property name = "SSAuditing"         value="true"></property>
            <property name = "SSAuditingLogSize"  value="100"></property>
            <property name = "SSAuditingLogFiles" value="2"></property>
 </settings>
 .
 .
</tdwc>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Modifying the number of archived plans displayed in the Dynamic Workload Console

You can modify the number of archived plans displayed in the Monitor Workload view of the Dynamic Workload Console. The default number is 30 plans.

To modify the default number, configure the following property in the **TdwcGlobalSettings.xml** file:

```
<monitor>
    <property name="maxArchivedPlan"value="30"></property>
  </monitor>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

Show or hide predecessors from What-if Analysis Gantt view

When you have hundreds of predecessors, you can optimize performance by excluding them from the What-if Analysis Gantt view. By default, all predecessors are loaded into the What-if Analysis Gantt view. To exclude them, uncomment this section and leave the default setting of the property **whatIfAutoLoadPreds** to "`false`" . To revert back to the default behavior either set the property to "`true`" or comment the section again in the **TdwcGlobalSettings.xml** file.

To modify the default setting, configure the following property in the **TdwcGlobalSettings.xml** file:

```
      <WhatifAnalysis>
            <property name = "whatIfAutoLoadPreds" value="false"></property>
      </WhatifAnalysis>
```

See TdwcGlobalSettings.xml sample on page 41 to view the complete syntax for the file.

For more information about how to customize global settings, see Customizing your global settings on page 25.

TdwcGlobalSettings.xml sample

The following example is a sample of the file:

```
 <?xml version="1.0"?>
<tdwc>

   <!--
   ##############################################################################
   #################         SETTINGS FOR ALL USERS              ##################
   ##############################################################################
   -->
   <settings>

   <!--
   ##############################################################################
   #################       CUSTOMIZE LINKS TO VIDEOS             ##################
   ##############################################################################
   -->
   <!--
   This section shows how you should customize your URLs that link video content in
   the Dynamic Workload Console so that you can link to a company intranet server
   to view help videos rather than a public video site.

   ### This prefix "_baseURL" will be added to all video URLs ###

   ### Links to videos, missing entries or empty (blank) values are not considered ###

   #### Graphical view: detect loop                      ####
   #### Graphical view: highlight and release dependencies ####
   #### Graphical view: reply to prompt dependency       ####
   #### Graphical view: using the impact view            ####
   #### Table: creating and using tasks                  ####
   #### Workload editor: add and remove a file dependency  ####
   #### Workload editor: create a dependency             ####
   #### Workload editor: add a job                       ####
   #### Workload editor: highlight dependencies          ####
   #### Workload editor: creating a z/OS job             ####
   -->

   <!--
       <videoGallery>
           <property name="_baseURL" value=""></property>
           <property name="depLoop" value=""></property>
           <property name="highlightRelDep" value=""></property>
           <property name="viewDepPrompt" value=""></property>
           <property name="usingImpactView" value=""></property>
           <property name="createUseTasks" value=""></property>
           <property name="weAddRemoveFile" value=""></property>
           <property name="weCreateDeps" value=""></property>
           <property name="weAddJob" value=""></property>
```

```
            <property name="weHighlightDeps" value=""></property>
            <property name="weCreateJCL" value=""></property>
        </videoGallery>
    -->
    <!--
    ################################################################################
    ##################           GRAPHICAL VIEW SETTINGS            ######################
    ################################################################################
    -->
    <!--
    This section specifies the maximum number of objects shown in each graphical view.
    Default value is 1000 for all properties.
    -->
    <!--
        <graphViews>
            <property name="planViewMaxJobstreams" value="1000"></property>
            <property name="preProdPlanViewMaxJobstreams" value="1000"></property>
        </graphViews>
    -->
    <!--
################################  AutoLayout configuration  ######################

 nodeSep: a number of pixels representing the separation between adjacent nodes in the same rank. Default is 30
 edgeSep: a number of pixels representing the separation between adjacent edges in the same rank. Default is 0
 rankSep: a number of pixels representing the separation between ranks. Default is 0
 rankDir: direction of the layout (Possible values are: "TB" (top-to-bottom),"BT" (bottom-to-top),"LR" (left-to-right),"RL" (right-to-left)). Default is TB
 marginX: a number of pixels representing the separation between adjacent nodes in the same rank. Default is 100
 marginY: number of pixels to use as a margin around the top and bottom of the graph. Default is 100
 setLinkVertices: If set to true the layout will adjust the links by setting their vertices. It defaults to false.
      If the option is defined as a function it will be used to set the vertices of links at the end of the layout.
    -->
<!--
  <AutoLayout>
        <property name="nodeSep" value="30"/>
  <property name="edgeSep" value="0"/>
  <property name="rankSep" value="0"/>
  <property name="rankDir" value="TB"/>
  <property name="marginX" value="100"/>
  <property name="marginY" value="1500"/>
  <property name="marginY" value="1500"/>
  <property name="setLinkVertices" value="false"/>
    </AutoLayout>
    -->
    <!--
    ################################################################################
    ####################           PLAN VIEW IN NEW WINDOW          ######################
    ################################################################################
    -->
    <!--
    This section is used to prevent Internet Explorer 7 from freezing while using the Plan View. To solve the problem, set value         to true.
    Default value is false
    -->
    <!--
        <graphViews>
            <property name="planViewNewWindow" value="true"/>
        </graphViews>
    -->
    <!--
    ################################################################################
    #############       DISABLE /CUSTOMIZE CREATION OF PREDEFINED TASKS         ############
    ################################################################################
    -->
    <!--
    To avoid or customize the creation of predefined tasks at first logon.
    Possible values are:
        all             both distributed and z/OS tasks are created. This is the default value
        none            no task is created
        distributed     only distributed tasks are created
        zos             only z/OS tasks are created
    -->
    <!--
        <application>
            <property name="precannedTaskCreation" value="all"/>
            <property name="updateWorkstationMaxNumber" value="20"/>
        </application>
    -->

    <!--
        <PositionSorting>
            <property name="enabled" value="true"></property>
        </PositionSorting>
    -->

    <!--
    ################################################################################
    #############       ADD A CUSTOM DOCUMENTATION URL TO JOB/JOBSTREAM         ############
    ################################################################################
```

```
    -->
    <!--
    This section contains URLs where you can store customized documentation about your jobs or job streams.
    By default this setting is not specified. If you want to associate customized documentation to a job or
    job stream, use this setting to specify the external address where this information is located.
    If you want to specify a URL to be opened as related documentation for jobs and job streams,
    uncomment the section lines so that a new action, Open Documentation, is inserted in the More Actions
    menu for Monitor Jobs and Monitor Job Streams tasks. The new action links to the specified URL

   You can customize the URL template by using variables. The variables have the following syntax
                  ${<variable_name>}

    For the complete list of variables, please refer to the documentation.

    -->
    <!--
        <twsObjectDoc>
            <property name="jobstreamUrlTemplate" value="http://www.yourhost.com/tws/docs/jobstream/${js_name_w}" />
            <property name="jobUrlTemplate"       value="http://www.yourhost.com/docs/jobs/${job_name_w}" />
            <property name="customActionLabel"    value="Custom Action" />
        </twsObjectDoc>
    -->

<!--
    ###############################################################################
    ############                          USER REGISTRY                    ###########
    ###############################################################################
       In this section you can configure properties related to the User Registry in use.

       The property groupIdMap is related to the groups of User Registry, and can be modified
       to map and display the specified value of each group. By default the common name
       of the group is displayed.

       The property importSettingsMaxFileSize is related to the "Manage settings" > "Import Settings"
       functionality and defines the max file size of the uploaded TDWCSettings.xml.
       KB is the unit of measure, and by default, it is set to 102400 KB (100 MB).
       If you need to upload a property file bigger than 100MB, you can increase this value, but
       for security purposes, it is strongly suggested to revert the file size back to the default
       value once the import has been performed.
-->

    <!--
        <security>
            <property name="groupIdMap" value="cn"></property>
            <property name="importSettingsMaxFileSize" value="102400"></property>
        </security>
    -->

    <!--
        ###############################################################################
        ##################               Z/OS HTTP CONNECTIONS                    ###########
        ###############################################################################

    Use this section to increase or decrease timeout for http connection in Z/OS
    environment. Change this setting if you receive a connection timeout
    using plugin actions/picklists.

    The setting is in milliseconds.
    -->
        <!--
            <http>
                <property name="zosHttpTimeout" value="90000" />
            </http>
        -->
    <!--
    ###############################################################################
    ##################    LIMIT THE NUMBER OF OBJECTS RETURNED BY THE QUERIES   ###########
    ###############################################################################

        Use this section to configure: the number of results displayed for Monitor tasks, the maximum number of rows
        to display on each page, and the number of direct queries to maintain in history.
        This setting applies to all tasks except for Monitor critical jobs and Monitor jobs on multiple engines.
        If you want to limit the number of results produced by your queries, you can specify the maximum number of items that must be retrieved.
        To set the maximum number of rows to display in a table view, configure the maxRowsToDisplay property.
        To set the maximum number of direct queries to maintain in history, configure the maxHistoryCount property.
        These queries are available from the pull-down for the Query field on the Direct Query page.
        The property maxRowInfoNumber indicates the maximum number of objects to which actions can be performed in the plan.

        -->


        <!--
        <monitor>
          <property name="monitorMaxObjectsPM" value="2000"></property>
    </monitor>
    <monitor>
```

```
        <property name="maxRowsToDisplay" value="25"></property>
    </monitor>
    -->
    <!--
     You modify the number of archived plans displayed in the Monitor Workload view of the Dynamic
     Workload Console. The default number is 30 plans.
    -->
    <!--
    <monitor>
        <property name="maxArchivedPlan"value="30"></property>
    </monitor>
    -->
    <!--
    <monitor>
        <property name="maxHistoryCount" value="100"></property>
    </monitor>
    -->
 <!--
  Custom SQL report HTML format maximum limit. The default limit is 10000.
 -->
 <!--
  <monitor>
        <property name="SQL_REPORT_HTML_FORMAT_RESULT_MAX_NUMBER" value="10000"></property>
    </monitor>
    -->
     <!--
  <monitor>
        <property name="PROMPT_FILTER" value="true"></property>
    </monitor>
 -->
    <!--
    <settings>
        <search>
            <property name="search_max_limit" value="500"></property>
        </search>
    </settings>
    -->
    <!--
    <monitor>
        <property name="maxRowInfoNumber" value="100"></property>
    </monitor>
    -->
    <!--
    Increase the share maximum limit to share engines, boards, queries, and data sources.
    -->
    <!--
    <ShareLimit>
    <property name = "MaxShareCount" value="1000"></property>
    </ShareLimit>
    -->
    <!--
    ################################################################################
    ##################          LIMIT TASK AND ENGINE SHARING              ###########
    ################################################################################

    Use this section to prevent users from sharing tasks and engines.
    By default there is no limit to task and engine sharing and all users are authorized to share
    their tasks and engine connections. If you want to change this behavior, preventing users from
    sharing tasks and engines, set this property to true. The property default value is false,
    set it to true to enable the limit:

    -->
    <!--
        <security>
            <property name="limitShareTask"     value="false" />
            <property name="limitShareEngine"   value="false" />
        </security>
    -->

<!--
    ################################################################################
    ##################      CHANGE DEFAULT BEHAVIOR FOR DEPENDENCIES PANEL      ###########
    ################################################################################

    Use this section to change the default behavior of the UI when displaying
    dependencies in the dependencies panel. By setting this value to true, by default,
    all dependencies are displayed, and not just the unsatisfied ones.
-->
<!--
        <ShowDependencies>
            <property name = "AlwaysShowAllDependencies" value="true"></property>
        </ShowDependencies>
-->

<!--
    ################################################################################
    ##################    CHANGE DEFAULT BEHAVIOR FOR SSC AND SSD AUDITING     ###########
```

```
    ################################################################################
    Use this section to change the default behavior of the auditing of activities performed
    using the Self-Service Catalog and the Self-Service Dashboards applications. By default,
    auditing is enabled. You can also set the maximum size of the log file before it rolls
    over to a new log file, and the maximum number of log files maintained.
    Note: This section is valid only for the Self-Service Catalog V9.x, not for the latest one.
-->
 <!--
        <SSCMAuditing>
            <property name = "SSAuditing"         value="true"></property>
            <property name = "SSAuditingLogSize"  value="100"></property>
            <property name = "SSAuditingLogFiles" value="2"></property>
        </SSCMAuditing>
 -->

    <!--
    ################################################################################
    ##################             URL FOR AGENT LICENSE                  ###########
    ################################################################################
    Use this section to change the default Agent License URL.
    -->
    <!--
        <AgentLicense>
            <property name = "URL" value="Workoad Automation SaaS agent license document"></property>
        </AgentLicense>
    -->

    </settings>

    <!--
    ################################################################################
    ##################       SETTINGS FOR ALL Administrators users    ##############
    ################################################################################
    -->
    <settings role="Administrator">
    <!-- Put here setting to be applied only to users with Administrator role  -->
    </settings>
    <!--
    ################################################################################
    ##################        SETTINGS FOR ALL Operators users      ################
    ################################################################################
    -->
    <settings role="Operator">
    </settings>
    <!--
    ################################################################################
    ##################     SETTINGS FOR ALL Configurator users      ################
    ################################################################################
    -->
    <settings role="Configurator">
    </settings>
    <!--
    ################################################################################
    ##################      SETTINGS FOR ALL Developer users        ################
    ################################################################################
    -->
    <settings role="Developer">
    </settings>
    <!--
    ################################################################################
    ##################        SETTINGS FOR ALL Analyst users        ################
    ################################################################################
    -->
    <settings role="Analyst">
    </settings>

</tdwc>
```

For more information about how to customize global settings, see Customizing your global settings on page 25.

Integrating AI Data Advisor (AIDA) in the Dynamic Workload Console

Following this procedure, you can integrate AIDA in the Dynamic Workload Console.

About this task

To integrate AIDA in the Dynamic Workload Console, proceed as follows:

1. Configure the engine on the AIDA server. For further information, see: Adding IBM Workload Scheduler engines to AIDA.
2. Stop the Dynamic Workload Console by using the following command: <DWC_home>/appservertools/stopAppServer.sh [-direct].

   Note: On Windows systems, the command is: <DWC_home>\appservertools\stopAppServer.bat [-direct].

3. Modify the *widget_config.xml* file as follows:

   a. Copy the *widget_config.xml* template from the following path:`<DWC_home>/usr/servers/dwcServer/configDropins/templates`

      Note: On Windows systems, the path is: `<DWC_home>\usr\servers\dwcServer\configDropins\templates`

   b. Paste the *widget_config.xml* template in the following path: `<DWC_DATA_dir>/usr/servers/dwcServer/configDropins/overrides`

      Note: On Windows systems, the path is: `<DWC_home>\usr\servers\dwcServer\configDropins\overrides`

   c. Insert a value for **aidaHostname** and **baseUrl**.

      Where

      aidaHostname is the IP address where AIDA is hosted.

      baseUrl is the server URL where AIDA is hosted, and it also includes the server port where AIDA is running.

```
<server description="widget_config_variables">

  <!-- Update the host name and base url AIDA -->
  <variable name="aidaHostname" value=""/>
  <variable name="baseUrl" value="https://{IP_AIDA_Server}:{Port_AIDA_SERVER}"/>

  <!-- DO NOT CHANGE -->
  <jndiEntry id="aidaHost" jndiName="aidaHost" decode="false" value="${aidaHostname}"/>
  <jndiEntry id="baseUrlAida" jndiName="baseUrlAida" decode="false" value="${baseUrl}"/>
  <!-- DO NOT CHANGE ENDS -->


  <!-- Update the value of the below variable with the alias of the keystore only if a new store
 is used other than provided store-->
<variable name="certificateAlias" value=""/>

<!-- DO NOT CHANGE -->
<jndiEntry id="certificateAlias" jndiName="certificateAlias" decode="false"
 value="${certificateAlias}"/>
<!-- DO NOT CHANGE ENDS -->
```

```
</server>
```

4. Start the Dynamic Workload Console by using the following command: <DWC_home>/appservertools/startAppServer.sh [-direct].

   Note: On Windows systems, the command is: <DWC_home>\appservertools\startAppServer.bat [-direct].

Results

You successfully integrated AIDA in the Dynamic Workload Console, and now you can start monitoring your workload, prevent problems and reach operational excellence.

Integrating AIDA after an environment update

Following this procedure, you can integrate AIDA in the Dynamic Workload Console.

About this task

To integrate AIDA in the Dynamic Workload Console, proceed as follows:

1. Configure the engine on the AIDA server. For further information, see: Adding IBM Workload Scheduler engines to AIDA.
2. Stop the Dynamic Workload Console by using the following command: <DWC_home>/appservertools/stopAppServer.sh [-direct].

   Note: On Windows systems, the command is: <DWC_home>\appservertools\stopAppServer.bat [-direct].

3. Modify the *widget_config.xml* file as follows:

   a. Copy the *widget_config.xml* template from the following path:`<DWC_home>/usr/servers/dwcServer/configDropins/templates`

      Note: On Windows systems, the path is: `<DWC_home>\usr\servers\dwcServer\configDropins\templates`

   b. Paste the *widget_config.xml* template in the following path: `<DWC_DATA_dir>/usr/servers/dwcServer/configDropins/overrides`

      Note: On Windows systems, the path is: `<DWC_home>\usr\servers\dwcServer\configDropins\overrides`

   c. Insert a value for **aidaHostname** and **baseUrl**.

      Where

aidaHostname is the IP address where AIDA is hosted.

baseUrl is the server URL where AIDA is hosted, and it also includes the server port where AIDA is running.

```
<server description="widget_config_variables">

  <!-- Update the host name and base url AIDA -->
  <variable name="aidaHostname" value=""/>
  <variable name="baseUrl" value="https://{IP_AIDA_Server}:{Port_AIDA_SERVER}"/>

  <!-- DO NOT CHANGE -->
  <jndiEntry id="aidaHost" jndiName="aidaHost" decode="false" value="${aidaHostname}"/>
  <jndiEntry id="baseUrlAida" jndiName="baseUrlAida" decode="false" value="${baseUrl}"/>
  <!-- DO NOT CHANGE ENDS -->


  <!-- Update the value of the below variable with the alias of the keystore only if a new store
 is used other than provided store-->
<variable name="certificateAlias" value=""/>

<!-- DO NOT CHANGE -->
<jndiEntry id="certificateAlias" jndiName="certificateAlias" decode="false"
 value="${certificateAlias}"/>
<!-- DO NOT CHANGE ENDS -->


 </server>
```

4. Copy the *aida.crt* certificate from `<aida_install_path>/nginx/cert` and paste it in any folder.
   For custom certificates, see the AIDA Readme Files.
5. Add the token as follows:

   a. Where the Dynamic Workload Console is installed, go to the java bin folder: `<DWC_home>/java/jre/bin`

   Note: On Windows systems, the path is: `<DWC_home>\java\jre\bin`

   b. Launch the following command:

```
keytool -importcert -alias aidaTest -file <Local path where certificate is stored>/aida.crt
 -keystore <DWC_home>/usr/servers/dwcServer/resources/security/TWSServerTrustFile.jks -storepass
 default -storetype jks
```

   Note: On Windows systems, the command is:

```
keytool -importcert -alias aidaTest -file <Local
 path where certificate is stored>/aida.crt -keystore
 <DWC_home>\usr\servers\dwcServer\resources\security\TWSServerTrustFile.jks -storepass
 default -storetype jks
```

6. Confirm to upload the certificate.

7. Start the Dynamic Workload Console by using the following command: <DWC_home>/appservertools/startAppServer.sh [-direct].

   Note: On Windows systems, the command is: <DWC_home>\appservertools\startAppServer.bat [-direct].

Results

You successfully integrated AIDA in the Dynamic Workload Console, and now you can start monitoring your workload, prevent problems and reach operational excellence.

Chapter 6. IBM Workload Scheduler Concepts

Conceptual information about IBM Workload Scheduler

This section provides conceptual information about IBM Workload Scheduler and the Dynamic Workload Console.

Scheduling environment

Main concepts that help you to understand what a scheduling environment is

This section contains the main concepts that help you to understand what a scheduling environment is and what it comprises.

Workstation

Using workstations for scheduling jobs and job streams.

Note: This section provides information relating to the use of workstations for scheduling jobs and job streams.

If, instead, you want to learn about workstations because you are planning your network, see *IBM Workload Scheduler: Planning and Installation* or *IBM Z Workload Scheduler: Planning and Installation*.

The computer system where you run your jobs and job streams is called a *workstation*.

Workstations can be grouped logically into *workstation classes* and organized hierarchically into *domains*, managed by *domain managers*.

When you create a workstation definition for a system in your network you define a set of characteristics that uniquely identify the system and that control how jobs run on it. For example, the IP address of the workstation, if it is behind a firewall, if communications with it must be secure, what time zone it is in, and the identity of its domain manager.

Workstations in the IBM Workload Scheduler scheduling network perform job and job stream processing, but can also have other roles. When your network is designed, these roles are assigned to these workstations to suit the specific needs of your business. The following types of workstation are available:

Master domain manager

A workstation acting as the management hub for the network. It manages all your scheduling objects. The master domain manager workstation must be installed with this role.

Backup master domain manager

A workstation that can act as a backup for the master domain manager when problems occur. It is a master domain manager, waiting to be activated. Its use is optional. This workstation must be installed as a master domain manager workstation.

Learn more about switching to a backup master domain manager in *IBM Workload Scheduler: Administration Guide*.

Domain manager

A workstation that controls a domain and that shares management responsibilities for part of the IBM Workload Scheduler network. It is installed as an agent, and then configured as a domain manager workstation when you define the workstation in the database.

Dynamic domain manager

An installed component in a distributed IBM Workload Scheduler network that is the management hub in a domain. All communication to and from the agents in the domain is routed through the dynamic domain manager. When you install a dynamic domain manager the workstation types listed below are created in the database:

fta

Fault-tolerant agent component manually configured as domain manager

broker

Broker server component

agent

Dynamic agent component

Backup dynamic domain manager

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

Learn more about switching to a backup dynamic domain manager in *IBM Workload Scheduler: Administration Guide*.

When you install a dynamic domain manager the workstation types listed below are created in the database:

fta

Fault-tolerant agent component.

broker

Broker server component

agent

Dynamic agent component

Fault-tolerant agent

A workstation that receives and runs jobs. If there are communication problems with its domain manager, it can run jobs locally. It is installed as an agent, and then configured as a fault-tolerant agent workstation when you define the workstation in the database. This workstation is recorded in the IBM Workload Scheduler database as **fta**.

Standard agent

A workstation that receives and runs jobs only under the control of its domain manager. It is installed as an agent, and then configured as a standard agent workstation when you define the workstation in the database.

Extended agent

A workstation that 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 is run by the host whenever the extended agent is referenced in the production plan. Extended agents are used to extend the job scheduling functions of IBM Workload Scheduler to other systems and applications. For example, to launch a job on an extended agent, the host runs the access method, passing it job details as command line options. The access method communicates with the external system or application to launch the job and return the status of the job.

Also it is a workstation where a IBM Workload Scheduler access method has been installed as a bridge so that you can schedule jobs in the SAP, PeopleSoft, z/OS, or custom applications. It must be physically hosted by a fault-tolerant agent (up to 255 extended agents per fault-tolerant agent) and then defined as an extended agent in the database.

For more information, see *IBM Workload Scheduler: User's Guide and Reference* and *IBM Workload Automation: Scheduling Job Integrations with IBM Workload Automation*.

Workload broker agent

A workstation that manages the lifecycle of Workload Broker jobs in Workload Broker. It is installed and configured as a dynamic workload broker workstation in the database.

z/OS agent

A distributed workstation that runs jobs scheduled from IBM Workload Scheduler for z/OS. Like fault-tolerant workstations, it is installed in a IBM Workload Scheduler distributed domain. Unlike fault-tolerant workstations, it does not:

- Have fault tolerance
- Require an end-to-end server
- Need topology definitions

Communication with the agents is handled directly by the controller. For more information about the end-to-end scheduling with fault tolerance capabilities, see *IBM Z Workload Scheduler: Scheduling End-to-end with Fault Tolerance Capabilities*.

Virtual workstation

A workstation that is created with the automatic reporting attribute and the virtual option, defining a list of destinations, for the workload submission, that are used to spread the workload across trackers. When the scheduler processes the jobs submitted to a virtual workstation, it distributes the workload according to a sequenced turn criteria, based on a round-robin algorithm. To submit the job, at least one of the destinations in the list must be available.

You can associate open intervals, parallel servers, and fixed resources to each destination belonging to the defined pool. The association is disabled at virtual workstation level, because the jobs that you submit on a virtual workstation are actually run on a single destination. When you associate parallel servers with a virtual workstation destination, you can specify a value up to 65535. The alternative workstation definition is not applicable either at workstation level or at single destination level.

Remote engine

A workstation that represents locally a remote IBM® Workload Scheduler engine. It is a workstation used to run only *shadow jobs*. A shadow job is a job that runs locally and is used to map another job running on a remote engine. This relationship between the two jobs is called a *cross dependency*. You define a remote engine workstation if you want to federate your environment with another IBM® Workload Scheduler environment, either distributed or z/OS, to add and monitor dependencies on jobs running in the other scheduling environment. This type of workstation uses a connection based on HTTP protocol to allow the two environments to communicate.

Dynamic agent

A workstation that manages a wide variety of job types, for example, specific database or FTP jobs, in addition to existing job types. This workstation is automatically created and registered when you install the dynamic agent. Because the installation and registration processes are performed automatically, when you view the agent in the Dynamic Workload Console, it results as updated by the Resource Advisor Agent. You can group agents in pools and dynamic pools.

In a simple configuration, dynamic agents connect directly to a master domain manager or to a dynamic domain manager. However, in more complex network topologies, if the network configuration prevents themaster domain manager or the dynamic domain manager from directly communicating with the dynamic agent, then you can configure your dynamic agents to use a local or remote gateway.

Pool

A workstation grouping a set of dynamic agents with similar hardware or software characteristics to submit jobs to. IBM Workload Scheduler balances the jobs among the dynamic agents within the pool and automatically reassigns jobs to available dynamic agents if an agent is no longer available. To create a pool of dynamic agents in your IBM Workload Scheduler environment, define a workstation of type **pool** hosted by the workload broker workstation, then select the dynamic agents you want to add to the pool. A computer system group is automatically defined in the workload broker database together with its associated dynamic agents.

Dynamic pool

A workstation grouping a set of dynamic agents that is dynamically defined based on the resource requirements you specify. For example, if you require a workstation with low CPU usage and the Windows operating system installed to run your job, you specify these requirements using the Dynamic Workload Console or the **composer** command. When you save the set of requirements, a new workstation is automatically created in the IBM Workload Scheduler database. This workstation is hosted by the workload broker workstation. This workstation maps all the dynamic agents in your environment that meet the requirements you specified. The resulting pool is dynamically updated whenever a new suitable agent

becomes available. Jobs scheduled on this workstation automatically inherit the requirements defined for the workstation.

Related information

Creating a task to Monitor Workstations on page 168

Workstation types on page 231

Domain

The domain.

All the workstations in a distributed IBM Workload Scheduler network are organized into one or more *domains*, each of which consists of one or more agents and a domain manager acting as the management hub. Most communication to and from the agents in the domain is routed through the domain manager. If the agent has the "behind firewall" designation, all of it is.

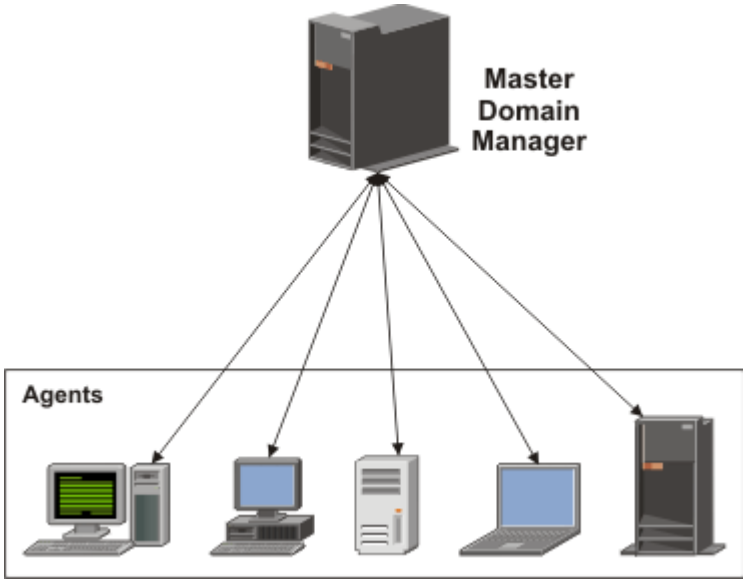
All the networks have a master domain where the domain manager is the master domain manager. It maintains the database of all the scheduling objects in the domain and the central configuration files. The master domain manager generates the plan and creates and distributes the Symphony file. In addition, logs and reports for the network are maintained on the master domain manager.

You can organize all the agents in your network into a single domain or into multiple domains.

Single-domain network

A single domain network consists of a master domain manager and any number of agents. Figure 1: Single-domain network on page 55 shows an example of a single-domain network. A single-domain network is well suited to companies that have few locations and business functions. All the communication in the network is routed through the master domain manager. With a single location, you are concerned only with the reliability of your local network and the amount of traffic it can handle.

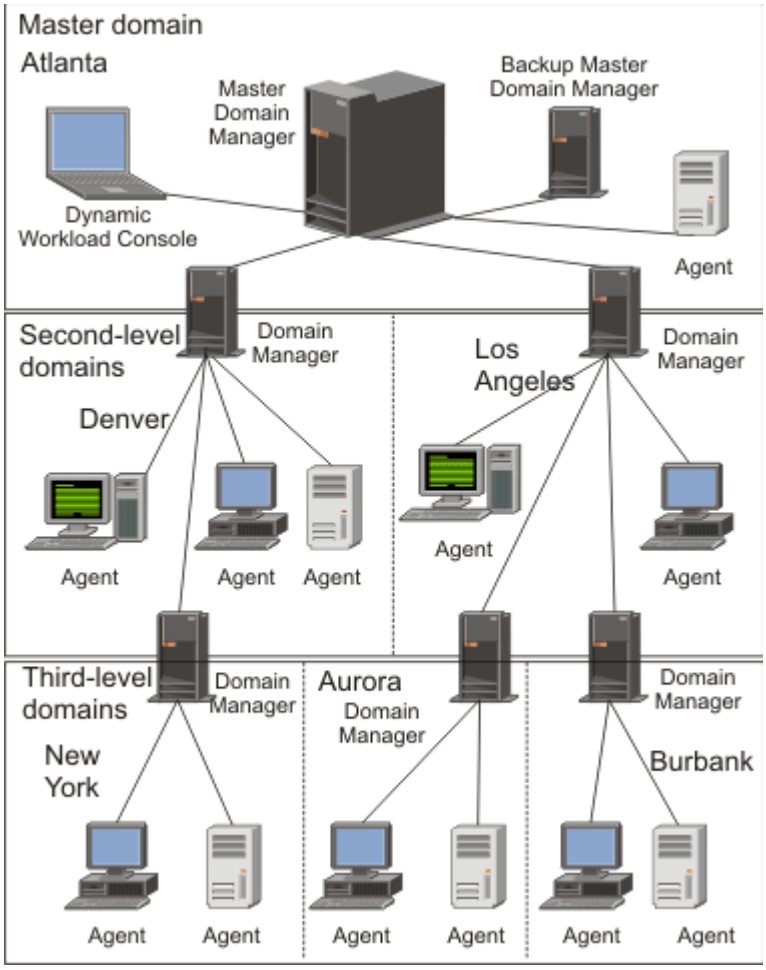
Figure 1. Single-domain network



Multiple-domain network

Multiple-domain networks are especially suited to companies that span multiple locations, departments, or business functions. A multiple-domain network consists of a master domain manager, any number of lower tier domain managers, and any number of agents in each domain. Agents communicate only with their domain managers, and domain managers communicate with their parent domain managers. The hierarchy of domains can have any number of levels.

Figure 2. Multiple-domain network



In Figure 2: Multiple-domain network on page 56, the master domain manager is located in Atlanta. The master domain manager contains the database files used to document the scheduling objects, and distributes the Symphony file to its agents and to the domain managers in Denver and Los Angeles. The Denver and Los Angeles domain managers then distribute the Symphony file to their agents and subordinate domain managers in New York, Aurora, and Burbank. The master domain manager in Atlanta is responsible for broadcasting inter-domain information throughout the network.

All the communication to and from the New York domain manager is routed through its parent domain manager in Denver. If there are schedules or jobs in the New York domain that are dependent on schedules or jobs in the Aurora domain, those dependencies are resolved by the Denver domain manager. Most inter-agent dependencies are handled locally by the lower tier domain managers, greatly reducing traffic on the network.

You can change the domain infrastructure dynamically as you develop your network. You move a workstation to a different domain, by changing the domain name in its database definition. The change takes effect when the master generates/extends the plan.

Note: You cannot schedule jobs or job streams to run on all workstations in a domain by identifying the domain in the job or job stream definition. To achieve this, you must create a *workstation class* that contains all the workstations in the domain.

For more information about domain definitions, see "*Defining objects in the database*" in the *User's Guide and Reference*.

For more information about workstation classes, see Workstation class on page 84

Related information

Managing job stream definitions on page 117

Scheduling objects

Scheduling objects that you can view and manage using the Dynamic Workload Console.

The set of scheduling objects described in the current plan is a subset of all the scheduling objects stored in the database. The scheduling objects accessible from the Dynamic Workload Console depend on your IBM Workload Scheduler environment.

For distributed environments, the scheduling objects reported in the production plan are:

- All the active workstations defined in the database. These are the workstations whose definition does not have the *ignore* flag set to on.
- All the domains.
- All the job streams scheduled to start in the production period and all jobs belonging to these job streams.
- All the resources, files, parameters, variables, and prompts defined in the job streams.

For z/OS environments, the scheduling objects reported in the current plan are:

- All the active workstations defined in the database.
- All the job streams scheduled to start in the production period and all jobs belonging to these job streams.
- All the resources that these jobs and jobs streams depend on.

To differentiate between jobs and job streams defined in the database and jobs and job streams scheduled to run within the production period, according to the IBM Workload Scheduler standard naming convention, each occurrence of a job or a job stream scheduled to run in the current plan is called an *instance*. A current plan can contain more than one instance of the same job or job stream.

Related information

Creating and managing engine connections on page 13

Designing your workload on page 110

Monitoring your Workload on page 170

Job

Information about jobs, which are scheduling objects used to define and run activities in the scheduling environment.

A *job* is a unit of work that specifies an action, such as a weekly data backup, to be performed on specific workstations in the IBM Workload Scheduler network.

In the IBM Workload Scheduler distributed environment, jobs can be defined either independently from job streams, within a job stream definition or in workflow folders.

In the IBM Workload Scheduler for z/OS environment, jobs can be defined only within a job stream and are called *operations*. You can have started-task operations, which are operations run on a computer workstation that are used to start and stop started tasks.

Regardless of whether the IBM Workload Scheduler engine is distributed or z/OS based, you can define locally a *shadow job* to map a remote job instance running on a different IBM Workload Scheduler engine.

For more information about the job definition, see the section about defining scheduling objects in *User's Guide and Reference*.

After job definitions have been submitted into the production plan, you still have the opportunity to make one-off changes to the definitions before they run, or after they have run. You can update the definition of a job that has already run and then rerun it. The job definition in the database remains unchanged.

Related information

Managing job definitions on page 114

Adding a job to a job stream

Status description and mapping for distributed jobs on page 222

Status description and mapping for z/OS jobs on page 225

Limit the number of objects retrieved by queries on page 36

Listing jobs and job streams

Managing job stream definitions on page 117

Prerequisite steps to create job types with advanced options on page 115

Job stream

A *job stream* is a sequence of jobs to be run, together with times, priorities, and other dependencies that determine the order of processing. Each job stream is assigned a time to run, represented by run cycle with type calendar, set of dates, or repetition rates.

Job streams can be organized in workflow folders and defined as part of the same line of business. Job streams can be easily monitored through the workflow folders and every workflow folder has its own security and notifications.

In an IBM® Z Workload Scheduler environment, job streams are called *applications*.

Related information

Managing job stream definitions on page 117

Status description and mapping for distributed job streams on page 227

Status description and mapping for z/OS job streams on page 229

Limit the number of objects retrieved by queries on page 36

Adding a job to a job stream

Listing jobs and job streams

Folder

A folder is an object that can be used to organize job streams into lines of business or other custom categories.

With the folder you can organize jobs and job streams in folders categorized by departments or adaptable tagging concepts.

You can also quickly move a set of jobs or job streams that use a specific naming convention into folders that are named after tokens contained in the object names. See the topic about organizing jobs and job streams into folders in the *User's Guide and Reference*.

Workload application

A workload application is one or more job streams together with all the referenced jobs that can be shared with other IBM Workload Scheduler environments through an easy deployment process.

A workload application is an IBM Workload Scheduler database object that acts as a container for one or more job streams. You can use workload applications to standardize a workload automation solution so that the solution can be reused in one or more IBM Workload Scheduler Workload Automation on Cloud environments thereby automating business processes.

You can prepare a workload application template in a source IBM Workload Scheduler environment and then export it so that it can be deployed in a target environment. The export process extracts from the source environment all of the elements necessary to reproduce the solution in another environment. It produces a compressed file containing a number of files required to import the workload application into the target environment. These files contain a definition of the objects in the source environment extracted from the IBM Workload Scheduler database. For those elements that depend on the topology of the target environment, some manual configuration is required. For example, the definitions extracted from the

source environment contain references to workstations that do not exist in the target environment. For this reason, before proceeding with the import, a mapping of some of the elements must be made associating the name of the object in the target environment.

The exported workload application template contains definitions or references for all of the following objects:

- Job streams
- Jobs
- Workstations, workstation classes
- Calendars
- Prompts
- Run cycles
- Run cycle groups
- Resources
- Internetwork dependencies
- External dependencies
- Event rules

For information about how to define workload application templates, see "Defining workload application" in the *User's Guide and Reference*.

Period

Periods are either *cyclic*, such as a week or 28-day period, or *noncyclic*, such as an academic semester.

Cyclic periods

Defined by their origin date and their length: a cyclic period starts on a specific date and has a specified number of days. There are two kinds of cyclic periods:

work-days-only cyclic periods

Only working days are considered.

all-days cyclic periods

All the days are considered.

Noncyclic periods

Defined by the origin date of each interval and can optionally have an end date for each interval.

Periods can be combined with offsets to create run cycles and define when a job stream runs. For example, an offset of 1 in a weekly period specifies Monday. An offset of 10 in a monthly period specifies the tenth day of each month.

The long-term planning process uses the calendar information, the period definitions, and the run cycle, to determine the days on which an application is scheduled to run.

If you run workload on fixed days of the week, month, or year, and take one of the standard IBM® Workload Scheduler for z/OS actions when this day falls on a non-working day, you do not need to create your own periods. You can describe most cases with rules such as:

- First Sunday in June
- First working day in the week
- Last Friday in the year
- Last non-working day in the month

If you use rules with their built-in calendar cycles (days of the week, months of the year, and so on), you probably need to create only special noncyclic periods, such as university semesters and tax years. The following sections show some examples of types of periods.

Cyclic period examples

Examples of cyclic periods are a day, a week, and a fortnight, with fixed intervals of 1 day, 7 days, and 14 days, respectively. An academic semester cannot be described as a cyclic period, because spring, summer, and fall semesters have different lengths. The following example shows a lunar calendar month, assumed to be 28 days:

Period name

Moon

Type

Cyclic based on all days

Interval

28 days

Interval origin

7 February 2009 (date of a new moon)

Non-cyclic period examples

Examples of non-cyclic periods are a quarter and a payroll period. You specify the start of each interval of a non-cyclic period with an origin date. This example shows a period for university semesters, with the interval origin and end specified for each semester:

Period name

Semester

Type

Non-cyclic

Interval origin

26 August 2009, 13 January 2010, 9 June 2010.

Interval end

13 December 2009, 16 May 2010, 28 June 2010

Non-cyclic periods have a once-a-year maintenance overhead when you must create the intervals for the coming months. For this reason, carefully consider how flexible your period definitions are, and remove potentially duplicated definitions.

Calendar

A *calendar* is a list of dates that define when a job stream runs.

Calendar in a distributed environment

A calendar can also be designated as a *non-working days* calendar in a job stream. A non-working days calendar is a calendar that is assigned to a job stream to represent the days when the job stream and its jobs do not run. It can also be used to designate Saturdays or Sundays, or both, as workdays. The default non-working days calendar for all job streams is called the *holidays* calendar.

Calendar in a z/OS environment

The calendar specifies normal working days and public holidays. IBM® Workload Scheduler for z/OS uses the calendar to determine when job streams are scheduled and to calculate dates for JCL tailoring.

You can specify the calendar when you create a job stream. If no calendar is specified for the job stream, IBM® Workload Scheduler for z/OS uses the calendar in the CALENDAR keyword of the BATCHOPT initialization statement, for batch services such as extending the long-term plan, or the calendar specified under the IBM® Workload Scheduler for z/OS options, for online services such as testing a rule with GENDAYS.

If no calendar is specified, a calendar with the name DEFAULT is used. If the DEFAULT calendar does not exist, all days are considered as working days. You can have several calendars, but always name your default calendar DEFAULT, and specify the same calendar name on BATCHOPT or in the IBM® Workload Scheduler for z/OS options. A calendar must contain at least one working day.

Related information

Designing your workload on page 110

Listing object definitions in the database

Run cycle

A *run cycle* specifies the days that a job stream is scheduled to run. Each run cycle is defined for a specific job stream and cannot be used by other job streams. You can specify the following types of run cycle:

simple

A specific set of user-defined days when a job stream is run.

daily

A run cycle that specifies that the job stream runs according to a day frequency and type that you set. For example, it might run daily, every three days, or just on working days.

weekly

A run cycle that specifies the days of the week when a job stream is run. For example, a job stream can be run every Monday, Wednesday, and Friday using a weekly run cycle.

monthly

A run cycle that specifies that the job stream runs according to a monthly day or date that you set. For example, it might run every 1st and 2nd day of the month, or every 1st Monday and 2nd Tuesday of the month.

It can also run, for example, every 1st and 2nd day of the month every two months.

yearly

A run cycle that specifies that a job stream runs, for example, yearly.

It can also run, for example, every three years.

exclusive

A run cycle that specifies the days and times when a job stream cannot be run. Exclusive run cycles take precedence over inclusive run cycles.

inclusive

A run cycle that specifies the days and times when a job stream is scheduled to run. Exclusive run cycles take precedence over inclusive run cycles.

offset-based

A run cycle that uses a combination of user-defined periods and offsets. For example, an offset of 3 in a period of 15 days is the third day from the beginning of the period. It is more practical to use offset-based run cycles when the cycle is based on cyclic periods. This term is used only in IBM Z Workload Scheduler, but the concept applies also to the distributed product.

rule-based

A run cycle that uses rules based on lists of ordinal numbers, types of days, and common calendar intervals (or period names in IBM Z Workload Scheduler). For example, the last Thursday of every month. Rule-based run cycles are based on conventional periods, such as calendar months, weeks of the year, and days of the week. In IBM Z Workload Scheduler, run cycles can also be based on periods that you define, such as a semester. This term is used only in IBM Z Workload Scheduler, but the concept applies also to the distributed product. You can also specify a rule to establish when a job stream runs if it falls on a free day.

Related information

Creating and managing run cycle groups and their run cycles

Run Cycle Preview

Run cycle group

You can optionally define a run cycle group for your job stream instead of, or in addition to, a number of single run cycles.

A run cycle group is a list of run cycles that are combined together to produce a set of run dates.

By using run cycle groups, you can benefit from the following advantages:

A run cycle group is a distinct database object

It is defined by itself and can be matched with one or more job streams. It is not defined as part of a specific job stream like single run cycles.

The same run cycle group can be used on different job streams

This improves the overall usability of the run cycles, because you can specify the same run cycle group in multiple job streams, avoiding the need to have multiple run cycle definitions for the same scheduling rules.

Run cycle groups enhance the use of exclusive run cycles

Exclusive (or negative) run cycles are used to generate negative occurrences, which identify the days when a job stream would normally be scheduled but is not required. The sum of the exclusive run cycles are subtracted from the inclusive ones. A negative occurrence always cancels any matching positive occurrences and you can specify a negative occurrence only if the positive equivalent already exists. An exact matching of the days, as well as any time restrictions, is required between the exclusive and inclusive run cycles for the cancellation to occur. Run cycle groups add much flexibility by allowing users to apply exclusive run cycles to a subset of the positive ones rather than to all of them. Group your run cycles into *subsets* so that the exclusive run cycles can be applied only to the positive occurrences generated by the run cycles belonging to the same set.

Run cycles must be organized into *subsets* within a run cycle group. The subsets are always in a logical **OR** relationship with each other. The result of the run cycle group is always a date or set of dates; it cannot be negative.

For example, you might want your job stream to run every day of the month except the last day of the month. But, you also want the it to be scheduled on the last day of the year (the last day of December). You can define a run cycle group using subsets, as follows:

Subset 1

- **Run cycle 1** - Inclusive run cycle every day of the month
- **Run cycle 2** - Exclusive run cycle on the last day of the month

Subset 2

- **Run cycle 3** - Inclusive run cycle on December 31st

where, run cycle 2 cancels the last day of each month in Subset 1, while run cycle 3 generates December 31st as a separate date and therefore you can schedule the job stream on Dec 31st.

Run cycle groups allow the use of a logical AND between individual run cycles in the subset

By default, the run cycles within a subset are in a logical **OR** relationship but you can change this to a logical **AND**, if the run cycle group result is a positive date or set of dates (Inclusive). For each run cycle, you can specify either operator (**AND** ,**OR**), obtaining the following behavior:

1. All the run cycles of the group that are in *AND* relationship are calculated first. The result of this calculation is a date or a set of dates.
2. Then, all the run cycles in an *OR* relationship are added to the result of the previous step.

A similar behavior is applied to inclusive and exclusive run cycles to determine the final date or set of dates of a group.

Inclusive (A)

Rule-based run cycle. Select days when the job stream is to be run if they belong to all A types of the set of run cycles.

Exclusive (D)

Exclusion rule-based run cycle. Select days when the job stream is NOT to be run if they belong to all D types of the set of run cycles.

For example, you can add two conditions together:

```
Run on Wednesday AND the 8th workday of the month.
```

In this way, the only scheduled dates are any 8th work day of the month that falls on a Wednesday.

Full compatibility with *traditional* run cycles

The *traditional* run cycles specified in the job stream definition can reference run cycle groups, with the possibility to specify shift or offsets on them (as with periods for z/OS or calendars for distributed systems).

A set of dates (interval starts) is created automatically either at run cycle level directly (inclusively or exclusively with offsets, or in the rule. This is a two-step process with run cycles:

1. Define the key "business event", such as, Month End, using run cycles and free day rules
2. Define rules that use the dates of the "business event" as the intervals against which the other batch run can be scheduled relative to.

For example, you have a *Month End process* that runs on the Last Friday of a month, but that moves forward to the next working day, except in December when it runs on the 3rd Friday of the month. This scheduling rule can be defined with a few rules, run cycles, and free day rules.

Two working days before Month End you need to run a pre-validation process to allow problems to be addressed before the run. You cannot choose the last Wednesday of the month, because in some months this might occur after the last Friday. Similarly, if the last Friday was a free day, the last Wednesday will not be 2 working days before it, because the Free Day rule applies ONLY to the day the rule falls on, it cannot look at anything else.

Many other batch runs might also need to be run on a certain number of days before or after the Month End, but the same restrictions apply.

You can now define work to run relative to something defined by a combination of run cycles and free day rules.

Use of calendars with run cycles within a run cycle group

Optionally, you can specify more than one calendar to calculate the working and non-working days definition for a run cycle. The primary calendar is used to calculate which working days are valid, and a secondary calendar is used to calculate specific non-working dates. If the dates calculated according to the secondary calendar match with the working days in the primary calendar, the job is scheduled; if they do not match, the job is not scheduled.

For example, a global company that runs workload in the United States for many other countries needs many calendar combinations to ensure that the batch jobs only run on a day that is a working day both in the United States and the other country. The calendar can be defined at job stream level and, if not specified, a default calendar is used. However, the calendar at run cycle level, whenever defined, can be used as secondary calendar and the job stream (or default) calendar can be used as the primary calendar.

For example, Primary calendar can be *WORKDAYS*, which is defined as MONDAY to FRIDAY excluding US holiday dates. You might also need to calculate the job runs based on calendar *HKWORK*, which is defined as Monday to Friday excluding Hong Kong holiday dates. The job might have several schedules:

- Run on working days, but not the last working day and not Mondays
- Run on Mondays, but not on the last working day
- Run on the last working day

Because each schedule is calculated against the *WORKHK* calendar it is also checked against the *WORKDAYS* calendar to ensure that it is scheduled on a US working day.

The use of time restrictions with run cycle groups

You can specify time constraints to define the time when processing must start or the time after which processing must no longer start. To do this, you can associate *time restrictions* to job, job streams, run cycles, and run cycle groups. When you define a time restriction, you basically obtain a *time*. Because you can associate time restrictions to multiple objects, the following hierarchy shows the order by which the different time restrictions are taken into consideration to actually define when to start the processing:

1. Time restriction defined in the run cycle into the job stream
2. Time restriction defined in the job stream

3. Time restriction defined in the run cycle contained in the run cycle group associated to the job stream.
4. Time restriction defined in the run cycle group associated to the job stream.
5. Start of Day

This means that:

Time restrictions in the job stream

Override and take precedence *over any other time restrictions* defined in the run cycles or run cycle groups associated to the job stream.

No time restrictions in the job stream nor in the run cycle group

The group generates only a date that is the *Start Of Day*. If offsets and free day rules are to be calculated, the calculation always starts from the *Start Of Day*.

Time restrictions in the run cycle group (not in the job stream)

Time restrictions (and possible offset) are calculated starting from the *Start Of Day* and the resulting date and time indicate the start of processing.

Examples

Table 5. Scenario 1. No time restriction in the run cycle group

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with offset (+ 3 days)	10/27 (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	0/29/ (Monday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday) Start of Day
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday) Start of Day

Table 6. Scenario 2. Time restriction in the run cycle group without offset

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with calendar offset (+ 3 days)	10/27/ (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	0/29/ (Monday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday) Start of Day
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday) Start of Day

Table 7. Scenario 3. Time restriction in the run cycle group with offset (+1 12:00)

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with calendar offset (+ 3 days)	10/27/ (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	10/29/ (Monday)
Run cycle group with offset +1 12:00	10/29/ (Monday)	10/30 12:00 (Tuesday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm

Table 7. Scenario 3. Time restriction in the run cycle group with offset (+1 12:00) (continued)

Run cycle group	Scheduled date	Earliest Start
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/03 12:00 (Saturday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/03 12:00 (Saturday)

Availability of the GENDAYS command at run cycle group level

Using GENDAYS, you can check the result of the combination of all the run cycles in the group.

Related information

Creating and managing run cycle groups and their run cycles

Run Cycle Preview

Operator instructions

In a IBM Workload Scheduler for z/OS environment, some jobs might require specific instructions about how they are to be handled. These instructions are called operator instructions.

An *operator instruction* can be permanent or temporary. A temporary instruction has a validity period associated with it, which specifies when the instruction is valid.

Parameter

A *parameter* is an object to which you assign different values to be substituted in jobs and job streams, either from values in the database or at run time. Parameters are useful when you have values that change depending on your job or job stream. Job and job stream definitions that use parameters are updated automatically with the value at the start of the production cycle. Use parameters as substitutes for repetitive values when defining jobs and job streams. For example, using parameters for user logon and script file names in job definitions and for file and prompt dependencies allows the use of values that can be maintained centrally in the database on the master.

For more information about how to define parameters, see the section about variable and parameter definition in the *User's Guide and Reference*.

Dependencies

Controlling processing using dependencies

When defining job streams and managing the workload in the plan, you can control process flow using dependencies.

You can specify the following types of dependencies:

Dependencies in a distributed environment:

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

Internal dependencies

These are dependencies established between jobs belonging to the same job stream.

External dependencies

These are dependencies between job streams, between job streams and jobs belonging to other job streams, or between jobs belonging to different job streams. The following resolution criteria are used to satisfy these dependencies:

Closest preceding

The closest preceding in time before the instance that includes the dependency.

Same scheduled date

The instance planned to run on the same day.

Within a relative interval

The closest preceding instance within the relative time interval you chose, or, if none is found, the closest following instance within the relative time interval you chose.

Within an absolute interval

The closest preceding instance within an absolute time interval you chose, or, if none is found, the closest following instance within the absolute time interval you chose.

Regardless of the used matching criteria, if multiple instances of potential predecessor job streams exist in the specified time interval, the rule used by the product to identify the correct predecessor instance is the following:

1. IBM Workload Scheduler searches for the closest instance that precedes the depending job or job stream start time. If such an instance exists, this is the predecessor instance.
2. If there is no preceding instance, IBM Workload Scheduler considers the correct predecessor instance as the closest instance that starts after the depending job or job stream start time.

Internetwork dependencies

These are dependencies on jobs or job streams running in another IBM Workload Scheduler network. Internetwork dependencies require a network agent workstation to communicate with the external IBM Workload Scheduler network.

Conditional dependencies (distributed) on page 74

A relationship between one job, named the *successor*, and one or more jobs or job streams, named *predecessors*, stating that the successor can run only when a specific combination of conditions occur or are satisfied by the predecessor job. Job output conditions are set on the job definition and when a conditional dependency is added to a job in a job stream, the output conditions that must be satisfied by the predecessor job are specified. Conditions can include if the job has started, the state of the job, and any number of custom defined conditions, often expressed as job return or exit codes. You can specify that only a single condition must be satisfied, all conditions must be satisfied, or specify a subset of conditions that must be satisfied.

When the conditions are not met by the predecessor, then any successor jobs with a conditional dependency associated to them are put in suppress state. Successor jobs with a standard dependency or no dependency at all defined run normally.

Note:

Conditional dependencies are evaluated after any standard dependencies in the job or job stream are satisfied.

Dependencies in a z/OS environment:

You can have different kind of dependencies between jobs and job streams. They can be:

Internal dependencies

These are dependencies established between jobs belonging to the same job stream.

External dependencies

These are dependencies between job streams, between job streams and jobs belonging to other job streams, or between jobs belonging to different job streams. The following resolution criteria are used to satisfy these dependencies:

Closest preceding

The closest preceding in time before the instance that includes the dependency.

Same scheduled date

The instance planned to run on the same day.

Within a relative interval

The closest preceding instance within the relative time interval you chose, or, if none is found, the closest following instance within the relative time interval you chose.

Within an absolute interval

The closest preceding instance within an absolute time interval you chose, or, if none is found, the closest following instance within the absolute time interval you chose.

Regardless of the used matching criteria, if multiple instances of potential predecessor job streams exist in the specified time interval, the rule used by the product to identify the correct predecessor instance is the following:

1. IBM Workload Scheduler searches for the closest instance that precedes the depending job or job stream start time. If such an instance exists, this is the predecessor instance.
2. If there is no preceding instance, IBM Workload Scheduler considers the correct predecessor instance as the closest instance that starts after the depending job or job stream start time.

Condition dependencies (z/OS) on page 76

It is a relationship between one job, named a *conditional successor*, and one or more jobs, named *conditional predecessors*, stating that the conditional successor can run only when a specific combination of conditional predecessor status and return code values occurs. You can define a conditional dependency where the conditional successor starts if its conditional predecessors are in ended-in-error or started status.

Note: Condition dependencies are always managed as external dependencies, even if they link jobs belonging to the same job stream occurrence in the plan.

Job streams in a z/OS environment do not support dependencies on files or prompts.

Cross dependencies on page 81

In multiple heterogeneous scheduling environments you can define dependencies on batch activities managed by other IBM Workload Scheduler environments. To define a cross dependency on a job running on a different IBM Workload Scheduler engine, you must define a dependency on a locally-defined *shadow job* pointing to the remote job instance and running on a *remote engine workstation*. The remote engine workstation manages the communication with the remote engine using an HTTP or HTTPS connection.

Dependencies on resources are supported by IBM Workload Scheduler in both the distributed and z/OS environments.

Prompt

Information about prompts, which are textual messages presented to the operator to pause the job or job stream processing.

A *prompt* identifies a text message that is displayed to the operator and halts processing of the job or job stream until an affirmative answer is received (either manually from the operator or automatically by an event rule action). After the prompt is replied to, processing continues. 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.

There are the following types of prompt:

global or named

A prompt that is defined in the database as a scheduling object. It is identified by a unique name and can be used by any job or job stream.

local or ad-hoc

A prompt that is defined within a job or job stream definition. It does not have a name, and it is not defined as a scheduling object in the database, therefore it cannot be used by other jobs or job streams.

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 called a *recovery job*.

For information about how to define prompts, see "Defining scheduling objects" in the *User's Guide and Reference*.

Related information

Designing your workload on page 110

Listing object definitions in the database

Resource

A *resource* is either a physical or logical system resource that you use as a dependency for jobs and job streams. A job or job stream with a resource dependency cannot start to run until the required quantity of the defined resource is available.

For information about how to define resources, see the section about resource definition in the *IBM Workload Scheduler User's Guide and Reference* .

Related information

Designing your workload on page 110

Listing object definitions in the database

File

A *file* is 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.

Related information

Designing your workload on page 110

Listing object definitions in the database

Applying conditional branching logic

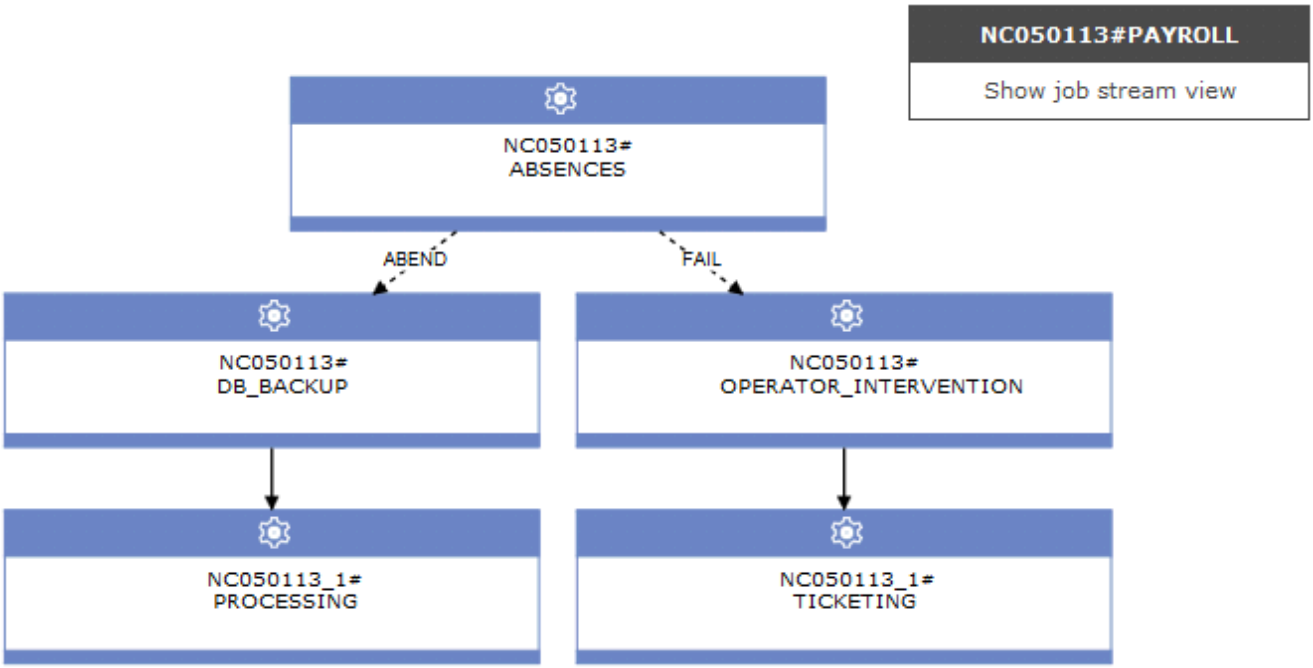
With IBM® Workload Scheduler you can define jobs to run when and as often as necessary. Sometimes some jobs might have to wait for other jobs to finish successfully before they start. Add even more flexibility to your job flows by choosing which job to run depending on the result of the job status or output of a previous job. Whenever you have conditions that specify whether or not a segment of your job flow should run, then that is a conditional dependency.

When specifying dependencies, you can define job flows with alternative branches based on conditions, specifically to achieve the same results as using IF/THEN/ELSE statements. You can use return codes, job status, output variables, and job log content as *conditional logic* elements to determine the start of a successor job. In addition to providing flexibility to your job flows, the Graphical View provides a graphical representation of the relationships between the jobs and job streams, including the dependencies and conditions. This at-a-glance view of your job flow is easy to read and you can also edit your job flow from this view.

Note:

Conditional dependencies are evaluated after any standard dependencies in the job or job stream are satisfied.

The following example shows the PAYROLL job stream that starts with the ABSENCES job, which is a predecessor job and is then followed by two possible branches of jobs that can run. The branch that runs depends on the outcome of the initial job, the predecessor ABSENCES job. Possible outcomes of the ABSENCES job are defined in *output conditions*. Any jobs in the flow that do not run, because the output conditions were not satisfied, are put in SUPPRESSED state, which is different from regular dependencies where jobs are put in Hold until the predecessor is in successful (SUCC) state. Predecessors can be either jobs or job streams.



Conditions can be **status conditions**, based on job status, or **other output conditions**, based on a mapping expression such as a return code, output variables, or output in a job log. When the predecessor is a job stream, the conditional dependency can only be a status condition.

Status conditions

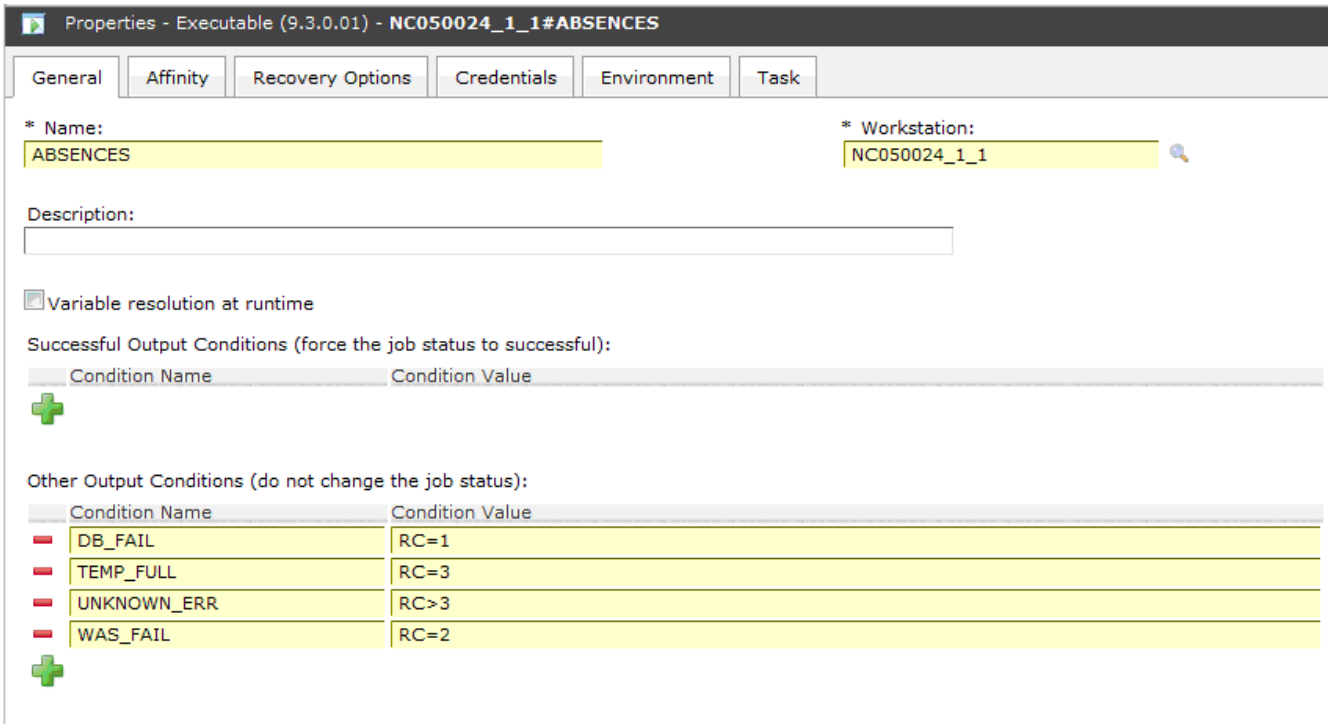
These are conditions based on job status, such as if the job started, or if the job completes in FAIL, ABEND, SUCC, or SUPPR state. For job streams, the valid statuses are SUCC, SUPPR, and ABEND.

Other output conditions

Other types of conditions, including successful output conditions, can be specified using a mapping expression, which can be:

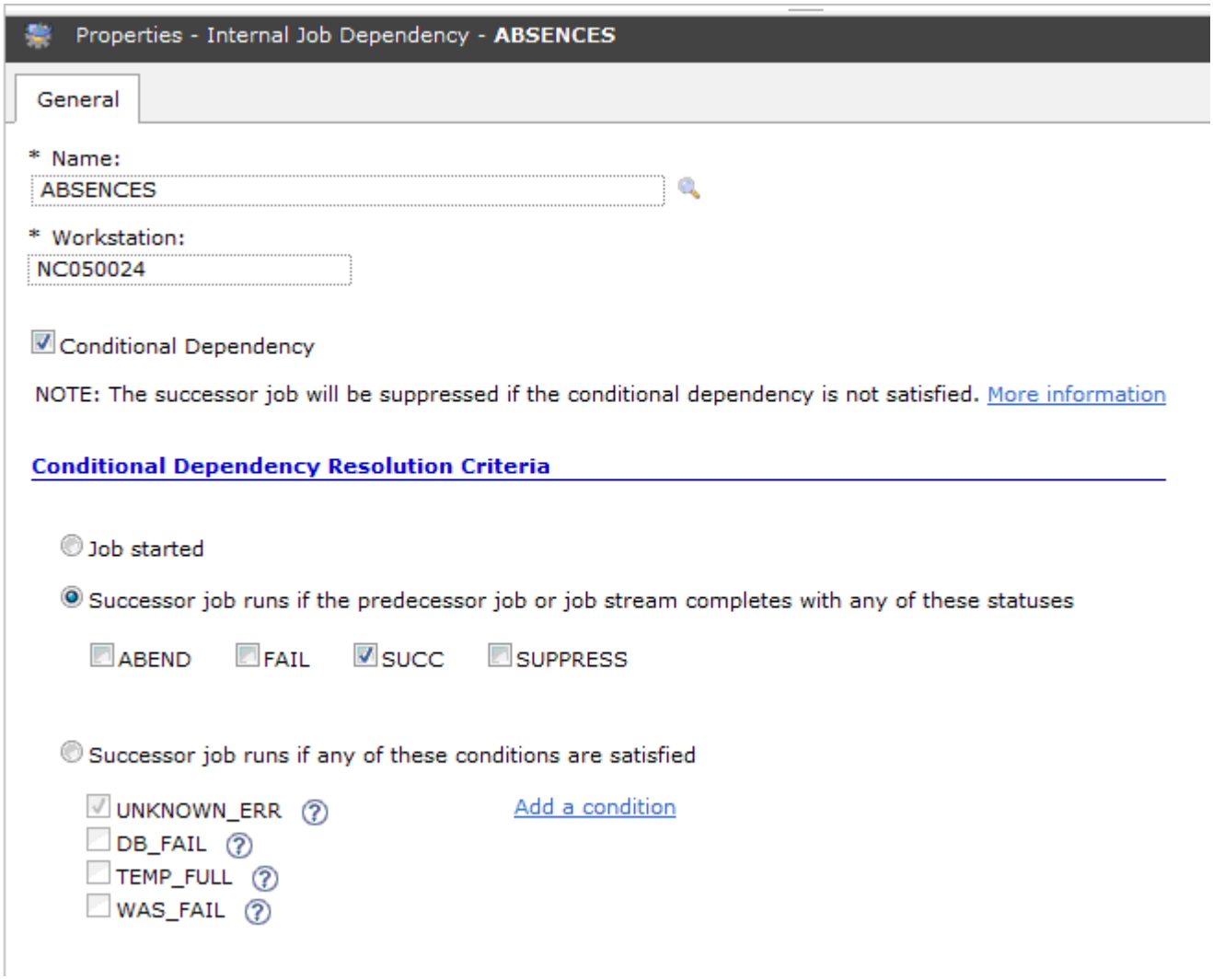
- A return code (fault-tolerant and dynamic agents)
- Output variables (dynamic agents)
- Job log content (dynamic agents)

A condition dependency relationship is set up by using a *condition*. You specify the output conditions in the job definition. You can define an unlimited number of conditional dependencies. When you choose to add a conditional dependency on this job, you select the status and output conditions that you want to be considered during the job processing. The following example shows other output conditions defined in the job definition.



You can define both successful output conditions, conditions that when satisfied signify that the job completed successfully, and other output conditions, which when satisfied determine which successor job to run. The output conditions are evaluated in "OR".

When this job is added to a job stream as a successor job, and a conditional dependency is added to the job preceding this job (predecessor job), then a selection of the conditions is made. The properties panel for the internal or external dependency is dynamically updated to include the conditions originally specified in the job definition. In addition to the conditions originating from the job definition, you can select conditions based on job status. If the selected conditions are satisfied during job processing, then the corresponding successor job runs.



You can also join or aggregate conditional dependencies related to different predecessors. A join contains multiple dependencies, but you decide how many of those dependencies must be satisfied for the join to be considered satisfied. You can define an unlimited number of conditional dependencies, standard dependencies, or both in a join.

Conditional dependencies are supported only for dependencies where the predecessor is a job or a job stream in the same network and where all the components are at least at the Version 9.3 Fix Pack 2 level. They are not supported on internetwork dependencies, nor on Limited Fault-Tolerant Agents for IBM i.

Condition dependencies (z/OS)

Conditional logic feature in IBM® Workload Scheduler for z/OS.

In IBM® Workload Scheduler for z/OS, you can specify that jobs are dependent on other jobs. For example, if job A1 must complete before job A2 can start, then A1 is a *predecessor* of A2, and A2 is a *successor* of A1. These relationships between jobs are called dependencies.

When specifying dependencies, you can also define work flows with alternative branches based on conditions, specifically to achieve the same results as using IF/THEN/ELSE statements in the job JCL. You can use both job return code and job status as *conditional logic* elements to determine the start of a successor job. The following example shows how this works.

A condition dependency relationship is set up by using a *condition*.

You can define condition dependencies at the following levels:

Job level

By conditioning the start of the successor to the check on job return code or status of the predecessor.

Step level

By conditioning the start of the successor to a specific step return code of the predecessor.

How condition dependencies work

A *condition dependency* is a specific check of the status or return code of a predecessor job or of the return code of a job step.

The job processing flow is affected by the conditions set and their final status.

The status of a condition is set based on the rule defined and on the statuses of its condition dependencies.

The condition dependency is evaluated only when a path in the plan exists, otherwise the condition dependency remains Undefined until a manual intervention or a rerun is done.

A possible path for the conditional predecessor exists when at least one of the following conditions occurs:

- The job has status **Completed** and a normal successor exists.
- There is at least one conditional successor that has *all* the subsets of conditions, referencing that conditional predecessor, set to **true**, according to the condition rules.

For example:

- A conditional predecessor (Job A) has several conditional successors (Jobs B, C, D)
- Each conditional successor has a set of condition dependencies, relating to job A, that must be satisfied to make it possible for the successor to start.

- Job A runs and changes its status.
- If at least one subset of conditions between job A and one of its successors is **true**, the path in plan exists and all the successors' condition dependencies related to job A are evaluated. Otherwise all condition dependencies are left undefined.

When specifying predecessors in the database, you can define a list of *conditions* by combining single *condition dependencies* on predecessor job status or return code. You cannot define a job both as a conditional and as a normal predecessor of another job. For each condition you can specify one of the following rules:

- At least *n* number of conditions out of all the condition dependencies must be satisfied. This rule corresponds to the OR operator in Boolean logic.
- All the condition dependencies in the list must be satisfied. This rule corresponds to the AND operator in Boolean logic.

At run time, the scheduler evaluates the condition status resulting from the condition dependencies statuses, based on the selected rule. The condition status can be:

True

When ALL condition dependencies are true.

If the rule is set to AND

When ALL condition dependencies are true.

If the rule is set to OR (at least *n* condition dependencies must be true)

When at least *n* condition dependencies are true.

False

The condition was not satisfied.

If the rule is set to AND

When at least one condition dependency is false.

If the rule is set to OR (at least *n* condition dependencies must be true)

When at least *n* condition dependencies cannot be true.

Undefined

When the rule cannot be evaluated yet.

A set of conditions results as satisfied if all the conditions are satisfied, according to the logic of the AND operator.

When a predecessor ends, the successor job status changes to one of the following statuses:

Waiting

Undefined, until the scheduler evaluates all the defined conditions. At least one normal predecessor is not in **Completed** or **Suppressed by Condition** status or at least one condition is U (Undefined). The scheduler processes all subsequent statuses as usual, up to a final status.

Ready

Ready, when all the defined conditions are satisfied. Job normal predecessors are in **Completed** or **Suppressed by Condition** status and all its conditions are **True**. The scheduler processes all subsequent statuses as usual, up to a final status.

Suppressed by Condition

Suppressed by condition, when the defined condition dependency is not satisfied. At least one condition is **False**.

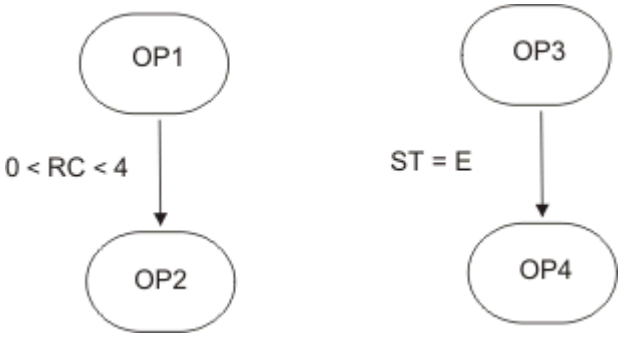
Note: When evaluating conditional successors status, predecessor jobs in status **Suppressed by Condition** are considered equal to predecessor operations in status **Completed**.

Examples of condition dependencies

Use a job-level condition dependency when you want a successor job to start depending on a combination of one or more return codes or statuses of predecessor jobs.

#unique_97_Connect_42_exconddepdef on page 79 shows the two different types of job level conditions, one based on the predecessor return code and the other based on the predecessor job status. For example, using the return code as condition type, you can define that job OP2 is dependent on job OP1, specifying that OP2 must run when OP1 ends with a return code in the range from 1 to 3. Similarly, using job status as condition, you can define job OP4 as dependent on job OP3, specifying that OP4 must run if OP3 ends with status **Error**.

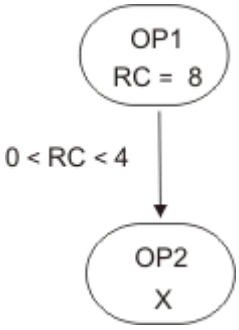
Figure 3. Example of a condition dependency definition



In this example, OP1 is a *conditional predecessor* of OP2 and OP3 is a *conditional predecessor* of OP4.

In the previous example, if OP1 ends with return code of 8, the scheduler sets OP2 to status **Suppressed by Condition**, because the condition is not satisfied.

Figure 4. Example of a condition dependency at run time



For further information about conditional logic, see *IBM Z Workload Scheduler: Managing the Workload*

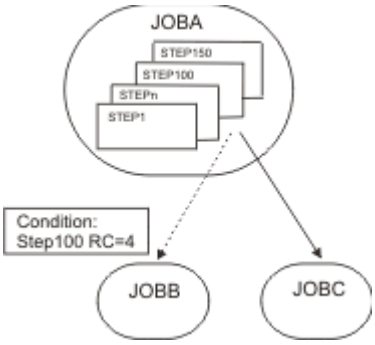
Step level dependency

If you configured IBM® Workload Scheduler for z/OS to track step-end events, then the step dependencies are checked at step-end time when the return code value is available.

This section contains an example showing how job processing flow is affected when using step-level conditions.

If the predecessor job is associated to a job comprising several steps, you can specify a dependency on step return codes. #unique_97_Connect_42_autorecjssld on page 80 shows an example of conditional dependency logic at job step level, to obtain auto-recovery applications with recovery jobs that can start without waiting for the end of predecessor jobs, depending on the result of specific steps.

Figure 5. Auto-recovery job stream with step level dependency



In this example:

- JOBB can start if STEP100, belonging to JOBA, ends with RC=4.
- JOBC is a normal successor of JOBA and therefore starts if JOBA status is **Completed**.

Handling recovery using condition dependencies

Using condition dependencies, the error status of a job can be used as a criteria for starting a successor, when this successor is used as a recovery job.

By specifying the conditional recovery job option, you can define that the job is used as the recovery job for a conditional predecessor.

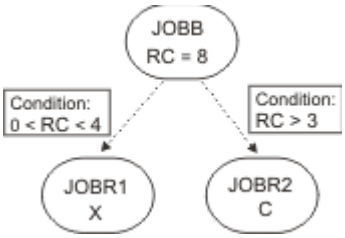
Any conditional predecessor that Ended-in-Error, with a status or error code matching a condition dependency defined for the job, does not prevent the daily plan process from removing the occurrence to which the predecessor belongs. To check if the status **Ended-in-error** can be ignored at occurrence removal phase, the daily plan process uses a field automatically set by the scheduler, corresponding to **Recovered by condition**.

Note: As soon as a recovery job becomes ready, the scheduler checks the predecessors in error status at that time. Any predecessor that ends in error after the recovery job runs cannot be flagged as **Recovered by condition**. The daily plan process removes the occurrence in the following cases:

- The occurrence status is **Completed**.
- The occurrence status is **Ended-in-error** , and includes only jobs in one of the following statuses:
  - **Completed**
  - **Suppressed by condition**
  - **Ended-in-error** with the **Recovered by condition** option specified.

For example, suppose that either JOBR1 or JOBR2 must run when JOBB ends with an error. You can specify JOBB as their conditional predecessor, as shown in #unique_97_Connect_42_examprecjobconddep on page 81.

Figure 6. Example of recovery job with condition dependencies



When defining JOBR1 and JOBR2 and specifying JOBB as conditional predecessor, you can also set the **Conditional recovery job** option to have the daily plan process remove the occurrence containing JOBB, because it ended with an error code matching one of the defined condition dependencies.

Cross dependencies

Using cross dependencies and *shadow jobs*.

A cross dependency is a dependency of a *local job* on a *remote job* running in a different scheduling environment. It is achieved by using a *shadow job*, which runs in the same environment as the local job and maps the remote job processing.

Cross dependencies help you integrate workload running on more than one engine. They can be both IBM® Workload Scheduler for z/OS engines (controller) and IBM® Workload Scheduler engines (master domain manager).

The following objects allow you to define and manage cross dependencies:

Remote engine

A workstation that represents locally a remote IBM® Workload Scheduler engine. It is a workstation used to run only *shadow jobs*. A shadow job is a job that runs locally and is used to map another job running on a remote engine. This relationship between the two jobs is called a *cross dependency*. You define a remote engine workstation if you want to federate your environment with another IBM® Workload Scheduler environment, either distributed or z/OS, to add and monitor dependencies on jobs running in the other scheduling environment. This type of workstation uses a connection based on HTTP protocol to allow the two environments to communicate.

Shadow job

A job running locally that is used to map a job running on the remote engine. This job is called a *remote job*. Shadow jobs can run only on remote engine workstations. The shadow job definition contains all the information needed to correctly match the remote job in the plan of the remote engine. The status transition of the shadow job reflects the status transition of the remote job.

Remote job

A job that runs on a remote scheduling environment and is mapped by a shadow job to become a dependency for a job that runs in a local environment.

To add a cross dependency to a local job on a job that is defined on a remote engine, you must define a normal dependency for your local job on a shadow job that:

- Points to the remote job on which you want to create the cross dependency
- Is defined on a local workstation of remote engine type, that points to the engine where the remote job is defined.

To do this, you must

1. Create a remote engine workstation where the shadow job runs.
2. Create a shadow job pointing to a specific job instance defined on a remote engine.

   Shadow jobs can be added to the plan by the plan creation process or dynamically at run time. The shadow job scheduled time identifies the remote job instance in the remote engine plan.

   The **bind** process is the process to associate a shadow job with a job instance in the remote engine plan.

   As soon as the bind is established, the remote engine sends back an HTTP notification containing the status of the bind and, if the bind was successful, the information to identify the remote job instance bound. This information is saved in the shadow job instance details.
3. Add the shadow job as a dependency of the local job.

The resolution of the cross dependency depends on the status of the shadow job, which reflects at any time the status of the remote job. Because the remote job status transition is mapped into the shadow job status transition, the status of the cross dependency is represented by the status of the normal dependency.

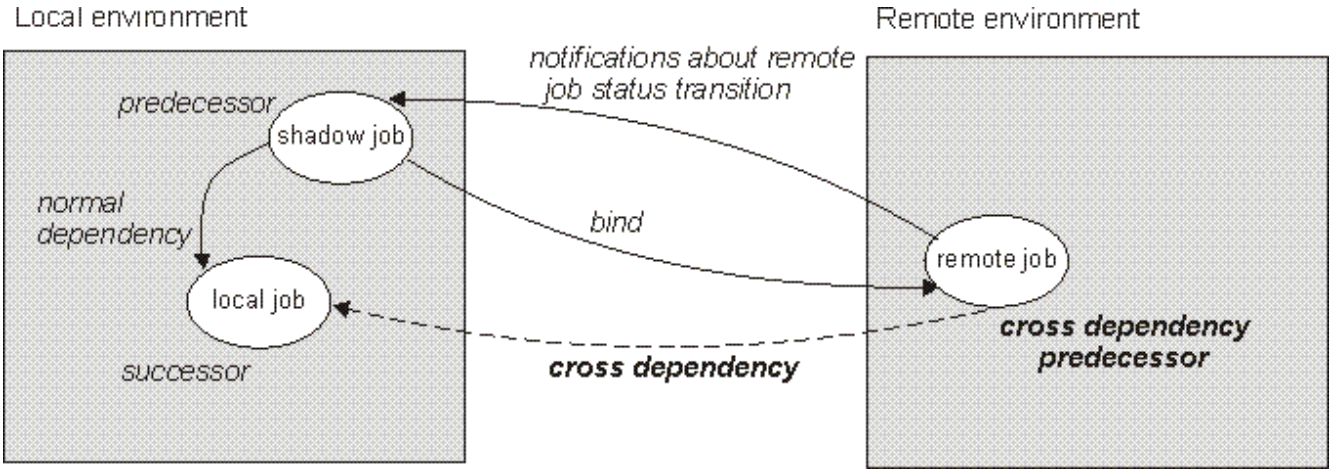
The key attributes to identify the remote job instance and the matching criteria depend on the type of remote engine where the remote job instance is defined. z/OS engines support only *closest preceding* matching criteria. Distributed shadow jobs, instead, support the four matching criteria available for external dependencies. See Dependencies on page 69 for more details.

The scheduled time of the job stream containing the shadow job is used to find the match.

To avoid incongruence, at plan creation or extension time, consistency checks are performed to ensure that no mismatch has occurred in between the definition of jobs and workstations in the database and their inclusion in the current plan.

Figure 7: Cross dependencies on page 83 summarizes how cross dependencies work.

Figure 7. Cross dependencies



For more information about cross dependencies, see the sections about defining and managing cross dependencies in *User's Guide and Reference* and in *Managing the Workload*.

Related information

Creating cross dependencies

User

A *User* is the user name used as the login value for several operating system job definition. Users must be defined in the database.

If you schedule a job on an agent, on a pool, or on a dynamic pool, the job runs with the user defined on the pool or dynamic pool. However, the User must exist on all workstations in the pool or dynamic pool where you plan to run the job.

Users can be defined in a specific folder if you want to organize them by line of business or some other custom category.

Related information

Designing your workload on page 110

Listing object definitions in the database

Workstation class

A *workstation class* is a group of workstations with similar job scheduling characteristics. Any number of workstations can be grouped in a class, and a workstation can be included in many classes. Jobs and job streams can be assigned to run on a specific workstation class and this makes the running of jobs and job streams across multiple workstations easier.

For example, you can set up the following types of workstation classes:

- Workstation classes that group workstations according to your internal departmental structure, so that you can define a job to run on all the workstations in a department
- Workstation classes that group workstations according to the software installed on them, so that you can define a job to run on all the workstations that have a particular application installed
- Workstation classes that group workstations according to the role of the user, so that you can define a job to run on all the workstations belonging to, for example, managers

In the previous example, an individual workstation can be in one workstation class for its department, another for its user, and several for the software installed on it.

Workstations can also be grouped into domains when your network is set up. Because the domain name is not one of the selection criteria used when choosing where to run a job, you might need to mirror your domain structure with workstation classes if you want to schedule a job to run on all workstations in a domain.

For more information about how to define workstation classes, see the section about workstation class definition in *IBM Workload Scheduler: User's Guide and Reference*.

Related information

Designing your workload on page 110

Listing object definitions in the database

Variable table

A *variable table* is a table containing multiple variables and their values. All global parameters, now called *variables*, are contained in at least one variable table.

You are not required to create variable tables to be able to use variables, because the scheduler provides a default variable table.

However, you might want to define a variable with the same name, but different values, depending on when and where it is used. You do this by assigning different values to the same variable in different variable tables. You can then use the same variable name in different job definitions and when defining prompts and file dependencies. Variable tables can be assigned at run cycle, job stream, and workstation level.

Variable tables can be particularly useful in job definitions when a job definition is used as a template for a job that belongs to more than one job stream. For example, you can assign different values to the same variable and reuse the same job definition in different job streams.

For information about how to define variable tables, see the section about variable table definition in the *IBM Workload Scheduler User's Guide and Reference* .

Related information

Designing your workload on page 110

Listing object definitions in the database

Production process

Generating the IBM Workload Scheduler production plan.

IBM Workload Scheduler production is based on a plan that runs in a production period.

You can define the production period when creating or extending the *production plan* and it can span from a few hours to multiple days (by default it lasts 24 hours).

The production plan contains information about the jobs to run, on which fault-tolerant agent, and what dependencies must be satisfied before each job can start.

You use the `JnextPlan` script to generate the production plan and distribute it across the IBM Workload Scheduler network. Then, if you want to extend your production plan at a fixed time interval, for example every day, you have the option to automate the extension by using the `final` job stream at the end of each production period. A sample job stream helps you to automate plan management and runs the sequence of script files included in `JnextPlan` to generate the new production plan.

When the production plan is generated, all of the required information about that production period is taken from the scheduling environment and object definitions and is included in the plan.

During the production period, the production plan is regularly updated to show what work is completed, in progress, and left to process.

In IBM Workload Scheduler for distributed environments or in a z/OS end-to-end network, a file called *Symphony* contains all the information about the production plan. This file is sent to all the subordinate domain managers and fault-tolerant agents in the scheduling environment. This allows the fault-tolerant agents throughout the network to continue their processing even if the network connection to their domain manager is down.

IBM Workload Scheduler processes monitor the production plan and make calls to the operating system to launch jobs as required. The operating system runs the jobs, and informs IBM Workload Scheduler if the job completed successfully. This information is used to update the production plan to indicate the status of the job.

From the Dynamic Workload Console or the command line interface, you can view and make changes in the current production plan.

Even once job definitions have been submitted into the production plan, you still have the opportunity to make one-off changes to the definitions before they run, or after they have run. You can update the definition of a job that has already run and then rerun it. You can update a job definition from either the Job Stream Graphical view, the job monitoring view, or from the conman command line. The job definition in the database remains unchanged.

Database

The IBM® Workload Scheduler database.

The IBM® Workload Scheduler database, hereafter referred to as the database, is a relational database that is accessible by the master domain manager and contains all the definitions for scheduling objects, such as jobs, job streams, resources, and workstations. It also holds statistics of job and job stream execution, as well as information about the user ID that created an object and when an object was last modified.

For more information about the types and versions of the supported relational database, see the IBM® Workload Scheduler documentation.

Related information

Designing your workload on page 110

Plans

Different types of plans you can access to manage the objects and the activities they contain.

A plan contains all jobs and job-related scheduling objects that are scheduled for a selected time interval. There are different types of plans based on the type of IBM® Workload Scheduler environment you are connected to.

Note: The only type of plan that you can access through the Dynamic Workload Console is the current plan.

The following plans are available:

Production plan (current plan)

A production plan (in distributed environment) or current plan (in z/OS environment) is the master control for all job scheduling activity planned for a user-defined time interval, named the *production period*. Scheduling object definitions stored in the database, such as jobs and job streams, become instances in the production plan, where they can be monitored and modified.

The production plan is created on the master domain manager and contains all the jobs and job streams that are scheduled to run within the production period together with their depending objects and all workstation definitions. You do have the opportunity to make additional changes to job definitions even once they are in the plan as long as the job has not yet started running. If the job has already run, you can update the definition and rerun the job. Only the job instance is modified. The job definition in the database remains the same. The production plan can be extended to cover future time intervals. Any job streams that did not complete successfully within the production period or that are either running or still waiting to be run, can be carried forward into the plan extension.

The production plan data is stored in the Symphony file and replicated in the database. With IBM® Workload Scheduler version 9.1, accessing this information from the Dynamic Workload Console directly queries the database thereby improving response times.

Preproduction plan

A preproduction plan is used to identify in advance the job stream instances and the job stream dependencies involved in a specified time period.

This improves performance when generating the production plan by preparing in advance a high-level schedule of the anticipated production workload.

The preproduction plan contains:

- The job stream instances to be run during the covered time period.
- The external dependencies that exist between the job streams and jobs included in different job streams.

Symnew plan

A Symnew plan is a temporary plan. It is an intermediate production plan that covers the whole time the new production plan that is being generated will cover. It is replaced by the production plan as soon as it starts.

Archived Plan

An archived plan is a copy of an old production plan that ran in the IBM® Workload Scheduler environment and that is now stored in the IBM® Workload Scheduler database.

Using this type of plan you can, for example, see the result of running a past production plan. The difference between using an archived plan and a forecast plan covering the same time interval is that an archived plan shows how the real production was based on the job and job stream processing results, while a forecast plan shows how the production was planned to be.

Trial plan

A trial plan is a projection of what a production plan would be if it covered a longer period of time. For example, if you generate a production plan that covers two days, but you want to know what the plan would be if it covered three days, you can generate a trial plan.

A trial plan is typically created to extend a production plan and to have an idea of future impacts on the scheduling environment. Therefore, if there is a valid production plan, the start time option is greyed out. 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 defined in the production plan, if available, or in the preproduction plan. Trial plans are based on the information contained either in the production or in the preproduction plan. If neither is available, a trial plan cannot be created.

Forecast plan

A forecast plan is a projection of what the production plan would be in a chosen time interval. For example, if you generate a production plan that covers two days and you want to know what the plan would be for the next week you can generate a forecast plan.

A forecast plan is typically created to anticipate and solve any kind of scheduling problems, therefore the start time is always enabled and it is a mandatory field.

Using this type of plan you can, for example, see how the production will be in a future time interval based on the job and job stream dependencies defined in the IBM® Workload Scheduler database. Based on this information, you can modify some information in the database, if needed, before extending the production plan.

When workload service assurance is enabled, it can calculate the predicted start time of each job in the job stream. You can enable and disable this feature using the **enForecastStartTime** global option. IBM® Workload Scheduler calculates the average run for each job based on all previous runs. For complex plans, enabling this feature could negatively impact the time taken to generate the forecast plan.

Note: Neither the trial nor the forecast plan takes into account any dynamic updates made to the Symphony file while the production plan is being processed. Therefore, all the job streams it contains are in one of the following states:

HOLD

If they are dependent on other job streams or if their start time is later than the plan start time.

READY

If they are free from dependencies and their start time has elapsed.

Related information

Display a graphical plan view on page 147

Graphical views in the plan on page 149

Selecting the working plan on page 191

Generating Trial and Forecast Plans on page 192

Workload Dashboard on page 161

Preproduction plan

The preproduction plan is used to identify in advance the job stream instances and the job stream dependencies involved in a specified time period.

This improves performance when generating the production plan by preparing in advance a high-level schedule of the predicted production workload.

The preproduction plan contains:

- The job stream instances to be 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.

A job or job stream that cannot start before another specific external job or job stream is successfully completed is named *successor*. An external job or job stream that must complete successfully before the successor job or job stream can start is named *predecessor*.

IBM Workload Scheduler automatically generates, expands, and updates, if necessary, the preproduction plan by completing the following steps:

- Removes any job stream instances that are in COMPLETE or CANCEL state.
- Selects all the job streams that are scheduled to run after the end of the current production plan and generates their instances.
- Resolves all job and job stream dependencies, including external follows dependencies, according to the defined matching criteria.

To avoid any conflicts, the database is locked during the generation of the preproduction plan and unlocked when the generation completes or if an error condition occurs.

At this stage only the job streams with the time they are scheduled to start and their dependencies are highlighted. All the remaining information about the job streams and the other scheduling objects (calendars, prompts, domains, workstations, resources, files, and users) that will be involved in the production plan for that time period are not included, but are retrieved from the database as soon as the production plan is generated.

When the production plan is extended, old job stream instances are automatically removed. The criteria used to remove these instances is based on the following conditions:

- The first job stream instance that is not in COMPLETE state when the new plan is generated (FNCJSI). This job stream instance can be both a planned instance, that is an instance added to the plan when the production plan is generated, and a job stream instance submitted from the command line during production using the **conman sbs** command.
- The time period (T) between the time FNCJSI is planned to start and the end time of the old production plan.

Assuming **T** is this time period, the algorithm used to calculate which job stream instances are removed from the preproduction plan is the following:

if T < 7

All job stream instances older than 7 days from the start time of the new production plan are removed from the preproduction plan; all job stream instances closer than 7 days to the start time of the new production plan are kept regardless of their states.

if T > 7

All job stream instances older than FNCJSI are removed from the preproduction plan; all job stream instances younger than FNCJSI are kept.

This algorithm is used to ensure that the preproduction plan size does not increase continuously and, at the same time, to ensure that no job stream instance that is a potential predecessor of a job stream newly added to the new preproduction plan is deleted.

For more information about how you can open the preproduction plan in view mode from the Dynamic Workload Console, see the section about how to view preproduction plan in the Dynamic Workload Console User's Guide.

Note: In the IBM Workload Scheduler for z/OS terminology the concept that corresponds to the preproduction plan is *long term plan* (LTP).

Related information

Display a graphical preproduction plan on page 193

Engine connections

You must define an engine connection to manage and monitor objects that run on a specific workstation in the IBM® Workload Scheduler environment in the network.

To manage and monitor objects you must connect the Dynamic Workload Console to IBM® Workload Scheduler distributed environments, z/OS environments, or both. You connect them by defining engine connections, and you can create as many engine connections as you need.

Note: Only Administration roles can manage engine connections.

When you create a new connection, you must provide an engine name, the host name, and port number of the engine you want to connect to.

If you connect to a IBM® Workload Scheduler for z/OS environment, the plan that you access is the current plan and the engine that you connect to is the controller workstation, which is the management hub of the IBM® Workload Scheduler for z/OS environment.

If you connect to a IBM® Workload Scheduler distributed environment, you can connect to:

The master domain manager workstation

A workstation acting as the management hub for the network. It manages all your scheduling objects. You can define and use different engine connections to the master domain manager, each accessing a different plan.

Backup master domain manager

A workstation that can act as a backup for the master domain manager when problems occur. It is a master domain manager, waiting to be activated. Its use is optional. This workstation must be installed as a master domain manager workstation.

Related information

Creating and managing engine connections on page 13

Event management

What the event management feature is and how to use it.

You can use the *event management* feature to launch a predefined set of actions in response to events that occur on the nodes where IBM® Workload Scheduler runs.

The main elements of event management are:

- Events on page 91
- Actions on page 92
- Event rules on page 93

You can use event management capabilities to:

- Create Event Rules
- Create and run Workload Events tasks

Events

An event represents a set of circumstances that match selected criteria. Events are divided into the following major categories:

Note: The topics listed below are .html files referenced by the PDF. Ensure you are connected to the Internet before clicking on the links.

IBM® Workload Scheduler object related events

All the events relating to scheduling objects such as jobs, job streams, workstations, and prompts.

This type of event is described in more detail in Workload Automation plan events.

Note: Any change performed on a workstation referenced in a rule is not reported in the rule. For example if you modify, update, or delete a workstation that is referenced in a rule, the rule ignores the change and continues to consider the workstation as it was when it was included in the rule.

File monitoring events

Events relating to changes to files and logs. File monitoring events are not supported on IBM i systems.

This type of event is described in more detail in File monitor.

Application monitoring events

Events relating to IBM® Workload Scheduler processes, file system, and message box. Application monitoring events are not supported on IBM i systems.

This type of event is described in more detail in Application Monitor.

SAP related events

These events are available only if you have installed IBM® Workload Scheduler for Applications and they are generated by external SAP systems.

This type of event is described in more detail in SAP Monitor.

Data set monitoring

These events are available only if you are using the agent for z/OS on IBM® Workload Scheduler.

This type of event is described in more detail in Data Set Monitoring.

Generic events

Events used to manage custom events sent by external applications. You can write an XML file to define a custom event. A schema is provided to validate your XML, as well as a basic event template that you can use as a starting point. For more information, see the schemas for generic events. Events of this category are:

- Changes in a resource of the operating system, such as processes and memory
- Email received

Actions

When one or more of the above events occurs, you can specify which actions to perform. Actions are divided into the following main categories:

Operational actions

Actions that cause a change in the status of one or more IBM® Workload Scheduler objects. Actions in this category include:

- Submitting jobs or job streams
- Submitting ad hoc jobs
- Replying to a prompt

This type of action is described in more detail in .

- Adding an application occurrence (job stream) to the current plan on IBM® Workload Scheduler for z/OS® in IBM® Workload Scheduler for z/OS end-to-end scheduling configurations.

This type of action is described in more detail in IBM Workload Scheduler for z/OS.

Notification actions

Actions such as:

- Sending email's or SMS. For details, see .
- Actions performed by running a command. This type of action is described in more detail in .
- Forwarding messages. For details, see .
- Opening an incident in ServiceNow incident management. For details see .

Event rules

Use *event rules* to associate one or more events to the response actions that you want to perform. When you create an event rule, you are actually creating an *event rule definition* in the database. While the event rule is in **Draft** status, it is not deployed to the IBM® Workload Scheduler. All new and modified non-draft rules saved in the database are periodically (by default every five minutes) found, built, and deployed by an internal process named rule builder. At this time they become active. Meanwhile, an event processing server, which is normally located in the master domain manager, receives all events from the agents and processes them. The updated monitoring configurations are downloaded to the agent and activated. The occurrence of an event rule that has performed the corresponding actions is called the *event rule instance*.

Related information

Event management configuration on page 13

Reports

How to customize and generate IBM® Workload Scheduler reports.

Create a *report* task to customize and generate IBM® Workload Scheduler reports, which you can then view, print, and save, in different kinds of output. Reports help you in many business-related activities, such as:

Tuning the workload on the workstations

- Workstation Workload Summary
- Workstation Workload Runtimes

Extract detailed information about the plan

- Planned Production Details
- Actual Production Details

Detect jobs with exceptions

- Job Run History
- Job Run Statistics

To generate your reports, you can use either the predefined reports or create your personalized reports created using Business Intelligent Report Tool (BIRT). For more information, see Reporting on page 205.

The following table shows the available reports within the Dynamic Workload Console and their details.

Table 8. Report types

Report Name	Description	Supported environment
Job Run History	Collects the historical job run data during a specified time interval. Use it to detect which jobs ended in error or were late. It also shows which jobs missed their deadline, long duration jobs, and rerun indicators for reruns.	Distributed and z/OS
Job Run Statistics Chart	Collects the job run statistics. Use it to detect success, error rates; minimum, maximum, and average duration; late and long duration statistics.	Distributed
Job Run Statistics Table	A report collecting the job run statistics, which returns output in table format. It is useful to detect success, error rates; minimum, maximum, and average duration; late and long duration statistics.	Distributed
Job Run Statistics	A report collecting the job run statistics, which returns output in table format. It is useful to detect success, error rates; minimum, maximum, and average duration; late and long duration statistics.	z/OS
Workstation Workload Summary	Shows the workload on the specified workstations. The workload is expressed in terms of number of jobs that ran on them. It helps for capacity planning adjustments (workload modelling and workstation tuning).	Distributed and z/OS
Workstation Workload Runtimes	Shows job run times and duration on the specified workstations. It helps for capacity planning adjustments (workload modelling and workstation tuning).	Distributed and z/OS

Table 8. Report types (continued)

Report Name	Description	Supported environment
Custom SQL	Allows you to create reports that best fit your business needs. You can specify an SQL query or import SQL scripts.	Distributed
Planned Production Details	Extracts information about planned production plans into either an XML or a CSV format, to be used respectively, with Microsoft Project and Microsoft Excel. This also allows users who do not know IBM® Workload Scheduler to access plan information in a familiar format.	Distributed
Actual Production Details	Extracts current plan information into either an XML or a CSV format, to be used, respectively, with Microsoft Project and Microsoft Excel. This also allows users who do not know IBM® Workload Scheduler to access plan information in a familiar format.	Distributed
Analysis Job Duration Estimation Error	A report that shows the average estimation error. It is useful to detect whether a job ends in frequent errors, ends in error, or if the jobs have unsatisfactory accuracy rates. You can then drill down to display all the jobs that are in that threshold and finally you can visualize charts that will help you to identify the jobs that have a high estimated error rate allowing you to intervene beforehand on those jobs.	Distributed and z/OS
Analysis Job Duration Standard Deviation	A report showing variances in job duration. The variance is calculated as a percentage and according to which variance level the jobs are they will be presented as follows: **High variability**, **Medium variability** or a **Low variability**. You can drill down to display all the jobs that are in that threshold which then returns output in a chart format. This report is useful to identify the run that had a greater duration.	Distributed and z/OS

The output of historical reports, which is extracted from the database, consists of the following main sections. The output of a Planned and Actual report is not structured because it is a file that must be opened with an external program.

Report header

Contains the report title, description, engine name, engine type, creation time, type, and the total number of the result set extracted.

Report of content

Contains a set of hyper-links to each section and subsection.

Report format

Depending on the kind of information you are processing, you can choose to view it in the most appropriate format. The report output can be in:

Table format

It shows information organized in rows and columns, in a CSV or HTML file.

Graphical format (HTML)

If you choose the graphical formats, depending on the report type, and on the information you choose to include, you can have data displayed in pie charts, bar charts, line charts, or tables.

Note: To see report output correctly, make sure that you configure your browser as follows:

- Allow pop-up windows.
- Remove any optional browser toolbar that you installed, if its settings prevent new windows from opening.
- To see CSV reports, configure the browser security settings to automatically prompt for file downloads.

Related information

Reporting on page 205

Regular expressions and SQL reports on page 236

Workload service assurance

Basic concepts about critical jobs and the critical path, which are fundamental to take advantage of this feature.

Workload service assurance is an optional feature that allows you to identify critical jobs and to ensure that they are processed in a timely manner.

When the workload service assurance feature is enabled, you can indicate that a job is critical and define a deadline by which it must be completed when you add the job to a job stream. Defining a critical job and deadline triggers the calculation of timings for all jobs that make up the critical network. The critical network includes the critical job itself and any predecessors that are defined for the critical job. When changes that have an impact on timings are made to the critical network, for example addition or removal of jobs or follows dependencies, the critical start times are automatically recalculated.

The critical network is constantly monitored to ensure that the critical job deadline can be met. When a critical network job completes, timings of jobs that follow it are recalculated to take account of the actual duration of the job. The system also acts automatically to remedy delays by prioritizing jobs that are actually or potentially putting the target deadline at risk. Some conditions that cause delays might require your intervention. A series of specialized critical job views, available on the Dynamic Workload Console, allow you to monitor critical jobs, display their predecessors and the critical paths associated with them, identify jobs that are causing problems, and drill down to identify and remedy problems.

Dynamic critical path

If a job is critical and must complete by the deadline set on the database you can mark it as a *critical job* thus specifying that it must be considered as the target of a *critical path*. The critical path consists of the critical job predecessors with the least *slack time*. In a critical job predecessor path, the slack time is the amount of time the predecessor processing can be delayed without exceeding the critical job deadline. It is the spare time calculated using the deadline, scheduled start, and duration settings of predecessors jobs. The calculation of critical path is performed dynamically. In this way, during daily planning processing, a critical path including the internal and external predecessors of the critical job is calculated, and a table of predecessors is cached (in the local memory for z/OS and on the master domain manager for distributed systems). Every time a predecessor of the critical job starts delaying, the scheduler *dynamically* recalculates the critical path, to check whether a new path, involving different jobs, became more critical than the path calculated at daily planning phase.

Hot list

The *hot list* contains a subset of critical predecessors that can cause a delay to the critical job because they are states such as error, late, fence (for distributed systems only), suppressed (for distributed systems only), or long duration. If these jobs do not complete successfully on time, they prevent the critical job from completing on time. Using the hot list view, you can quickly see which jobs need you to take appropriate recovery actions. Jobs included in the hot list are not necessarily also included in the critical path.

Related information

Limit the number of objects retrieved by queries on page 36

Using workload service assurance to monitor z/OS critical jobs on page 212

Processing and monitoring critical jobs

Automatic tracking and prioritizing of critical network jobs.

Workload service assurance provides automatic tracking and prioritizing of critical network jobs and online functions that allow you to monitor and intervene in the processing of critical network jobs.

Automatic tracking and prioritizing

To ensure that critical deadlines can be met, workload service assurance provides the following automated services for critical jobs and for predecessor jobs that form their critical networks:

Promotion

When the critical start time of a job is approaching and the job has not started, the promotion mechanism is used. A promoted job is assigned additional operating system resources and its submission is prioritized.

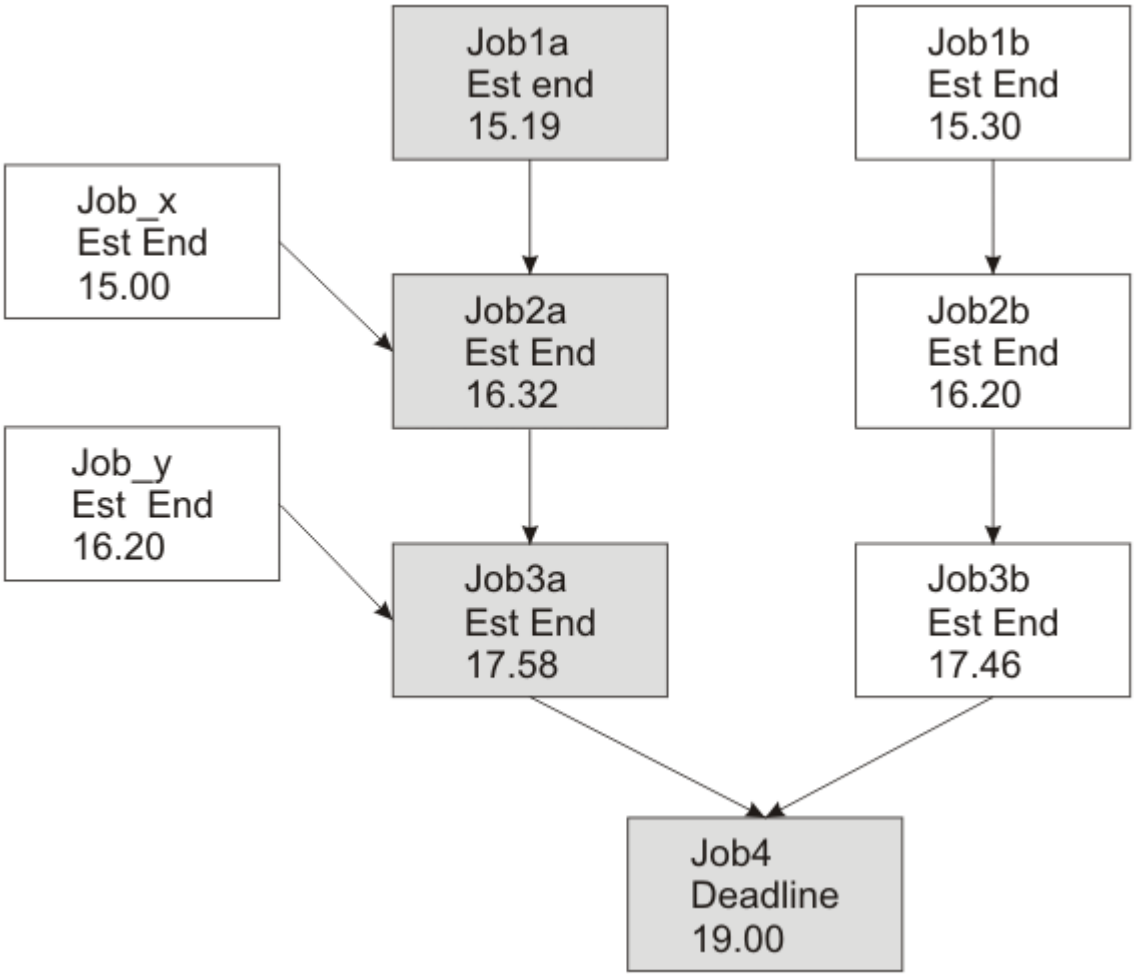
The timing of promotions is controlled by the global option `promotionoffset`. Promoted jobs are selected for submission after jobs that have priorities of "high" and "go", but before all other jobs. Prioritizing of operating system resources is controlled by the local options `jm promoted nice` (UNIX™ and Linux™) and `jm promoted priority` (Windows™).

Calculation of the critical path

The critical path is the chain of dependencies, leading to the critical job, that is most at risk of causing the deadline to be missed at any given time. The critical path is calculated using the estimated end times of the critical job predecessors. Working back from the critical job, the path is constructed by selecting the predecessor with the latest estimated end time. If the actual end time differs substantially from the estimated end time, the critical path is automatically recalculated.

Figure 8: Critical path on page 99 shows the critical path through a critical network at a specific time during the processing of the plan.

Figure 8. Critical path



At this specific time, the critical path includes Job3a, Job2a, and Job1a. Job3a and Job3b are the immediate predecessors of the critical job, job4, and Job3a has the later estimated end date. Job3a has two immediate predecessors, Job2a and Job_y. Job2a has the later estimated end time, and so on.

Addition of jobs to the hot list

Jobs that are part of the critical network are added to a hot list that is associated to the critical job itself. The hot list includes any critical network jobs that have a real or potential impact on the timely completion of the critical job. Jobs are added to the hot list for the one or more of the following reasons. Note that only the jobs that begin the current critical network, for which there is no predecessor, can be included in the hot list.

- The job has stopped with an error. The length of time before the critical start time is determined by the `approachingLateOffset` global option.
- The job has been running longer than estimated by a factor defined in the `longDurationThreshold` global option.
- The job has still not started, although all its follows dependencies have either been resolved or released, and at least one of the following conditions is true:

- The critical start time has nearly been reached.
- The job is scheduled to run on a workstation where the limit is set to zero.
- The job belongs to a job stream for which the limit is set to zero.
- The job or its job stream has been suppressed.
- The job or its job stream currently has a priority that is lower than the fence or is set to zero.

Setting a high or potential risk status for the critical job

A critical job can be set to the following risk statuses:

High risk

Calculated timings show that the critical job will finish after its deadline.

Initially, estimated start and end times are used. As jobs are completed, timings are recalculated to take account of the actual start and end times of jobs.

Potential risk

Critical predecessor jobs have been added to the hot list.

Online tracking of critical jobs

The Dynamic Workload Console provides specialized views for tracking the progress of critical jobs and their predecessors. You can access the views from the following sources:

- Workload Dashboard: dedicated widgets to monitor the critical status: high risk, no risk, potential risk
- **Monitor Critical Jobs** tasks: lists all critical jobs for a selected engine with the possibility to run actions against the results. View jobs with a high risk level along a horizontal time axis.
- What-if analysis view: from a Gantt view, highlight the critical path and show the impact on critical jobs.

late or in high risk state, right-click the job in the list of results and select **What-if** from the table toolbar to open the What-if Analysis and view them in a Gantt chart for further investigation.

Planning critical jobs

Planning critical jobs.

Workload service assurance provides the means to identify critical jobs, define deadlines, and calculate timings for all jobs that must precede the critical job.

If it is critical that a job must be completed before a specific time, you can flag it as critical when you add it to a job stream using the Workload Designer functions on the Dynamic Workload Console. You can define the deadline either at job or job stream level.

Jobs can also be flagged as critical by including the **critical** key word on the job statement when you create or modify a job stream using the **composer** command line.

When you run the command to include the new job in the production plan, all jobs that are direct or indirect predecessors of the critical job are identified. These jobs, together with the critical job itself, form a `critical network`.

Because timing of jobs in the critical network must be tightly controlled, Time Planner calculates the following timing benchmarks for each critical network job:

Critical start

Applies to distributed systems only and represents the latest time at which the job can start without causing the critical job to miss its deadline.

Critical start times are calculated starting with the deadline set for the critical job and working backwards using the estimated duration of each job to determine its critical start time. For example, if the critical job deadline is 19:00 and the estimated duration of the critical job is 30 minutes, the critical job will not finish by the deadline unless it starts by 18:30. If the immediate predecessor of the critical job has an estimated duration of 20 minutes, it must start at latest by 18.10.

Note: Only the deadline of the critical job is considered when calculating critical start times for jobs in the critical network. If other jobs have deadlines defined, their critical start times might be later than their deadlines.

Earliest start

Represents the earliest time at which a job in the critical network can start, taking into consideration all dependencies and resource requirements.

Estimated start times are calculated starting with the earliest time at which the first job or jobs in the critical network can start and working forward using the estimated duration of each job to estimate the start time of the job that follows it.

Estimated start and end time

For the initial calculations, these values are set to the planned start and end time. They are subsequently recalculated to take into consideration any changes or delays in the plan.

Estimated duration

The estimated duration of a job is based on statistics collected from previous runs of the job. If the job has never run before, a default value of one minute is used. Take this into account when considering the accuracy of calculated timings for critical job networks that include jobs running for the first time.

The timings for each job in the critical network are added to the `Symphony` file, which includes all the plan information and is distributed to all workstations on which jobs are to be run.

As the plan is run, Plan Monitor monitors all critical networks: subsequent changes to the critical network that affect the timing of jobs trigger the recalculation of the critical and estimated start times. Changes might include manual changes, for example, releasing dependencies or rerunning jobs, and changes made automatically by the system in response to a potential or actual risk to the timely completion of the critical job.

Specific views for critical jobs and their predecessors, available from the Dynamic Workload Console, allow you to keep track of the processing of the critical network. The views can immediately identify problems in your planning of the critical job. For example, if the estimated start time of a job in the critical network is later than the critical start time, this is immediately signaled as a potential risk to the critical job.

IBM Workload Scheduler for SAP

Use IBM Workload Scheduler with SAP support

With SAP support, you can use IBM Workload Scheduler to do the following tasks:

- Use IBM Workload Scheduler standard job dependencies and controls on SAP jobs.
- Create SAP jobs using the IBM Workload Scheduler interface.
- Schedule SAP jobs to run on specified days and times, and in a defined order.
- Define inter-dependencies between SAP jobs and jobs that run on different platforms.
- Define the national language support options.
- Use the SAP Business Warehouse Support function.
- Customize job execution return codes.
- Use SAP logon groups for load balancing and fault-tolerance.
- Work with SAP variants and placeholders.
- Use Business Component-eXternal Interface Background Processing (BC-XBP 2.0) interface support to:
    - Intercept jobs
    - Track child jobs
    - Keep all job attributes when you rerun a job
    - Raise events

Note: For more information about SAP, see the topic about the intergation with SAP in *IBM Workload Automation: Scheduling Job Integrations with IBM Workload Automation*.

Scheduling process for the SAP extended agent

IBM Workload Scheduler launches jobs in SAP by using jobs defined on the following workstations that support the r3batch access method:

- A IBM Workload Scheduler extended agent workstation. A workstation that is hosted by a fault-tolerant agent or master workstation.
- A dynamic agent workstation.
- A dynamic pool.
- A z-centric workstation.

These supported workstations use the r3batch access method to communicate with the SAP system. The access method is used to pass SAP job-specific information to predefined SAP instances. The access method uses information provided in an options file to connect and launch jobs on an SAP instance.

Note: For more information about this, see the section about the integration with SAP in *IBM Workload Automation: Scheduling Job Integrations with IBM Workload Automation*.

You can define multiple agent workstations to use the same host, by using multiple options entries or multiple options files. Using the SAP agent name as a key, r3batch uses the corresponding options file to determine which instance of SAP will run the job. It makes a copy of a template job in SAP and marks it as able to run with a start time of start immediate. It then monitors the job through to completion, writing job progress and status information to a job standard list found on the host workstation.

Chapter 7. Creating and editing items in the Database

This section describes creating and editing items in the database.

In this section you can find information about creating and editing scheduling items from the Dynamic Workload Console by using the Graphical Designer, the right place to start designing your environment and define the automation of your business processes.

By using the Graphical Designer, you have full control over your orchestration: you can create job streams, add new jobs, create internal or external dependencies, add join conditions, triggers and constraints to your job streams. You can save items in the database, edit definitions, reuse existing items and perform any operation useful to make your orchestration as smooth as possible.

Designing your scheduling environment

To begin working with IBM® Workload Scheduler you must design your scheduling environment.

About this task

The scheduling environment is composed of the following items on distributed engines:

- Workstations
- Domains

To design your environment, you need to create workstations and domains. Depending on the business needs of your organization and the complexity of your network, you can decide to have a hierarchical domain structure.

To create a scheduling environment perform the following actions.

Creating a workstation

About this task

You can create multiple workstation definitions.

For more information about the main workstation types and their attributes, see the section about workstation definition in the *User's Guide and Reference*.

To add a workstation definition to the database and to assign it to a domain, perform the following procedure.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. From the **Design** menu, click **Graphical Designer** page.
2. Select a engine from the list.
3. Go to the **Assets** tab, click the **+** icon, and from the drop-down menu select **Workstation**.
4. In the workstations properties panel, specify the attributes for the workstation you are creating.

   a. In a distributed environment, depending on the type of workstation you select, some attributes are mandatory.
   b. In a z/OS environment, specify the workstation attributes using the General, Resources, and Open Time Intervals tabs as appropriate. Depending on the type of workstation you select, some attributes are mandatory.
5. To assign the workstation to an existing domain:
   a. Go to the **Domain** field.
   b. Select the domain from the available domains, or click **View all** and search for the required domain.
   c. Click the **Add** button at the end of the properties panel.

Results

The workstation is now added to the database.

Note: You can add workstation definitions to the database at any time, but you must run JnextPlan -for 0000 to add the workstation definition to the plan, so that you can run jobs on this workstation. For dynamic agent workstation definitions, if you have the `enAddWorkstation` global option set to `"yes"`, the workstation definitions are automatically added to the plan after they are defined in the database.

What to do next

You can also edit and manage workstation and domain definitions.

To edit a workstation definition in the database and to assign it to a domain, perform the following steps.

1. From the Graphical Designer page, select the **Assets** tab, expand the **Workstation** section and search for the workstation definition that you want to edit. You can also use the search bar to find the workstation definition to edit.
2. Click on the three-dot menu next to the workstation definition and click edit.
3. Change workstation information in the properties panel and click **Save**.

To add a new domain definition, perform the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. Select an engine from the list.
3. Go to the **Assets** tab, click the **+** icon and select **Domain**.
4. In the properties panel that appears on the right, specify domain information.
5. Click **Add** at the end of the properties panel.

Creating a pool of agents

Creating distributed workstation definitions in the IBM® Workload Scheduler database.

About this task

You can define and schedule dynamic jobs to perform application specific operations, such as database, file transfer, Java, and Web Services operations. You can customize the sample files provided with the product to match the requirements of your environment.

To run these job types, you can use dynamic agents, a workstation type that you create by running the related installation process. The dynamic agents are automatically created and registered at installation time. You can also organize the dynamic agents into groups, called pools or dynamic pools.

To add this kind of workstation definition to the database and to assign it to a domain, perform the following steps.

Note: For all the details about options and fields displayed in the panels, expand the contextual help placed at the bottom of the properties panel.

1. From the **Design** menu, click **Graphical Designer** page.
2. Select an engine from the list and create a new **Workstation**.
3. Go to the **Assets** tab, click the **+** icon, and from the drop-down menu select **Workstation**.
4. In the workstations properties panel, in general information, specify the attributes for the pool of dynamic agents you are creating. In the **Type** drop-down menu, select **Pool** or **Dynamic Pool**, depending on the set of dynamic agents you are defining.
   Choose from:
   - Select **Pool** to define a cluster of dynamic agents with similar hardware or software characteristics to submit jobs to. Then, in the **Pool** section of the properties panel that displays the dynamic agents that belong to the pool, click **Add worskation** to add new dynamic agents or select the bin icon to remove unwanted dynamic agents.

     When dynamic agents are added to a pool, the agent is registered with the pool and this registration is written to the `pools.properties` file located in *`TWS_home`*`/ITA/cpa/config`. As an alternative method, you can add dynamic agents to a pool by directly editing this file.
   - Select **Dynamic Pool** to define a set of dynamic agents that is dynamically defined based on the resource requirements you specify. In **Requirements** field, you can specify the requirements necessary for running your jobs. All your selections produce an XML file, which is used to select a workstation with the characteristics you require, to run Workload Broker jobs. When you provide the requirements, you specify a list of workstation candidates to be included in the **Dynamic Pool** of dynamic agents and the preferred order in which they must be considered. You also specify the best criteria, which is used to change the workstation (workload balance, CPU utilization, or its use of logical resources).
5. Optionally, you can associate the new pool to a variable table.
6. Specify the Workload Broker hosting the workstation.

Graphical Designer overview

The Graphical Designer presents drag and drop functionalities that enable you to add elements from a palette to a canvas workspace.

To open the Graphical Designer from the Dynamic Workload Console, open the **Design** menu and click Graphical Designer.

Graphical Designer interface

The Graphical Designer interface is composed of two main elements:

- **Palette**
- **Workspace**

Palette

In the palette you can find everything you need to create and deploy functioning job streams. The palette is divided in two tabs:

Blocks

The **Blocks** tab houses job streams, jobs and join conditions. Job streams can be dragged and dropped in the workspace, while the other items can only be dropped inside job streams.

In this tab you can also find all the available job types divided in categories. You can add any number of jobs to job streams.

The **Most used** section displays the most frequently used items.

Assets

In this tab you can specify item definitions, add defined items to a job stream, or edit definitions saved in the database. You can define new assets by clicking on the add icon **+** and selecting an asset from the drop-down menu.

The items you can find in the **Assets** tab are:

- Calendar
- Credentials
- Domain
- Folder
- Job definition
- Job stream
- Prompt
- Resource
- Run cycle group
- Variable table
- Workstation
- Workstation class

You can use the search bar to find and reuse previously defined assets in your current workspace.

Note: When adding blocks and assets to the workspace, you can specify their properties in their contextual properties panel.

Workspace

The workspace is the canvas on which you can place, design and connect your job streams.

Navigation

The workspace offers a flexible navigation system. You can pan freely across the canvas, zoom in and out, and use the fit-to-screen feature to adjust the view to see all your blocks and assets at the same time. A mini-map provides an overview of the entire workspace, highlighting the current visible area.

Controlling jobs and job streams processing

To control the processing of jobs and job streams in your environment, you can add triggers, dependencies, and constraints.

Triggers

You can associate three types of triggers to a job stream:

Service

You can link a service to a job stream, and publish the service on the Self-Service Catalog.

Run cycle

A run cycle specifies the days and times when a job stream is scheduled to run.

Excluding run cycle

An excluding run cycle specifies the days and times when a job stream must not run. Excluding run cycles take precedence over run cycles.

To add a trigger to your job stream, click the **Add triggers** button in the job stream box in the workspace, then click **Add new** and select the trigger from the drop-down menu. Then, specify the required information in the property panel.

Dependencies

You can create internal or external dependencies between job streams or jobs in your workspace.

Job streams and jobs are represented visually as rectangular blocks in the workspace, each featuring four connection points. To establish dependencies between these blocks, drag connecting arrows from one block's connection point

to another. Arrows are visually represented differently depending on whether the dependency is internal or external:

- Internal dependencies are represented by a single arrowhead
- External dependencies are represented by a double arrowhead

When you create an internal dependency, a panel where you can specify dependency information appears, and you can eventually change the dependency type from internal to external.

Join conditions

To establish a set of dependencies between job streams or jobs, you can define a join condition. This condition specifies that a job stream or job must wait for the join condition to be met before it can start running. The join condition is satisfied only when one or more dependencies have been satisfied.

You can drag a join condition from the **Blocks** tab and drop it in a job stream or in a job. Then, connect the join condition block to the items that must be part of that join condition. In the properties panel, specify a name for the join condition and optionally define the minimum number of predecessor dependencies required to satisfy it. If you do not specify a value, all predecessor dependencies must be satisfied.

Constraints

To control the job or job stream processing, you can add the following constraints:

- Prompt
- Resource
- File
- Internetwork dependency

Click the **Add constraints** button on the job or job stream box in the workspace, then click **Add new** and select the type of constraint from the drop-down menu. Then, specify the required information in the property panel.

For more information about dependencies, see Using dependencies to control job and job stream processing on page 185.

Saving and exporting

The items you defined in a workspace can be saved to the database by selecting **Deploy**.

Note: After the successful deployment of a workspace, you are prompted to decide whether you want to delete it or not. Deleting a deployed workspace clears the canvas, and all the items previously defined in the workspace are safely saved to the database and can be used in future workspaces.

You can export workspaces as JSON files or upload previously exported workspaces to keep working on them. You can also save the workspace visuals as Portable Network Graphics (PNG) files.

Note: If you close the Graphical Designer page before deploying a workspace, you can continue to edit your job streams when you log in again, but changes are not saved in the database until you complete the deployment.

Designing your workload

Create and edit workload definitions in the database.

Before you begin

To create and edit scheduling items, open the Graphical Designer, by performing the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. In the displayed panel, specify the engine connection you want to use. Only the categories of items supported by the engine that you selected are available.

Important: To ensure compatibility, the Dynamic Workload Console version installed must always be equal to or greater than the version of any engine it connects to.

You can open multiple occurrences of the Graphical Designer. The following scenarios are supported:

- Same user connected to multiple engines
- Multiple users connected to the same engine.

Creating a job stream

To start designing your environment, you can drag a job stream from the **Blocks** tab, and drop it on the workspace. A property panel appears, and you can specify information about the job stream. In the job stream box on the workspace, you can find the **Add triggers** and the **Add constraints** button; by clicking on them, you can add items that aim at controlling the processing of jobs and job streams in your environment.

Creating a job

There are two types of job that you can create from the Graphical Designer:

Job definitions

From the **Assets** tab, you can create a new job definition by clicking on the add icon **+** and selecting **Job definition** from the drop-down menu. After you have defined job information in the properties panel, you can save the job definition in the database by clicking **Add**.

Embedded jobs

From the **Blocks** tab, you can create an embedded job by dragging and dropping a job element into a job stream that is displayed on the workspace. In the properties panel, you must untoggle the **Use job definition** button, select the job type, and then specify information about the job. An embedded job can only be visible when opening the job stream that contains the embedded jobs, therefore you cannot find embedded jobs listed inside the **Assets** tab.

Adding jobs to a job stream

There are two ways of adding a job to a job stream from the Graphical Designer:

From the Blocks tab

You can drag and drop a new job into the job stream, and decide whether to reference an existing job definition or create an embedded job by toggling the **Use job definition** button.

From the Assets tab

You can search an existing job definition, then drag and drop it into the job stream.

Creating assets and adding them to a job stream

You can create assets from the **Assets** tab by clicking the add icon **+**, and selecting an asset from the drop-down menu. You can specify information about the selected asset in the properties panel, and then click **Add** to save the item in the database.

After you have created an asset, you can drag and drop it inside a job stream.

Managing assets

From the **Assets** tab, you can search for the asset that you want to manage, and then click on the three-dot menu next to the asset definition. The following actions are available:

- Edit
- Delete
- Duplicate
- Run

Related information

Scheduling objects on page 57

Managing calendar definitions

This topic shows how to manage a calendar.

About this task

The purpose of the following scenario is to show how to create a calendar in the context of an example.

Angela is a IBM® Workload Scheduler user and she is responsible for internal communication in her Company. Every Monday, Angela sends a newsletter that contains business updates and news to employees. A big marketing campaign is planned for Tuesday, November 26, 2024. To celebrate the occasion, Angela decides to send a newsletter dedicated to the campaign, with interviews to managers, board members and everyone involved in the making, exceptionally on that Tuesday. Therefore, she needs to create a calendar to make the `wf_newsletter_1` job stream run during that particular day that the run cycle does not define.

1. Create a calendar:
    a. From the Graphical Designer, select the **Assets** tab and click on the add **+** icon.
    b. From the drop-down menu, select **Calendar**.
    c. In **Name**, type the name **Campaign_celebration**.
    d. In **Date selection**, select the day **26th of November**.
    e. Click **Add**.
2. Add the calendar to the `wf_newsletter_1` job stream:
    a. Using the map, go to the workspace area where the `wf_newsletter_1` job stream is placed.
    a. From the **Assets** tab, in the search bar, type **Campaign**.
    b. Expand the **Calendar** menu, and then drag the **Campaign_celebration** calendar into the `wf_newsletter_1` job stream in the workspace.
3. Deploy the workspace.

Results

You created a calendar that makes the job stream run on November 26, 2024 as an exception.

Managing a credentials definitions

This topic shows how to manage credentials definition.

About this task

The purpose of the following scenario is to create credentials for an existing job definition.

Alfred works in the IT department, and he is authorized to perform different operations. John, a colleague, is an IBM® Workload Scheduler user, and Alfred has asked him to create some job definitions that can automatically perform some of Alfred's activity in the form of REST API calls. To simplify the creation or the editing of job definitions, John decides to create credentials related to Alfred. In this way, John can create valid job definitions without knowing Alfred's password.

You can find the steps that John follows to create the credentials with a user name **Alfred** and how John can use these credentials on an existing job definition. For information about creating job definitions, see Managing job definitions on page 114.

1. Create the credentials:
    a. From the Graphical Designer page, select the **Assets** tab and click the add **+** icon.
    b. From the drop-down menu, select **Credentials**.
    c. In **General information**, in **Username**, type **Alfred**.
    d. In **Secret**, enter the password.
    e. Click **Add**.
2. Add the user to an existing job definition that performs REST API calls:
    a. From the Graphical Designer page, select the **Assets** tab.
    b. In the search bar, type the name of the existing job definition of type **RESTful Web services** named **API_task**.
    c. Click on the overflow menu and select **Edit**. A property panel appears on the right side of the page.
    d. Go to the **Credentials** section, then enter the user name **Alfred** that you previously created.
    e. In **Password**, select the **User** checkbox.
3. Save.

Results

You created the credentials for a user and referenced these credentials on an existing job definition.

Managing folder definitions

You can manage folders to define a line of business.

About this task

The purpose of the following scenario is to show how to create a folder for mortgage loan openings.

Sarah works in a bank that uses IBM® Workload Scheduler to automate processes. Every time a mortgage loan request is approved, information regarding their opening must be stored in folders. The mortgages are organized in different folders that are named according to the year of the mortgage loan was opened, and also in sub-folders that are named after the account holder that has requested the mortgage loan. In January, a new mortgage loan has been opened; therefore, Sarah needs to create a folder for the current year, 2024, and a a sub-folder with the surname of the account holder, **Anderson**.

1. Create a folder named **loans_2024**, that will be used to store mortgage loan openings in 2024:
    a. In the Graphical Designer page, from the **Assets** tab, click the add icon **+**.
    b. From the drop-down menu, select **Folder**.
    c. In **General information**, in **Parent**, select a parent folder from the drop-down menu.
    d. In **Name**, type **loans_2024**.
    e. Click **Add**.
2. Create a sub-folder of the **loans_2024** folder named **Anderson**:
    a. In the **Assets** panel, click **Add new** and choose **Folder**.
    b. In **General information**, in **Parent**, select **loans_2024** from the drop-down menu.

c. In **Name**, type the name **Anderson**.
d. Click **Add**.

Results

You defined two folders with different names according to the year in which the mortgage loan was opened and the surname of the account holder.

Managing job definitions

The purpose of the following scenario is to show how to create a job definition in the context of an example.

About this task

Jobs can either be an embedded job, or they can reference a job definition. For information about how to define an embedded job from the UI, see Creating embedded jobs on page 118.

David is an IBM® Workload Scheduler user who needs to upgrade the Java version on the existing backup server named **BK_server**. To automate the process, David decides to create a job stream that references a job definition. David prefers to reference a job definition instead of creating an embedded job because he needs to use the same job definition in other job streams and he wants to avoid to duplicate the definition. In this way, if David needs to modify any parameter, he can modify just one job definition and every job stream that references that job definition will be affected by the change.

To create a job definition named **UPGRADE_JAVA_TEMPLATE** and then reference the job definition inside the existing **UPGRADE_JAVA** job stream, perform the following steps:

1. Create a job definition named **UPGRADE_JAVA_TEMPLATE** and enter the command that performs the Java upgrade procedure.
    a. From the Graphical Designer page, select the **Assets** tab and click the add icon **+**.
    b. From the drop-down menu, select **Job definition**.
    c. In the search bar, type **Executable**, then select the executable job type and click **Next**.
    d. In **General information**, in **Workstation**, select the existing workstation **BK_server**.
    e. In **Folder**, select **IT_operations**.
    f. In **Name**, type `UPGRADE_JAVA_TEMPLATE`.
    g. In **Task**, from the drop-down menu, select **Inline script**.
    h. In **Command text or script name**, enter `apt install --only-upgrade openjdk-11-jre-headless -y`.
    i. Click **Add** to add the job definition to the database.
2. Reference the **UPGRADE_JAVA_TEMPLATE** job definition into the existing **UPGRADE_JAVA** job stream.
    a. From the **Assets**, use the search bar to find the `UPGRADE_JAVA_TEMPLATE` job definition.
    b. Drag the `UPGRADE_JAVA_TEMPLATE` job definition into the `UPGRADE_JAVA` job stream.
    c. Click **Deploy** to deploy the workspace.

    Result

    You created a job definition to upgrade the Java version on the backup server named **BK_server** and you referenced the `UPGRADE_JAVA_TEMPLATE` job definition into a job stream.

Related information

Job on page 58

Status description and mapping for distributed jobs on page 222

Status description and mapping for z/OS jobs on page 225

Limit the number of objects retrieved by queries on page 36

Workstation on page 50

Variable table on page 84

Run cycle on page 62

Dependencies on page 69

Prerequisite steps to create job types with advanced options

How to define a new job definitions using the Dynamic Workload Console.

About this task

Perform the following steps before you define and schedule job types with advanced options.

1. **Install a number of dynamic agents and add the Java run time**

   To install dynamic agents, run the installation program. You can install the dynamic agent during the full installation of IBM Workload Scheduler or in a stand-alone installation of just the agent. During the installation, you have the option of adding the Java run time to run job types with advanced options, both those types supplied with the product and the additional types implemented through the custom plug-ins.

   Follow the installation wizard to complete the installation.

   For a description of the installation parameters and options, see the section about installation options in *IBM Workload Scheduler: Planning and Installation*.

2. **Organize the dynamic agents in pools and dynamic pools.**

   Pools and dynamic pools help you organize the environment based on the availability of your workstations and the requirements of the jobs you plan to run.

   a. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.

   b. Select a distributed or z/OS engine.
   The workstations you can create vary depending on the engine type you select.

   c. Select the workstation type you want to create.

- To create a pool, define the dynamic agents you want to add to the pool and the workload broker workstation where the pool is hosted.
- To create a dynamic pool, specify the requirements that each dynamic agent must meet to be added to the dynamic pool.

3. **Grant the required authorization for defining job types with advanced options.**

   The IBM Workload Scheduler administrator has to grant specific authorizations in the security file to allow the operators to create job types with advanced options.

   Choose from:

   - In the distributed environment, perform the following steps:
     a. Navigate to the *TWA_home*/TWS directory from where the **dumpsec** and **makesec** commands must be run.
     b. Run the **dumpsec** command to decrypt the current security file into an editable configuration file.

        For more information, see the section about **dumpsec** in IBM Workload Scheduler Administration.

     c. Add display and run access to the workstation, as follows:
        - If the operation is performed on the IBM Workload Scheduler Connector, display and run access is required on the CPU corresponding to the workstation where the job is created.
        - If the operation is performed on the workstation where the job runs, display access is required on the workload broker workstation.

        For more information, see the section about configuring the security file in IBM Workload Scheduler Administration.

     d. Close any open **conman** user interfaces by using the **exit** command.
     e. Stop any connectors on systems running Windows operating systems.
     f. Run the **makesec** command to encrypt the security file and apply the modifications.

        For more information, see the section about **makesec** in IBM Workload Scheduler Administration.

     g. If you are using local security, the file is immediately available on the workstation where it has been updated.
        i. If you are using a backup master domain manager, copy the file to it.
   - In the z/OS environment, perform the following steps:
     a. Define the fixed resource that owns the subresource and give universal read access to it:

        ```
        RDEFINE (CLASS_NAME) FIXED_RESOURCE UACC(READ)
        ```

     b. Give user *USER_ID* update access to the *FIXED_RESOURCE* fixed resource:

        ```
        PERMIT FIXED_RESOURCE ID(USER_ID) ACCESS(UPDATE) CLASS(CLASS_NAME)
        ```

     c. Define a RACF resource, JSO*RACF_RESOURCE*, to RACF and give universal read access to it:

        ```
        RDEFINE (OPCCLASS) JSORACF_RESOURCE UACC(READ)
        ```

        JSO is the 3-character code that RACF uses for JS.OWNER.

d. Give user *USER_ID* update access to JSO*RACF_RESOURCE*:

```
PERMIT JSORACF_RESOURCE ID(USER_ID) ACCESS(UPDATE) CLASS(CLASS_NAME)
```

4. Define the job types with advanced options as described in Managing job definitions on page 114.

   You can define job types with advanced options also using the **composer** command.

   For more information, see the section about job definition in the IBM Workload Scheduler User's Guide and Reference

   You can define job types with advanced options also using the statement.

Related information

Job on page 58

Variable table on page 84

Run cycle on page 62

Dependencies on page 69

Managing job stream definitions

You can create a job stream, add a run cycle, and add jobs from the Graphical Designer.

About this task

The purpose of the following scenario is to show how to create a job stream in the context of an example.

The scope of the scenario is to create a job stream named “PAYROLL_PROC_24” inside the **2024** folder that monthly creates and uploads payrolls for the year 2024 into the payroll management system. The job stream references the existing calendar **monthly_payroll**, and runs on the workstation **MDM_production**.

1. Create a job stream and define general information about it:
   a. From the **Design** menu, click **Graphical Designer** page, and then select the engine.
   b. From the **Blocks** tab, drag a job stream into the workspace.
   c. In the properties panel that appears on the right, in **Folder**, select **2024**.
   d. In **Name**, type `PAYROLL_PROC_24`.
   e. In **Workstation**, select the existing workstation **MDM_production**.
2. Add a run cycle to the job stream:
   a. From the workspace in the canvas, select the `PAYROLL_PROC_24` job stream.
   b. Click **Add triggers**, select **Add new**, and then select **Add run cycle** from the drop-down menu. A properties panel appears on the right.
   c. In **Name**, type `RC1`.
   d. In **Rule**, select **Calendar**.
   e. In **Calendar**, select the existing **monthly_payroll** calendar.
3. Add the existing jobs to the job stream:

a. From the **Assets** tab, drag the following jobs and drop them into the PAYROLL_PROC_24 job stream:
   - OPERATOR_INTERVENTION
   - DB_BACKUP
   - ABSENCES
   - RESTART_DB
   - RESTART_WAS
   - TEMP_CLEANUP
   - ABSENCES_2

Results

You created a job stream that monthly creates and uploads payrolls for the year 2024 into the payroll management system.

Related information

Domain on page 54

Job stream on page 59

Status description and mapping for distributed job streams on page 227

Status description and mapping for z/OS job streams on page 229

Limit the number of objects retrieved by queries on page 36

Job on page 58

Managing job definitions on page 114

Creating embedded jobs

The purpose of the following scenario is to show how to create an embedded job in the context of an example.

About this task

Jobs can either be an embedded job, or they can reference a job definition. For information about how to create a job definition from the Dynamic Workload Console, see Managing job definitions on page 114.

David is an IBM® Workload Scheduler user who needs to upgrade the Java version on the existing backup server named **BK_server**. To automate the process, David decides to create a job stream that contains an embedded job. David prefers to create an embedded job instead of referencing a job definition because he only needs to use the job in that job stream.

To create an embedded job named **UPGRADE_JAVA_VERSION** inside the **UPGRADE_JAVA** job stream, perform the following steps:

1. Create a job stream named **UPGRADE_JAVA** and define general information about it.
   a. From the **Blocks** tab in the Graphical Designer page, drag a job stream and drop it into the workspace.
   b. In **General information**, in **Folder**, select a folder.

c. In **General information**, in **Workstation**, select the existing workstation **BK_server**.
d. In **General information**, in **Name**, type **UPGRADE_JAVA**.

2. Create an embedded job named **UPGRADE_JAVA_VERSION** into the **UPGRADE_JAVA** job stream.
    a. From the **Blocks** tab in the Graphical Designer page, drag an executable job and drop it into the **UPGRADE_JAVA** job stream.
3. Define information about the **UPGRADE_JAVA_VERSION** embedded job:
    a. In **General information**, in **Workstation**, select **BK_server**.
    b. In **General information**, in **Name**, type `UPGRADE_JAVA_VERSION`.
    c. In **General information**, in **Description**, type `Java upgrade to version 11`.
    d. In **Task**, in **Command text or script name**, type `apt install --only-upgrade openjdk-11-jre-headless -y`.
4. Click **Deploy** to deploy the workspace.

Results

You created an embedded job to upgrade the Java version on the backup server named **BK_server**.

Managing variable table definitions

The purpose of the following scenario is to show how to create a variable table definition in the context of an example.

FineBusiness is a new B2B company that sells mobile devices. The company is creating a team dedicated to solve any technical issue that clients might face. To automate the process, FineBusiness IT has asked Louis, who is an IBM® Workload Scheduler user, to create a job stream that forwards customer requests to the technical team e-mail. Louis has already created a job stream named **customer_requests**. Now, he has to create a variable table that sends e-mails to the technical team during working days, and add that variable table to the run cycle defined into **customer_requests**.

The existing job stream **customer_requests** references a job definition of executable type. The job definition definition contains the following command:

```
echo "This is message body" | mail -s "This is Subject" ${var.<mail>}
```

To create a variable table to send the customer e-mails to the technical team, proceed as follows:

1. Create a variable table:
    a. From the Graphical Designer page, select the **Assets** tab.
    b. Click the add icon **+** and select **Variable table**
    c. In **General information**, type the name **workingdays_requests**.
    d. In **Variables**, click on the add symbol (+).
    e. Type the name **mail** and the value **technical_team@mail.com**.
    f. Save.
2. Edit the existing job stream **customer_requests**:
    a. From the workspace area, select the `customer_requests` job stream and then click the **Add trigger** icon.
    b. Click **Add new**, and then select **Run cycle**.
    c. In **General information**, type the name **working_days**.
    d. In **Variable table**, search for and select the **workingdays_requests** variable table.

e. In **Rule**, from the **Type** drop-down menu, select **Calendar**.
f. Save.

You created a variable table to send the requests of the customers to the technical team during working days.

Managing workstation definitions

You can create a workstation from the Graphical Designer.

About this task

The scope of this scenario is to create a workstation of type **Pool**, set on the **America/New_York** time zone.

1. From the Graphical Designer page, select the **Assets** tab and click the add icon **+**.
2. From the drop-down menu, select **workstation**.
3. In **General information**, from the **Type** drop-down menu, select **Pool**.
4. In **Folder**, select one of the available folders.
5. In **Name**, type the name of the workstation as **NEWYORK1**.
6. From the **Time zone** drop-down menu, select **America/New_York**.
7. From the **Operating system** drop-down menu, select **Unix**.
8. In **Host or Broken workstation**, select the workload broker workstation.
9. In **Pool**, click **+ Add workstation** and select the the dynamic agent that you want to add to the pool; you can add multiple members. Then, click **Select** to confirm the selection.
10. Click **Add**.

Results

You have created a Pool workstation, added members to the pool and set the time zone to **America/New_York**.

Specifying the parameter format for date, time, and time stamp

When defining reports either using the Dynamic Workload Console, or **composer**, specify parameters of type date, time, and time stamp, using a specific syntax.

The following table shows the syntax you must use when defining reports containing date, time, and time stamp formats as parameters.

Table 9. Examples to use for parameters of date, time, and time stamp formats

Prompt type	Cognos® parameter format	Cognos® parameter format examples		
		Single value	List of values	Interval values
Date	CCYY-MM-DD	2012-02-03	2012-02-03-Value:2012-03-14	Between 2012-02-03 and 2012-04-15
Time	hh:mm:ss	01:00:00	01:00:00-Value:01:01:01	Between 01:00:00 and 23:59:30
Time Stamp	`CCYY-MM-DDThh:mm:ss` or `CCYY-MM-DD hh:mm:ss`	`2012-02-03 15:05:00`	2012-02-03 15:05:00-Value:2012-02-03T16:01:00-Value:2012-02-03T16:00:00	`Between 2012-02-03 15:05:00 and 2012-04-15T16:00:00`

Note: You must specify the parameter format exactly as they are shown in the table respecting lower case and upper case formats.

Creating an event rule

How you can create an event rule.

About this task

The purpose of the following scenario is to show how to create a new event rule in the context of an example.

The scope of the following draft event rule is to monitor log files named "messages.log*" for error messages written to the log containing the text "ERROR: Authentication did not succeed for user ID dwcadmin. An invalid user ID or password was specified", every 60 seconds on a workstation named "MDM_production". When an event that corresponds to this rule occurs, a new incident is automatically opened in Service Now containing a description of the error.

1. Create a new event rule and define general information about it:
    a. Go to **Workload Designer** and choose an engine.
    b. In the *Explore area*, select *Create new* +.
    c. In *General Info*, in **Name** type: SERVICENOW
    d. In *General Info*, set **Save as draft:** on
2. Define the event to be triggered:
    a. Select *Add events*
    b. Select the *Log message written* event from the *File Monitor* category
    c. In **Event name** type: logMessWritEvt1
    d. In **File name** type: matches `C:\Users\DWC\stdlist\appserver\dwcServer\logs\messages.log*`
    e. In **Match expression** type: matches ERROR: Authentication did not succeed for user ID dwcadmin. An invalid user ID or password was specified

f. In **Sample interval**: equal to 60
g. In **Workstation:** matches MDM_production

3. Define the action to be performed when the defined event occurs:
   a. Select *Add actions*
   b. Select the *Open Incident* action from the *ServiceNow* category
   c. In **Short Description** type: In the %{logMessWritEvt1.FileName} file, the following error has been found: %{logMessWritEvt1.MatchExpression}
   d. Set **Priority:** 3
   e. In **Description** type: At %{logMessWritEvt1.TimeStamp} on the %{logMessWritEvt1.Hostname} workstation, the following error %{logMessWritEvt1.MatchExpression} has been found in the %{logMessWritEvt1.FileName} file
   f. Type the **ServiceNow URL:**
   g. Enter the **ServiceNow User:** TWS_user
   h. In **Assignment Group:** L3_Busines_Unit_Team
4. Select the event rule and click **Save**.

Results

You have now created a draft event rule that monitors the log file, and when a new error string is added, an event is triggered and an action automatically opens an incident in Service Now. The ticket contains the error message, the name of the log file in which the error is written, and the name of the workstation on which the log file is located. Furthermore, the ticket is assigned to the specified user of the specified group.

What to do next

To activate the rule, you need to deploy it in the scheduling environment by switching the draft toggle off, and save again.

Go to *Manage Event Rule* to verify that the event rule is active.

Related information

Event management on page 91

Event management configuration on page 13

Event rule on page 245

Editing event rules

Editing event rules in the IBM® Workload Scheduler database.

About this task

The purpose of the following scenario is to show how to edit an existing event rule in the context of an example.

To avoid the SAP system overload when operators work, you want to set a low SAP job limit during the day, and instead increase the SAP job limit during the night. Thus, you created an event rule named SAP_OPERATOR_MODE that increases the number of running SAP jobs.

The number of SAP jobs increases when an SAP event, which ID is SAP_OPMODE_SWITCH and the parameter is equal to NIGHT, is raised. In the following scenario, the NIGHT parameter corresponds to 8pm.

After few months, the number of SAP jobs is increased, so you need to edit the event rule to make all jobs run. To achieve that, you need to anticipate the running time from NIGHT to AFTERNOON. In the following scenario, the AFTERNOON parameter corresponds to 6pm.

1. Select the engine where the event rule has been created.
2. Select the event rule named SAP_OPERATOR_MODE and select *Edit*.
3. The SAP_OPERATOR_MODE event rule with the following parameters opens:
   Choose from:
   - Selecting SAP_OPERATOR_MODE, you can see the following parameters in the *General Info* panel:

     Name: SAP_OPERATOR_MODE

     Save as draft: off

     Description: Increase workstation limit
   - Selecting the SAP Event Raised on XA Workstations event, you can see the following parameters in the *Properties* panel:

     Event name: R3EventRaised1

     Workstation matches: S4HANA

     SAP Event ID equal to: SAP_OPMODE_SWITCH

     SAP Event parameter equal to: NIGHT
   - Selecting the Submit job action, you can see the following parameters in the *Properties* panel:

     Job name: Increase_limit_on_S4HANA_wks

     Job workstation name: fta_production

     Description: definition to increase the SAP job limit
4. Select the SAP Event Raised on XA Workstations event and change the SAP Event parameter from NIGHT to AFTERNOON.
5. Select the event rule checkbox and then select Save.

Results

You have correctly edited the event rule that increases the SAP job limit at 6pm instead of 8pm.

What to do next

Monitor your event rule definition. Go to *Monitor Event Rules* to verify that the event rule runs correctly.

Chapter 8. Managing Workload Security

Managing security settings in your environment by using Dynamic Workload Console.

If you work with the role-based security model, you can manage security settings in your environment by using Dynamic Workload Console.

From the navigation toolbar click **Administration > Security > Manage Workload Security**. Here you can create and manage security objects in the database, according to a role-based security model.

You can work with the following security objects:

Security roles

Each role represents a certain level of authorization that defines the set of actions that users or groups can perform on a set of object types.

Security domains

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

Access control lists

Each access control list assigns security roles to users or groups, in a certain security domainor folder.

Folders

Each folder has its own level of authorisation that defines the set of actions that users or groups can perform on each folder.

If you want to specify different security attributes for some or all of your users, you can create additional security domains based on specific matching criteria. For example, you can define a domain that contains all objects named with a prefix 'AA' and specify the actions that each role can do on that domain. You can then associate the roles to users or group by defining an access control list.

To create or modify security objects with Dynamic Workload Console, you must have permission for the **modify** action on the object type **file** with attribute **name=security**.

Note:

After saving a new object in the Graphical Designer, the object is automatically placed in edit mode. When saving an object for which you have the **create** permission, but not the **modify** permission, an error message informs you that you are not authorized to perform the **modify** action on that object.

When working with the role-based security from Dynamic Workload Console, be aware that access to security objects is controlled by an "optimistic locking" policy. When a security object is accessed by user "A", it is not actually locked. The security object is locked only when the object update is saved by user "A", and then it is unlocked immediately afterwards. If in the meantime, the object is accessed also by user "B", he receives a warning message saying that the object has just been

updated by user "A", and asking him if he wants to override the changes made by user "A", or refresh the object and make his changes to the updated object.

For more information about enabling a role-based security model for your installation, see the section about getting started with security in *Administration Guide*.

Managing access control list

About this task

Create an access control list by assigning security roles to users or groups, in a certain security domain or in one or more folders.

You can:

- Give access to user or group.
- View access for user or group.
- View access for Security Domain or folders.
- Manage accesses.

Give access to user or group

About this task

To give access to users or groups complete the following procedure:

1. From the navigation toolbar, click **Administration**.
2. In the **Workload Environment Design**, select **Manage Workload Security**.
   Result
   The Manage Workload Security panel opens.
3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
4. In the Access Control List section, click **Give access to user or group**.
   Result
   The Create Access Control List panel opens.
5. Enter the user name or the group name, the assigned roles, and the security domain or enter the folders assigned. For each Access Control List you can associate one or more folders.
6. Click **Save** to save the access definition in the database.
7. Click **Save and Create New** to save the access definition in the database and proceed to create a new access definition.
8. Click **Save and Exit** to save the access definition in the database and return to the Manage Workload Security panel.

Results

The access definition has now been added to the database. If the **optman** enRoleBasedSecurityFileCreation global option is set to *yes*, the access definition is activated in your security file.

View access for user or group

About this task

From Manage Workload Security, you can also view the access for users or groups.

1. In the Access Control List section of the Manage Workload Security panel, click **View access for user or group**.
   Result
   The input field for the user or group name is displayed.
2. Enter the user or group name and click **View**.
   Result
   The user or group access, with the assigned roles, to the related security domains is displayed.

View access for Security Domain

About this task

From Manage Workload Security, you can also view the access to a certain security domain.

1. In the Access Control section of the Manage Workload Security panel, click **View access for Security Domain**.
   Result
   The input field for the security domain name is displayed.
2. Enter the security domain name and click **View**.
   Result
   The list of users or groups, with the assigned roles, that have access to the specified security domain is displayed.

Manage accesses

About this task

From Manage Workload Security, you can also **remove** and **edit** existing access control lists.

1. In the Access Control List section of the Manage Workload Security panel, click **Manage Accesses**.
   Result
   The list of users or groups, with the assigned roles, that have access to the different security domains is displayed.
2. Select the access control list that you want to manage.
3. Select the action that you want to run on the selected access control list.

   If you select the **edit** action, you can change only the roles associated with the access control list. You cannot change the associated domain. If you want to change the domain, you must **remove** the access control list and redefine the access control list with a new domain.

Managing security domains

About this task

A security domain represents the set of objects that users or groups can manage. For example, you can define a domain that contains all objects named with a prefix 'AA'. If you want to specify different security attributes for some or all of your users, you can create additional security domains based on specific matching criteria.

You can filter objects by specifying one or more attributes for each security object type. You can include or exclude each attribute from the selection. For example, you can restrict access to a set of objects having the same name or being defined on the same workstation, or both.

For the attributes that you can specify for each security object type, seeAttributes for security object types on page 139.

For the values that you can specify for each object attribute, see Specifying object attribute values on page 140.

You can create new security domains or manage existing security domains.

Create new security domain

About this task

To create a new security domain from the Dynamic Workload Console, complete the following procedure:

1. From the navigation toolbar, click **Administration**.
2. In the **Security**, select **Manage Workload Security** .
   Result
   The Manage Workload Security panel opens.
3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
4. In the Security Domains section, click **Create new Security Domain**.
   Result
   The security domain creation panel opens.
5. Enter the name of the security domain that you are creating and, optionally, the domain description.
6. Select the type of security domain that you want to define:

   Simple

   To define a filtering rule that applies to all object types. Events and actions are excluded from this filtering rule.

   Complex

   To define different filtering rules for different object types.
7. Use object filtering to select the set of security objects that users or groups can manage in the security domains that you are defining. You can use the wildcard character (*) when defining object attributes.
8. Click **View** to see the mapping between the set of security objects that you are assigning to the domain and the corresponding set of security objects in the classic security model.
9. Click **Save** to save the security domain definition in the database.
10. Click **Save and Exit** to save the security domain definition in the database and then exit.

Results

The security domain has now been added to the database. If the **optman**enRoleBasedSecurityFileCreation global option is set to *yes*, the security domain is activated in your security file.

Edit security domain

About this task

From Manage Workload Security, you can also **remove**, **edit**, and **duplicate** existing security domains.

1. In the Security Domains section of the Manage Workload Security panel, click **Manage Security Domain**.
   Result
   The list of the available security domains is displayed.
2. Select the security domains that you want to manage.
3. Select the action that you want to run on the selected security domains.

Managing security roles

About this task

A security role represents a certain level of authorization and includes the set of actions that users or groups can perform on a set of object types.

For the list of actions that users or groups can perform on the different objects, for each IBM Workload Scheduler task, see Actions on security objects on page 134.

A set of predefined security roles is available in the master domain manager database after the product has been installed:

- A full access definition for the user who installed the product, TWS_user with the default security role assigned named `FULLCONTROL`.
- An access definition for the system administrator, root on UNIX or Administrator on Windows.

You can create new security roles or manage existing security roles.

Create new role

About this task

To create a new security role from the Dynamic Workload Console, complete the following procedure:

1. From the navigation toolbar, click **Administration**.
2. In the **Security** select **Manage Workload Security**.
   Result
   The Manage Workload Security panel opens.
3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
4. In the Roles section, click **Create new role**.

Result

The Create Role panel opens.

5. Enter the name of the security role that you are creating and, optionally, the role description.
6. For each of the IBM Workload Scheduler tasks, assign the level of access for performing certain actions on specific object types to the security role. You can assign a predefined or a custom level of access.
7. Click **Show Details** to see the permissions associated to a predefined level of access, or to define your custom level of access. Tooltips are available to explain what a certain permission means for a particular object type.
8. Click **View** to see the mapping between the set of permissions that you are assigning and the corresponding set of permissions in the classic security model.
9. Click **Save** to save the security role definition in the database.
10. Click **Save and Exit** to save the security role definition in the database and return to the Manage Workload Security panel.

Results

The security role has now been added to the database. If the **optman** enRoleBasedSecurityFileCreation global option is set to *yes*, the security role is activated in your security file.

Manage roles

About this task

From Manage Workload Security, you can also **remove**, **edit**, and **duplicate** existing roles.

1. In the Roles section of the Manage Workload Security panel, click **Manage roles**.
   Result
   The list of the available security roles is displayed.
2. Select the security roles that you want to manage.
3. Select the action that you want to run on the selected roles.

Authenticating the command line client using API Keys

You can use API Keys to authenticate the command line client

About this task

The purpose of the following scenario is to show how to create and use an API Key to successfully authenticate the command line client.

The scope of the scenario is to list every folder defined in an engine.

1. Login into the Dynamic Workload Console and navigate to **Manage API Keys**.
2. In **Manage API Keys**, select an engine from the **Engine name** list and click **Apply**.
3. Create an API Key:

a. Click **Add new**.
b. Select either *Personal* or *Service* from the **Type** list.
c. In **Name**, type **API_Key_Name**.
d. If you set the API Key type to *Service*, select the relevant groups from the **Group Name** list.
e. Click **Submit**.
f. The API Key token is displayed in the dialogue box. Save it and store it in a secure place.

Note: API Keys expire by default after 365 days. You can set a custom duration for API Keys by adding the following parameters to the `{INST_DIR}/usr/servers/engineServer/resources/properties/TWSConfig.properties` file:

```
com.ibm.tws.dao.rdbms.apikey.expiring.timeout = n.days
com.ibm.tws.util.jwt.apikey.expiration.date = n.days
```

where `com.ibm.tws.dao.rdbms.apikey.expiring.timeout = n.days` determines the duration in days of any future API Key created, and `com.ibm.tws.util.jwt.apikey.expiration.date = n.days` determines when the status of future API Keys changes from **Valid** to **Expiring**.

You need to restart WebSphere Application Server Liberty after adding the custom parameters and before creating new API Keys.

This process must be repeated every time you want to change the duration and status change interval of new API Keys. API Keys that have been already created cannot be modified.

4. Authenticate the command line client with the API Key token:

a. Open the command line client configuration file:

On Windows

```
%userprofile%\.OCLI\config.yaml
```

On Linux

```
$HOME/.OCLI/config.yaml
```

b. Add or replace the `JWT` property, entering the API Key token you generated in **Step 3** as the property value.

c. Save the file. The command line client is now authenticated.

5. Access the list of available folders using command line client.
a. Enter the command: `composer list folder=@`.

Results

You have accessed an engine from a command line client authenticated with an API Key.

Service API Key with no group associated

This scenario covers the specific case where a Service API Key is created without assigning a group to it.

About this task

Creating a Service API Key without assigning a group to it results in the generation of a token that cannot authenticate any user. There is a procedure in place to enable the Service API Key:

1. Copy the Service APY Key name
2. Edit or create the Access Control List or Security file adding a username/logon using the name of the Service API Key copied in step 1. Associate a role to the username

Results

The authentication of the Service API Key token is now based on the username assigned to it, rather than on a group.

Managing folders

About this task

Folders help you to organize jobs and job streams into different categories. You can create folders with different levels of authorization that define the set of actions that users or groups can perform on each folder. More than one folder can be associated to the same Access Control List, and the level of security is also applied to the sub-folders.

You can also grant a user administrator privileges on a folder and its sub-folders so that this user can then create access control lists, with a dedicated role to manage the objects contained in the folder. See Granting administrator permissions to a user on a folder on page 133.

Creating, renaming, or deleting a folder

About this task

To create, rename, or delete a folder:

1. From the navigation toolbar, click **Administration**.
2. In the **Security**, select **Manage Workload Security**.
3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
   Result
   The Manage Workload Security panel opens.
4. In the folders section, click **Manage Folder**.
   Result
   The **Manage Folders** panel opens. From this panel you can:
   - Use the search box to search folders and job streams in the current view.
   - Create a folder or subfolder, rename or delete a folder.

Granting administrator permissions to a user on a folder

About this task

The IBM Workload Scheduler administrator can grant administrator permissions to a user on a folder so that the user can freely define access control lists for other users on the same folder or any sub-folders. Users can then access the objects in the folder or sub-folders in accordance with the access permissions they have on the objects.

Tip: Users with the FULLCONTROL security role assigned automatically have administrator rights on folders.

The following scenario demonstrates how Tim, the IBM Workload Scheduler administrator, grants Linda, the application administrator (`app1_admin` user), permissions on the folder named `/PRD/APP1/`, and how Linda grants access to Alex, the application user, to work with the objects defined in `/PRD/APP1/FINANCE`:

1. Tim, the IBM Workload Scheduler administrator, grants Linda, the `app1_admin` user, administrator permissions on the folder, `/PRD/APP1/`, through the definition of an access control list and by modifying her currently assigned role, `APPADMIN`. Optionally, Tim can create a new role with the appropriate permissions to achieve the same result.
    a. From the **Manage Workload Security** page, Tim selects **Manage roles**.
    b. He then selects her current role from the list, `APPADMIN` and clicks **Edit**.
    c. He gives this role administrator permissions on folders by selecting **Delegate folder permission (folder - acl)** in the **Administrative Tasks** section and clicks **Save and Exit**.
    d. Tim then creates an access control list for Linda, the `app1_admin` user. From the **Manage Workload Security** page, Tim selects **Give access to users or groups**.
    e. From the **Create Access Control** List page, Tim selects **User name** from the drop-down and enters Linda's user name, `app1_admin` in the text box.
    f. In the **Role** text box, Tim enters the `APPADMIN` role he modified earlier.
    g. In the text box next to the **Folder** selection, Tim enters the folder path of the folders on which he wants to grant Linda permissions, `/PRD/APP1/`.

    Result

    Linda, the `app1_admin` user, with the `APPADMIN` role assigned, can now access the entire `/PRD/APP1/` hierarchy, can create new folders in this path, and can assign access to these folders to other users.
2. Linda needs to give application users such as Alex, access to the objects in the `/PRD/APP1/FINANCE` folder. She creates a new access control list on the folder for the application user and assigns a role to this user.
    a. From the **Manage Workload Security** page, Linda selects **Give access to users or groups** from the **Access Control List** section.
    b. On the **Create Access Control List** page, Linda selects **User name** from the drop-down and enters the user name for Alex, the application user, `app1_user`.
    c. Since Linda cannot create new roles, she specifies an existing role in the Role text box. Only Tim, the IBM Workload Scheduler administrator, can create new roles.
    d. In the text box next to the **Folder** selection, Linda enters the folder path to the new sub-folder she created and to which Alex requires access: `/PRD/APP1/FINANCE`.

    Result

    Alex now can access the `/PRD/APP1/FINANCE` folder. He does not have access permissions on the `/PRD/APP1` folder.

Actions on security objects

The following tables show the actions that users or groups can perform on the different object types, for each IBM Workload Scheduler task. See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with **composer** command line interface.

Table 10. Actions that users or groups can perform when designing and monitoring the workload

Design and Monitor Workload	
Actions that users or groups can perform	**Security object types**
List (list)	Jobs (job)
Display (display)	Job Streams (schedule)
Create (add)	User Objects (userobj)
Delete (delete)	Prompts (prompt)
Modify (modify)	Resources (resource)
Use (use)	Calendars (calendar)
Unlock (unlock)	Run Cycle Groups (runcygrp)
Actions on remote workstations while modeling jobs (cpu-run) **Note:** See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command-line interface.	Variable Tables (vartable) Workload Application (wkldappl) Workflow Folders (folder) Parameters (parameter)

Table 11. Actions that users or groups can perform when modifying current plan

Modify current plan
Actions that users or groups can perform on the current plan
Add job stream dependency (schedule - adddep)
Add job dependency (job - adddep)
Remove job dependency (job - deldep)
Remove job stream dependency (schedule - deldep)
Change job priority (job - altpri)

Table 11. Actions that users or groups can perform when modifying current plan

(continued)

Modify current plan
Actions that users or groups can perform on the current plan
Change job stream priority (schedule - altpri)
Cancel job (job - cancel)
Cancel job stream (schedule - cancel)
Rerun job (job - rerun)
Confirm job (job - confirm)
Release job (job - release)
Release job stream (schedule - release)
Kill jobs (job - kill)
Reply to prompts (prompt - reply)
Reply to job prompts (job - reply)
Reply to job stream prompts (schedule - reply)
Alter user password (userobj - altpass)
Change jobs limit (schedule - limit)
Actions on job remote system (job - run)
Change resource quantity (resource - resource)

Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 12. Actions that users or groups can perform when submitting workload

Submit Workload
Workload definitions that can be added to the current plan
Only existing job definitions (job - submitdb)
Existing jobs definitions and ad hoc jobs (job - submit)
Existing job stream definitions (schedule - submit)

Table 12. Actions that users or groups can perform when submitting workload

(continued)

Submit Workload
Workload definitions that can be added to the current plan

Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 13. Actions that users or groups can perform when managing the workload environment

Manage Workload Environment
Actions that users or groups can perform on workstations, domains, and workstation classes
List workstations (cpu - list)
Display workstation details (cpu - display)
Create workstations (cpu - add)
Delete workstations (cpu - delete)
Modify workstations (cpu - modify)
Use workstations (cpu - use)
Unlock workstations (cpu - unlock)
Start a workstation (cpu - start)
Stop a workstation (cpu - stop)
Change limit (cpu - limit)
Change fence (cpu - fence)
Shutdown (cpu - shutdown)
Reset FTA (cpu - resetfta)
Link (cpu - link)
Unlink (cpu - unlink)
Use 'console' command from conman (cpu - console)
Upgrade workstation (cpu - manage)

Table 13. Actions that users or groups can perform when managing the workload environment

(continued)

Manage Workload Environment
Actions that users or groups can perform on workstations, domains, and workstation classes

Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 14. Actions that users or groups can perform when managing event rules

Manage Event Rules
Actions that users or groups can perform on event rules
List event rules (eventrule - list)
Display event rules details (eventrule - display)
Create event rules (eventrule - add)
Delete event rules (eventrule - delete)
Modify event rules (eventrule - modify)
Use event rules (eventrule - use)
Unlock event rules (eventrule - unlock)
Display actions in the event rules (action - display)
Monitor triggered actions (action - list)
Use action types in the event rules (action - use)
Submit action (action - submit)
Use events in the event rules (event - use)
Use a File Monitor event on the workstation where the file resides. (event - display)

Table 14. Actions that users or groups can perform when managing event rules

(continued)

Manage Event Rules
Actions that users or groups can perform on event rules
Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 15. Administrative tasks that users or groups can perform

Administrative Tasks
Administrative tasks that users or groups can perform
View configuration (dump security and global options) (file - display)
Change configuration (makesec, optman add) (file - modify)
Delete objects definitions (file - delete)
Unlock objects definitions (file - unlock)
Allow planman deploy, prodsked and stageman (file - build)
Delegate security on folders (folder - acl)
Note: See in parenthesis the corresponding *action* and *object* values that you must use when defining role-based security with the **composer** command-line interface.

Table 16. Actions that users or groups can perform on workload reports

Workload Reports	
Actions that users or groups can perform on workload reports	
Generate workload reports (display report)	**Reports in Dynamic Workload Console** **RUNHIST** Job Run History **RUNSTATS** Job Run Statistics **WWS** Workstation Workload Summary **WWR** Workstation Workload Runtimes

Table 16. Actions that users or groups can perform on workload reports

(continued)

Workload Reports Actions that users or groups can perform on workload reports
SQL Custom SQL **ACTPROD** Actual production details (for current and archived plans) **PLAPROD** Planned production details (for trial and forecast plans)

Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 17. Actions that users or groups can perform on folders.

Folders Actions that users or groups can perform on folders
Access folders chfolder (display) listfolder (list or list and display) mkfolder (modify) rmfolder (delete) renamefolder (add)

Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Attributes for security object types

Table 18: Attributes for security object types on page 140 shows the attributes that you can specify for each security object type (see in parenthesis the corresponding object type and object attribute that you must use when defining security objects with the **composer** command line interface).

Table 18. Attributes for security object types

Attribute / Security object type	Name (name)	Workstation (cpu)	Custom (custom)	JCL (jcl)	JCLtype (jcltype)	Logon (logon)	Provider (provider)	Type (type)	Host (host)	Port (port)	Folder (folder)	CPU Folder (cpufolder)
Actions (action)							✓	✓	✓	✓		
Calendars (calendar)	✓										✓	
Workstations (cpu)	✓							✓			✓	
Events (event)			✓				✓	✓				
Event rules (eventrule)	✓										✓	
Files (file)	✓											
Jobs (job)	✓	✓		✓	✓	✓					✓	✓
Parameters (parameter)	✓	✓									✓	✓
Prompts (prompt)	✓										✓	
Reports (report)	✓											
Resource (resource)	✓	✓									✓	
RunCycle groups (runcygrp)	✓										✓	
Job streams (schedule)	✓	✓									✓	✓
User objects (userobj)		✓				✓						✓
Variable tables (vartable)	✓										✓	
Workload applications (wkldappl)	✓										✓	
Folders (folder)	✓											

For the values that are allowed for each object attribute, see Specifying object attribute values on page 140.

Specifying object attribute values

The following values are allowed for each object attribute (see in parenthesis the corresponding object type and object attribute for the **composer** command line interface):

Name (name)

Specifies one or more names for the object type.

- For the Files (file) object type, the following values apply:

  globalopts

  Allows the user to set global options with the `optman` command. The following access types are allowed:

  - Display access for `optman ls` and `optman show`
  - Modify access for `optman chg`

  prodsked

  Allows the user to create, extend, or reset the production plan.

  security

  Allows the user to manage the security file.

  Symphony

  Allows the user to run **stageman** and **JnextPlan**.

  trialsked

  Allows the user to create trial and forecast plans or to extend trial plans.

  Note: Users who have restricted access to files should be given at least the following privilege to be able to display other object types that is, Calendars (calendar) and Workstations (cpu):

  ```
  file  name=globalopts  action=display
  ```

- For the **Variable Tables (vartable)** object type, you can use the $DEFAULT value for the **Name (name)** attribute to indicate the default variable table. This selects the table that is defined with the `isdefault` attribute.

Workstation (cpu)

Specifies one or more workstation, domain, or workstation class name. Workstations and workstation classes can optionally be defined in a folder. If this attribute is not specified, all defined workstations and domains can be accessed. Workstation variables can be used:

$MASTER

The IBM Workload Scheduler master domain manager.

$AGENTS

Any fault-tolerant agent.

$REMOTES

Any standard agent.

$THISCPU

The workstation on which the user is running the IBM Workload Scheduler command or program.

If you use the **composer** command line to define security domains, the following syntax applies:

```
cpu=[folder/]workstation]...
```

folder=*foldername*

Scheduling objects such as, jobs, job streams, and workstations, to name a few, can be defined in a folder. A folder can contain one or more scheduling objects, while each object can be associated to only one folder. The default folder is the root folder (/).

cpufolder=*foldername*

The folder within which the workstation or workstation class is defined.

Custom (custom)

Use this attribute to assign access rights to events defined in event plug-ins. The precise syntax of the value depends on the plug-in. For example:

- Specify different rights for different users based on SAP R/3 event names when defining event rules for SAP R/3 events.
- Define your own security attribute for your custom-made event providers.
- Specify the type of event that is to be monitored. Every event can refer to an event provider.

If you use **composer** command line to define security domains, the following syntax applies:

```
custom=value[,value]...
```

JCL (jcl)

Specifies the command or the path name of a job object's executable file. If omitted, all defined job files and commands qualify.

You can also specify a string that is contained in the task string of a JSDL definition to be used for pattern matching.

If you use **composer** command line to define security domains, the following syntax applies:

```
jcl="path" | "command" | "jsdl"
```

JCL Type (jcltype)

Specifies that the user is allowed to act on the definitions of jobs that run only scripts (if set to scriptname) or commands (if set to docommand). Use this optional attribute to restrict user authorization to actions on the definitions of jobs of one type only. Actions are granted for both scripts and commands when JCL Type (jcltype) is missing.

A user who is not granted authorization to work on job definitions that run either a command or a script is returned a security error message when attempting to run an action on them.

If you use **composer** command line to define security domains, the following syntax applies:

```
jcltype=[scriptname | docommand]
```

Logon (logon)

Specifies the user IDs. If omitted, all user IDs qualify.

You can use the following values for the **Logon (logon)** attribute to indicate default logon:

$USER

Streamlogon is the conman/composer user.

$OWNER

Streamlogon is the job creator.

$JCLOWNER

Streamlogon is the OS owner of the file.

$JCLGROUP

Streamlogon is the OS group of the file.

If you use **composer** command line to define security domains, the following syntax applies:

```
logon=username[,username]...
```

Provider (provider)

For **Actions (action)** object types, specifies the name of the action provider.

For **Events (event)** object types, specifies the name of the event provider.

If **Provider (provider)** is not specified, no defined objects can be accessed.

If you use **composer** command line to define security domains, the following syntax applies:

```
provider=provider_name[,provider_name]...
```

Type (type)

For **Actions (action)** object types, is the `actionType`.

For **Events (event)** object types, is the `eventType`.

For **Workstations (cpu)** object types, the permitted values are those used in composer or the Dynamic Workload Console when defining workstations, such as `manager`, `broker`, `fta`, `agent`, `s-agent`, `x-agent`, `rem-eng`, `pool`, `d-pool`, `cpuclass`, and `domain`.

Note: The value `master`, used in conman is mapped against the `manager` security attributes.

If **Type (type)** is not specified, all defined objects are accessed for the specified providers (this is always the case after installation or upgrade, as the type attribute is not supplied by default).

If you use **composer** command line to define security domains, the following syntax applies:

```
type=type[,type]...
```

Host (host)

For **Actions (action)** object types, specifies the TEC or SNMP host name (used for some types of actions, such as sending TEC events, or sending SNMP). If it does not apply, this field must be empty.

If you use **composer** command line to define security domains, the following syntax applies:

```
host=host_name
```

Port (port)

For **Actions (action)** object types, specifies the TEC or SNMP port number (used for some types of actions, such as sending TEC events, or sending SNMP). If it does not apply, this field must be empty.

If you use **composer** command line to define security domains, the following syntax applies:

```
port=port_number
```

Chapter 9. Changing user password in the plan

How you change the user password in the plan.

About this task

A User is the user name used as the login value for several operating system job definitions. Users are defined in the database and are associated to a password.

Users need to access the workstation where IBM Workload Scheduler launches jobs. If you need to change the user password after having already generated the plan, you can change the password in the plan. However, the changed password is only relevant to the current plan; Jnextplan restores the user password the next time it is run.

To change a user password in the plan, perform the following steps:

1. From the navigation toolbar, click **Administration > Security > Alter User Password in Plan**.
2. Select the engine.
3. Enter the following information in the **Alter User Password in Plan** panel:

   Workstation

   The name of the IBM® Workload Scheduler workstation where the user can launch jobs.

   User name

   The user name. A user needs access to the workstation where IBM® Workload Scheduler launches jobs, and have the right to **Log on as batch**.

   The following formats are supported when specifying the value of the user name:

   username

   The Windows user. For example `user1`.

   domain\username

   The user belongs to a Windows domain. Specify the Windows domain name to which the user belongs. For example `MYDOMAIN\user1`.

   username@internet_domain

   The user belongs to an internet domain. The user name is in User Principal Name (UPN) format. UPN format is the name of a system user in an email address format. The user name is followed by the "at sign" followed by the name of the Internet domain with which the user is associated.

   For example `administrator@bvt.com`.

   When a name is not unique it is considered to be a local user, a domain user, or a trusted domain user, in that order. If you schedule a job on a pool or a dynamic pool, the job runs with the user defined on the pool or dynamic pool. However, the user must exist on all workstations in the pool or dynamic pool where you plan to run the job.

Maximum length is 47 characters.

Password

The user password as defined on the computer. Maximum length is 31 characters.

Confirm password

The user password again for confirmation.

Versions

Use this section to view the history of object changes and work with the different versions.

You can run the following actions:

Compare

Select two different versions and compare them.

Restore...

Select a previous version and start the restore process.

4. Modify the values as required and click **Save** to save the modified task.

Chapter 10. Monitoring your environment

Through the Dynamic Workload Console, you can monitor your plans, items in plans, scheduling activities, workstations and domains in your environment.

The IBM® Workload Scheduler enables you to monitor your environment in three ways:

Graphical Designer

Through the Graphical Designer, you can have an overall picture of the environment and how it's structured. At a glance, you can analyze one or more job streams, jobs, dependencies and also analyze the impact a job stream and its jobs can have on the rest of the plan. This helps you to focus where necessary, detect and resolve issues quickly.

Monitoring the process and troubleshooting is simple and immediate thanks to the Graphical View. You can also export the displayed graphic to either a Scalable Vector Graphics (SVG) file or a Portable Network Graphics (PNG) file.

Furthermore, you can tailor the view as you prefer thanks to the flexible layout of the Graphical Designer that enables you to reposition items in whichever way is most useful or meaningful for you.

For further information about the Graphical Designer, see Graphical Designer overview on page 106.

Orchestration Monitor

To monitor your environment you can easily define a monitoring plan query by clicking **Monitoring & Reporting > Orchestration Monitor**. For details about the complete procedure, see Orchestration Monitor overview on page 171.

Dashboard

From the Workload Dashboard you can view the whole status of your workload at a glance for one or more of the engines you have configured. You can check the status of workstations, jobs, critical jobs, prompts, and other relevant information.

For more information about the dashboard monitoring see: Workload Dashboard on page 161

Display a graphical plan view

Specify filter criteria and display a graphical representation of the plan in the Plan View.

About this task

To display a graphical representation of the plan, you can specify multiple filter criteria to match objects in the plan and display them in the plan view.

The IBM Workload Scheduler engine must be version 9.3 or later.

An enhanced graphical view of the plan is also available in the Plan View and Job Stream View, see Graphical views in the plan on page 149.

Tip: You can also request a graphical view of jobs and workstations from a mobile device. Refer to the *Mobile Applications Users Guide*.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

To display a Show Plan View, perform the following steps:

1. In the navigation bar, click **Monitoring & Reporting > Workload Monitoring > Show Plan View**
2. Specify the engine on which you want to search for the scheduling objects to display in the view.
3. Use the fields in the Filter Criteria section to limit the number of job streams displayed in the result. You can filter based on the names of job streams and workstations (distributed systems only).
4. You can also filter on the starting times of the job streams and decide if you want to include predecessors and successors in the Plan View.
5. Select **Auto refresh view on a timer** to refresh the Plan View at regular intervals. By default, the view is refreshed every 300 seconds (5 minutes). You can control the auto refresh through timer controls on the toolbar to pause, stop, and resume the auto refresh. You configure the default auto refresh interval by setting the **DefaultTime** property in the **PlanViewAutorefresh** section of the `TdwcGlobalSettings.xml` configuration file. See Plan View auto refresh interval on page 30.
6. Click **Go** when you are ready to see the results displayed in the graphical plan view.

Results

The Plan View is a graphical representation of a filtered set of the job streams that make up the plan. Using the Plan View toolbar, you can perform several actions on the objects displayed. You can also modify the original filter criteria to update the display. For more information about the objects in the view, see Graphical views in the plan on page 149.

What to do next

There are several actions you can perform on the objects displayed in the Plan View:

- Use the actions available from the toolbar that include modifying the original filter criteria that was set to change the objects displayed.
- Right-clicking a job stream reveals a number of actions that you can perform on the job stream:
  - Perform an action such as Cancel, Hold, Release, and add or remove a dependency.
  - Launch and display the job stream in the What-if analysis Gantt chart.
  - View the jobs in the job stream by launching the in-context link to the Job Stream View.
  - Open the job stream definition within the Workload Designer

    - Perform actions on the workstation such as Link, Unlink, Start, and Stop.
    - View workstation properties.
- Single-click a job stream to reveal an info-rich tooltip. The tooltip also includes a color-coded status bar that indicates the status of jobs such as, the number of failed jobs, successful jobs, jobs in abend, or jobs in waiting.

Related information

Plans on page 86

Graphical views in the plan on page 149

Graphical views in the plan

Views to monitor the progress of your job streams in the plan in a graphical map.

You can use these views to monitor the progress of your job streams in the plan from a graphical map.

You can also take several actions on the items displayed in the view. Almost all of the actions and information available in the traditional views resulting from Monitor queries are also graphically available from these views.

All of the views provide a toolbar that you can use to act upon the views and the objects displayed. For more information about the actions available from the toolbar, see the specific panel help.

You can also export the displayed graphic to either a Scalable Vector Graphics (SVG) file or a Portable Network Graphics (PNG) file . In these formats, vector-based images can be scaled indefinitely without degrading the quality of the image.

The graphical views contain a combination of textual (tooltips, labels) and visual elements (icons, colors, shapes, progress bar). See the following representations of the graphical elements in the views and what each represents:

Table 19. Graphical representation of scheduling objects

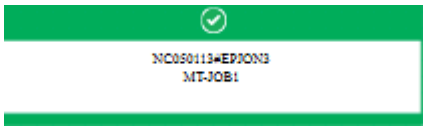
Object	Shape	Icon
Job		In the Workload Designer, the job is represented by this icon: . In the plan, this icon is replaced with the icon representing the status of the job, see Table 20: Graphical representation of statuses on page 151. If the job has a deadline set, then the deadline is visible in the lower-left corner of the job shape. An icon in the lower-left corner indicates if a job is time-dependent.

Table 19. Graphical representation of scheduling objects (continued)

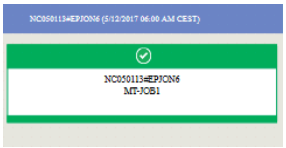
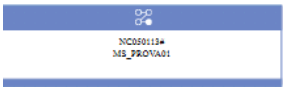
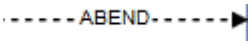
Object	Shape	Icon
Job stream	**Workload Designer Graphical View** **Job Stream View**  **Plan View** 	In the Workload Designer, the job stream is represented by this icon: . In the plan, the job stream is represented as a container containing jobs.
Prompt		In the Workload Designer, the prompt is represented by this icon:
Resource		In the Workload Designer, the resource is represented by this icon:
File		In the Workload Designer, the file is represented by this icon:
Internetwork dependency		In the Workload Designer, the internetwork dependency is represented by this icon:
Dependency		In the Workload Designer, the arrow represents a dependency.
Conditional dependency		In the Workload Designer, the dashed arrow represents a conditional dependency.

Table 19. Graphical representation of scheduling objects (continued)

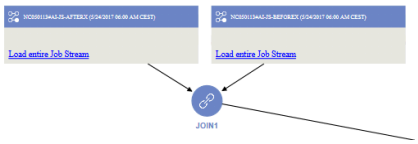
Object	Shape	Icon
Join dependency		In the Workload Designer, the join dependency is represented by this icon:

Table 20. Graphical representation of statuses

Status	Icon	Color
Blocked		Red
Error		
Running		Blue
Successful		Green
Canceled		Grey
Suppressed		
Held		Light grey
Undecided		
Waiting		
Ready		

Table 21. Quick actions on scheduling objects

Icon	Description
	Remove the selected object from the view. Use it, for example, to remove a job from a job stream or a dependency from a job. The removal becomes effective only when you save the item.

Table 21. Quick actions on scheduling objects (continued)

Icon	Description
	Remove all the dependencies of the item. Available in the Workload Designer graphical view only.
	Create a dependency. Click the icon, click an item, and then draw a line to the job stream or to the job that represents the depending item. The lines are drawn from the dependency to the depending item.
	View the job log. Not available in the Workload Designer graphical view.
	Rerun job. Not available in the Workload Designer graphical view.
	View the properties. Not available in the Workload Designer graphical view.
	Load successors. Not available in the Workload Designer graphical view.
	Release dependency. Not available in the Workload Designer graphical view.
	Open the actions menu which can also be accessed by right-clicking an item. Not available in the Workload Designer graphical view.

The following graphical views are currently available in the plan:

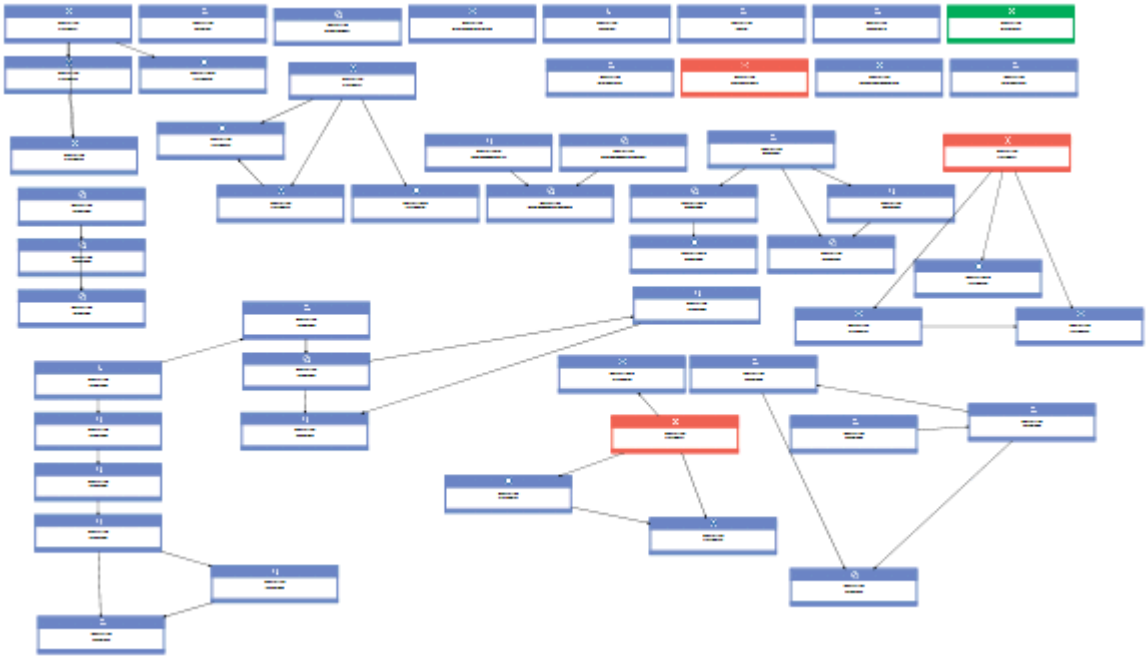
- Plan View on page 152
- Job Stream View on page 153

Plan View

Use this view to get an overall picture of your plan.

This view shows a collapsed plan; it shows only the job streams, hiding any jobs and internal dependencies. External dependencies, both at job and job stream levels, are shown as arrows between job streams. If a job stream has multiple dependencies on another job stream, these are shown as one single arrow.

The following is an example of the view:



Because potentially a plan can contain a large number of job streams, by default the Plan View displays a limited number of objects. You can determine which job streams to display by specifying filtering criteria in the **Show Plan View** page before launching the view or from the toolbar within the view itself.

The maximum number of job streams displayed by the Plan View is 1000. You can modify the default maximum number of job streams to display in the Plan View by modifying the **planViewMaxJobstreams** property in the `TdwcGlobalSettings.xml` global settings file for the Dynamic Workload Console.

To launch the Plan View:

1. From the portfolio, click: **Monitoring and Reporting > Workload Monitoring > Show Plan View**.
2. Specify any filter criteria and then click **Go**.

You can take multiple actions on the job streams in the view by right-clicking them and choosing an option from the context menu such as the following:

- Perform an action such as Cancel, Hold, Release, and add or remove a dependency.
- Launch and display a job stream in the What-if analysis Gantt chart.
- View the jobs in the job stream by launching the in context link to the Job Stream View.

Job Stream View

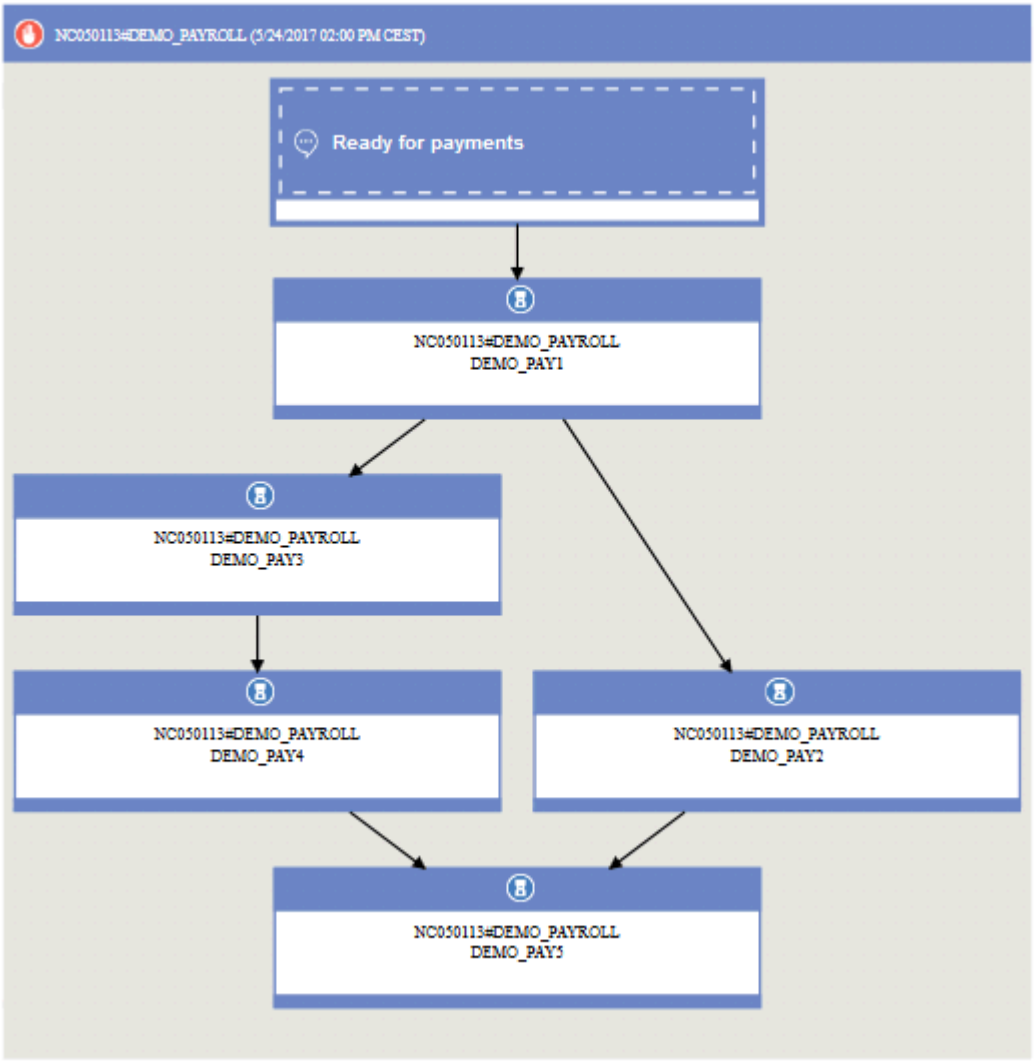
This view displays a picture of the job stream with all its jobs and related dependencies. You can navigate through the displayed jobs and job streams, choosing the level of predecessors and successors to display. By default, this view displays all the objects in the job stream and the first-level of external predecessors.

You can open this view in the following ways:

- From the table of results of a Monitor Jobs or Monitor Job Streams task, select an item and select **Job Stream View** from the toolbar.
- From the Plan View, right-click a job stream and select **Open > Job Stream View**.

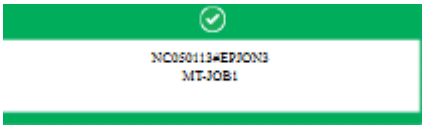
Use the **Rename** icon to change the name of the view. The **Rename** icon is displayed when you hover over the Job Stream View string.

The following is an example of a Job Stream View:



The main elements in the Job Stream View are:

- **Jobs**



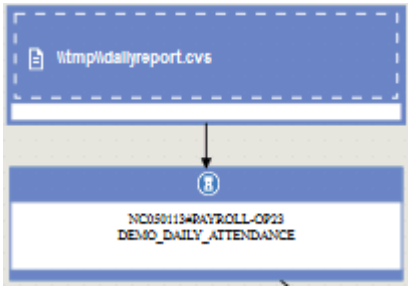
The status of the job is indicated by the color and the small icon in the header. See Table 20: Graphical representation of statuses on page 151 for the various icons representing the status. Additional information available about the job include:

  - The deadline written in the lower-left of the rectangle representing the job.
  - An icon is displayed in the lower-left indicating if the job is time dependent.
  - The tooltip for the job indicates if the job is a critical job or a shadow job. Shadow jobs are jobs running locally that are used to map jobs running on the remote engine.

Further details are available in the tooltip displayed when you click on the job.

You can take multiple actions on the job by right-clicking it and choosing options from the context menu. The actions available depend on the selected engine and on the type of job. From this menu, you can view and print the job log and the job properties, or act on the job and its dependencies. You can also take actions on the job workstation or open the job and job stream definitions in the database to modify them.

- **Dependencies**



They are shown as boxes different from those used to represent jobs, connected to the depending items by arrows that represent the dependency relationships. A joined conditional dependency instead is represented by the circle icon connected to the dependency arrows. To take actions on dependencies, you can right-click either the box or the arrow and choose an option from the context menu. Icons displayed in the box to the left describe the dependency type. Further details are available in the tooltip displayed when you click on the dependency. If dependencies within a job stream form a loop, this is highlighted in the view and a message appears, so that you can take the appropriate actions to resolve it.

Related information

Plans on page 86

Display a graphical plan view on page 147

Graphical View - modelling

Use this panel to graphically view the selected job or job stream with its dependencies in the Workload Designer.

This view shows the jobs contained in a job stream and their associated dependencies or the job stream dependencies. When you select an object in the graphical view, the object properties are displayed at the bottom of the properties pane. In distributed environments, the flexibility of the layout in this view enables you to reposition objects in whichever way is most useful or meaningful for you. If you save changes to the layout, the changes are maintained the next time you open the same job stream in the Graphical View, and also if you open the same job stream in the Job Stream View. In a z/OS environment, and in the case of a connection to a previous version of the engine, any changes made in the layout persist for the current browser session only. Changes to the layout cannot be saved to the database in a z/OS environment or when the engine connection is an engine of a release previous to 9.4.

For a description of the shapes and icons available in this view, as well as in the Job Stream View and Plan View, see Graphical views in the plan on page 149.

The same information and actions available from the **Details** pane are also available from the **Graphical View**.

You can select or open objects in the Welcome page and work on them in these panes to edit your definitions as required.

In this view, you can use the following toolbar, icons, and buttons:

Use this toolbar to rapidly manage the view.

Use these icons or the slider bar to zoom in or out from the view. The percentage of the view is displayed.

Use this icon to adjust the zoom level of the view so that all the objects are shown at the maximum resolution.

Use this icon to adjust the view to full screen size.

Use this icon to dynamically calculate the best size and position of all the objects in your view.

Click this icon to highlight all the dependencies of the selected object.

Click this icon to display or hide the job stream and job external dependencies in the graphical view. It can be useful to customize the picture granularity, especially in complex and large job streams.

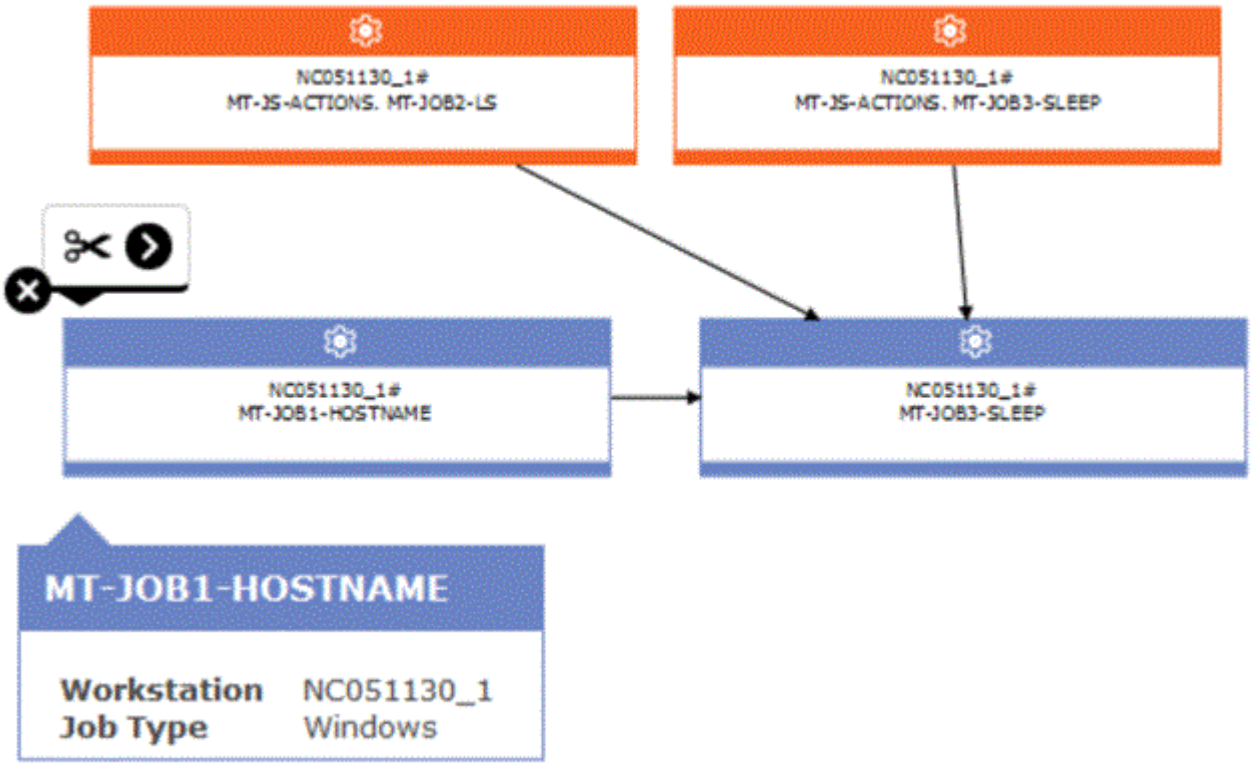
Click this icon to show or hide the job stream dependencies.

Click this icon to open the graphic in a Scalable Vector Graphics (SVG) file. With this type of file, vector-based images can be scaled indefinitely without losing any quality. A vector graphic program uses these mathematical formulas to construct the screen image, building the best quality image possible, for the given screen resolution.

Click this icon to open the graphic in a Portable Network Graphics (PNG) file. A PNG file is compressed in lossless fashion, which means that all image information is restored when the file is decompressed during viewing.

Use this icon to print the view.

Click an item to view a tooltip containing more item details. You can also view a quick actions menu above the selected item that enables you to perform the following actions:



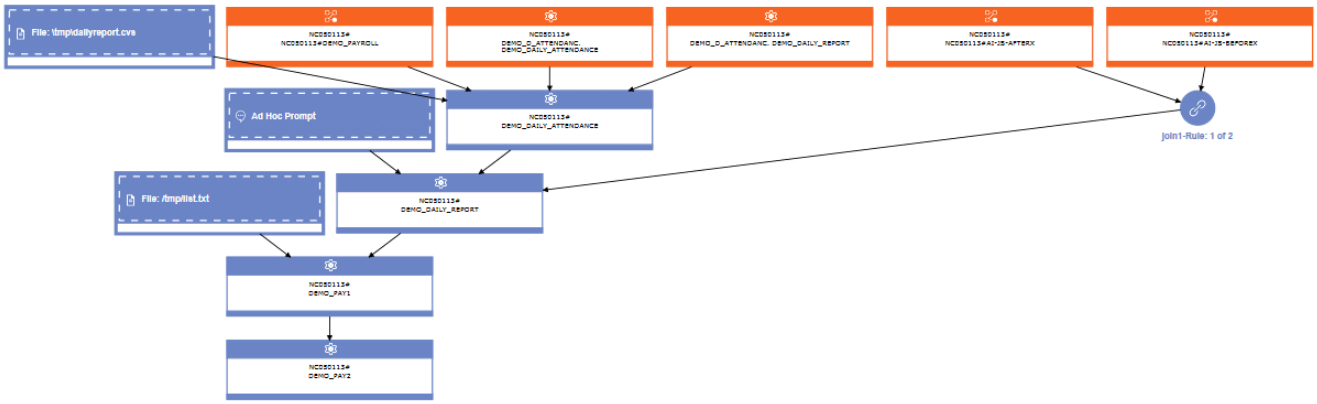
Click this icon to remove the selected item from the view. Use it, for example, to remove a job from a job stream or a dependency from a job. The removal becomes effective only when you save the object.

Use this icon to remove all the dependencies of the item.

Use this icon to create dependencies. Click the icon, click an item, and then draw a line to the job stream or to the job that represents the depending item. You can use this icon only to create dependencies from items displayed in the view and by drawing lines in the correct direction (from the dependency to the depending item).

Use this icon to also create a conditional dependency on jobs internal to the job stream. However, in this case, the job stream cannot be saved until you have manually updated the conditional dependency table, by specifying all the required information about the condition.

The following graphic is an example of a graphical representation of a job stream.



Dependencies

When you click a job stream or job dependency, you select its dependency relationship and you can remove it. If this item is a dependency for multiple items, click it again to select the next dependency relationship. If dependencies within a job stream form a loop, this is highlighted in a light yellow color in the view so that you can take the appropriate actions to resolve it. If you delete the dependency causing the loop, the highlighting disappears.

The arrows represent the dependency relationships; where the arrow goes from the dependency to the item that depends on it.

Dependencies can also be conditional. In the graphical view, this type of dependency is represented by a dash arrow.

Related information

Designing your workload on page 110

Analyzing the impact of changes on your environment

How to apply changes to your plan and see the related impact displayed in a Gantt chart.

The What-if Analysis visualizes the current plan in real time, displaying the current status of jobs and job streams, the planned start and end times, deadline, and the risk level associated to each job or job stream. The analysis is launched from any job or job stream in a distributed environment, or from any critical job or job that is part of a critical network in a z/OS environment. This feature is supported on version 9.3 engines or later. The current plan is visually represented in a Gantt chart. A Gantt chart is a time and activity bar chart that is used for planning and controlling projects or programs that have a distinct beginning or end. In a Gantt chart, each main activity that is involved in the completion of the overall project or program is represented by a horizontal bar. The ends of the bar represent the start and end of the activity. In IBM® Workload Scheduler, each activity represents a job or job stream.

Use the What-if Analysis to simulate and evaluate the impact of changes on the current plan. For example, if you know that a required file will be made available later than expected, you can evaluate the effects on the whole plan of the delay in the delivery of the file and see in detail which jobs and job streams risk missing their deadlines. For example, if you know that

a specific workstation is scheduled to undergo a programmed maintenance operation, you can simulate the impact of the missing workstation on the overall plan.

Drag and drop each job or job stream along the horizontal time axis to see how this changes the job or job stream status with respect to 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 move a job or job stream, its dependencies are maintained and automatically recalculated. You can also add and remove successors and predecessors for each job or job stream.

If you need to enlarge the scope of your analysis, you can add more job streams and jobs to the Gantt view using the **Show Jobs** and **Show Job Streams** (distributed only) buttons. This operation performs a search on the current plan and adds the selected jobs or job streams with the related predecessors and successors.

You can also highlight the critical path for a selected job or job stream along with its successors and predecessors. Modify the job duration, start time, and end time, add or delete successors and predecessors and see how this affects the whole critical path.

Under **More Actions** tab of the context menu you can find the **Workstation Unavailability Intervals** page where you can select intervals of time when the workstation will be unavailable. Use **Simulate Job Stream Submit** (distributed only) to simulate the impact of submitting a job stream. The predecessors and successors will not be matched automatically.

If you want to view the current plan again, click **More Actions > Reset** to revert the What-if Analysis to the current plan status.

Running the What-if Analysis before carrying out the actual plan shows your predicted results at a glance, and, by seeing the results beforehand, you can plan for any potential problems before you start.

Any changes that you make in What-if Analysis are applied for simulation purposes until you use the option **Apply changes** to the actual plan with exception to a z/OS environment where changes are made for simulation purposes only.

You can disable the What-if Analysis in your environment by setting the **optman** `enWhatIf | wi` global option to *no* (default value is *yes*). If you change this global option value, run "JnextPlan" to make the change effective. For additional information about **optman** global options , see the section about setting global options in *Administration Guide*.

The `enWhatIf | wi` global option interacts with the `enWorkloadServiceAssurance | wa` global option, which enables or disables privileged processing of mission-critical jobs and their predecessors. For details about the interaction between the What-if Analysis and the workload service assurance, see the section about disabling the What-if Analysis in *Administration Guide*.

If you want to extend the What-if Analysis to plans other than the current plan, consider that the maximum number of plans that you can analyze simultaneously is 5.

If you want to use the What-if Analysis on the backup master domain manager, ensure that on the backup master you have the same user definition as on the master domain manager.

If the administrator has enforced the related policy, when you apply the changes to the actual plan, a panel is displayed, where you enter the reason why the change was implemented, the ticket number, if any, and the category of the change. For more information about the justification policy, see Keeping track of changes to scheduling objects on page 199.

Workload Dashboard

Monitor your environment using the Workload Dashboard.

You can monitor the progress of your plan by using a dashboard.

To open the Workload Dashboard, in the navigation bar at the top, click **Boards > Workload Dashboard**. The panel opens showing a number of predefined widgets which return the results for the most widely-used queries. The Workload Dashboard provides a single, consolidated view for monitoring the workload status. By selecting the engine on the list, you can see the information related to the engine.

To customize the **Workload Dashboard** duplicate it by clicking the Overflow menu at the top left of the panel.

To create you own dashboard, refer to Creating a customized dashboard for monitoring on page 164.

Workload Dashboard

In the Workload Dashboard you can view the whole status of your workload at a glance for one or more of the engines you have configured. You can check the status of workstations, jobs, critical jobs, prompts, and other relevant information.

By clicking on selected widgets, you open the table listing the information in the widget in tabular format. The following widgets support the link to the table of results:

Engines

This table lists the available engines. By selecting the engine you can see the details in each widget.

Available workstations

This widget shows the number of available workstations for the selected engine. By clicking on the widget the **Monitor Workstations** view is displayed with detailed information about available workstations.

Unavailable workstations

This widget shows the number of unavailable workstations for the selected engine. By double clicking on the widget the **Monitor Workstations** view is displayed with detailed information about unavailable workstations.

Pending Prompts

This panel shows prompts for the selected engine. By clicking on the widget the **Monitor Prompts** view is displayed.

Critical job status

This widget shows in a bar chart the number of jobs in high, potential, and no risk status for the selected engine. By clicking on the widget the **Monitor Jobs** view is displayed with detailed information about jobs in critical status.

Job status

This pane shows the status of the jobs. By clicking on the widget the **Monitor Jobs** view is displayed with detailed information.

Jobs in late

This widget shows how many jobs have completed in late status for the selected engine. By clicking on the widget the **Monitor Jobs** view is displayed with detailed information about jobs in late status.

Jobs in error

This widget shows how many jobs have completed in error status for the selected engine. By clicking on the widget the **Monitor Jobs** view is displayed with detailed information about jobs in error status.

Min duration

This widget shows the number of jobs that have not reached the minimum duration for the selected engine. By clicking on the widget the **Monitor Jobs** view is displayed with detailed information about jobs that have not reached the minimum duration.

Max duration

This widget shows the number of jobs that have exceeded the maximum duration for the selected engine. By clicking on the widget the **Monitor Jobs** view is displayed with detailed information about jobs that have exceeded the maximum duration.

Resources

This widget shows the status of your resources. By clicking on the widget the **Monitor Resources** view is displayed with detailed information about the resources.

Related information

Running IBM Workload Scheduler from a mobile device on page 17

Plans on page 86

Widgets and Datasources

A *Widget* is an element of the custom boards, it can be static or associated to a datasource. the *Datasource* is the source of information and can be created by retrieving data from a query in the plan or from Rest APIs.

Widgets can be static, such as **Web content**, or associated to datasources. According to the datasource you create, a different widget can be associated.

Table widget has a limit of 1000 objects but the drill down is active, therefore by clicking on the widget it is possible to see more results in the monitoring panel.

The Bubble chart retrieves data from any associated data source showing a series of bubbles of different sizes placed in descending order. For example, when monitoring the critical jobs, data is gathered together according to the job status, and you can see three bubbles:

- The biggest red one represents data on critical jobs at high risk.
- The medium yellow one represents data on critical jobs at potential risk.
- The small green one represents data on critical jobs at potential risk.

Note: The bubble dimension and position does not change according to the number of jobs. They are placed according to the importance of the critical job status.

Gauge Chart and KPI widgets have a simplified process of creation. You can just select the datasource and insert a name. In addition you can set notification both for thresholds and advanced properties. Notifications are also available as pop-up notifications and they can be turned off by selecting **Browser notification** in the drop-down list of at the top right of the navigation bar.

During the creation of a widget, at the selection of the datasource the system suggests you which datasource can be associated.

A set of default datasources are provided in the list. These datasources only retrieve data from the engine specified from the list in the dashboard. To create a datasource associated to multiple engine, you can duplicate and edit the existing datasources or create new datasources.

To customize the workstation of a Rest API datasources with a z/OS engine use the filter `WORKSTATION_NAME`.

For information about how to add widgets to the custom pages, see the section about Creating a custom page in the Dynamic Workload Console User's Guide.

Creating a customized dashboard for monitoring

Create a dashboard to monitor your scheduling environment.

About this task

For monitoring purposes, you can create a dashboard to monitor your environments. For instance, in the new dashboard, you can have at a glance information about:

- Critical jobs using a KPI widget on page 164
- List of your engines in a table on page 165
- Monitor of jobs in waiting status on page 165
- Monitor your AIDA environments on page 166
- Company's website on page 166

The dashboard is composed of widgets that can be customized to show the data that you need; the data are collected from datasources that must be previously created. To start with a customized dashboard, first create the datasource or use our default, then create a board where you add the widgets.

Widgets and Datasources on page 163

At the top right of the Dynamic Workload Console there is a icon with a drop-down list to have all of the notifications at a glance. You can set the notifications with a threshold for each widget during the creation of a custom board, or set different type of notifications for the Workload Dashboard. You can also manage the pop up notifications by selecting **Browser notification** in the drop-down list.

Monitor critical jobs

1. From the navigation toolbar, click the icon and select **Manage Datasources**.
   Result
   The **Manage Datasource** panel opens.
2. Click **Add new plan query**.
3. Enter the **Engine** from the list, the **object type** and the **Plan**.
4. Enter the query in the related field or click **Edit** to select the filter that needs to be added in the query. Ensure the query has a valid result otherwise the widget could not work.
5. Enter a name and click **Save**.
   Result
   You can find your new datasource in the list.

Create the board and add the KPI widget to monitor your critical jobs

6. From the navigation toolbar, click the icon Board and **create board**.
7. Insert a name and click **Save**.
   Result
   The new board has been created.

8. Click the icon , select **datasource widget** and choose from the list the datasource that you have created and you want to associate.
9. Select **KPI** from the list of widgets.
10. Insert **widget name** and add a **threshold**.
11. Click **Save**.
    You can edit, duplicate and remove the widget from the hamburger menu and **save** the changes in the board.

List of your engines

Add the Table widget to the Board

1. From the **Board** menu, select the custom board that you have created.
2. Click the icon to enter edit mode in the board and then click the icon and select **Datasource widgets**.
3. Select the **Datasource widget** and choose **Engine List** from the drop-down menu.
4. Select the widget **Table** and insert a name.
5. Select the field you want to add and insert a name for each field.
6. Click **Save**.
   You can edit properties by clicking the Overflow menu.

Monitor of jobs in waiting status

1. From the navigation toolbar, click the icon and select **Manage Datasources**.
   Result
   The **Manage Datasources** panel opens.
2. Select the **Jobs count by status** and click duplicate.
3. Enter a name for the Datasource and then edit.
4. Type engine name and engine owner in the URL where requested.
5. **Save**.

Add the KPI widget to the Board to monitor your jobs in waiting

6. From the **Board** menu, select the custom board that you have created.
7. Click the icon to enter edit mode in the board and then click the icon and select **Datasource widgets**.
8. Select the **Datasource** previously edited.
9. Select KPI and insert a widget name.
10. In the **Select tracked properties** section click **Add new field**.
11. Insert the name and select `items[].waiting` from the drop-down menu.
12. Click **Add** and then **Save**.

Monitor your AIDA environments

1. From the **Board** menu, select the custom board that you have created.
2. Edit the board by clicking the edit icon and then the plus icon .
3. Add the AIDA widget.
    a. Go to **Special widgets** tab.
    b. Specify the name of the widget.
    c. Select an engine where AIDA is installed. If needed, you can change the engine later from the widget properties.
    d. Optionally, you can define a threshold to receive notifications.
4. Click **Save** to add the widget to your dashboard.

Add your company website to the dashboard

1. From the **Board** menu, select the custom board that you have created.
2. Click the icon to enter edit mode in the board and then click the icon and select **Static widgets**.
3. Select **Web content**.
4. Insert a name, select the datasource and insert the link of your company's website.
5. Click **Save**.

Exporting and importing a dashboard

The purpose of this scenario is to show how to export and import a dashboard.

About this task

You have just created a dashboard on the test environment that perfectly suits your needs; you can easily monitor your environment and quickly obtain the information you need.

Now, you want to reproduce it on the development and production environments, but creating a new dashboard on each environment with the same configuration, requires substantial effort.

With the import/export feature, you can quickly reproduce your dashboard anywhere in your environment.

Procedure

1. From the dashboard to be exported, click on the options menu next to the name of the dashboard and select **Export**. A JSON file is downloaded.
2. Log in to the Dynamic Workload Console that runs on the development environment.
3. Select the **Boards** menu and then click **Import Boards**.
4. Optionally, add a prefix to be added to the board name, for example, "Dev-". Although it is not required, it helps you identify the dashboard on the development environment.
5. Select the JSON file of the board to be imported and **Save**.

6. Reopen the **Boards** menu, and select the imported dashboard. The board automatically displays the retrieved data for monitoring purposes.
   You then replicate the same steps on the production environment.

Results

With just a few clicks, you have successfully reproduced the dashboard on all your environments.

Monitoring your Scheduling Environment

Tasks to monitor the workstations and domains of your environment.

To monitor workstations and domains in your environment, you create and run *monitor tasks*.

Note: You must create a connection to a remote IBM Workload Scheduler engine, before you can run tasks on it to obtain data.

When you create a task, you are actually defining a query where you specify multiple criteria to search for items and to display the search results. You can then save, reuse, and share this task with other users, and modify it at any time. When you run the task, you are actually running the query on the plan to retrieve the information according to the filters and level of detail you specified when you created the task.

Note: To add a task to your favorite bookmarks, from the panel displaying your task results, click the user icon and select **Favorites**.

To create a task, perform the following steps.

1. In the navigation bar, click **Monitoring and Reporting > Workload Monitoring > All Configured Task > New**.
2. In the Select Task Type panel, select the task you want to create, and click **Next**. You must select a task type to make the corresponding list active.
3. Follow the procedure relating to the specific task you are creating.

Alternatively, you can also create and run your task by specifying a query, as described in Creating a monitoring task query on page 178.

Each task you create and save is included under the **All Configured Tasks** menu.

Related information

Monitoring your Workload on page 170

Creating a task to Monitor Workstations

How you can create a Monitor Workstations task.

About this task

To create a Monitor Workstations task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. In the navigation bar, click **Monitoring and Reporting** > **Workload Monitoring** > **Monitor Workload** and follow the steps described in Monitoring your items in the plan on page 178.

   If you are familiar with conman, in the **Query** text box specify a query based on the conman showcpus syntax. Alternatively, click **Edit** to select the filter criteria from the list of options that is displayed.

2. In the General Filter section, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, in some of the results tables in the Periodic Refresh Options section, you can customize how often to refresh the information by specifying the refresh interval in seconds in hh:mm:ss format, with a minimum of 30 seconds and a maximum of 7200 seconds. For example, 00:01:10 means 70 seconds. If the value specified is not valid, the last valid value is automatically used. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table when the timer is in stop status. In this case, the value specified at task creation time is temporarily overwritten.

   You can filter the task results based on the workstation and domain names, or part of names (using wildcard characters).

   You can filter the task results based on the workstation types and reporting attributes.

3. In the Columns Definition section, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their link statuses, domains, and type, or you might want to see their statuses and the number of jobs successful or running on them. You can then drill down into this information displayed in the table and navigate it.

   In the Columns Definition section, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on:

   jobs, job streams domains, files, and resources. For example, you are creating a task to search for all the workstations of a domain. From the resulting list of workstations, you can navigate to see (secondary query) a list of all the jobs running on each of them.
   jobs. For example, you are creating a task to search for all the virtual workstations that are also fault-tolerant. From the resulting list of workstations, you can navigate to see (secondary query) a list of all the jobs running on each of them.

Results

After specifying all the required criteria, you can save your task or immediately run it to create a list of workstations that satisfies your filtering settings. For details, see Monitoring your items in the plan on page 178.

Related information

Workstation on page 50

Workstation types on page 231

Creating a task to Monitor Domains

Why and how you can create a Monitor Domains task.

About this task

To create a Monitor Domains task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. In the navigation toolbar, click **Monitoring & Reporting > Workload Monitoring > Monitor Workload** and follow the steps described in Monitoring your items in the plan on page 178.

   If you are familiar with conman, in the **Query** text box specify a query based on the conman showdomains syntax. Alternatively, click **Edit** to select the filter criteria from the list of options that is displayed.

2. In the General Filter section, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, in some of the results tables in the Periodic Refresh Options section, you can customize how often to refresh the information by specifying the refresh interval in seconds in hh:mm:ss format, with a minimum of 30 seconds and a maximum of 7200 seconds. For example, 00:01:10 means 70 seconds. If the value specified is not valid, the last valid value is automatically used. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table when the timer is in stop status. In this case, the value specified at task creation time is temporarily overwritten. You can filter the task results based on the domain name, or part of name (using wildcard characters). You can also configure the automatic refresh of the task results in the table.
3. In the Columns Definition section, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on workstations. Starting from the Monitor Domains task table of result, you can start further queries on the workstations associated to one of the domains in the table; the information to be retrieved with these secondary queries is specified in this panel.

Results

After specifying all the required criteria, you can save your task or immediately run it to create a list of domains that satisfies your filtering settings. For details, see Monitoring your items in the plan on page 178.

Monitoring your Workload

Controlling and managing scheduling activities and objects on plans with *tasks*.

To control and manage scheduling activities and objects in plans, you create and run *tasks*.

Note: You must create a connection to a remote IBM Workload Scheduler engine, before you can run tasks on it to obtain data.

You can create the following types of task:

Monitor Task

When you create a Monitor task, you define a query where you specify multiple criteria to search for items and to display the search results. You can then save, reuse, and share this task with other users, and modify it at any time.

Task sharing is enabled only if your role is authorized to share tasks. If you are not, contact the administrator or see: Limit task and engine sharing on page 38.

When you run the task, you launch the query, according to the filtering criteria, on all the objects associated to the IBM® Workload Scheduler connection you specified. A list of the objects that satisfy the search criteria is displayed when you run the task. You can view the objects resulting from your query, and their properties, and take actions on some of them.

Event Monitoring Task

When you create an Event Monitoring Task you define a query where you specify multiple criteria to search for specific monitoring objects stored on the database and to display the search results. Available monitoring objects are event rules, triggered actions, and operator messages. You can then save, reuse, and share this task with other users, and modify it at any time.

When you run the task, you launch the query, according to the filtering criteria, on all the objects associated to the IBM® Workload Scheduler connection you specified. A list of the objects that satisfy the search criteria is displayed when you run the task. You can view the objects resulting from your query, and their properties, and take actions on some of them.

Report Task

For information about this type of task, see Reporting on page 205.

Note:

To create a monitoring task, you can also define a monitoring task query by clicking **Monitoring & Reporting > Workload Monitoring > Monitor Workload**. For details about the complete procedure, see Monitoring your items in the plan on page 178.

Related information

Scheduling objects on page 57

Monitoring your Scheduling Environment on page 167

Orchestration Monitor overview

The Orchestration Monitor gives you granular control over your plan, enabling you to take direct action over every item in your plan.

The Orchestration Monitor is a Dynamic Workload Console section that functions like a control hub that oversees all of your workload associated with a specific engine. From there, you can monitor:

- Workstations
- Job streams
- Jobs
- Resources

Queries

You can use queries to search for specific items, and from there you can take direct control by performing manually a range of actions, depending on which item you have queried. Queries can also be saved and shared with other users or groups, to speed up your monitoring operations.

Tree view

A tree view is available to let you monitor your workload through the folders that contain the items in your plan. The tree view is updated in real time, so that you can always be aware of the folder system that contains the items you are monitoring.

Dependencies

Using the *Dependencies* action, you can monitor all the dependencies of job or a job stream displayed in a graphical way through a tile system. You can manually take action on the dependencies, releasing them or editing them for the job and job streams in your plan.

Operator instructions

When monitoring jobs and job streams, operators can access the **Operator instructions** panel from the **More actions** menu to retrieve definition information from the **Description** and **Documentation** fields. The availability of this information within the Orchestration Monitor facilitates rapid comprehension of job or job stream parameters during monitoring activities.

The **Operator instructions** panel displays definition information based on the selected item. For a job stream, the panel presents the content of the **Description** and **Documentation** fields from the job stream definition. For a job, the panel displays three sections:

Job stream

Displays the **Description** and **Documentation** fields of the job stream to which the selected job instance belongs.

Job instance

Displays the **Description** and **Documentation** fields of the selected job instance.

Job definition

Displays the **Description** field of the job definition referenced by the selected job instance.

Workflow details

Using the *Workflow details* panel, you can monitor information about plug-in jobs and list the operations that you have performed on the whole job or on one or more files in the job.

For z/OS users: To monitor your plug-in jobs from a Dynamic Workload Console different from the console where the job was defined, ensure that you stored the plug-in job' s `jar` file into the `<DWC_home>/usr/servers/dwcServer/resources/lib/zconn/applicationJobPlugIn` folder.

Monitor certificate expiration

The certificate expiration date for workstations is now visible in the results table when querying for workstations. You can make the certificate expiration column visible from the **Table configuration** menu.

For more information on how to use the Orchestration Monitor, see this Orchestration Monitor scenario on page 172

For details about how to add a z/OS engine to the list of discovered engines that are available from the Orchestration Monitor, see Mirroring the z/OS current plan to enable the Orchestration Monitor on page 175.

Monitoring scenario

About this task

Company StockKing needs to restock its distribution centers in Europe automatically. The company has automated the restocking process using IBM® Workload Scheduler and it can keep an eye on its operations thanks to the monitoring features. In this scenario, an operator monitors the job stream that restocks the German distribution center, called EU-GERSTOCK. The job stream runs daily and it includes the following jobs:

- UNITCHECK: it checks the number of units present in the distribution center and the distribution center' maximum capacity, calculating the difference.
- UNITORDER: it orders the number of units calculated by UNITCHECK.

- UNITCOURIER: it finds the first available courier within a certain cost parameter that is willing to deliver the units to the German distribution center.
- UNITCONFIRMATION: it saves and archives the delivery note generated by the courier after it collects the units for delivery.

In this scenario, the UNITORDER job does not complete successfully. The operator must take note of the number of units that need to be ordered and manually place an order for them. In addition, the operator must release the predecessor dependency of the UNITORDER job from the UNITCOURIER job to ensure that the UNITCOURIER job can launch automatically.

1. From the **Monitoring & Reporting** menu, click **Orchestration Monitor** page.
2. Click on the **Saved queries** tab and select the EU-GER query
3. In the results table select the EU-GERSTOCK job stream and click on the **Jobs** action
4. In the results table you can see the four jobs that constitute the job stream. The UNITORDER job is in **ERROR**. Manually order the number of units found by the UNITCHECK job
5. Select the UNITCOURIER and click on the **Dependencies** action
6. The job UNITORDER is the predecessor to the selected UNITCOURIER job. From the UNITORDER tile, **Release** the dependency

Results

The operator managed to restock its distribution center.

Configuring the Federator to mirror data on a database

Starting from the Dynamic Workload Console V10.2.3, the Federator component is automatically installed with the console to mirror (replicate) the events related to the z/OS current plan on a configured database.

Before you begin

The mirroring function has the following prerequisites:

- IBM Z Workload Scheduler v10.2, or later.
- **If you are using Db2 for z/OS, ensure that you have:**
  - Installed IBM Db2 for z/OS V13.1.503, or later.
  - Installed IBM Db2 Accessories Suite for z/OS for Db2 13 for z/OS V04.02.00, or later.
  - Defined a Workload Manager (WLM) policy for the required Db2 for z/OS environment and that a Db2 procedure for the WLM Application environment is up and running.

About this task

Data mirroring enables you to:

- Monitor your objects from the Orchestration Monitor page of the Dynamic Workload Console (for details, see Orchestration Monitor overview on page 171).

  Starting from DWC 10.2.4, you can also run OQL queries (for details, see Using Orchestration Query Language).
- Perform actions on the monitored objects through the REST APIs V2, with which you can also run OQL queries. (For details, see Using Orchestration Query Language.)

To configure the Federator, perform the following steps:

1. According to your operating system, browse to the `fed_variables.xml` file, which is created at installation time in the following folder:

   On Windows

   ```
   <DATA_dir>\usr\servers\dwcServer\configDropins\overrides
   ```

   On UNIX and z/OS

   ```
   <DATA_dir>/usr/servers/dwcServer/configDropins/overrides
   ```

2. Edit the file by setting the following variables:

   federator.secret

   The secret key string used to register the z/OS engine to the Federator, in the format UTF-8. Ensure that you use a character set that is compatible with the CODEPAGE parameter specified in the OPCOPTS initialization statement (for details about OPCOPTS, see *IBM Z Workload Scheduler: Customization and Tuning*).

   This variable is case-sensitive. You can specify up to 36 alphanumeric characters. Special characters, except blanks and single quotation marks, are allowed.

   mirroring.cleanup.interval

   Number of days after which the following records are deleted from the database:
   - Records flagged as `archived`
   - Records not updated over that time range which are no longer in the current plan.

   The default is 0, meaning that the database stores only the records that are present in the current plan.

   uno.federator.engine.jwt.expiration.duration

   The interval of time after which the authentication token expires, set in the ISO 8601 format. For example, to specify an interval of 50 days, set `P50D`. To specify an interval of 45 hours, set `PT45H`. The default is 30 days.

3. According to your operating system, browse to the `jwtsso.xml` file, which is created at installation time in the following folder:

   On Windows

   ```
   <DWC_home>\usr\servers\dwcServer\configDropins\templates
   ```

On UNIX and z/OS

```
<DWC_home>/usr/servers/dwcServer/configDropins/templates
```

4. Edit the following variable in the `jwtsso.xml` file by replacing the value 9443 with the actual port number that you are using to connect with the Dynamic Workload Console:

```
<variable name="dwc.portNumber" value="9443"/>
```

5. Copy the `jwtsso.xml` file to the following path:

   On Windows

   ```
   <DATA_dir>\usr\servers\dwcServer\configDropins\overrides
   ```

   On UNIX and z/OS

   ```
   <DATA_dir>/usr/servers/dwcServer/configDropins/overrides
   ```

6. If you are upgrading from any version earlier than 10.2.3, browse to the `datasource_db_vendor.xml` file, which is created at installation time in the following folder:

   On Windows

   ```
   <DWC_home>\usr\servers\dwcServer\configDropins\templates\datasources
   ```

   On UNIX and z/OS

   ```
   <DWC_home>/usr/servers/dwcServer/configDropins/templates/datasources
   ```

7. In the file, locate the section named `jndiName="jdbc/feddb"`.
8. Copy the section and paste it into the `datasource.xml` file located in the following path:

   On Windows

   ```
   <DATA_dir>\usr\servers\dwcServer\configDropins\overrides
   ```

   On UNIX and z/OS

   ```
   <DATA_dir>/usr/servers/dwcServer/configDropins/overrides
   ```

9. Stop and restart the Dynamic Workload Console server.

Mirroring the z/OS current plan to enable the Orchestration Monitor

You can mirror (replicate) a z/OS current plan to the database referenced by the Federator to enable the monitoring of your jobs, job streams, and workstations from both the Orchestration Monitor and REST APIs V2. In this way you run your queries on the database through the Federator without affecting the Z controller, which results in a performance enhancement of the product.

Registering a z/OS engine to the Federator

Before you begin

Ensure that you have configured the Federator, as explained in Configuring the Federator to mirror data on a database on page 173.

The mirroring function has the following prerequisites:

- IBM Z Workload Scheduler v10.2, or later.
- **If you are using Db2 for z/OS, ensure that you have:**
  - Installed IBM Db2 for z/OS V13.1.503, or later.
  - Installed IBM Db2 Accessories Suite for z/OS for Db2 13 for z/OS V04.02.00, or later.
  - Defined a Workload Manager (WLM) policy for the required Db2 for z/OS environment and that a Db2 procedure for the WLM Application environment is up and running.

About this task

To add your z/OS engine to the list of discovered engines that are available from the Orchestration Monitor, register the engine to the Federator by performing the following steps (for details about the initialization statements, see *Customization and Tuning*):

1. On the Z controller:
   a. Define the required parameters of the MIRROPTS statement.
   b. Set the CODEPAGE and TIMEZONE parameters of the OPCOPTS statement, to set the Z controller code page and time zone respectively.
2. On the z/OS server started task:
   a. In the TCPOPTS statement, set the following parameters:
      - SSLKEYSTORETYPE(SAF)
      - SSLKEYSTORE(MDMCSTMRING)
      - SSLLEVEL(FORCE)
   b. In the `connectionFactory.xml` file, set `useSsl="true"`.

Note: Modifying the Federator by adding, deleting, or editing an engine might take up to five minutes to be shown in the list of discovered engines.

Upon registration completion, the Federator generates the authentication token to secure communications with the Z controller. Before expiration time, the token is automatically renewed by the Federator and sent to the controller: in this way, theoretically, the token never expires. The only exception occurs if the connection between the Federator and controller is not established for a long period of time; to resolve this issue see Troubleshooting on page 178.

When a z/OS engine is registered to the Federator, data is mirrored to the configured database from the DB Filler subtask through the Federator, as shown in Figure 9: Mirroring data onto the database on page 176.

Figure 9. Mirroring data onto the database

When any job, job stream or workstation is created, updated, or deleted (either dynamically or by a daily planning EXTEND or REPLAN), the Z controller sends the related events to the Federator, which stores the information on the configured database. The objects that no longer are in the current plan are archived.

The supported databases are those supported by the Dynamic Workload Console.

Enabling the usage of the Orchestration Monitor

Before you begin

Ensure that you are connected to a Dynamic Workload Console V10.2.3 or later.

About this task

By registering a z/OS engine to the Federator, it is added to the list of discovered engines in the Dynamic Workload Console. You can then monitor your objects from both the following interfaces:

- Orchestration Monitor page of the Dynamic Workload Console (for details, see Orchestration Monitor overview on page 171).
- REST APIs V2, with which you can also run OQL queries. For details, see Using Orchestration Query Language.

Note: You can monitor only the objects for which you are granted permission by RACF.

You monitor the objects on the configured database from the Dynamic Workload Console through the Federator, as shown in Figure 9: Mirroring data onto the database on page 176.

Figure 10. Monitoring data from the Orchestration Monitor

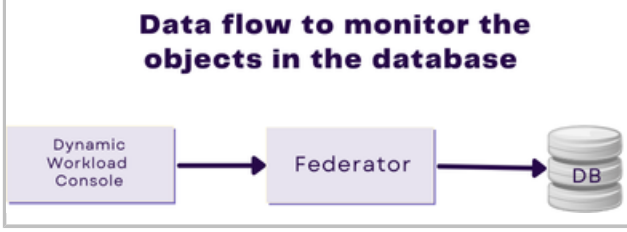
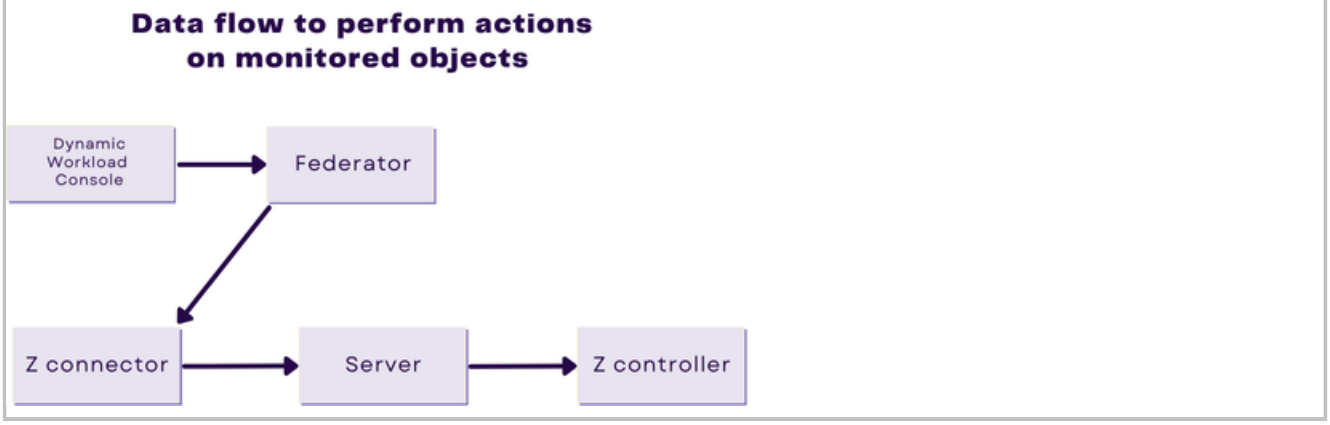


Figure 11: Data flow to perform actions on the monitored objects on page 178 shows the communication flow to perform actions on the objects monitored from the Orchestration Monitor.

Figure 11. Data flow to perform actions on the monitored objects



Troubleshooting

About this task

This section describes how troubleshoot possible issues.

The Z controller cannot connect to the Federator because the authentication token has expired

Before expiration time, the authentication token is automatically renewed by the Federator and sent to the controller to prevent communication interruptions. If, for connection problems, the Federator cannot send the renewed token the following scenarios might occur:

The secret key string that is set in MIRROPTS(SECRET) is still coincident with the value set in the `federator.secret` property of the Federator

When the connection is established again, the Federator generates a new token and sends it to the controller to be stored locally.

The secret key string that is set in MIRROPTS(SECRET) is no longer coincident with the value set in the `federator.secret` property of the Federator

When the controller tries to connect to the Federator, the communication is not established. You are required to modify the SECRET parameter of the MIRROPTS to be coincident with the value set in the `federator.secret` property. Then restart the controller to be registered again on the Federator.

Monitoring your items in the plan

Define a query to monitor items in the plan. The items for which you can create a monitoring query are: jobs, job streams, and workstations.

About this task

To monitor items in the plan, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. In the navigation toolbar, click **Monitoring & Reporting > Orchestration Monitor**.
2. From the **Engine** drop-down list, select the check box related to the engine where the task must run.
3. From the item drop-down list, select the type of item you want to monitor.

   For a distributed engine, you can create a monitoring task query for: **Job**, **Job Stream**, **Workstation**.
4. From the **List Plans** drop-down list, you can select the current plan or an archived plan related to the selected engine.
5. In the **Query** text box, type the query that specifies the filter criteria to apply to the object type you selected. If you are familiar with the conman syntax, the query syntax is quite similar; for example, the syntax for filtering jobs is similar to conman showjobs. If you are not familiar with conman, click `Edit` to create your query by selecting the options from the filter criteria that are displayed.

   The following rules apply to the query syntax:
   - To separate filter parameters, use the plus symbol (+).
   - To separate the workstation name from the job stream name, use the hash key symbol (#) for a distributed engine or the exclamation mark (!) for a z/OS engine.
   - To replace one or more alphanumeric characters, use the at sign symbol (@).

     Note: For a z/OS engine, when you use special characters you must enclose the whole string between single quotation marks (').

   - To separate the workstation name, job stream name from the job name, use the period (.) .
   - To separate the folders, use the slash (/) .

   Example

   For a *distributed* engine, to display the status of all the jobs in the `acctg` job stream on workstation `site3`, type the following string in the **Query** text box:

   ```
   site3#/foo/acctg.@
   ```

   or:

   ```
   site3#/foo/acctg
   ```

   For a *z/OS* engine, to display the status of all the jobs in the `acctg` job stream on workstation `site3`, type the following string in the **Query** text box:

   ```
   'site3!/foo/acctg.@'
   ```

   For more example about queries and syntax, see Example on page 180.
6. Press **Enter** to run the query immediately.

   Result

The results are displayed in table format. You can click **Advanced Query** to specify further information regarding the query.

7. Click **Save**.
8. Type a name for the monitoring query in the **Insert query name** text box.
9. Optionally, check the **Share with** box and select the users you want to share the query with.
10. Click **Save** to save the query.

Results

You have created your query that, when selected, creates a list of results satisfying your filtering criteria and showing, for each item in the list, the information contained in the columns you selected to view.

Example

The following examples show the syntax used to create some specific queries:

Job query in a distributed environment

To query for all jobs, in all job streams and on all workstations on a specific distributed engine, with the following characteristics:

- Having a dependency on a workstation with a name beginning with `FTA_1`
- Beginning with `Job_A` in a job stream beginning with `JS_A`

  in FOLDERA folder
- Scheduled to run at `10 a.m. on October 31, 2015`
- Currently in `Waiting` state with a priority in the range of `1 - 50`

specify the following query in the query line:

```
/@/@#/@/@.@+follows=/@/FTA_1#/FOLDERA/JS_A@(1000
 10/31/2015).JOB_A@+state=#Waiting,#Ready+priority=1,50
```

Job query in a z/OS environment

To query for all jobs in a job stream ending in `001`, on all workstations beginning with `HR` in a z/OS environment, with the following characteristics:

- Having an internal status of `Interrupted` and `Error`
- Having a priority of `1`

specify the following query in the query line:

```
HR@!@001.@+jsfrom=1000 10/10/2015+state=E,I+priority=1
```

Monitoring event rules

Why and how you can create a monitor event rule instances task.

About this task

To create a Monitor Event Rules task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. In the navigation bar, click **Monitoring and Reporting** > **All Configured Tasks** > **New**.
2. In the Create Task panel, under **Event Monitoring Task**, select **Monitor Event Rules** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a IBM Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. Here you also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter section, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, in some of the results tables in the Periodic Refresh Options section, you can customize how often to refresh the information by specifying the refresh interval in seconds in hh:mm:ss format, with a minimum of 30 seconds and a maximum of 7200 seconds. For example, 00:01:10 means 70 seconds. If the value specified is not valid, the last valid value is automatically used. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table when the timer is in stop status. In this case, the value specified at task creation time is temporarily overwritten. You can search for rule instances based on their status, type, or trigger timestamps.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, and the kind of rule that generated them. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on event rules (the event rule definition stored in the database). The information to be retrieved with these secondary queries is specified in this panel.
7. Click **next** to finalize the task creation, you can choose either to run the task immediately or finish to save the task and exit the wizard.

Results

You have created your query that, when run, lists the event rule instances satisfying your filtering criteria and shows, for each event rule in the list, the information you selected to view.

Related information

Creating an event monitoring tasks on page 182

Event management configuration on page 13

Creating an event monitoring tasks

Information about event monitoring tasks and how to create them.

You create and run an event monitoring task by specifying a filter and running a search to get information about event monitoring related objects.

The information retrieved when running event tasks is stored in the IBM Workload Scheduler databases, therefore, to run event tasks you must be connected to a IBM Workload Scheduler engine and must be authorized in the IBM Workload Scheduler security file to access those objects in the database.

You can create event monitoring tasks to query for:

Event rule definitions

The template that defines an event rule consists of:

- One or more events, defined with its properties.
- The relationship between the specified events (they can be randomly grouped or listed in chronological order).
- The actions that must be performed when all the event conditions are satisfied.

Event rules

The instance of a rule definition in the plan.

Triggered action

The actual occurrence of an action defined in the event rule and triggered when the event conditions have been satisfied.

Operator messages

The instance of the **MessageLogger** action specified in the event rule definition. It provides information about the result of an event rule instance on a repository stored in the IBM Workload Scheduler relational database.

Related information

Monitoring event rules on page 180

Monitor Triggered Actions on page 182

Monitor Operator Messages on page 184

Monitor Triggered Actions

Why and how you can create a Monitor Triggered Actions task.

About this task

To create a Monitor Triggered Actions task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. In the navigation bar, click **Monitoring & Reporting > Monitor Triggered Actions > New**
2. In the Create Task panel, select **Event Monitoring Task** > **Monitor Triggered Actions** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a IBM Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation.
5. In the General Filter section, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, in some of the results tables in the Periodic Refresh Options section, you can customize how often to refresh the information by specifying the refresh interval in seconds in hh:mm:ss format, with a minimum of 30 seconds and a maximum of 7200 seconds. For example, 00:01:10 means 70 seconds. If the value specified is not valid, the last valid value is automatically used. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table when the timer is in stop status. In this case, the value specified at task creation time is temporarily overwritten. You can search for triggered actions based on the type of rule instance that triggers them or on their scope. The scope of an action (or an event) is the set of properties that most characterize it.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for each of the actions resulting from your query, you might want to see the status, type, or associated message. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on event rule instances. The information to be retrieved with these secondary queries is specified in this panel.
7. Click **Next** to finalize the task creation, you can choose either to run the task immediately or finish to save the task and exit the wizard.

Results

You have created your query that, when run, lists the event rule instances satisfying your filtering criteria and shows, for each event rule in the list, the information you selected to view.

Related information

Creating an event monitoring tasks on page 182

Event management configuration on page 13

Monitor Operator Messages

Why and how you can create a Monitor Operator Messages task.

About this task

To create a Monitor Operator Messages task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. In the navigation bar, click **Monitoring & Reporting > Monitor Operator Messages > New**
2. In the Create Task panel, select **Event Monitoring Task** > **Monitor Operator Messages** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a IBM Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. You also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. In the General Filter section, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, in some of the results tables in the Periodic Refresh Options section, you can customize how often to refresh the information by specifying the refresh interval in seconds in hh:mm:ss format, with a minimum of 30 seconds and a maximum of 7200 seconds. For example, 00:01:10 means 70 seconds. If the value specified is not valid, the last valid value is automatically used. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table when the timer is in stop status. In this case, the value specified at task creation time is temporarily overwritten. You can search for operator messages based on their severity, time stamp, or scope. The scope of an operator message is the set of properties that most characterize it.
5. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for each of the operator messages resulting from your query, you might want to see the severity, the type of associated event, or the group in whose queue the message is. You can then drill down into this information displayed in the table and navigate it.
6. Click **Next** to finalize the task creation, you can choose either to run the task immediately or finish to save the task and exit the wizard.

Results

You have created your query that, when run, lists the event rule instances satisfying your filtering criteria and shows, for each event rule in the list, the information you selected to view.

Related information

Creating an event monitoring tasks on page 182

Event management configuration on page 13

Controlling job and job stream processing

How you can control job and job stream processing.

In the Dynamic Workload Console, you can control job and job stream processing by specifying dependencies, setting properties, and running actions against the job or job stream.

Using dependencies to control job and job stream processing

A *dependency* is a prerequisite that must be satisfied before processing can proceed. You can define dependencies for both jobs and job streams to ensure the correct order of processing. You can use these types of dependencies:

On completion of jobs and job streams

A job or a job stream must not begin processing until other jobs and job streams have completed. May be defined to require success or just completion.

On satisfaction of specific conditions by jobs and job streams

A job or a job stream, named *successor*, must not begin processing until other jobs and job streams, named *predecessor*, have met one, all, or a subset of specific conditions that can be related to the status of the job or job stream, the return code, output variables, or job log content. When the conditions are not met by the predecessor, then any successor jobs with a conditional dependency associated to them are put in suppress state. Successor jobs with a standard dependency or no dependency at all defined run normally.

Resource

A job or a job stream needs one or more resources available before it can begin to run.

File

A job or a job stream needs to have one or more files meet the specified criteria before it can begin to run.

Prompt

A job or a job stream needs to wait for an affirmative response to a prompt before it can begin to run.

You can define up to 40 dependencies for a job or job stream. If you need to define more than 40 dependencies, you can group them in a join dependency. In this case, the join is used simply as a container of standard dependencies and therefore any standard dependencies in it that are not met are processed as usual and do not cause the join dependency to be considered as suppressed. For more information about join dependencies, see the section about joining or combining conditional dependencies and the **join** keyword in the *User's Guide and Reference*. In a IBM Workload Scheduler network, dependencies can cross workstation and network boundaries.

On completion of jobs and job streams

A job or a job stream must not begin processing until other jobs and job streams have completed. May be defined to require success or just completion.

Resource

A job or a job stream needs one or more resources available before it can begin to run.

In addition to this, each job needs the workstation where it is scheduled to run to be available.

To add a dependency to a job or to a job stream from the Graphical Designer, see Controlling jobs and job streams processing on page 108.

You can also add a dependency from the panel displayed as result of your monitor task related to jobs or job streams by performing the following steps:

1. In the query result panel select a job or a job stream and click **Dependencies**.
2. In the Dependencies panel expand the section related to the dependency type you want to add and click **Add predecessor**.
3. Choose between **Job** or **Job stream**
4. Enter the required information and click **OK**.

   For all the details about options and fields displayed in the panel, expand the **Contextual help** tab at the bottom of the properties panel.

Using time restrictions to control job and job stream processing

Time restrictions can be specified for both jobs and job streams.

For a specific job or job stream you can specify the time that processing begins, **earliest start**, or the time after which processing can no longer be started, **latest start**. By specifying both, you define a time interval within which a job or job stream runs. You can use them as time dependencies.

You can also specify a **deadline** to specify the time within which a job or a job stream must complete. Jobs or job streams that have not yet started or that are still running when the deadline time is reached, are considered late in the plan. The deadline does not prevent jobs or job streams from starting.

You can also specify a **maximum duration** or a **minimum duration** for a job defined within a job stream. If a job is running and the maximum duration time has been exceeded, then the job can either be killed or can continue to run. If a job does not run long enough to reach the minimum duration time, then the job can be set to Abend status, to Confirm status awaiting user confirmation, or it can continue running.

For jobs you can also specify a **repeat range**; for example, you can have IBM® Workload Scheduler launch the same job every 30 minutes between 8:30 a.m. and 1:30 p.m.

To specify time restrictions for a job or a job stream, perform the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. In the workspace, select the job stream you want to modify. For information about editing an object, see the topic about designing your workload in *Dynamic Workload Console User's Guide*.
3. Go to the **Time restrictions** section in the properties panel on the right.
4. Enter the time restriction properties.

   For all the details about options and fields displayed in the panel, expand the **Contextual help** tab at the bottom of the properties panel.

Using job priority and workstation fence to control distributed job processing

IBM® Workload Scheduler has its own queuing system, consisting of levels of *priority*. Assigning a priority to jobs gives you added control over their precedence and order of running.

The *fence* provides another type of control over job processing on a workstation. When it is set to a priority level, it only allows jobs whose priority exceeds the fence value to run on that workstation. Setting the fence to 40, for example, prevents jobs with priorities of 40 or less from being launched.

To specify job priority for a job, perform the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. In the workspace, select the job stream you want to modify. For information about editing an object, see the topic about designing your workload in *Dynamic Workload Console User's Guide*.
3. Go to the **Scheduling options** section in the Properties panel.
4. Enter the job priority and save the job stream.

   For all the details about options and fields displayed in the panel, expand the **Contextual help** tab at the bottom of the properties panel.

You can also add a job priority from the panel displayed as result of your monitor task related to jobs by performing the following steps:

1. In the query result panel select a job and click **More Actions** > **Priority**.
2. In the Set Priority panel specify a priority value and click **OK**.

To set a workstation fence perform the following steps:

1. In the panel displayed as results of your monitor workstation task, select the workstation and click **More Actions** > **Fence**.
2. In the Set Fence panel specify a fence value and click **OK**.

Using limits to control job and job stream processing

The *limit* provides a means of setting the highest number of jobs that IBM Workload Scheduler is allowed to launch. You can set a limit:

- In the job stream definition
- In the workstation definition

Setting the limit on a workstation to 25, for example, allows IBM Workload Scheduler to have no more than 25 jobs running concurrently on that workstation.

To specify a limit for a job stream, perform the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. In the workspace, select the job stream you want to modify. For information about editing an object, see the topic about designing your workload in *Dynamic Workload Console User's Guide*.
3. Go to the **Scheduling options** section in the Properties panel.
4. Enter the limit value and save the job stream.

   For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

You can also add a limit from the panel displayed as result of your monitor job stream task by performing the following steps:

1. In the query result panel select a job stream and click **More Actions** > **Limit**.
2. In the Set Limit panel specify a new limit value and click **OK**.

To set a workstation limit perform the following steps:

1. In the panel displayed as result of your monitor workstation task, select the workstation and click **More Actions** > **Limit**.
2. In the Set Limit panel specify a new limit value and click **OK**.

Using job confirmation to control job processing

There might be scenarios where the completion status of a job cannot be determined until you have performed some tasks. You might want to check the results printed in a report, for example. In this case, you can set in the job definition that the job *requires confirmation*, and IBM Workload Scheduler waits for your response before marking the job as successful or failed.

To specify that a job requires confirmation, perform the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. In the workspace, select the job stream you want to modify. For information about editing an object, see the topic about designing your workload in *Dynamic Workload Console User's Guide*.
3. Go to the **Scheduling options** section in the Properties panel.

4. Check **Requires confirmation**.

   For all the details about options and fields displayed in the panel, expand the **Contextual help** tab at the bottom of the properties panel.

Using job recovery actions to control job processing

When you create a job definition, you can specify the type of recovery you want performed by IBM Workload Scheduler if the job fails. The predefined recovery options are:

- Continue with the next job. You can also continue with the next job after a prompt is issued which requires a response from the operator.
- Stop and do not start the next job. You can also continue with the next job after a prompt is issued which requires a response from the operator.
- Run the failed job again. You can specify also how often you want IBM Workload Scheduler to rerun the failed job and the maximum number of rerun attempts to be performed. If any rerun in the sequence completes successfully, the remaining rerun sequence is ignored and any job dependencies are released. You can optionally decide to also rerun the successors of the parent job, either all successors in the same job stream, or all successors, both in the same job stream and in other job streams, if any. In the Dynamic Workload Console, you can easily view the list of all job successors before rerunning them from the **Orchestration Monitor**, by selecting the job and clicking **More Actions> Rerun with successors**.

In addition, you can specify other actions to be taken in terms of recovery jobs and recovery prompts. For example, if a job fails, you can have IBM Workload Scheduler automatically run a recovery job, issue a recovery prompt that requires an affirmative response, and then run the failed job again. For more information about recovery actions, see the section about defining job rerun and recovery actions in *User's Guide and Reference*.

To specify the job recovery actions, perform the following steps:

1. From the **Design** menu, click **Graphical Designer** page.
2. In the workspace, select the job you want to modify. For information about editing an object, see the topic about designing your workload in *Dynamic Workload Console User's Guide*.
3. Select the **Recovery options** section in the Properties panel.
4. Enter the recovery **Action** and the remaining information, as necessary. Then save the job.

   For all the details about options and fields displayed in the panel, expand the **Contextual help** tab at the bottom of the properties panel.

Modifying job definitions from job instances

After your jobs have been submitted into the current plan, you can control their processing by modifying the job definition directly from the **Orchestration Monitor** or **Job Stream View**, without having to go back to the original job definition in the database by using the following actions:

- Set hold for definition
- Remove hold for definition
- Set `No operation` for definition
- Remove `No operation` for definition

Note: These actions do not apply to the selected job-in-job-stream instance, but to the future instances of the job.

You can find the description of what each action does below:

Set no operation for definition

Sets the **No operation** property of the selected job-in-job-stream definition in the database, so that any future instance of the job is created with this property, which sets the job instances status to **success** without the job actually running.

Remove no operation for definition

Removes the **No operation** property of the selected job-in-job-stream definition in the database, so that any future instance of the job is created without this property and it can run and perform its function as designed.

Set hold for definition

Sets the priority of the selected job-in-job-stream definition in the database to 0, so that any future instance of the job is created with this priority.

Remove hold for definition

Sets the priority of the selected job-in-job-stream definition in the database to 10, so that any future instance of the job is created with this priority.

To modify job definitions starting from a job instance, perform the following steps:

From the Orchestration Monitor

1. **Monitoring and Reporting > Orchestration Monitor**.
2. Run a query on your jobs.
3. From the resulting list of jobs, select the job instance that you want to modify and click **More Actions**.
4. From the drop-down menu, select the desired action.

From the Job Stream View

1. **Monitoring and Reporting > Orchestration Monitor**
2. Run a query on your job streams.
3. From the resulting list of job streams, select a job stream and click **More Actions > Job Stream View**.
4. Select the block corresponding to the job instance that you want to modify and click the **... More Actions** button.
5. From the drop-down menu, select the desired action.

Chapter 11. Working with Plans

The main tasks that involve plans.

This section contains the main tasks that involve plans. You can find information about selecting the working plan, creating trial and forecast plans, monitoring the progress of a plan, and generating a graphical plan view or a preproduction plan view.

Selecting the working plan

How to list all the plans that are available on a specific engine.

From the **Manage Available Plans**, you can specify some filtering criteria to retrieve a list of plans. You can also generate a trial plan or a forecast plan.

Follow these steps to define a filter and run a query to create a list of available plans:

1. From the navigation toolbar, click **Planning** > **Workload Forecast** > **Manage Available Plans**.
2. In the Manage Available Plans panel:
    a. Under **Select an Engine**, select the engine from which you want retrieve the list of plans.
    b. Under **Select Plan Type**, click the corresponding check box to select the type of plan you want to list. Selections are mutually exclusive. By default, all available plans are listed.
    c. Under **Select Plan Name**, specify the name of the file containing the plan you want to search. You can use wildcard characters.
    d. Click **Display Plans List** to generate a list of plans.

Clicking on **Display Plans List**, you can see a table containing the list of plans you are searching for.

On top of the table that lists the available plans, you can see the engine connection you are using together with the default and active plan.

The list of available plans shows the following set of information for each plan:

Type

The type of the plan. Possible settings are:

Blank

Current plan that is in process in the IBM® Workload Scheduler environment.

Symnew

Temporary files containing the information about the next production period based on the job and job stream dependencies defined in the IBM® Workload Scheduler database.

Forecast

Forecast plan. It contains a projection over a selected time interval of the current production plan based on the job and job stream dependencies defined in the IBM® Workload Scheduler database.

Trial

Trial plans. It contains a projection of the current plan for a different period.

Archived

Archived plan. It is a copy of an old production plan that ran in the IBM® Workload Scheduler environment and that is now stored in the database.

File Name

The name of the file containing the plan.

Plan Start

The date and time when the plan starts.

Plan End

The date and time when the plan ends.

Schedule Date

The date when the plan was created.

Earliest Start

The actual date and time the plan began running. This column is blank for trial and forecast plans.

Last Updated

The time that the plan file was last updated.

Run Number

The run number associated to the plan. The run number is the total number of times the plan was generated. This number is zero for trial and forecast plans.

Size

The size of the file that contains the plan, in records.

From the header of the table, you can also generate a trial plan or a forecast plan. For further information about the creation of these two plans, see Generating Trial and Forecast Plans on page 192.

Related information

Plans on page 86

Generating Trial and Forecast Plans

How you create either a new trial or forecast plan.

About this task

To create a new plan, perform the following steps:

1. From the navigation toolbar, click **Planning** > **Workload Forecast**

   Create Trial Plan

   To create a trial plan. The Create Trial Plan panel is displayed.

   Create Forecast Plan

   To create a new forecast plan. The Create Forecast Plan panel is displayed.

2. Under the **Plan Information** section, enter the required information:

   Engine Name

   In the drop-down menu, select the engine where you want to create the plan. Only the engine connections that you created are available in the menu.

   Plan Filename

   Assign a name to the file that contains the plan. This is a mandatory field.

3. Under the **Plan Start** section, assign the date and time when the plan starts. Because the trial plan is mainly an extension of an existing and processing current plan, if you selected to create a new trial plan and a current plan is available on the engine, these fields are grayed out and their values are the same as the current plan end date. If you selected to create a new trial plan and a current plan is not available on the engine, or if you selected to create a new forecast plan, you can enter a date and time for the plan to start.
4. Under the **Plan End** section, assign one of the following values:
   Choose from:
   - A date and time when the plan ends.
   - The number of days and hours the plan lasts.

   By default the plan covers a one-day time interval.
5. Under the **Plan Time Zone** section, choose the time zone used in the plan.
6. Click **OK** to create the plan.

Related information

Plans on page 86

Display a graphical preproduction plan

How to view a graphical representation of a preproduction plan from the Dynamic Workload Console.

About this task

The preproduction plan is used to identify in advance the job stream instances and the job stream dependencies involved in a specified time period.

This improves performance when generating the production plan by preparing in advance a high-level schedule of the predicted production workload.

The preproduction plan contains:

- The job stream instances to be run during the time interval covered by the plan.
- The external dependencies that exist between the job streams and jobs included in different job streams.

From the Dynamic Workload Console you can view the preproduction plan graphically. You can open the preproduction plan in view mode only; you cannot extend it from this panel.

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.

To open the preproduction plan view, perform the following procedure:

1. From the navigation toolbar, click **Planning** > **View Preproduction Plan**.
2. In the displayed panel, select the distributed engine whose preproduction plan you want to view.
3. Optionally, specify a filter to reduce the number of job streams shown in the view. Only the job streams matching the string 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 panel.
4. 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.

Results

The graphical view of the preproduction plan is displayed. In this view, you can view the job streams included in the plan together with their job stream dependencies. Each box represents a job stream, whose color indicates the status. By default, it shows a maximum number of 1,000 job streams.

If you want to change this setting, modify the property **preProdPlanViewMaxJobstreams** in the Override graphical view limits on page 29 file.

There are several actions you can perform on the objects displayed in the Preproduction Plan View:

- Right-click a job stream, to open the job stream definition within Workload Designer. You can modify a job stream from the Workload Designer and then reload the updated preproduction plan view.
- Use the actions available from the toolbar:
  - Zoom in and zoom out to better view segments of the plan.
  - Print the plan or export it to either .svg (Scalable Vector Graphics) or .png (Portable Network Graphics) format.
  - Highlight the dependencies in the view.
  - Modify the original filter criteria that was set to change the objects displayed.

Related information

Preproduction plan on page 89

Chapter 12. Submitting Workload on Request in Production

How you can insert jobs or job streams into your workflow at any time.

About this task

In addition to the jobs and job streams that are scheduled to run in production, you can also submit at any time other jobs and job streams. However, these jobs and jobs streams are not considered when identifying predecessors for jobs and job stream dependencies.

In production you can:

Submit Ad Hoc Jobs

This is a job that is:

- Not defined in the database.
- Used to run a command or a script in production.

Submit Predefined Jobs

This is a job that is:

- Defined in the database.

Submit Predefined Job Streams

This is a job stream that is:

- Defined in the database.

See the following sections for instructions about inserting each of these types.

Submitting ad hoc jobs

How to add an ad hoc job to the current plan.

About this task

To add an ad hoc job to the current plan, perform the steps listed below. You cannot submit ad hoc jobs to the <env_ID>_CLOUD pool which is automatically created when you request your trial. To submit an ad hoc job, you need to install an agent.

1. From the navigation toolbar, click **Planning > Workload Submission > Submit Ad Hoc Jobs** .
2. In the displayed panel, select, from the drop-down list, the engine where you want to run the job, and click **Go**.

3. Enter all the required information about the job that you want to add. For more details about the information to enter in this panel, see the screen help, which you can open by clicking the question mark located at the top-right corner of the panel.
4. Click **OK** to save your changes, exit the panel and submit the job.

Submitting predefined jobs

How to insert a predefined job into the production plan.

About this task

To add a predefined job to the current plan, perform the steps listed below. You cannot submit predefined jobs to the <env_ID>_CLOUD pool which is automatically created when you request your trial. To submit predefined jobs, you need to install an agent.

1. From the navigation toolbar, click **Planning > Workload Submission > Submit Predefined Jobs** .
2. In the displayed panel, select, from the drop-down list, the engine where you want to run the job, and click **Go**.
3. Enter all the required information about the job that you want to add. For more details about the information to enter in this panel, see the screen help, which you can open by clicking the question mark located at the top-right corner of the panel.
4. Click **OK** to save your changes, exit the panel and submit the job.

Submitting predefined job streams

How to insert a predefined job stream into the current plan.

About this task

To add a predefined job stream to the current plan, perform the following steps:

1. From the navigation toolbar, click **Planning > Workload Submission > Submit Predefined Job Streams**.
2. In the displayed panel, select, from the drop-down list, the engine where you want to run the job.
3. Enter all the required information about the job stream that you want to add. For more details about the information to enter in this panel, see the screen help, which you can open by clicking the question mark located at the top-right corner of the panel. To find the job stream that you want to submit, you can launch searches based on part of the job stream name, workstation name, alias or variable table associated to it.
4. Optionally, specify a scheduled time to submit the job stream. The scheduled time refers to the submission of the job stream in the current production plan to evaluate dependencies resolution.
5. **Note:** if the job stream has no dependencies it will start immediately. If you want to have an earliest start time specify it in the properties panel.

   After completing the panel, click **Submit** to submit your job stream in the plan. Close the tab to exit discarding your changes.

Setting properties for ad hoc jobs and predefined jobs and job streams

This topic describes how to specify all the settings and properties required to add jobs or job streams to the current plan.

About this task

To set the properties required to add jobs or job streams to the current plan, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. Enter the required information in the **General** section.
2. For predefined and ad hoc jobs only: under the **Task** tab, enter the task properties for the job in the displayed panel.
3. Select the **Time Restrictions** tab, and enter the required information in the displayed panel to set time restrictions for the job or job stream.
4. Select the **Resources** tab to set resource dependencies.
   Choose from:
   - To create a new resource, click **New** and enter the required information in the Info panel.
   - To delete an existing resource, select it from the list and click **Delete**.
   - To modify a resource listed in the table, double-click its name and edit its properties in the Info panel.
5. Select the **Prompts** tab to set prompts as dependencies for the job or job stream.
   Choose from:
   - To create a new prompt, click **New** and enter the required information in the Info panel.
   - To delete an existing prompt, select it from the list and click **Delete**.
   - To modify a prompt listed in the table, double-click its name and edit its properties in the Info panel.
   - **Note:** When submitting ad hoc prompt no name is associated, therefore it is not possible to search for the name in the monitoring panel.
6. Select the **Files** tab to set file dependencies for the job or job stream.
   Choose from:
   - To create a new file, click **New** and enter the required information in the Info panel.
   - To delete an existing file, select it from the list and click **Delete**.
   - To modify the file properties, double-click the file and edit the settings in the displayed table.
7. Select the **Internetwork Predecessors** tab to add predecessor dependencies from a remote IBM Workload Scheduler network. The displayed panel shows existing internetwork predecessor properties.
   Choose from:
   - To create a new internetwork predecessor, click **New** and enter the required information in the Info panel. Click the **...** (Browse) button to search for and select the name of the network agent. Internetwork dependencies require that a network agent is configured to communicate with the external scheduler network.
   - To delete an existing internetwork predecessor, select it from the list and click **Delete**.
   - To modify an existing internetwork predecessor properties, double-click it, and edit its settings.

8. Select the **Predecessors** tab to set predecessor dependencies for the job or job stream. The displayed panel shows existing predecessor properties. Select **Conditional Dependencies** option to specify the type of the conditional dependency.

   Choose from:

   - To create a new predecessor, click **New** and enter the required information in the displayed panel.
   - To delete an existing predecessor, select it from the list and click **Delete**.
   - To modify an existing predecessor properties, double-click it, and edit its settings in the displayed table.

Chapter 13. Keeping track of changes to scheduling objects

How to keep track of changes in your environment

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

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

Historical information is available for each scheduling object in the Workload Designer in the **Versions** tab and in the results of the **Monitor Workload** query.

Auditing justification and reporting

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

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

They can produce a report containing information on all changed objects or on a subset of such objects.

You can audit both information available in the plan and in the database. By default, auditing is enabled. To disable auditing, use the following global options:

enDbAudit

Enables auditing of the information available in the database.

enPlanAudit

Enables auditing of the information available in the plan.

You can store auditing information in a file, in the IBM® Workload Scheduler database, or in both. To define in which type of store to log the audit records, use the **auditStore** global option. For more information about global options, see the related section in the *Administration Guide*.

To enforce a policy by which each user making a change to an object from the Dynamic Workload Console must provide a justification for the change, see Auditing justification and reporting on page 200.

To enforce the policy when making changes from the command-line programs see:

- The composer command-line program in the *User's Guide and Reference* .
- The conman command-line program in the *User's Guide and Reference*.
- The optman command-line program in the *Administration Guide*.
- The wappman command-line program in the *User's Guide and Reference*.

Versioning

Application developers and schedulers can compare and restore previous versions of each changed object because IBM® Workload Scheduler Workload Automation on Cloud maintains all previous versions for the whole set of objects.

Schedulers can also view the differences between versions using a comparison viewer, compare the differences, and rollback to a previous version.

Previous versions of changed objects are available in scheduling language.

Release management

Application developers and schedulers can promote the job workflow from development to test or production environments. An enhanced versioning capability keeps track of differences between different versions in the test and production environments to avoid conflicts.

They can also promote changes from production to pre-production or development environments to apply in the preproduction environments last-minute changes implemented into the active plan.

To move their workflows they can create and export a workload application template, a compressed file containing one or more job streams with all the related jobs and internal or external dependencies (such as files, resources, prompts). A workload application template is a self-contained workflow that you can import into a target environment.

Any mismatches in the versions of objects between development and production environments are seamlessly managed and resolved. For more information about creating and importing workload application templates, see Reusing a workload in another environment.

Auditing justification and reporting

Enforcing justification policies and maintaining auditing reports

About this task

From the Dynamic Workload Console, administrators can enforce a policy by which each user making a change to an object must provide a justification for the change. From the Dynamic Workload Console, administrators perform the following steps:

1. From the navigation toolbar, click **Administration**.
2. In the **Security** section, select **Auditing Preferences**.
3. For each displayed engine, you can enable the justification policy. After enabling the justification policy, decide which fields (Category, Ticket, and Description) are required.
4. Optionally create a new category to be added to the default ones in the lower section of the panel.
5. Optionally click on **Ticket required** to specify the address of the ticketing server and the specific syntax supported on the server, for example: `https://ticket.server.com/tickets/{$ticketnumber}`, where:

https://ticket.server.com/tickets/

is the ticketing server sample address

{ticketnumber}

is the ticket number to be provided by the user

After the justification policy is enabled, the justification panel is displayed each time a user performs a change in the Graphical Designer however, the justification will not be documented in the file. Justification can be viewed in the database. All other related changes are stored for auditing purposes in the IBM® Workload Scheduler database, in a file or in both, depending on the value you set in the **auditStore** global option.

For more information about global options, see the *IBM Workload Scheduler: Administration Guide*.

For more information about enabling the storage of auditing information from the command line, see the following commands:

- The composer command-line program in the *User's Guide and Reference*.
- The conman command-line program in the *User's Guide and Reference*.
- The optman command-line program in the *Administration Guide*.
- The wappman command-line program in the *User's Guide and Reference*.

Checking version information

Work with previous versions of your scheduling and security objects.

About this task

IBM Workload Scheduler maintains all previous versions, both for scheduling and security objects. From the Dynamic Workload Console you can view and edit object definitions and check the related version information.

Depending on your object type, select the **Versions** tab or the **Show versions** button. For each object, you can view the history of changes, discover which user performed a specific change, the time stamp, and the reason for the change. You can also display previous versions in scheduling language.

For the object in the database, you can compare different versions in a comparison viewer. Perform the following steps:

- Open the object in edit mode and, depending on your object type, select the **Versions** tab or the **Show versions** button.
- Select the two versions that you want to compare.
- Click on **Compare**.
- A comparison viewer opens that highlights the differences between the two versions.

For the object in the Workload Designer, you can also restore a previous version. Perform the following steps:

- From the Workload Designer, open the object in edit mode and select the **Versions** tab.
- Select the version that you want to revert to.
- Click on **Restore...** to start the restore process.
- The selected version is opened in edit mode. Review the object definition and save it.

Auditing justification and reporting- a business scenario

Keep your scheduling environment under control

Jason and Tim both work for S&L, a primary brokering company. Tim is a IBM® Workload Scheduler administrator and he must be in control of every aspect of the scheduling process. Just when he is getting ready to leave, his boss calls him with an urgent request: he needs a complete audit report of last year's activities for the Toronto branch. Tim is in a hurry and it is getting late.

However, with IBM® Workload Scheduler , he can easily trace all changes and quickly find out who did what and why.

In the Dynamic Workload Console, Tim requires that justification information be provided for each change to an object by setting the **Auditing Preferences**. The requirement applies to all objects on the specified engines. Each time a user performs a change on an object, they have to specify a reason for making the change. Based on this information, Tim can produce a report containing all information about who, when, why, and what has changed for each modification on each scheduling object.

Tim's colleague, Jason, is an IBM® Workload Scheduler operator and he needs to make sure that his workflows run smoothly. However, this is not always the case: an error has occurred and Jason must understand what happened. The PAYROLL job stream has failed unexpectedly. In the Workload Designer, Jason can see that some jobs were modified yesterday and why. He then applies the same correction to the one remaining job and restarts the job stream. With the correction applied, the job stream now runs correctly.

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

Streamline release management - a business scenario

Deploy new products, features, and fixes faster.

If you can reduce the amount of time it takes to push a code change into production, then you also reduce costs and maximize productivity, efficiency, and reliability. Code changes are usually triggered by a new feature or release, a minor enhancement stemming from a change request, or an emergency situation where a hot fix needs to be implemented.

Code changes also need to be continuously tested, integrated and built, and they often need to undergo an approval process before they can be promoted. To deliver business value, your release management process cannot suffer lag time from manual, error-prone processes that would slow down the development team's productivity.

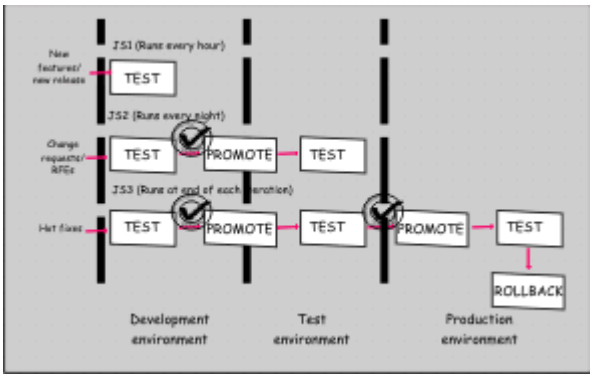
IBM Workload Scheduler can automate and control the entire process from the introduction of the code change to the promotion into production. With IBM Workload Scheduler workload application templates, you can easily replicate a job flow from one environment to another through a quick and easy import and export operation.

Business scenario

Rob is an application developer for an insurance company. He writes code for new releases, new features and enhancements, and hot fixes. His company applies agile methodology to their business development model. Agile is both iterative and incremental, with continuous improvement, that means that at the end of each abbreviated, repetitive cycle, a shippable deliverable is produced and can be deployed. This also means that in a short, abbreviated time slot, the development team is developing and testing and then promoting into production. The teams need an up-to-date environment with the latest code changes running at all times. Running a process like this manually is time consuming and error prone. Here's where Rob's colleague Marnie can help. Marnie is a scheduler. She suggests that rather than implement and run the release management system manually, that they could automate and schedule the development, test, and promotion into production of the process.

Rob maps the process to jobs, job streams, dependencies, prompts and any other scheduling objects needed to replicate the process. Here's what his workload includes:

- A job stream, JS1, is created to run a job every hour that runs an automation test suite that tests the code change.
- A job stream, JS2, that runs every night that contains jobs that, through a script, export the environment to a workload application template and then imports the template to deploy the environment into the test environment. Approval is obtained through a prompt. Another job is triggered to run a new automation test suite in the test environment.
- A third job stream, JS3, that runs at the end of each iteration and automatically tests and promotes after approval into the test environment and then again in the production environment. Finally, a job that performs a rollback if the test job fails.



Standardizing and automating your business processes ensures your deliverable is consistent every time guarantees a high level of quality, frees up human resources, saves money, and guarantees faster time to market.

Version control - a business scenario

Gain full control of changes in your scheduling environment.

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

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

Business scenario

Rob works as an application developer in a large healthcare company. Marnie works as a scheduler in the same company. Rob is working on the new release of an application for smart diagnostics. He prepares the full set of scheduling objects, jobs and job streams, associated to the application, and validates them in the test environment. The deadline has arrived, Rob is confident that the new release can be deployed in the production environment. He just makes a last minute change to the input file path for a java class in the DefineDiagnostics job. Time is running out and Rob leaves the office without verifying his last change.

The day after, Marnie, the scheduler, opens the Monitor Workload view and checks the job results for the daily plan. She is surprised to find out that the healthcare application job stream is blocked. The DefineDiagnostics job that has always completed successfully, today has ended abnormally. From the Workload Designer user interface, Marnie opens the failing job definition in edit mode. On the Versions page, she looks at the history of changes and realizes that Rob made a last minute change the day before.

Marnie can compare the failing job definition with the previous one, with no manual effort. By opening the comparison viewer and by looking at the differences between the two versions, she can determine the error introduced by Rob: a wrong file path. Marnie decides to restore the previous version of the job definition. From the Monitor Workload page, she reruns the failing job. This time, the healthcare application job stream completes successfully.

By using the version control feature, Marnie can quickly and easily solve a problem that otherwise would have compromised the release of the application into production. Version control enables audit readiness, reporting, collaboration, and change management within your company: all challenges that you must overcome in today´s business world.

Chapter 14. Reporting

You can use IBM Workload Scheduler reports to retrieve data from the IBM Workload Scheduler database. You can then view, print, and save the results in different kinds of output. You can also save, reuse, and share the reports with other users and modify them at any time.

You can choose to use **predefined reports** or import your **personalized reports** created using Business Intelligent Report Tool (BIRT). By using **predefined reports** it is possible to generate historical reports in a panel completely integrated in the Dynamic Workload Console. The **personalized reports** allow the administrator to generate reports from other databases by importing BIRT templates in the Dynamic Workload Console.

After you analyze the results of the reports, you can plan and assess changes and adjustments to your workload with the What-if Analysis. The What-if Analysis simulates and evaluates the impact of any changes to the current plan. For more information on What-if Analysis, see Analyzing the impact of changes on your environment on page 159.

Related information

Reports on page 93

Regular expressions and SQL reports on page 236

Predefined Reports

This topic lists the reports you can generate using the predefined reports section.

Before you can run a report to retrieve data from an IBM Workload Scheduler engine, you must complete the following steps:

1. Create a connection to an IBM Workload Scheduler engine.
2. Configure the Dynamic Workload Console to view reports, as described in the section about reports in the *User's Guide and Reference*.

You can generate the following reports from the Dynamic Workload Console using the predefined reports in Monitoring and Reporting from the navigation toolbar in the Dynamic Workload Console.

Creating a task to generate a Job Run Statistics report

How to generate and run a Job Run Statistics report task.

About this task

To create a task to run a Job Run Statistics Report, perform the following steps:

1. From the navigation toolbar, click **Monitor & Reporting > Reporting > Manage Predefined Reports** and click **Create**.
2. In the Task Information tab, select the **Report Type** > **Job Run Statistics**.
3. In the Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify

whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.

4. In the **Report Header** Tab, choose the title and the description of the output of the report. Check the boxes **Append report selection criteria to the report header** and **Include table of contents**, if needed.
5. In the Filter Criteria tab, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help.
6. In the Report Output Content tab, select the layout of your report. You can view the information either as a chart or as a table. The chart view displays the statistics for each job run in pie charts. You can select the report format. If you select HTML format, you can also limit the size of the report. You can also select the job details and the statistics you want to include in your report. After you made your selection, click **Save** to create the task.

What to do next

Next, run the report. For further details, see the section about running reports and batch reports in *User's Guide and Reference*.

Related information

Reports on page 93

Samples of report output

Creating a task to generate a Job Run History report

How to generate and run a Job Run History report task.

About this task

To create a task to run a Job Run History report, perform the following steps:

1. From the navigation toolbar, click **Monitor & Reporting > Reporting > Manage Predefined Reports** and click **Create**.
2. In the Task Information tab, select **Job Run History** as report Type.
3. In the Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the **Report Header** Tab, choose the title and the description of the output of the report. Check the boxes **Append report selection criteria to the report header** and **Include table of contents**, if needed.
5. In the Filter Criteria tab, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help.
6. In the Report Output Content tab, select the view of your report. You can view the information only as a table, but you can format it as an HTML or CSV file. If you select HTML format, you can also limit the size of the report. You can also select the job details you want to include in your report. After you make your selection, click **Save** to create the task.

What to do next

Next, run the report. For further details, see the section about running reports and batch reports in *User's Guide and Reference*.

Related information

Reports on page 93

Samples of report output

Creating a task to generate a Workstation Workload Summary report

How to create and run a Workstation Workload Summary report.

About this task

To create a task to run a Workstation Workload Summary report, perform the following steps:

1. From the navigation toolbar, click **Monitor & Reporting > Reporting > Manage Predefined Reports** and click **Create**.
2. In the Task information tab, select **Workstation Workload Summary Report** as **Report Type**.
3. In the Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the **Report Header** Tab, choose the title and the description of the output of the report. Check the boxes **Append report selection criteria to the report header** and **Include table of contents**, if needed.
5. In the Filter Criteria tab, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help.
6. In the Report Output Content tab, select the layout of your report. You can view the information either as a chart or as a table. The chart view shows the workload of all the specified workstations, aggregated by time. You can also choose to view the workloads of all the specified workstations in a single line chart. Aggregating all the information, you have a comparative view of the workstation workloads. You can select the report format. If you select HTML format, you can also limit the size of the report. You can also select the granularity with which to extract the data (by day or hour) and to order the report. After you make your selection, click **Save** to complete the task creation using all the default values.

What to do next

Next, run the report. For further details, see the section about running reports and batch reports in *User's Guide and Reference*.

Related information

Reports on page 93

Samples of report output

Creating a task to generate a Workstation Workload Runtimes report

How to create and run a Workstation Workload Runtimes report.

About this task

To create a task to run a Workstation Workload Runtimes report, perform the following steps:

1. From the navigation toolbar, click **Monitor & Reporting > Reporting > Manage Predefined Reports** and click **Create**.
2. In the Task information tab, select **Workstation Workload Runtimes Report** as Report Type.
3. In the Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the **Report Header** Tab, choose the title and the description of the output of the report. Check the boxes **Append report selection criteria to the report header** and **Include table of contents**, if needed.
5. In the Filter Criteria tab, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help.
6. In the Report Output Content tab, select the layout of your report. You can view the information either as a chart or as a table. The chart view shows in a bar chart the number of jobs that are running on the workstations. Selecting the chart view, you can also specify how many jobs to display in each chart. You can specify the format of your report. If you select HTML format, you can also limit the size of the report. You can also select the information you want to include in the report and how you want it ordered. After you make your selection, click **Save** to complete the task creation.

What to do next

Next, run the report. For further details, see the section about running reports and batch reports in *User's Guide and Reference*.

Related information

Reports on page 93

Samples of report output

Creating a task to Create Plan Reports

How to generate a Plan Report task.

About this task

Perform the following steps to create one of the following reports:

Actual Production Details Report

A report based on the information stored either in the current or in an archived plan. The information contained in these plans is retrieved from the Symphony files. Actual Production Details Report can be run on distributed engines (master domain manager, backup domain manager, domain manager with connector, and fault-tolerant agent with connector).

Planned Production Details Report

A report based on the information stored either in a trial or in a forecast plan. The information contained in these plans is retrieved from the IBM Workload SchedulerWorkload Automation on Cloud database. A Planned Production Details Report can be run on distributed engines (master domain manager and backup domain manager). A real production report extracted from a fault-tolerant agent might contain different information with respect to a plan extracted from a master domain manager. For example, the number of jobs and job streams is the same, but their status can change, because a job successful on the master can be in hold or ready on the agent. The update status rate is the same only on the full status agent that runs on the domain master.

1. From the navigation toolbar, click **Monitor & Reporting > Reporting > Manage Predefined Reports** and click **Create**.
2. In the Task information tab, select **Actual Production Details** or **Planned Production Details** as Report Type.
3. In the Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the **Report Header** tab, choose the title and the description of the output of the report. Check the boxes **Append report selection criteria to the report header** and **Include table of contents**, if needed.
5. In the Filter Criteria tab, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help.
   Wild cards are not allowed for **Actual Production Details Report** and **Planned Production Details Report** report types.
6. In the Report Output Content panel, select the job information that you want to display in the report output. After you make your selection, click **Save** to complete the task creation.

What to do next

Next, run the report. For further details, see the section about running reports and batch reports in *User's Guide and Reference*.

Related information

Reports on page 93

Samples of report output

Creating a task to Create Custom SQL Reports

How to create an SQL query report task to run on a distributed IBM® Workload Scheduler environment.

About this task

Use this task to define your own reports by writing or importing SQL queries for extracting data in HTML or CSV format. To create an SQL report task, perform the following steps:

1. From the navigation toolbar, click **Monitoring and Reporting** > **Reporting** > **Manage Predefined Reports**, and click **Create**.
2. In the Task Information panel, from the **Report type** drop-down menu, select **Custom SQL Report**.
3. In the Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report and click **Next** to proceed or **Finish** to complete the task creation using all the default values. The **Custom SQL Report** supports only the **Tables View** in either HTML or CSV format.
5. In the Filter Criteria panel, enter the SQL statement on which you want to base your report. You can write the query in the text pane or load an existing query from a file by browsing to the required file and clicking **Load**. Click **Next** to proceed or **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the job information that you want to display in the report output. After you make your selection click **Next** to proceed or **Finish** to complete the task creation using all the default values.

What to do next

Next, run the report. For further details, see the section about running reports and batch reports in *User's Guide and Reference*.

Related information

Reports on page 93

Samples of report output

SQL report examples on page 242

Personalized Reports

This topic lists the reports you can generate using the personalized reports created with Business Intelligent Report Tool (BIRT).

To create a personalized report, customized with Business Intelligent Report Tool (BIRT), the administrator must import and every user can run it.

Before you can run a report to retrieve data from an IBM Workload Scheduler engine, complete the following steps:

1. Create a report with Business Intelligent Report Tool (BIRT) and import it in the section **Monitoring & Reporting > Manage Personalized Reports**.
2. Run your personalized reports in the section **Monitoring & Reporting > Personalized Reports**.

Manage personalized reports with BIRT

Create and run your personalized report with Business Intelligent Report Tool (BIRT).

About this task

With this functionality the administrator can manage a report starting from a custom template that has been previously created with Business Intelligent Report Tool (BIRT). The administrator can either put the information in the datasource before importing it in the Dynamic Workload Console or perform the steps below and add the information during the process of import or run using the engine connection data.

To create a Personalized Report perform the following steps:

1. From the Navigation toolbar click **Monitoring & Reporting > Reporting > Manage Personalized Reports**.
2. Click **Import** and insert a **name** and a **description** for the report.
3. You can also specify whether to share the task with others or not.
4. Check the box **Run on dynamic workload engine** to use the database connection data that are on the Dynamic Workload engine. Selecting it, you can choose between the following two options:

   Run on engine

   Select one or more engines to include in the report.

   Let users choose the engine

   Postpone the choice to a later time when the report runs.
5. Click **Choose a .rtpdesign file** to import a template created with Business Intelligent Report Tool (BIRT).
6. If needed, click **Add property.file** to assign a resource file to the report.
7. click **Import** to upload the report.

Run your personalized reports created with BIRT

About this task

To run your personalized perform the following steps:

1. From the Navigation toolbar click **Monitoring & Reporting > Reporting > Personalized Reports**.
2. Click the specified icon to run the report as HTML, PDF, XSLX, DOC or PPTX format.

Chapter 15. Scenarios

Scenarios demonstrating practical application of IBM Workload Scheduler in a business environment.

View these scenarios to help you get familiar with IBM Workload Scheduler and learn how to use the product to achieve your business goals.

You can find additional scenarios in the Workload Automation channel in YouTube.

Using workload service assurance to monitor z/OS critical jobs

How to monitor jobs that are critical for the customer business and that must complete by their deadline.

This scenario shows how an operator can monitor the jobs that are critical for the customer business and that must complete by their deadline.

Overview

The operator uses the Dynamic Workload Console to meet a Service Level Agreement (SLA) that requires a DB2 database to be up and running each day by 3 p.m., after its backup.

The operator must be informed whether critical jobs risk missing their deadline, to take the appropriate actions if needed. While the plan is running, the operator expects that the scheduler dynamically controls the network of submitted jobs, detecting when a critical job predecessor is late, long running, or ended with an error.

Roles

The scheduling administrator and the operator are involved in this scenario:

IBM Workload Scheduler for z/OS scheduling administrator

When planning the operations, the administrator defines:

- Scheduled time, duration, and deadline times.
- Critical jobs.

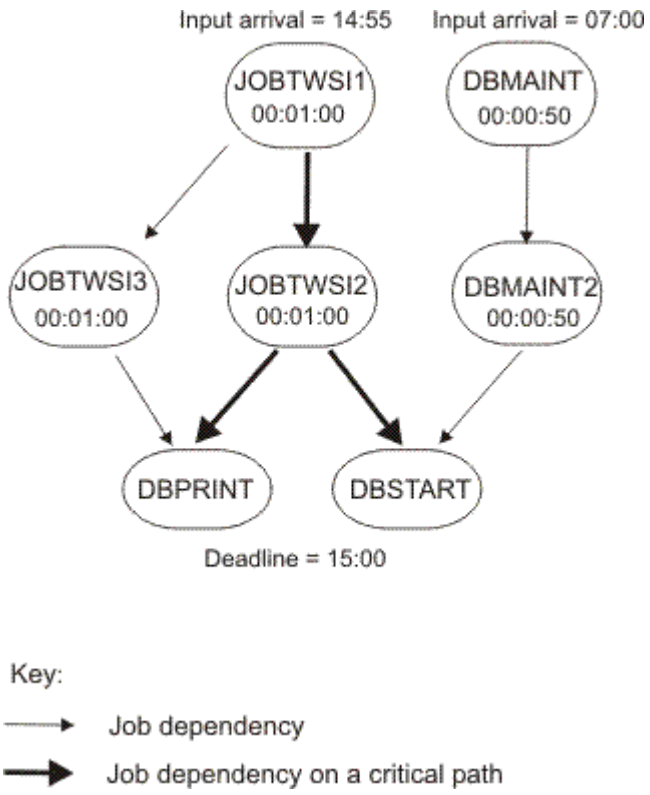
IBM Workload Scheduler operator

Controls the submitted workload by using Critical Jobs and Hot List views.

Setting up the environment

Complete the following tasks when planning your operations:

1. Mark your critical jobs in the z/OS database. Set DBSTART and DBPRINT as critical jobs, using a job network with the following structure:



2. Run a daily planning job. The daily planning process calculates the critical paths in your job network, using the deadline, scheduled time arrival, and duration settings.

Running the scenario

After you updated your current plan, you can monitor your critical workload by using **Critical Path** and **Hot List** views:

1. In the navigation bar, click **Monitoring & Reporting > Workload Monitoring > Monitor Workload**.
2. Under the **Engine** drop-down list, select the check-box related to the engine or engines where the task must run.
3. Under the **Object Type** drop-down list select **Critical job**.
4. Click **Edit**.
5. In the General Filter panel, specify DB* as **Job Name** and set a **Risk Level** different from None as filter criteria, because you want to monitor the critical jobs that risk missing their deadlines.
6. Click **Save** to complete the task, leaving the default values in the remaining panels.
7. **Run** the task.
8. Select the DBSTART job and click **Critical Path** to view the path of the DBSTART predecessors with the least slack time. The **Critical Path** view does not show any cause for the delay, because no problems occurred for any of the DBSTART predecessors in the critical path. Return to the Monitor Workload task output.

9. Click **Hot List** or the **Potential Risk** hyperlink to get a list of any critical job predecessor that is late, has been running for too long, or has ended with an error. The returned **Hot List** shows DBMAINT as a late job. It is scheduled to run on the CPU2 workstation.
    a. Click the CPU2 hyperlink.
    b. After verifying that CPU2 is offline, activate the workstation. The DBMAINT job starts to run.
10. Refresh the **Monitor Workload** task output. It shows that the **Risk Level** for DBSTART job is now **No Risk**.

Related information

Workload service assurance on page 96

Limit the number of objects retrieved by queries on page 36

Monitoring jobs running on multiple engines

Using the Dynamic Workload Console to create a task to simultaneously monitor jobs that run on multiple engines.

This scenario describes how you use the Dynamic Workload Console to create a task to simultaneously monitor jobs that run on multiple engines, which can be in mixed distributed and z/OS environments.

Overview

High efficiency batch processing relies on powerful monitoring capabilities. The need for a single operator to monitor systems is continuously increasing. Up until about 10 years ago, only a limited amount of workload was monitored, but this is increasing to the monitoring of an entire division, and even to an entire company.

Today operators frequently have to monitor multiple large divisions or multiple companies for service providers. These operators work in shifts in multiple geographical locations according to a "follow-the-sun" approach, in some cases. They must try to balance what must be monitored with the size of the monitored environment.

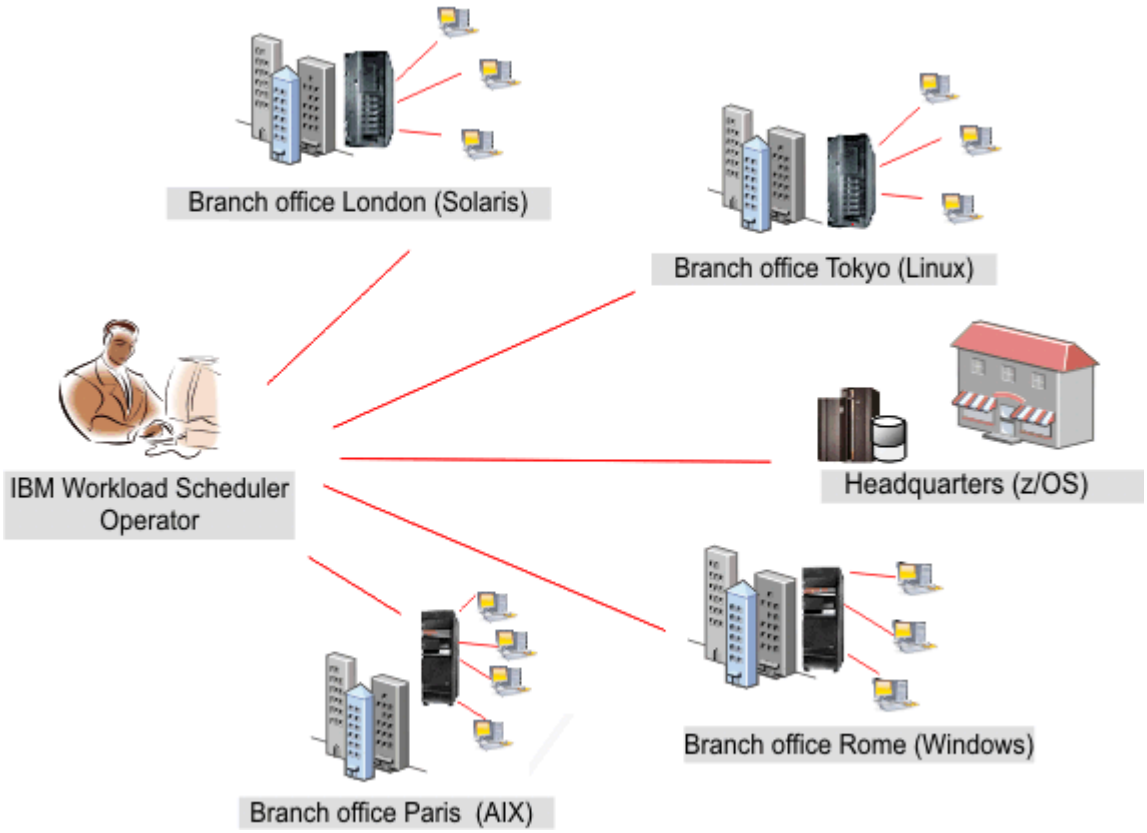
Business scenario

In this scenario, an insurance company, named *Starbank*, consists of a headquarters where its central office accounting department is located, and multiple branch offices located all over the world, where several administrative departments perform accounting activities.

The central office is in charge of the company's entire accounting workload. Therefore, the IBM Workload Scheduler operator must verify that all the workload processing for the *Starbank* company proceeds smoothly and without errors and needs a comprehensive workload management solution.

To achieve this goal, the operator needs to create a task that he can run every day to monitor all the administrative jobs, detecting in real time any possible failures.

However, although the sales department of the company runs its jobs in a z/OS environment, the single business units run their jobs in distributed environments. The operator needs a single console panel, from which he can control all the jobs, both z/OS and distributed, at the same time.



The operator creates a task to monitor jobs that run on multiple engines, including both the environments. He does this by creating and running a task to Monitor Jobs on multiple engines.

Creating a Monitor Jobs task for multiple engines

The operator logs in to the Dynamic Workload Console and, from the navigation bar, he clicks **Monitoring & Reporting > Workload Monitoring > Monitor Workload**.

To create a task using Monitor Workload, see Monitoring your items in the plan on page 178.

Selecting the engines

In the Enter Task Information panel, the operator specifies a name for the task, for example *AccError*, and defines the scheduler engines on which to run the task.

According to a company naming convention policy, all the engine names have a prefix specifying to which department they belong. Therefore, the operator includes in the **Selected Engines** list all the engines named *acc_**. The operator then organizes the list by importance, placing the engines belonging to the most critical company departments (like *Finance* and *Sales*) at the beginning of the list, so as to have their results displayed as the first rows of the table. The task runs following the engine sequence, but the results are displayed altogether, only after the task has run on all the engines in the list.

Defining the filter

In the General Filter panel, the IBM Workload Scheduler operator specifies some filtering criteria to limit the results retrieved by the query. Here he starts refining the scope of the query by also considering the amount of information to retrieve. Defining a meaningful filter is very important to avoid unnecessary overhead, considering that the task runs on multiple engines. First, the operator sets the automatic refresh time to *600* so as to receive the updated monitoring results every 600 seconds (10 minutes). He then filters for jobs based on their job streams. According to a company policy, all administrative job streams begin with the company name followed by the department code. In our scenario, the operator looks for all the job streams whose identifier starts with *Starb** that did not complete successfully.

Selecting the columns

In the Columns Definition panel, the operator selects the information to display in the table containing the query results. According to the columns he chooses, the corresponding information is displayed in the task results table. In our scenario, for all the jobs resulting from the query, the operator wants to see their statuses, the job streams they belong to, when they were scheduled to run, and the engines on which they ran. Then, if more details are necessary, he can drill down into this information displayed in the table of results and navigate it.

Results

In the All Configured Tasks panel, the operator can see the main details about the task that he has just created and launch the task immediately. The task is now in the list of saved tasks from where the operator can open and modify it any time. To find the task from the displayed task lists, he clicks the following options: **Monitoring and Reporting** > **Workload Monitoring > Monitor Workload**.

The operator has created a task that can be run every day to highlight possible critical failure in real time. If there is a failure in any of the administrative jobs run by the selected offices, the operator discovers it no later than 10 minutes after the error occurs.

Running the Monitor Jobs task for multiple engines

To launch the task, the operator clicks **Monitoring & Reporting > Workload Monitoring > Monitor Workload**.

The operator clicks *AccError* task to launch it. Because some engine connections do not work correctly, the Checking engine connections panel reports some errors on two of the eight engines defined. The failing connections are the *Tokyo* and *Paris* offices. The operator could ignore the failed connections and proceed, running the task on the successful engines only. However, monitoring the entire workload running in all the branch offices is crucial to his activity, and he does not want to skip any engine connection. Therefore, by clicking **Fix it** next to each failing engine connection, the operator opens a dialog where he can enter the credentials required for that engine. After entering the correct credentials, also the remaining engine connections work successfully and the operator clicks **Proceed** to run the task against all the engines.

Viewing results and taking corrective actions

Viewing the results of the *AccError* task, the operator realizes that there is a job in error, named *PayAcc1*. He right-clicks the job to open its job log, to better determine the cause and effects of this error.

From the job log, he finds out that only the last step of the job failed, which was a data backup process. This step can be done manually at a later time. The most important part of the job, consisting of the accounting processes related to payrolls, completed successfully.

Now the operator needs to determine the impact that this job in error has on the overall plan. To do this, he selects the *PayAcc1* job and clicks **Job Stream View**. From this view, he realizes that this job is a predecessor dependency of another job, named *Balance1*. The operator releases the failing job dependency so as to make it possible for the successor *Balance1* to start and the whole workload processing to complete.

A second job in error results from the *AccError* task. It is a z/OS job, named *Info*. The operator selects this job from the list and right-clicks it to open the **Operator Instructions** that give him important information about what to do. According to the instructions, this is an optional procedure, which can be skipped without consequence for the entire processing. Therefore, the operator right-clicks the job and cancels it.

The operator then refreshes the view to ensure that there are no other jobs in error.

To view connection status information and statistical information about the engines against which the task was run, the operator clicks the statistical icon on the table toolbar.

A pie chart showing the number of query results and job status is displayed for each engine on which the task ran successfully. By clicking the pie sections, he can see further details. If the task did not run successfully run on one or more engines, he sees a message containing details about the errors.

Related information

Limit the number of objects retrieved by queries on page 36

Chapter 16. Troubleshooting the Dynamic Workload Console

Accessing troubleshooting information.

About this task

You can find information about how to troubleshoot problems with the Dynamic Workload Console, related to connections, performance, user access, reports, and others at the following link: IBM Workload Scheduler Troubleshooting, under the section about troubleshooting Dynamic Workload Console problems.

Note: If you print PDF publications on other than letter-sized paper, set the option in the File -> Print window that enables Adobe Reader to print letter-sized pages on your local paper.

Chapter 17. Reference

Reference information to perform the main tasks and activities from the Dynamic Workload Console.

This section provides some reference information that can be useful to perform the main tasks and activities from the Dynamic Workload Console.

Users and groups

How a system of users and groups allows different types of access to the Dynamic Workload Console.

In the Dynamic Workload Console you can manage the users to authorize them to the different views to which every role is associated. From the **Manage Roles** panel you can also create customized roles and share the boards. Users are granted access to resources based on the role to which they have been assigned. The roles in which users are defined or the role assigned to users determines the operations they can perform and which resources are visible to them. This means that depending on your user role designation, you might not see all of the items described in this help system. The groups or predefined roles available are:

API User

Users in this group can use only the Dynamic Workload Console APIs to perform the available actions. Logging to the Dynamic Workload Console through a Web browser would not give them access to any feature. For more details about the Dynamic Workload Console APIs, see `https://<DWC_hostname>:<port>/dwc/api`.

Administrator

Users with this role can see the entire portfolio and use all features of the Dynamic Workload Console.

Users with this group can also access and use all features of the Self-Service Catalog and theSelf-Service Dashboards mobile applications. From the Self-Service Catalog mobile application, these users can create and edit catalogs, create and edit services, add services to catalogs, submit services associated to job streams, and share catalogs and services with other users. From the Self-Service Dashboards mobile application, these users can create and edit dashboards to filter for jobs and workstations, display a dashboard of results, perform recovery actions on a single result.

From the Manage Roles panel, the administrator can add entities, manage pinned pages and shared boards.

Analyst

Users in this group can manage Dynamic Workload Console reports and user preferences.

Broker

Users in this group can define Broker settings, create and manage Broker jobs and resources, and monitor Broker computers and resources.

Developer

Users in this group can create, list, and edit workload definitions, workstations, and event rule definitions in the IBM Workload Scheduler database.

Mobile User

Users in this group can manage the Self-Service Catalog and theSelf-Service Dashboards mobile applications but the actions they can perform are limited to submitting service requests (job streams) from the Self-Service Catalog and , from the Self-Service Dashboards mobile application, displaying a dashboard of results and performing recovery actions on them.

Operator

Users in this group can see Dynamic Workload Console:

- All Monitor tasks.
- Jobs and job streams to be submitted on request
- Set User Preferences

Example

The following table lists some entries of the navigation toolbar, and some activities you can perform from the Dynamic Workload Console. Beside each item, the table shows the groups whose users are authorized to access them.

Table 22. Menu and Group Permissions

Menu Item	Groups with Permission
Quick Start	Administrator
All Configured Tasks	Administrator, Operator
Manage Workload Reports	Administrator, Analyst
Design -> Workload Definitions	Administrator Developer
Planning & Submission -> Workload Forecast	Administrator, Operator
Administration -> Workload Submission	Administrator Operator
Monitoring & Reporting	Administrator, Operator
Design -> Workload Definitions	Administrator

Table 22. Menu and Group Permissions (continued)

Menu Item	Groups with Permission
Reporting	Administrator, Analyst
Security ->Manage Engines	Administrator
Security -> Manage Settings	Administrator
Administration → Broker Settings; Design → Broker Design; Monitoring & Reporting → Broker Monitoring	Administrator, Broker

Type of communication based on SSL communication options

How the SSL communication options for distributed workstations affect communication between workstations.

Based on the authentication types that you defined for the workstations in your network, communication between workstations is different. The following table summarizes the types of connection for the different authentication type settings.

Table 23. Type of communication based on workstation SSL communication options

Fault-tolerant Agent (Domain Manager)	Domain Manager (Parent Domain Manager)	Connection Type
Disabled	Disabled	TCP/IP
Allow Incoming	Disabled	TCP/IP
Upward Forced	Disabled	No connection
All Forced	Disabled	No connection
Disabled	Upward Forced	TCP/IP
Allow Incoming	Upward Forced	TCP/IP
Upward Forced	Upward Forced	SSL
All Forced	Upward Forced	SSL
Disabled	Allow Incoming	TCP/IP
Allow Incoming	Allow Incoming	TCP/IP
Upward Forced	Allow Incoming	SSL
All Forced	Allow Incoming	SSL
Disabled	All Forced	No connection

Table 23. Type of communication based on workstation SSL communication options (continued)

Fault-tolerant Agent (Domain Manager)	Domain Manager (Parent Domain Manager)	Connection Type
Allow Incoming	All Forced	SSL
Upward Forced	All Forced	SSL
All Forced	All Forced	SSL

For details about how to create SSL certificates and how to set local options for SSL communication, see *IBM Workload Scheduler: Administration Guide*.

Status description and mapping for distributed jobs

The job status as described in Dynamic Workload Console and how it is defined in IBM Workload Scheduler.

There are the following types of status for distributed jobs:

Job status on page 222

A subset of internal status that is common for both IBM® Workload Scheduler for distributed and IBM® Z Workload Scheduler environments.

Job internal status on page 223

The IBM Workload Scheduler job status registered on the workstation where the job is running. The internal status uniquely identifies a job status in IBM Workload Scheduler.

Job status

Table 24: Job status on page 222 lists the job statuses.

Table 24. Job status

This job status ...	Means that ...
Waiting	The job is waiting for its dependencies to be resolved.
Ready	The dependencies of the job have been resolved and the job is ready to run.
Running	The job is running.
Successful	The job completed successfully.
Error	The job stopped running with an error.
Canceled	The job was canceled.
Held	The job was put on hold.
Undecided	The job status is currently being checked.

Table 24. Job status (continued)

This job status ...	Means that ...
Blocked	The job was blocked because of unfulfilled dependencies.
Suppressed by condition	The job is suppressed because the condition dependencies associated to its predecessors are not satisfied.

Job internal status

Table 25: Job internal status on page 223 lists the job internal statuses.

Note: The + flag written beside the INTRO and EXEC statuses means that the job is managed by the local **batchman** process.

Table 25. Job internal status

This job status ...	Means that ...
ABEND	The job terminated with a nonzero exit code or with an exit code outside the defined RC mapping.
ABEND P	An ABEND confirmation was received, but the job is not completed.
ADD	The job is being submitted.
+AUTOGEN+	A monitoring job automatically created to monitor conditions in condition-based workload automation. For more information, see the section about condition-based workload automation in *User's Guide and Reference*.
BOUND	For shadow jobs, it means that the shadow job matched a remote job instance in the remote plan. For IBM Z Workload Scheduler Agent, it means that the job is on the JES queue.
CANCEL	The job was canceled.
CANCEL P	The job is pending cancellation. Cancellation is deferred until all of the dependencies, including **at** time dependencies, are resolved.
DONE	The job completed in an unknown status.
ERROR	For internetwork and cross dependencies only, an error occurred while checking for the remote status.
EXEC	The job is running.
EXTRN	For internetwork dependencies only, the status is unknown. An error occurred, a rerun action was just performed on the job in the **external** job stream, or the remote job or job stream does not exist.
FAILED	Unable to launch the job.
FENCE	The job's priority is below the fence.

Table 25. Job internal status (continued)

This job status ...	Means that ...
HOLD	The job is awaiting dependency resolution.
INTRO	The job is introduced for launching by the system.
PEND	The job completed, and is awaiting confirmation.
READY	The job is ready to launch, and all dependencies are resolved. If a job fails because the agent is not available, the job is automatically restarted and set to the READY status, waiting for the agent to connect again. As soon as the agent connects again, the job is submitted.
R JOB	The job is running.
SCHED	The job's **at** time has not arrived.
SUCC	The job completed with an exit code of zero.
SUCC P	A **SUCC** confirmation was received, but the job is not completed.
SUSP	The job was blocked because of unfulfilled dependencies. For IBM i jobs only, this status indicates that the IBM i job is waiting for a reply to a message. For more information, see the section about scheduling jobs on IBM i systems in *User's Guide and Reference*.
USER STAT	The job was put on hold by the user.
WAIT	The job is waiting to fulfill its dependencies.
WAITD	The job is waiting to fulfill its dependencies.
SUPPRESS	The job is suppressed because the condition dependencies associated to its predecessors are not satisfied.

Job status mapping

Table 26: Job status mapping on page 224 describes how a job status is mapped to the corresponding job internal status.

Table 26. Job status mapping

This job status ...	Maps to this job internal status
Waiting	ADD, PEND, WAIT, WAITD, INTRO, HOLD
Ready	READY
Running	EXEC, SUCC P, ABEND P, R JOB, BOUND
Successful	SUCC
Error	ABEND, FAILED

Table 26. Job status mapping (continued)

This job status ...	Maps to this job internal status
Canceled	Status of the job when it was canceled. Canceled flag is set.
Held	Priority = 0, WAIT, READY, USER STAT
Undecided	ERROR, EXTRN
Blocked	SUSP
Suppressed by condition	SUPPR

Related information

Job on page 58

Managing job definitions on page 114

Adding a job to a job stream

Status description and mapping for z/OS jobs

z/OS job status displayed by Dynamic Workload Console and how it is defined in IBM® Workload Scheduler for z/OS.

There are the following types of status for z/OS jobs:

z/OS job status on page 225

A subset of internal statuses common for both IBM® Workload Scheduler distributed and z/OS environments.

z/OS job internal status on page 226

The job status registered on the IBM® Workload Scheduler controller. The internal status uniquely identifies the status of a z/OS job.

z/OS job status

Table 27: z/OS job status on page 225 shows the z/OS job statuses that are displayed by the Dynamic Workload Console.

Table 27. z/OS job status

This job status ...	Means that ...
Waiting	The job is waiting for its dependencies to be resolved.
Ready	The dependencies of the job have been resolved and the job is ready to run.
Running	The job is running.

Table 27. z/OS job status (continued)

This job status ...	Means that ...
Successful	The job completed successfully.
Error	The job has stopped running with an error.
Canceled	The job was canceled.
Held	The job was put in hold.
Undefined	The job status is currently being checked.
Suppressed by Condition	The job is suppressed because the condition dependencies associated to its predecessors are false.

z/OS job internal status

Table 28: z/OS job internal status on page 226 shows the z/OS job internal statuses that are displayed by the Dynamic Workload Console and how they map to the status displayed on the IBM® Workload Scheduler for z/OS controller.

Table 28. z/OS job internal status

This job internal status ...	Means that ...	Maps to ...
Arriving	The job is ready for processing; no predecessors were defined.	A
Complete	The job has completed	C
Deleted	The job has been deleted from the plan	D
Error	The job has ended-in-error.	E
Interrupted	The job is interrupted.	I
Ready	The job is ready for processing; all predecessors are complete.	R
Started	The job has started	S
Undefined	The job status is being evaluated.	U
Waiting	The job is waiting for a predecessor to complete.	W
Ready - non-reporting workstation	Ready - At least one predecessor is defined on a *nonreporting* workstation; all predecessors are complete.	*
Suppressed by Condition	The condition dependencies associated to its predecessors are not satisfied.	X

z/OS job status mapping

Table 29: z/OS job status mapping on page 227 describes how a z/OS job status is mapped to the corresponding job internal status.

Table 29. z/OS job status mapping

This job status ...	Maps to this job internal status
Waiting	W
Ready	A, R, *
Running	S
Successful	C
Error	E
Canceled	I, D
Held	A,R,* manually held
Undefined	U
Suppressed by Condition	X

Related information

Job on page 58

Managing job definitions on page 114

Adding a job to a job stream

Status description and mapping for distributed job streams

The job stream status in Dynamic Workload Console and how it is defined in IBM Workload Scheduler.

There are the following types of status for job streams:

Job stream status on page 227

A subset of internal statuses that is common for both IBM Workload Scheduler for distributed and IBM® Z Workload Scheduler environments.

Job stream internal status on page 228

The IBM Workload Scheduler job stream status registered on the workstation where the job stream is running. The internal status uniquely identifies a job stream status in IBM Workload Scheduler.

Job stream status

Table 30: Job stream status on page 228 lists the job stream statuses.

Table 30. Job stream status

This job stream status ...	Means that ...
Waiting	The job stream is waiting for its dependencies to be resolved.
Ready	The dependencies of the job stream have been resolved and the job stream is ready to run.
Running	The job stream is running.
Successful	The job stream completed successfully.
Error	The job stream has stopped running with an error.
Canceled	The job stream was canceled.
Held	The job stream was interrupted.
Undefined	The job stream status is currently being checked.
Blocked	The job stream was blocked because of unfulfilled dependencies.
Suppressed by condition	The job stream is suppressed because the condition dependencies associated to its predecessors are not satisfied.

Job stream internal status

Table 31: Job stream internal status on page 228 lists the job stream internal statuses.

Table 31. Job stream internal status

This job stream status ...	Means that ...
ABEND	The job stream terminated with a nonzero exit code.
ADD	The job stream was added with operator intervention.
CANCEL	The job stream was canceled.
CANCEL P	The job stream is pending cancellation. Cancellation is deferred until all of the dependencies, including an at time, are resolved.
EXEC	The job stream is running.
EXTRN	The job stream is in a remote IBM Workload Scheduler network and its status is unknown. An error occurred, a Rerun action was performed on the **EXTERNAL** job stream, or the **INET** job or job stream does not exist.
HOLD	The job stream is awaiting dependency resolution.
READY	The dependencies for the job stream have been met but the time restrictions for the job stream have not.
STUCK	The job stream was interrupted. No jobs are launched without operator intervention.
SUCC	The job stream completed with an exit code of zero.

Table 31. Job stream internal status (continued)

This job stream status ...	Means that ...
SUPPRESS	The job stream is suppressed because the condition dependencies associated to its predecessors are not satisfied.
Get Job Status Error	This is for internetwork job streams and specifies that an error occurred while checking for the remote status.

Job stream status mapping

Table 32: Job stream status mapping on page 229 describes how a job stream status is mapped to the corresponding job stream internal status.

Table 32. Job stream status mapping

This job stream status ...	Maps to this job stream internal status
Waiting	ADD, PEND, WAIT, WAITD, INTRO, HOLD
Ready	READY
Running	EXEC
Successful	SUCC
Error	ABEND, FAILED
Canceled	CANCEL, HOLD, CANCEL P
Held	HOLD
Undefined	ERROR, EXTRN
Blocked	STUCK
Suppressed by condition	SUPPR

Related information

Job stream on page 59

Managing job stream definitions on page 117

Status description and mapping for z/OS job streams

The z/OS job stream status in Dynamic Workload Console and how it is defined in IBM® Z Workload Scheduler.

There are the following types of status for z/OS job streams:

z/OS job stream statuses on page 230

A subset of internal statuses that is common for both IBM® Workload Scheduler for distributed and IBM® Z Workload Scheduler environments.

z/OS job stream internal statuses on page 230

The IBM® Workload Scheduler job stream statuses registered on the controller. The internal status uniquely identifies the status of a z/OS job stream in IBM® Workload Scheduler.

z/OS job stream statuses

Table 33: z/OS job stream status on page 230 shows the z/OS job stream statuses that are displayed by the Dynamic Workload Console.

Table 33. z/OS job stream status

This job stream status ...	Means that ...
Waiting	No job in the job stream has started.
Running	The job stream is running.
Successful	The job stream completed successfully.
Error	The job stream has stopped running with an error.
Canceled	The job stream was canceled.

z/OS job stream internal statuses

Table 34: z/OS job stream internal status on page 230 shows the z/OS job stream internal statuses that are registered on the Dynamic Workload Console controller.

Table 34. z/OS job stream internal status

This job stream status ...	Means that ...
Waiting	No job in the job stream has started.
Started	The job stream is running.
Completed	The job stream completed successfully.
Error	The job stream has stopped running with an error.
Deleted	The job stream was deleted.
Undefined	The job stream status is not known.

z/OS job stream status mapping

Table 35: z/OS job stream status mapping on page 231 describes how a z/OS job stream status is mapped to the corresponding job stream internal status.

Table 35. z/OS job stream status mapping

This job stream status ...	Maps to this job stream internal status
Waiting	Waiting
Running	Started
Successful	Completed
Error	Error
Canceled	Deleted

Related information

Job stream on page 59

Managing job stream definitions on page 117

Workstation types

The following are the main workstation types and their attributes.

Table 36. Attribute settings for management workstation types

Attributes	Master domain manager	Domain manager	Backup domain manager
cpuname	The name of the workstation.		
description	A description for the workstation enclosed within double quotes. This attribute is optional.		
vartable	The name of a variable table associated with the workstation. Variables used with the workstation are defined in this table. This attribute is optional.		

os	The operating system installed on the system. Specify one of the following values:
	`UNIX`
	`WNT`
	`OTHER`
	`IBM_i`		

node	The system host name or IP address.		
tcpaddr	The value assigned to *nm port* in the `localopts` file. For multiple workstations on a system, enter an unused port number. The default value is 31111.		

Table 36. Attribute settings for management workstation types

(continued)

Attributes	Master domain manager	Domain manager	Backup domain manager
secureaddr	The value assigned to *nm ssl port* in the `localopts` file. Specify it if **securitylevel** is set to `on`, `force` or `enabled`.		
timezone | tz	The time zone in which the system is located. It is recommended that the value matches the value set on the operating system.		
domain	`MASTERDM`	The name of the managed domain.	
host	Not applicable		
access	Not applicable		
type	`manager`		`fta`
ignore	Use this attribute if you do not want this workstation to appear in the next production plan.		

autolink	It indicates if a link between workstations is automatically opened at startup. Specify one of the following values:
	`ON`
	`OFF`
	This is an optional attribute. The default value is `ON`.		

behindfirewall	This setting is ignored.	It indicates if there is a firewall between the workstation and the master domain manager. Specify one of the following values:
		`ON`
		`OFF`
		The default value is `OFF`.	

securitylevel	The type of SSL authentication to use:
	`enabled`
	`on`
	`force`		

fullstatus	`ON`		
server	Not applicable		This setting is ignored.
protocol	Not applicable		
members	Not applicable		
requirements	Not applicable		

Table 37: Attribute settings for target workstation types on page 233 describes the values you set for each attribute for target workstation types. Following the table you find additional details about each attribute.

Table 37. Attribute settings for target workstation types

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool
cpuname	The name of the workstation.						
description	A description for the workstation enclosed within double quotes. This attribute is optional.						
vartable	The name of a variable table associated with the workstation. Variables used with the workstation are defined in this table. This attribute is optional.						

os	The operating system installed on the system. Specify one of the following values:
	`UNIX`
	`WNT`
	`OTHER`
	`IBM_i`
	Specify `OTHER` for IBM® i systems running as limited fault-tolerant agents.
		`OTHER`	The operating system installed on the machine. Specify one of the following values:
			`UNIX`
			`WNT`
			`OTHER`
			`IBM_i`
				This value setting is discovered on the system.	The operating system installed on the machine. Specify one of the following values:
					`UNIX`
					`WNT`
					`ZOS`
						The operating system installed on the machine. Specify one of the following values:
						`UNIX`
						`WNT`
						`OTHER`
						`IBM_i`	

node	The system host name or IP address.		The system host name or IP address. Specify `NULL` when **host** is set to `$MASTER`, or when defining an extended agent for PeopleSoft, MVS or Oracle.	Agent host name or IP address.	Remote engine host name or IP address.	Not applicable	
tcpaddr	The value assigned to *nm port* in the `localopts` file. When defining multiple workstations on a system, enter an	The value assigned to *nm port* in the `localopts` file. When defining multiple	See the selected access method specifications.	The port number to communicate with the agent when the protocol is `http`.	The port number to communicate with the remote engine when the protocol is `http`.	Not applicable	

Table 37. Attribute settings for target workstation types

(continued)

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool
	unused port number. The default value is 31111.	workstations on a system, enter an unused port number. The default value is 41114.					
secureaddr	The value assigned to *nm ssl port* in the `localopts` file. Specify it if **securitylevel** is set to `on`, `force` or `enabled`.	Not Applicable	Not Applicable	The port number to communicate with the agent when the protocol is `https`.	The port number to communicate with the remote engine when the protocol is `https`.	Not applicable	
timezone | tz	The time zone in which the system is located. It is recommended that the value matches the value set on the operating system.		The time zone set on the workstation specified in the **host** attribute.	The time zone set on the agent.	The time zone set on the remote engine.	The time zone set on the pool agents.	The time zone set on the dynamic pool agents.
domain	Specify an existing domain. The default value for fault-tolerant agents is `MASTERDM`. This setting is mandatory for standard agents.	Specify an existing domain. This setting is mandatory.	This setting is needed only if the value assigned to **host** is `$MANAGER`.	Not applicable			
host	Not Applicable		The host workstation. It can be set to `$MASTER` or `$MANAGER`.	The broker workstation.			
access	Not Applicable			Select the appropriate access method file name.	Not Applicable		
agentID				The unique identifier of the dynamic agent			

Table 37. Attribute settings for target workstation types

(continued)

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool

type	`fta`
	`s-agent`
	The default value is `fta`.
	Specify `fta` for IBM® i systems running as limited fault-tolerant agents.
		`broker`	`x-agent`	`agent`	`rem-eng`	`pool`	`d-pool`

ignore	Use this attribute if you do not want this workstation to appear in the next production plan.						

autolink	It indicates if a link between workstations is automatically opened at startup. Specify one of the following values:
	`ON`
	`OFF`
	This is an optional attribute. The default value is `ON`.
			`OFF`	Not applicable			

behindfirewall	It indicates if there is a firewall between the workstation and the master domain manager. Specify one of the following values:
	`ON`
	`OFF`
	The default value is `OFF`.
			`OFF`	Not applicable			

securitylevel	The type of SSL authentication to use:
	`enabled`
	`on`
	`force`
	Not applicable for IBM® i systems running as limited fault-tolerant agents.
		Not Applicable					

fullstatus	It indicates if the workstation is updated	`OFF`		Not applicable			

Table 37. Attribute settings for target workstation types

(continued)

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool

	about job processing status in its domain and subdomains. Specify one of the following values:
	`ON`
	`OFF`
	Specify `OFF` for standard agents.						

server	0-9, A-Z. When specified, it requires the creation of a dedicated mailman processes on the parent workstation.		Not Applicable				

protocol	Not applicable			Specify one of the following values:
				`http`
				`https`
				This attribute is optional. When not specified, it is automatically determined from the settings specified for **tcpaddr** and **secureaddr**.
						Not applicable	

members	Not applicable					Required value	Not applicable
requirements	Not applicable						Required value

Related information

Workstation on page 50

Creating a task to Monitor Workstations on page 168

Regular expressions and SQL reports

This section contains examples of regular expressions and SQL reports. Some samples of report output are also shown.

Related information

Reports on page 93

Reporting on page 205

Regular Expressions

Examples of useful regular expressions supported by IBM Workload Scheduler.

This section gives some examples of useful regular expressions, together with a table that defines the expressions supported by IBM Workload Scheduler. Further information about regular expressions is also widely available in the Internet.

Useful regular expressions

The following table shows some useful regular expressions for use with the plan extractor, both for filtering jobs and job streams, and for configuring business unit names.

Table 38. Useful regular expressions

Requirement	Regular expression	Example
To obtain the same effect as using the "@"character in the Tivoli Workload Scheduler command line	.*	`-JScpu .*` Used as a parameter to the plan extractor, filters for all job stream workstations.
To join different criteria in an "OR" relationship	|	`(XYZ.*)|(.*ABC.*)` Filters for all items that begin with the string "XYZ or that contain the string "ABC" (regular expressions are case-sensitive).
To select objects that begin with one of several characters	`[<the characters to be included>]`	`[ABC].*` Filters for all items that begin with either "A", "B", or "C".
To select objects that do not begin with one of several characters	`[^<the characters to be excluded>]`	`[^ABC].*` Filters for all items that do not begin with either "A", "B", or "C".
To select objects in which certain characters appear a certain number of times	`<the character to be counted>{<the character count>}`	`A{3}.*` Filters for all items that begin with the string "AAA".

Table 38. Useful regular expressions (continued)

Requirement	Regular expression	Example
To select objects in which certain characters appear at least a certain number of times	`<the character to be counted>{<the character count>}`	`A{3,}.*` Filters for all items that begin with the string "AAA", "AAAA", "AAAAA", and so on.
To select objects in which certain characters appear at least a certain number of times, but not more than a certain number of times	`<the character to be counted>{<the lower character count>,<the upper character count>,}`	`A{3,4}.*` Filters for all items that begin with the string "AAA", or "AAAA"; a string that began with "AAAAA" would not be selected.

Complex expressions

These individual regular expressions can be combined to make a complex expression, as shown in the following table.

Table 39. Complex expressions

Example requirement	Regular expression
Select all strings that begin with "AA", "AB", "AC", "BA", "BB", "BC", "CA", "CB", or "CC", and also those that do not end in "X", "Y", or "Z".	([ABC]{2}.*)|(.*[^XYZ])
Select all strings that begin with "AA" followed by either one or more numbers or one or more letters, and then by the character "_". It can finish with any characters. This would, for example, select the string `AA11_XYZ76` and the string `AAFGH_43KKK`, but not the string `AA8H_3232IHSDG`, because this latter has both numbers and letters between the "AA" and the "_".	`A{2}([0-9]+|[A-Z]+)_.*`

Regular expressions supported by the plan extractor

The following tables provide full details of the regular expressions supported by the plan extractor.

Table 40. Regular expressions supported by the plan extractor: character

Syntax	Filters for?
`x`	The character `x` (regular expressions are case-sensitive)
`\\`	The backslash character
`\0n`	The character with octal value `0n` (0 <= n <= 7)
`\0nn`	The character with octal value `\0nn` (0 <= n <= 7)
`\0mnn`	The character with octal value `\0mnn`(0 <= m <= 3, 0 <= n <= 7)
`\0xhh`	The character with hexadecimal value `0xhh`

Table 40. Regular expressions supported by the plan extractor: character (continued)

Syntax	Filters for?
`\uhhhh`	The character with hexadecimal value `0xhhhh`
`\t`	The tab character ('`\u0009`')
`\n`	The newline (line feed) character ('`\u000A`')
`\r`	The carriage-return character ('`\u000D`')
`\f`	The form-feed character ('`\u000C`')
`\a`	The alert (bell) character ('`\u0007`')
`\e`	The escape character ('`\u001B`')
`\cx`	The control character corresponding to `x`

Table 41. Regular expressions supported by the plan extractor: character classes

Syntax	Filters for?
`[abc]`	`a, b`, or `c` (simple class)
`[^abc]`	Any character except `a, b`, or `c` (negation)
`[a-zA-Z]`	`a` through `z` or `A` through `Z`, inclusive (range)
`[a-d[m-p]]`	`a` through `d`, or `m` through `p: [a-dm-p]` (union)
`[a-z&&[def]]`	`d, e`, or `f` (intersection)
`[a-z&&[^bc]]`	`a` through `z`, except for `b` and `c: [ad-z]` (subtraction)
`[a-z&&[^m-p]]`	`a` through `z`, and not `m` through `p: [a-lq-z]` (subtraction)

Table 42. Regular expressions supported by the plan extractor: predefined character classes

Syntax	Filters for?
`.`	Any character (might or might not match line terminators)
`\d`	A digit: `[0-9]`
`\D`	A non-digit: `[^0-9]`
`\s`	A whitespace character: `[ \t\n\x0B\f\r]`
`\S`	A non-whitespace character: `[^\s]`
`\w`	A word character: `[a-zA-Z_0-9]`
`\W`	A non-word character: `[^\w]`

Table 43. Regular expressions supported by the plan extractor: POSIX character classes (US-ASCII only)

Syntax	Filters for?
`\p{Lower}`	A lowercase alphabetic character: `[a-z]`
`\p{Upper}`	An uppercase alphabetic character:`[A-Z]`
`\p{ASCII}`	All ASCII:`[\x00-\x7F]`
`\p{Alpha}`	An alphabetic character: `[\p{Lower}\p{Upper}]`
`\p{Digit}`	A decimal digit: `[0-9]`
`\p{Alnum}`	An alphanumeric character: `[\p{Alpha}\p{Digit}]`
`\p{Punct}`	Punctuation: One of !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
`\p{Graph}`	A visible character: `[\p{Alnum}\p{Punct}]`
`\p{Print}`	A printable character: `[\p{Graph}]`
`\p{Blank}`	A space or a tab: `[ \t]`
`\p{Cntrl}`	A control character: `[\x00-\x1F\x7F]`
`\p{XDigit}`	A hexadecimal digit: `[0-9a-fA-F]`
`\p{Space}`	A whitespace character: `[ \t\n\x0B\f\r]`

Table 44. Regular expressions supported by the plan extractor: classes for Unicode blocks and categories

Syntax	Filters for?
`\p{InGreek}`	A character in the Greek block (simple block)
`\p{Lu}`	An uppercase letter (simple category)
`\\p{Sc}`	A currency symbol
`\P{InGreek}`	Any character except one in the Greek block (negation)
`[\p{L}&&[^\p{Lu}]]`	Any letter except an uppercase letter (subtraction)

Table 45. Regular expressions supported by the plan extractor: boundary matchers

Syntax	Filters for?
`^`	The beginning of a line
`$`	The end of a line
`\b`	A word boundary
`\B`	A non-word boundary
`\A`	The beginning of the input
`\G`	The end of the previous match

Table 45. Regular expressions supported by the plan extractor: boundary matchers (continued)

Syntax	Filters for?
`\Z`	The end of the input but for the final terminator, if any
`\z`	The end of the input

Table 46. Regular expressions supported by the plan extractor: greedy quantifiers

Syntax	Filters for?
`X?`	X, once or not at all
`X*`	X, zero or more times
`X+`	X, one or more times
`X{n}`	X, exactly *n* times
`X{n,}`	X, at least *n* times
`X{n,m}`	X, at least *n* but not more than *m* times

Table 47. Regular expressions supported by the plan extractor: reluctant quantifiers

Syntax	Filters for?
`X??`	X, once or not at all
`X*?`	X, zero or more times
`X+?`	X, one or more times
`X{n}?`	X, exactly *n* times
`X{n,}?`	X, at least *n* times
`X{n,m}?`	X, at least *n* but not more than *m* times

Table 48. Regular expressions supported by the plan extractor: possessive quantifiers

Syntax	Filters for?
`X?+`	X, once or not at all
`X*+`	X, zero or more times
`X++`	X, one or more times
`X{n}+`	X, exactly *n* times
`X{n,}+`	X, at least *n* times
`X{n,m}+`	X, at least *n* but not more than *m* times

Table 49. Regular expressions supported by the plan extractor: logical operators

Syntax	Filters for?
`XY`	`X` followed by `Y`
`X|Y`	Either `X` or `Y`
`(X)`	`X`, as a capturing group

Table 50. Regular expressions supported by the plan extractor: back references

Syntax	Filters for?
`\n`	Whatever the nth capturing group matched

Table 51. Regular expressions supported by the plan extractor: quotation

Syntax	Filters for?
`\`	Nothing, but quotes the following character
`\Q`	Nothing, but quotes all characters until `\E`
`\E`	Nothing, but quotes all characters until `\Q`

Table 52. Regular expressions supported by the plan extractor: special constructs (non-capturing)

Syntax	Filters for?
`(?:X)`	`X`, as a non-capturing group
`(?idmsux-idmsux)`	Nothing, but turns match flags on - off
`(?idmsux-idmsux:X)`	`X`, as a non-capturing group with the given flags on - off
`(?=X)`	`X`, via zero-width positive lookahead
`(?!X)`	`X`, via zero-width negative lookahead
`(?<=X)`	`X`, via zero-width positive lookbehind
`(?<!X)`	`X`, via zero-width positive lookbehind
`(?>X)`	`X`, as an independent, non-capturing group

SQL report examples

Examples of queries that can be run using the SQL custom reports.

This section provides some examples of queries that can be run using the SQL custom reports.

Jobs grouped by return codes

For each return code, this query returns the number of jobs that ended with the corresponding return code:

```
SELECT DISTINCT return_code AS RC count(job_name) AS ,#JOB

FROM mdl.job_history_v

GROUP BY return_code
```

Table 53. Example of query outcome

RC	#JOB
0	1670
5	11
6	1
50	2
127	352

Job statistics grouped on job status

For each job status, this query returns the number of jobs that ended with the corresponding job status and also the planned duration time, the total elapsed time, and total CPU time:

```
SELECT job_status, count(job_name) AS job count, floor(sum(planned_duration/1000)) AS

planned duration, floor(sum(total_elapsed_time/1000)) AS total elapsed,

floor(sum(total_cpu_time/1000)) AS total cpu

FROM mdl.job_history_v GROUP BY job_status

FROM mdl.job_history_v

GROUP BY return_code
```

Table 54. Example of query outcome

JOB_STATUS	JOB COUNT	PLANNED DURATION	TOTAL ELAPSED	TOTAL CPU
A	366	0	21960	0
S	1670	1413360	1423500	183

Jobs in a range of return code

This query returns the number of job in a range of return codes

```
SELECT *

FROM (select DISTINCT return_code, count(job_name) AS #JOB

FROM mdl.job_history_v

GROUP BY return_code) AS temp

WHERE return_code > 0 AND return_code < 6
```

Table 55. Example of query outcome

RETURN_CODE	#JOB
5	11

Jobs that ran within a time range and finished with a specific job status

```
SELECT WORKSTATION_NAME, JOB_NAME, JOB_RUN_DATE_TIME

FROM MDL.JOB_HISTORY_V

WHERE JOB_RUN_DATE_TIME BETWEEN '2008-05-19 10:00:00.0' AND '2008-05-19

21:00:00.0' AND JOB_STATUS <> 'S'

ORDER BY JOB_RUN_DATE_TIME
```

Table 56. Example of query outcome

WORKSTATION_NAME	JOB_NAME	JOB_RUN_DATE_TIME
NC122072	PEAK_A_06	2008–08–03 23:23:00.0
NC122072	JOB_RER_A	2008–08–03 23:23:00.0
NC122072	PEAK_A_13	2008–08–03 23:23:00.0
NC122072	PEAK_A_20	2008–08–03 23:23:00.0
NC122072	PEAK_A_27	2008–08–03 23:23:00.0
NC122072	PEAK_A_43	2008–08–03 23:23:00.0
NC122072	PEAK_B_19	2008–08–03 23:24:00.0

Related information

Creating a task to Create Custom SQL Reports on page 210

Event rule

An *event rule* defines a set of actions that run when specific event conditions occur. An event rule definition correlates events and trigger actions.

For information about how to define event rules, see the topic about defining event rules in the *User's Guide and Reference* .

Related information

Event management on page 91

Event management configuration on page 13

Creating an event rule on page 121

Action properties

How to specify and use action properties.

When you select an action, its properties are displayed on the right of the table, where you can edit them.

Mandatory property fields are marked by asterisks. If you try to save an action without specifying one or more mandatory values, the required fields turn red and the rule cannot be saved. If available, default values are automatically entered in the fields.

Using variable information into action properties

To better qualify your action, you can use some event properties as variable information that can be added to the action properties. To use variables, select the variable icon above the field. If you select the **machine-readable format** check box the variable is used as input to a command or a script; otherwise, if you don't select it, the variable is included in normal text information (e.g. in a message or an email).

For example, you can include the job name in the mail body, if you have mail notification as a response action to a job-related event. You can include this variable information together with normal text in action properties that require a string value. For those properties that require a numeric value, you can enter either the variable information or a number.

Otherwise, you can click the lookup icon , enter your filter criteria and click **Search**. All the model items that reflect the inserted criteria are shown. Selecting an item, the fields are automatically filled in with its values.

Event properties

How to use event properties.

When you select an event, its properties are displayed on the right of the table, where you can edit them.

When you choose the event properties, you define a *filter* for all the events that you want to monitor and manage. The most meaningful properties you choose are logically correlated and represent the *event scope*, which is displayed in the related event row.

Mandatory property fields are marked by asterisks. If you try to save an event without specifying one or more mandatory values, the required fields turn red and the rule cannot be saved. If available, default values are automatically entered in the fields.

Some property fields support wildcard characters. If supported, the wildcard icon appears above the field.

You can also specify multiple values in the same field, separating them by semicolon (**;**). In this case, the properties are logically correlated by the conjunction *or*. When available, you see the multiple filters icon above the field.

For example, if you create a **Job Status Changed** event specifying the **Job name** property as "A;B;C", an action is triggered each time either of the specified jobs changes its status.

Furthermore, by clicking on **Add Properties** at the end of the *Properties* panel, you can add the same property field multiple times and assign different values to it. The properties are logically correlated by the conjunction *and*, creating a cumulative filter.

To remove an added properties, click on the close icon . You cannot remove mandatory properties.

For example, when you define a **Job Status Changed** event, if you want to exclude some jobs from it, you can define as event properties all the job names that *match* `production_*`, and all the job names that *do not match* `production_*_1`.

Chapter 18. Glossary

A

access method

An executable file used by extended agents to connect to and control jobs on other operating systems (for example, z/OS) and applications (for example, Oracle Applications, PeopleSoft, and SAP R/3). The access method is specified in the workstation definition for the extended agent. This term applies only to IBM® Workload Scheduler distributed environments. See also "extended agent on page 251".

actual start

The time that a job or job stream, planned to run during current production, actually starts. See also:

- "earliest start on page 250"
- "latest start on page 253"
- "planned start on page 255"
- "scheduled time on page 257"

ad hoc job

A job used to run a command or a script file that is inserted into the current plan. These jobs are not saved in the IBM® Workload Scheduler database. See also:

- "database on page 249"
- "plan on page 255"

ad hoc prompt dependency

A prompt dependency that is defined within the properties of a job or job stream and is unique to that job or job stream. This term applies only to IBM® Workload Scheduler distributed environments. See also "prompt dependency on page 256".

availability

The degree to which a system or resource is ready when needed to process data.

B

batchman

A process running on IBM® Workload Scheduler workstations. This process accesses the copy of the Symphony file distributed to workstations at the beginning of the production period and updates it, resolving dependencies. It is the only process that can update the Symphony file. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "processes on page 256"
- "production period on page 256"
- "symphony file on page 258"

backup domain manager

A full status fault-tolerant agent in a distributed network that can assume the responsibilities of its domain manager. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "fault-tolerant agent on page 251"
- "full status on page 252"
- "domain manager on page 250"

backup master domain manager

A full status fault-tolerant agent in a distributed network that can assume the responsibilities of the master domain manager. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "fault-tolerant agent on page 251"
- "full status on page 252"
- "master domain manager on page 254"

C

calendar

A list of scheduling dates used either to identify the dates when job streams or jobs can be run (when used with inclusive run cycles), or when they cannot be run (when used with exclusive run cycles). See also:

carry forward

A property of a job stream that, if not completed before the end of the current production period, is carried forward to the next and then to subsequent periods, until the latest start time is reached or the job stream completes. See also "latest start on page 253".

computer workstation

In the IBM® Z Workload Scheduler, either:

- A workstation that performs z/OS processing of jobs and started-task jobs, and that usually reports status to Dynamic Workload Console automatically.

or:

- A processor used as a workstation. It can refer to single processors or multiprocessor complexes serving a single job queue, for example JES2 or JES3 systems.

connector

An installed component that provides the interface between the Dynamic Workload Console and the engine. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "engine on page 250"

controller

In the IBM® Z Workload Scheduler, the component that runs on the management hub, and that contains the tasks that manage the plans and the database.

cpu time

The processor time used by a job when running. See also "duration on page 250".

current plan

In the IBM® Z Workload Scheduler the detail plan of activity that covers a period of at least 1 minute, and not more than 21 days. A current plan typically covers 1 or 2 days.

D

database

In IBM® Workload Scheduler for distributed environments it consists of a set of tables defined in relational database, such as DB2 or Oracle, containing the definitions for all scheduling objects (jobs, job streams, resources, workstations, domains, parameters, prompts and files), job and job stream statistics, user data, and the last time an object was modified.

In the IBM® Z Workload Scheduler it consists of collection of the following data: calendar, period, workstation description, JCL variable table, application description, and operation instruction.

deadline

The time by which a job or job stream is set to complete. When a job or job stream passes the deadline, notifications are sent to users and integrated applications, but the job or job stream is not prevented from running if all time restrictions and dependencies are satisfied. Jobs or job streams that have not yet started or that are still running after the deadline time has expired are marked as "late" in the plan. See also "plan on page 255".

dependency

A prerequisite condition that must be satisfied before a job or job stream can start or continue to run. See also:

- "external dependency on page 251"
- "file dependency on page 251"
- "follows dependency on page 251"
- "prompt dependency on page 256"
- "resource dependency on page 256"

distributed network

A connected group of workstations defined using IBM® Workload Scheduler for distributed environments. See also:

- "engine on page 250"
- "workstation on page 259"

distributed workstation

A workstation in the IBM® Workload Scheduler distributed environment on which jobs and job streams are run. See also:

- "engine on page 250"
- "workstation on page 259"

domain

A named group of workstations in a IBM® Workload Scheduler distributed network, consisting of one or more agents and a domain manager acting as the management hub. All domains have a parent domain except for the master domain. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "domain manager on page 250"
- "master domain manager on page 254"

domain manager

An installed component in a IBM® Workload Scheduler distributed network that is the management hub in a domain. All communication to and from the agents in the domain is routed through the domain manager. This term applies only to IBM® Workload Scheduler distributed environments. See also "workstation on page 259"

duration

The elapsed time that a job is expected to take to complete (estimated duration) and actually takes (actual duration). See also:

- "cpu time on page 249"
- "time restriction on page 258"

E

earliest start

The time before which a job or job stream cannot start. The job or job stream can start after the earliest start time provided that all other time restrictions and dependencies are satisfied. See also:

- "actual start on page 247"
- "latest start time on page 253"
- "planned start on page 255"
- "scheduled time on page 257"

engine

The core software for the scheduling environment. The engine can be either a z/OS engine (installed as part of IBM Z Workload Scheduler), or a distributed engine (installed as part of IBM Workload Scheduler).

extended agent

An agent used to integrate Tivoli® Workload Scheduler job control features with other entities such as operating systems (for example, z/OS) and applications (for example, Oracle Applications, PeopleSoft, and SAP R/3). Extended agents must be hosted by a master domain manager, domain manager, or an agent (not another extended agent), and use access methods to communicate with the other entities. This term applies only to IBM® Workload Scheduler distributed environments. See also "access method on page 247".

external dependency

A dependency defined in one job or job stream that refers to another job stream or to a job in another job stream.

external job

A job referred to in an external dependency. See also "external dependency on page 251".

F

fault-tolerant agent

A installed agent component in a distributed network capable of resolving local dependencies and launching its jobs. This term applies only to IBM® Workload Scheduler distributed environments.

fence

An attribute on a workstation that regulates whether a job can be run on a workstation. The job fence is a priority level that the priority of a job must exceed before it can run. This term applies only to IBM® Workload Scheduler distributed environments.

file dependency

A dependency where a job or job stream cannot start until it finds a specific file is present in a specific path on a specific workstation. Sometimes called an opens file dependency. This term applies only to IBM® Workload Scheduler distributed environments. See also "dependency on page 249".

follows dependency

A dependency where a job or job stream cannot start until other jobs or job streams have completed successfully. See also "dependency on page 249".

forecast plan

A projection over a selected time interval based on the job streams and dependencies defined in the database. This term applies only to IBM® Workload Scheduler distributed environments. See also

- "database on page 249"
- "plan on page 255"

FTA

See "fault-tolerant agent on page 251"

full status

An attribute of an agent that enables it to be updated with the status of jobs and job streams running on all other workstations in its domain and in subordinate domains, but not on peer or parent domains. A backup domain manager or master domain manager must be full status. See also:

- "backup domain manager on page 248"
- "domain on page 250"
- "master domain manager on page 254"

G

general workstation

In the IBM® Z Workload Scheduler this is a workstation where activities other than printing and processing are carried out. A general workstation is usually manual, but it can also be automatic. Manual activities can include data entry and job setup.

H

host

A workstation required by extended agents. It can be any IBM® Workload Scheduler workstation except another extended agent. This term applies only to IBM® Workload Scheduler distributed environments.

I

internal status

The current status of jobs and job streams in the IBM® Workload Scheduler engine. The internal status is unique to IBM® Workload Scheduler. See also "status on page 258".

internetwork dependencies

A dependency between jobs or job streams in separate IBM® Workload Scheduler networks. This term applies only to IBM® Workload Scheduler distributed environments. See also "network agent on page 254".

internetwork job or job stream

A job or job stream in a remote IBM® Workload Scheduler network that is referenced by an internetwork dependency defined for a job or job stream in the local network. This term applies only to IBM® Workload Scheduler distributed environments. See also "network agent on page 254".

J

job

A unit of work scheduled for a specific run date in the plan that is processed at a workstation. In IBM® Z Workload Scheduler, a job is an operation performed at a computer workstation.

job limit

See "limit on page 253"

job status

See "status on page 258".

job stream

A list of jobs that scheduled to run as a unit in the plan (such as a weekly backup application), along with run cycles, times, priorities, and other dependencies that determine the exact order in which the jobs run.

jobman

A process running on IBM® Workload Scheduler workstations. This process controls the launching of jobs under the direction of **batchman** and reports job status back to **mailman**. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "batchman on page 247"
- "jobmon on page 253"
- "mailman on page 254"

jobmon

An additional process running on IBM® Workload Scheduler Windows workstations. It manages and monitors job processing. A separate jobmon process is created to launch and monitor each job. This term applies only to IBM® Workload Scheduler distributed environments. It reports job status back to **jobman**. See also:

- "jobman on page 253"
- "processes on page 256"

L

latest start

The time before which the job or job stream must start. The job or job stream can start before the latest start time provided that all other dependencies are satisfied. See also:

- "actual start on page 247"
- "earliest start time on page 250"
- "planned start on page 255"
- "scheduled time on page 257"

limit

A means of allocating a specific number of job slots into which IBM® Workload Scheduler is allowed to launch jobs. A job limit can be set for each job stream, and for each workstation. For example, setting the workstation job limit to 25 permits IBM® Workload Scheduler to have no more than 25 jobs running concurrently on the workstation.

list

A means of filtering plan and objects and presenting them in a table. Lists on Dynamic Workload Console are the result of running tasks.

M

mailman

The mail management process running on IBM® Workload Scheduler workstations. It routes messages about jobs processing status to local and remote workstations. This term applies only to IBM® Workload Scheduler distributed environments. See also "processes on page 256".

master domain manager

The management hub of the top-level domain in the IBM® Workload Scheduler distributed network. It maintains the database of all scheduling objects in the domain and the central configuration files. The master domain manager generates and distributes the current plan through the network as a program control file named Symphony file. In addition, logs and reports for the network are maintained on the master domain manager. See also:

- "backup master domain manager on page 248"
- "database on page 249"
- "domain on page 250"
- "plan on page 255"

MDM

See "master domain manager on page 254".

N

netman

The network management process that is started first when the IBM® Workload Scheduler workstations starts up. It processes **start**, **stop**, **link** or **unlink** commands or requests to run actions impacting the job processing on that workstation. The **netman** process examines each request received and either implements the request itself or create a local IBM® Workload Scheduler process to do so. This term applies only to IBM® Workload Scheduler distributed environments. See also "processes on page 256".

network agent

A logical extended agent used to create dependencies between jobs and job streams on separate IBM® Workload Scheduler networks. This term applies only to IBM® Workload Scheduler distributed environments. See also "internetwork dependencies on page 252".

O

opens file dependency

See "file dependency on page 251".

open interval

The time interval during which a workstation is active and can process work.

owner ID

In IBM® Z Workload Scheduler, the identifier that represents the job stream owner.

P

parameter

An entity that enables job instance-specific values to be substituted in job and job stream scripts, either from values set in the database or at run time. Parameters cannot be used when scripting extended agent jobs. This term applies only to IBM® Workload Scheduler distributed environments.

plan

The means of scheduling jobs. See also:

- "database on page 249"
- "current plan on page 256"
- "trial plan on page 258"

planned start

The time that IBM® Workload Scheduler estimates a job instance will start. This estimate is based on start times of previous instances of the job. See also:

- "actual start on page 247"
- "earliest start on page 250"
- "latest start on page 253"
- "scheduled time on page 257"

portlet

A pluggable user interface components that are managed and displayed in a web portal.

predecessor

A job or job stream that must complete successfully before successor jobs or job streams can be started. See also "successor on page 258".

predefined prompt dependency

A prompt dependency that is defined in the database and can be associated to any job or job stream. This term applies only to IBM® Workload Scheduler distributed environments. See also "prompt dependency on page 256".

printer workstation

In IBM® Z Workload Scheduler, a workstation that prints output and usually reports status automatically.

priority

A way of determining the order in which jobs and job streams start. Priorities for each job and job stream range from 0 to 101. A job or job stream with a priority of 0 does not run.

production period

The time interval covered by the current plan. See also "current plan on page 256".

current plan

Contains all job scheduling activity planned for a production period. The plan is stored in the Symphony file, and consists of all the jobs, job streams, and dependency objects that are scheduled to run for that period, including any jobs or job streams carried forward from the previous plan. See also:

- "carry forward on page 248"
- "plan on page 255"

processes

IBM® Workload Scheduler workstation processes that control the scheduling environment and network traffic. This term applies only to IBM® Workload Scheduler distributed environments. See also:

- "batchman on page 247"
- "jobman on page 253"
- "jobmon on page 253"
- "mailman on page 254"
- "netman on page 254"

prompt dependency

A dependency where an operator must respond affirmatively to a prompt so that the dependent job or job stream can run. See also:

- "ad hoc prompt dependency on page 247"
- "predefined prompt dependency on page 255"

R

resource

Either physical or logical system resources. Resources are used as dependencies for jobs and job streams. See also "resource dependency on page 256".

resource dependency

A dependency where a job or job stream cannot start until the required quantity of the defined resource is available. See also "resource on page 256".

restart and cleanup

A recovery function that ensures the restart of a job and the related cleanup actions, for example, deleting temporary files or dataset created in a job run.

rule-based run cycle

A run cycle that uses rules based on lists of ordinal numbers, types of days, and common calendar intervals (or period names in IBM® Z Workload Scheduler). For example, the last Thursday of every month. Rule-based run cycles are based on conventional periods, such as calendar months, weeks of the year, and days of the week. In IBM® Z Workload Scheduler, run cycles can also be based on periods that you define, such as a semester. This term is only used as such in IBM® Z Workload Scheduler, but the concept applies also to the distributed product. See also:

- "run cycle on page 257"

run cycle

Specifies the days that a job stream is scheduled to run. The specification can be in the form of a rule or as a combination of period and offset. See also:

- "calendar on page 248"

S

schedule

See "job stream on page 253".

scheduled time

The time when a job or job stream is scheduled to run. See also:

- "actual start on page 247"
- "earliest start on page 250"
- "latest start on page 253"
- "planned start on page 255"

slack time

In a critical job predecessor path, the slack time is the amount of time the predecessor processing can be delayed without exceeding the critical job deadline. It is the spare time calculated using the deadline, input arrival, and duration settings of predecessors jobs

special resource

In IBM® Z Workload Scheduler, a resource that is not associated with a particular workstation, such as a data set.

standard agent

An installed agent component in a distributed network that runs jobs, but requires a domain manager to resolve local dependencies and launch the jobs.

started-task operation

In IBM® Z Workload Scheduler, an operation that starts or stops started tasks. This operations are run at a computer workstation with the STC option specified.

status

The current job or job stream status within the Dynamic Workload Console. The Dynamic Workload Console status is common to IBM® Workload Scheduler and IBM® Z Workload Scheduler. See also "internal status on page 252".

successor

A job that cannot start until all of the predecessor jobs or job streams on which it is dependent are completed successfully. See also: "predecessor on page 255".

Symphony file

A file containing the scheduling information needed by the production control process (**batchman**) to run the plan. The file is built and loaded when the current plan is created or extended on the master domain manager. During the production phase, this file is continually updated to indicate the current status of production processing: work completed, work in progress, and work to be done. The tasks run from the Dynamic Workload Console display the contents of the Symphony file (plan) and are run against it. See also:

- "batchman on page 247"
- "plan on page 255"

T

task

A filter that returns, when run, a list of scheduling objects of the same type whose attributes satisfy the criteria set in the task definition.

time restriction

Determines the times before which, after which, or both, that a job or job stream cannot be run. Specifying both defines a time interval within which a job or job stream runs. Jobs can also have a repetition rate. For example, IBM® Workload Scheduler can launch the same job every 30 minutes between the hours of 8:30 a.m. and 1:30 p.m.

tracker

In IBM® Z Workload Scheduler, a component that runs on every system in your complex. It acts as the communication link between the z/OS system that it runs on and the controller.

trial plan

A projection of the current plan for a different period, using the same start date. It is used to determine the effect of different plan decisions. This term applies only to IBM® Workload Scheduler distributed environments. See also "plan on page 255".

V

virtual workstation

It is a computer workstation where you define a pool of destinations for the workload submission. When the scheduler processes the operations submitted to a virtual workstation, it distributes the workload according a balanced turn criteria. To obtain the job submission, at least one of the destinations in the pool must be available.

W

workstation

A definition of an individual computer or computer partition on which jobs and job streams are run. Types of workstation vary depending on the type of engine. See also:

- "distributed workstation on page 250"
- "z/OS workstation on page 259"

workstation resource

In IBM® Z Workload Scheduler, a physical resource, such as a tape driver, that must be allocated among jobs. When you define a workstation, you can specify the quantity of each of the two resources (R1 and R2) that are available to jobs. When defining jobs to that workstation, you can specify the number of these resources that must be available for the operation to start on that workstation.

write-to-operator workstation

In IBM® Z Workload Scheduler, a general workstation that lets you use scheduling facilities to issue a write-to-operator (WTO) message at a specific operator console defined by the workstation destination.

X

x-agent

See "extended agent on page 251".

Z

z/OS network

A connected group of workstations that use the z/OS engine to perform workload scheduling. See also:

- "engine on page 250"
- "workstation on page 259"

z/OS workstation

A representation of system configuration elements in the IBM® Z Workload Scheduler network. For the IBM® Z Workload Scheduler engine, workstations can be:

- Computer
- General
- Printer

IBM® Z Workload Scheduler requires that you define for every workstation: the type of work it does, the quantity of work it can handle at any particular time, and the time it is active. See also "workstation on page 259".

Notices

This document provides information about copyright, trademarks, terms and conditions for product documentation.

© Copyright IBM Corporation 1993, 2016 / © Copyright HCL Technologies Limited 2016, 2025

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

© (HCL Technologies Limited) (2025).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. 2016

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM® or other companies. A current list of IBM® trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

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

Linear Tape-Open™, LTO™, the LTO™ Logo, Ultrium™, and the Ultrium™ logo are trademarks of HP, IBM® Corp. and Quantum in the U.S. and other countries.

Intel™, Intel™ logo, Intel Inside™, Intel Inside™ logo, Intel Centrino™, Intel Centrino™ logo, Celeron™, Intel Xeon™, Intel SpeedStep™, Itanium™, and Pentium™ are trademarks or registered trademarks of Intel™ Corporation or its subsidiaries in the United States and other countries.

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

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

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

Cell Broadband Engine™ is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

ITIL™ is a Registered Trade Mark of AXELOS Limited.

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

A

abend prompt 72
access control list
  security 126
access method jobs 114, 115
accessibility viii
action
  generic 92
  notification 92
  operational 92
  properties
    using variable information 245
actions on security objects
  specifying actions on security objects 134
ad-hoc prompt 72
adding
  security role 129
agents pool
  creating 105
analyzing
  current plan 159
  working plan 159
api keys 132
application monitoring event 91
archived plan 87
archived plans list
  modifying number 40
archived plans number
  displayed in Monitor Workload view 40
auditing
  Self-Service Catalog 39
  Self-Service Dashboards 39

B

backup master domain manager 50
beacon 30

C

calendar 112
  definition 62, 112
  holidays 62
call to a web service
  sample JSDL files 114
call to a Web service
  sample JSDL files 115
changes
  keeping track of 199
changing
  settings repository user 21
  user password 145
Cloud & Smarter Infrastructure technical training viii
Cognos
prompt type
  date syntax 120
  time stamp syntax 120
  time syntax 120
communication
  type based on SSL communication options 221
complex expressions 238
condition dependencies
  definition 72, 77
  examples 79
  handling recovery 80
  step level 80
conditional dependencies
  definition 74
conditional logic 74, 77
conditional recovery job option 80
configuring
  federator 173
console
  customizing 23
  portfolio 10
  start 10
controlling
  job processing
    by using dependencies 185
    by using job confirmation 188
    by using job priority 187
    by using job recovery actions 189
    by using limits 188
    by using time restrictions 186
    by using workstation fence 187
    edit job definition 189
  job stream processing
    by using dependencies 185
    by using job priority 187
    by using limits 188
    by using time restrictions 186
    by using workstation fence 187
create
  security domains 128
  show plan view 147
Create and customize your board 25
creating
  API Keys
    scenario 130
  customized dashboard 164
  database items 110
  event rule 121
  items in the database 104
  objects in the database 125
  page 164
  pool of agents 105
  task
    generate Custom SQL report 210
    generate Plan report 208
    generate Workstation Workload Summary report 207
    monitor domains 169
    monitor event rules 180
    monitor job streams 178
    monitor jobs 178
    monitor operator messages 184
    monitor triggered actions 182
    monitor workstations 168, 178
creating task generate Job Run History report 206
creating task generate Job Run Statistics report 205
creating task generate Workstation Workload Runtimes report 208
credentials 112
  definition 112
credentials definition 112
critical job
  high risk status 100
  monitoring 98
  monitoring scenario
    by using workload service assurance 212
  planning 101
  potential risk status 100
  processing 98
  tracking 100
critical path
  calculation of 98
cross dependencies 72, 81
current plan 86
  analyzing 159
customization 23
customizing
  console 23
  dashboard 164
  news notification beacon 30
  page 164
  portfolio 24
  startup page 25
cyclic periods
  definition 60
  examples 61

D

daily run cycle 62
dashboard
  customizing 164
database
  definition 86
  items 112, 112, 113, 119, 120
database data extract 115
database data extraction 114
database data validation 114, 115
database items
  creating 110
  credentials 112
  editing 110
  variable table 119
  workstations 120
database objects
  file 73
  job 58
  job stream 59
  prompt 72
  user 83
  workstation class 84
database operations
  sample JSDL files 114, 115
database repository
  changing user 21
database stored procedure
  database jobs
    sample JSDL files 114, 115
  sample JSDL files 114, 115
date syntax
Cognos
  prompt type 120
DB2 repository
  changing user 21
defining
  credentials 112
  database items
    credentials 112
    variable tables 119
    workstations 120
  folder 113
  workstation 120
defining a
  calendar 112
  folder 113
defining a line of business
  line of business 120
defining an

user 112
definition
variable table 119
dependencies
condition 72, 77
conditional 74
cross 72, 81
definition 69
external 70, 71
file 73
internal 70, 71
internetwork 70
prompt 72
using
to control job processing 185
to control job stream processing 185
designing 106
disabling
news notification beacon 30
display
plan view 147
domain 54
monitoring 169
domain manager 51
domains
monitoring 178
drag and drop 159
duplicate security domains 129
duplicate security role 130
dwc 106, 112, 119, 120
DWC 112, 113
exporting database data 19
importing database data 19
switching database vendor 19
DWC leak 28
dynamic agent
53
dynamic agent workstation 103
dynamic critical path 97
dynamic pool 53, 103
creating 105
dynamic scheduling 53, 53, 53
workstation definition 120
dynamic workload console 112, 112, 113, 119, 120
Dynamic Workload Console
accessibility viii
getting started 10
troubleshooting 218
Dynamic Workload Console
global settings
25
Dynamic Workload Console
customizing
25
dynamic workstations 53, 53, 53

E

editing
database items 110
event rules 122
items in the database 104
objects in the database 125
education viii
embedded task 118
engine
connection 90
event
application monitoring 91
definition 91
generic 91
management 91
task 182
objects related 91
properties 246
rules 93
event rule 245
creating 121
monitoring 180
event rules
editing 122
managing 122
examples
SQL report 242
exclusive run cycle 62
executable jobs 114, 115
exporting
settings 19
exporting database data
DWC 19
expression
complex 238
regular 237, 238
extended agent workstation 52, 103
external dependencies 70, 71

F

fault-tolerant agent 51
federator 173
feed 30
file
definition 73
monitoring event 91
file transfer jobs
sample JSDL files 114, 115
file transfer operations
sample JSDL files 114, 115
files
monitoring 178
FNCJSI
preproduction plan 89
folder 113
definition 113
folder definition 113
forecast plan 88
generating 192

G

generating
forecast plan 192
report
Custom SQL 210
Plan 208
Workstation Workload Summary 207
trial plan 192
generating report Job Run Statistics 205
generating report Workstation Workload Runtimes 208
generating reportJob Run History 206
generic
action 92
event 91
generic Java job
template 114, 115
generic web service call
template 114
generic Web service call
template 115
Give access
Give access to user or group 126
global prompt 72
global settings
customizing 25
graphical designer 118
Graphical designer 106
graphical view
Workload Designer 156
Graphical view
in the plan 149
groups 219

H

high risk status 100
holidays calendar 62
hot list 97
addition of jobs 99

I

IBM i jobs 114, 115
AS400 jobs 114, 115
importing
settings 19
importing database data
DWC 19
inclusive run cycle 62
installation
Dynamic Workload Console
9, 15
installing
Dynamic Workload Console
9, 15
internal dependencies 70, 71
internal status
distributed job stream 228
distributed jobs 223
z/OS job stream 230
z/OS jobs 226
internetwork dependencies 70
item 113
items 112, 112, 119, 120

J

J2EE jobs 114, 115
Java jobs
sample JSDL files 114, 115
Java operations
sample JSDL files 114, 115
JnextPlan 85
job
ad hoc
setting properties for 197
submitting 195
addition to the hot list 99
controlling process
by using dependencies 185
by using job confirmation 188
by using job priority 187
by using job recovery actions 189
by using limits 188
by using time restrictions 186
by using workstation fence 187
edit job definition 189
critical
monitoring 98
processing 98
definition 58
internal status 223, 226
monitoring scenario 214
predefined
setting properties for 197
submitting 196
return codes 242
statistics
return codes 243
status

description 222, 225
mapping 224, 226
time range 244
job definition
creating 114, 115
job stream 59
controlling process
by using dependencies 185
by using job priority 187
by using limits 188
by using time restrictions 186
by using workstation fence 187
internal status 228, 230
predefined
setting properties for 197
submitting 196
status
description 227, 230
mapping 229, 230
job stream definition
creating 117
job stream view 153
job types
template 114, 115
job types with advanced options
sample JSDL files 114, 115
template 114, 115

L

limits
using
to control job processing 188
to control job stream processing 188
local prompt 72
logical resource 73
login page 23
login page Private Label 23
logo 23
long term plan
preproduction plan 89

M

manage access 127
Manage access control list
access control list 127
managing
settings 19
settings repository 19, 22
master domain manager 50
Memory leak 28
mirroring data 173
monitor 171, 172
monitoring 171, 172
creating dashboard 164
domain 169
domains 178
event rules 180
files 178
items in the plan 147
job streams 178
jobs 178
jobs on multiple engines
scenario 214
new page 164
operator messages 184
plan 161
prompts 178
resources 178
scheduling environment 167
triggered actions 182
workload 170
workstation 168, 178
z/OS critical jobs
by using workload service assurance 212
monitoring scenario 172
monitoring task
query line 178
monthly run cycle 62
MSSQL jobs 114, 115
multiple engines
monitor jobs scenario 214
multiple-domain network 55

N

named prompt 72
network
multiple-domain 55
single-domain 54
new executors
template 114, 115
new plug-ins 114, 115
template 114, 115
news notification beacon
customize 30
disabling 30
non-cyclic periods
examples 61
noncyclic periods
definition 60
notification
news
disabling 30
enabling 30
notification action 92

O

object
creating 110
editing 110
related event 91
object attribute values
specifying object attribute values 140
object attributes
attributes for object types 139
offset-based run cycle 62
operational action 92
operator instruction
definition 69
operator message
monitoring 184
option
conditional recovery job 80
recovered by condition 80
orchestration monitor 171
overview 106

P

page
customizing dashboard 164
parameter 69
path
critical
calculation of 98
period
cyclic 60
definition 60
noncyclic 60
Personalized Reports 210
physical resource 73
plan 171
archived 87
current 86
analyzing 159
forecast 88
generating 192
monitoring 161
monitoring items 147
preproduction 87
production 86
Symnew 87
trial 87
generating 192
working
analyzing 159
selecting 191
plan extractor
regular expressions supported 238
plan view 152
display 147
Plan View
auto refresh
Plan View 30
Show Plan View
auto refresh 30
planning
critical job 101
pool 53
creating 105
portfolio
console 10
customizing 24
potential risk status 100
predecessor
successor 89
predecessors, hide
What-if Analysis view 41
preproduction plan 87
description 89
FNCJSI 89
long term plan 89
Private Label 23
Private Labeling 23
production
plan 85, 86
process 85
submitting workload in 195
promotion 98
prompt
abend 72
ad-hoc 72
global 72
local 72
named 72
recovery 72
prompt type
Cognos
date syntax 120
time stamp syntax 120
time syntax 120
prompts
monitoring 178

Q

query line
monitoring task 178

R

recovered by condition option 80
recovery prompt 72
regular expressions 237, 238
remote engine workstation 53, 72, 82
remove security domains
edit security domains 129
remove security role
edit security role 130

report
Custom SQL
generating 210
definition 93
format 96
header 96
Plan
generating 208
Workstation Workload Summary
generating 207
report Job Run History generating 206
report Job Run Statistics generating 205
report Workstation Workload Runtimes
generating 208
repository
changing user 21
settings 19
resource
logical 73
physical 73
scheduling 73
resources
monitoring 178
return codes
job statistics grouped by 243
jobs grouped by 242
rule 245
rule-based run cycle 62
run cycle
daily 62
exclusive 62
inclusive 62
monthly 62
offset-based 62
rule-based 62
simple 62
weekly 62
yearly 62

S

SAP
dynamic agent 103
dynamic pool 103
extended agent 103
IBM Workload Scheduler
102
z-centric agent 103
scenario 172
scenarios 212
scheduling environment
monitoring 167
scheduling objects
keeping track of changes 199
scheduling resource 73
security domain 127
security domains
create 128
security role
adding 129
security roles
security 129
selecting
working plan 191
Self-Service Catalog
auditing 39
Self-Service Dashboards
auditing 39
service api 132
service api keys 132
service keys 132
session timeout 28
setting
properties
for ad hoc jobs 197
for predefined job streams 197
for predefined jobs 197
settings
exporting 19
importing 19
managing 19
repository 19
settings repository
managing 19, 22
shadow job 58, 72
simple run cycle 62
single-domain network 54
specific job types
sample JSDL files 114, 115
SQL report
examples 242
SSL communication options 221
standard agent 52
starting
console 10
startup page
customizing 25
status description
distributed job stream 227
distributed jobs 222
z/OS job stream 230
z/OS jobs 225
status mapping
distributed job stream 229
distributed jobs 224
z/OS job stream 230
z/OS jobs 226
streamlogon
credentials definition 112
submitting
ad hoc jobs 195
predefined job streams 196
predefined jobs 196
workload in production 195
successor
predecessor 89
switching database vendor
DWC 19
Symnew plan 87
Symphony file 85, 102
syntax for date
Cognos
prompt type 120
syntax for parameterized filter
Cognos
parameterized filter 120
prompt type 120
syntax for time
Cognos
parameterized filter 120
prompt type 120
syntax for time stamp
Cognos
prompt type 120

T

task 118
event management 182
generate Custom SQL report 210
generate Plan report 208
generate Workstation Workload Summary
report 207
monitor domains 169
monitor event rules 180
monitor job streams 178
monitor jobs 178
monitor operator messages 184
monitor triggered actions 182
monitor workstations 168, 178
task generate Job Run History report 206
task generate Job Run Statistics report 205
task generate Workstation Workload Runtimes
report 208
TdwcGlobalSettings.xml.template file 25
technical training viii
time range 244
time restrictions
using
to control job processing 186
to control job stream processing 186
time stamp syntax
Cognos
prompt type 120
time syntax
Cognos
prompt type 120
tracking
critical job 100
training
technical viii
trial plan 87
generating 192
trigger action 245
triggered actions
monitoring 182
troubleshooting
Dynamic Workload Console
218

U

ui 118
user 83
change password 145
users 219

V

variable 84
definition 119
table 119, 119
variable information
using into action properties 245
variable table 84
definition 119
variable table definition 119
view
job stream 153
plan 152
view access 127
View access for security domain
view access 127
View access for users or groups
view access control list 127
viewing
job 171
job streams 171
virtual workstation 52

W

web service jobs
sample JSDL files 114
Web service jobs
sample JSDL files 115
weekly run cycle 62
What-if Analysis
hide predecessors 41

widgets 163
working plan
analyzing 159
monitoring 191
workload
monitoring 170
submitting in production 195
workload application
definition 59
workload broker
agent workstation 52
Workload Designer
graphical view 156
workload service assurance 96
scenario 212
workstation
backup master domain manager 50
class 84
defining 120
definition 50, 120
domain manager 51
dynamic agent
53, 103
dynamic pool 53
extended agent 52, 103
fault-tolerant agent 51
managing 120
master domain manager 50
monitoring 168, 178, 178, 178, 178
pool 53
remote engine 53, 82
standard agent 52
virtual 52
workload broker agent 52
z-centric agent 103
workstation definition 120, 120
workstation fence
using
to control job processing 187
to control job stream processing 187
workstation
IBM Z Workload Scheduler Agent
52

Y

yearly run cycle 62

Z

z-centric agent workstation 103