

IBM InfoSphere DataStage
Version 11 Release 3

*Operations Console Guide and
Reference*



IBM InfoSphere DataStage
Version 11 Release 3

*Operations Console Guide and
Reference*



Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 113.

Contents

Chapter 1. Administering workload management. 1

Introduction to workload management	1
Getting started with the workload management system	2
Configuring workload management	2
Configuration options for workload management	3
Starting and stopping the workload management process	4

Chapter 2. Monitoring jobs. 5

Introduction to monitoring jobs	5
Getting started with the Operations Console.	6
Creating and configuring the operations database	7
Operations database	7
Creating the operations database schema	7
Identifying an existing operations database.	17
Configuring the operations database	20
Managing the data collection processes	33
Data collection processes	33
Starting and stopping the AppWatcher process	34
Managing the data collection processes using the DSAppWatcher.sh script	36
Handling data collection process errors	41
Job runtime errors	43
Monitoring jobs and job runs by using the Operations Console.	43
Overview of the Operations Console	43
Scenarios for using the Operations Console.	44
User access to the Operations Console	45
Operations Console system requirements	46
Starting the Operations Console	46
Viewing log messages in the Operations Console	47
Configuring the display of data in the console.	47
Tabs and panels in the Operations Console	50
Extracting monitoring data from the operations database	57
Extract the names of all the jobs running on a specific engine	58
Extract the full status of all the jobs started after a certain time on any host	58
Extract the details of all jobs ever run, showing the run type as readable string	58
Extract the job run logs for a particular job run	59
Extract the details of all the job runs with a particular parameter set to a given value	59
Extract the details of all job runs that were active after a given time on a particular host	60
Extract the details of all job runs that were active in a given period on a particular host	60
Extract the slowest jobs in a project based on their last runs	61
Maintenance of the operations database	61
Reporting database content	62

Deleting historical data from the operations database	63
Changing the user password for the operations database	68
Deleting the operations database	71

Chapter 3. Operations database schema reference 73

Job run audit data tables and views	74
JobRun table	74
JobExec table	76
Host table	77
JobRunParams table	78
JobRunParamsView view	78
JobRunLog table.	79
ParallelConfig table.	79
ParallelConfigNodes view	80
Job run metrics tables and views	80
JobStage table	80
JobRunStage table	81
JobLink table	82
JobRunLink table	83
DataLocator table	83
Reference tables and views	84
MasterRef table	85
JobTypeRef view.	86
RunTypeRef view	86
RunMajorStatus view	86
RunMinorStatus view	86
LogTypeRef view	87
StageStatusRef view	87
LinkTypeRef view	88
Job run time-series data tables and views	88
JobRunUsage table	88
JobRunTotalRowUsage view.	89
System resource usage tables and views.	89
HostDetail table	89
Current system resource usage tables and views	91
Past system resource usage tables and views	95

Appendix A. Product accessibility . . . 105

Appendix B. Contacting IBM 107

Appendix C. Accessing the product documentation. 109

Appendix D. Providing feedback on the product documentation 111

Notices and trademarks 113

Index 119

Chapter 1. Administering workload management

You can use the workload management queues to control the starting of parallel and server jobs.

By default, the workload management system is enabled. Review the default configuration of the workload management system before using it. Additionally, review the workload management system policies that are specified using the IBM® InfoSphere® DataStage® and QualityStage® Operations Console. For more information, see System Policies panel.

If you do not want to use the workload management system, you must disable it by setting the configuration option `WLMON=0`. For more information, see “Configuring workload management” on page 2.

Introduction to workload management

The installation of the IBM InfoSphere Information Server engine tier installs the InfoSphere DataStage components, including executables and configuration files, that are required to use the workload management system on an engine system.

Using the workload management system on an engine system involves these components:

- A server process called `WLMServer`, which provides information about the workload management queues to the IBM InfoSphere DataStage and QualityStage Operations Console.
- The Operations Console, which displays information about the workload management queues for each engine system.

Workload management queues

The workload management system uses a set of user-defined queues to control the starting of parallel and server jobs. When a job is started, it is submitted to a designated queue. The queue to which a job is submitted can be specified for that job, or it can be inherited from the project that contains the job, or it can be the default queue.

Workload management system policies

The workload management system uses a set of system policies to determine whether another job should be allowed to start processing. The workload management queues are used to determine which specific job is allowed to start processing.

Workload management predefined queues

A number of workload management queues are predefined, some of which are used by product modules in IBM InfoSphere Information Server. The following workload management queues are predefined:

Table 1. Predefined workload management queues

Queue name	Description
ISD	Used by InfoSphere Information Services Director
IA	Used by IBM InfoSphere Information Analyzer
DataClick	Used by IBM InfoSphere Data Click
LowPriorityJobs	Low priority queue
MediumPriorityJobs	Medium priority queue
HighPriorityJobs	High priority queue

Getting started with the workload management system

Before you use the workload management system, you must complete these tasks.

About this task

You use the Operations Console to define and monitor your workload management queues.

Procedure

1. Configure the workload management configuration properties. See [Configuring workload management](#).
2. Start the workload management process, called `WLMServer`, that provides information about the workload management queues. See [Starting the workload management process](#).
3. Start the Operations Console. See [Starting the Operations Console](#).
4. Specify the workload management system policies. See [System Policies panel](#).
5. Define the workload management queues. See [Queue Management panel](#).
6. Specify the workload management queue to be used for your projects and jobs. See [Setting the workload management queue and Job Run Option window](#).
7. Run your InfoSphere jobs.

Configuring workload management

You must configure the workload management system before you can use the workload management queues to control the starting of parallel and server jobs.

About this task

The workload management configuration options are set in the `DSOdbConfig.cfg` configuration file. This configuration file is stored in the `Server/DSOdb` directory in the installation directory.

Procedure

1. Specify whether the workload management system is enabled or disabled.
2. If the workload system is enabled, configure the workload management configuration options.

Configuration options for workload management

You can configure the options that control the workload management system.

The workload management configuration options are set in the DSODBConfig.cfg configuration file. This configuration file is stored in the Server/DSODB directory in the installation directory.

As a minimum, you must set WLMON=1 if you want to use the workload monitoring system. The default values for the other options are sufficient to enable the use of the workload management system in most environments.

Table 2. Options that control the workload management system

Option	Values	Purpose
WLMON	0 or 1	Controls if the workload management system is enabled. By default, the workload management system is enabled. Set WLMON=0 if you do not want to use the workload management system. Set WLMON=1 if you want to use the workload management system.
WLM_CONTINUE_ON_COMMS_ERROR	0 or 1	Controls whether jobs are run if communication with the workload management system is not available. Set WLM_CONTINUE_ON_COMMS_ERROR=0 if you do not want jobs to run. Set WLM_CONTINUE_ON_COMMS_ERROR=1 if you want jobs to run. The default value is 1. Setting this value is equivalent to running with workload management set to disabled.
WLM_CONTINUE_ON_QUEUE_ERROR	0 or 1	Controls whether jobs are run if the specified queue name does not exist. Set WLM_CONTINUE_ON_QUEUE_ERROR=0 if you do not want jobs to run. Set WLM_CONTINUE_ON_QUEUE_ERROR=1 if you want jobs to run. The default value is 1. The default queue for the project is used if it is specified and it exists, otherwise the workload management default queue is used.
WLM_QUEUE_WAIT_TIMEOUT	integer	Specifies the number of seconds a job remains in the queue before it is stopped and removed from the queue. Set WLM_QUEUE_WAIT_TIMEOUT=0 if you want jobs to remain in the queue indefinitely.

Starting and stopping the workload management process

The workload management process, WLMServer, is automatically started and stopped as part of the DataStage Engine startup.

About this task

If you enable or disable the workload management system, you must restart the DataStage Engine.

Chapter 2. Monitoring jobs

You can use the IBM InfoSphere DataStage and QualityStage Operations Console to monitor the job runs, services, system resources, and workload management queues on several IBM InfoSphere Information Server engines.

You can use the IBM InfoSphere DataStage and QualityStage Operations Console to start and stop job runs, and to define and administer your workload management queues.

The Operations Console provides complementary monitoring capabilities to the capabilities provided by the IBM InfoSphere DataStage and QualityStage Director client.

Introduction to monitoring jobs

You must configure an IBM InfoSphere Information Server engine system before you can monitor it using the IBM InfoSphere DataStage and QualityStage Operations Console.

The installation of the engine tier installs the InfoSphere DataStage components, including executables and configuration files, that are required for an engine system to be monitored. The installation also checks or sets permissions on some directories.

The monitoring of an engine system involves these components:

- The operations database, which holds information about DataStage jobs that are running, or have been run, on one or more engine systems. The database also holds information about how system resources are used over time on those systems.
- Extensions to the DataStage job runtime system to capture job runtime events.
- A server process called EngMonApp, which transfers the job runtime events to the operations database.
- A server process called ResMonApp, which monitors system resource usage and record the results in the operations database.
- A server process called ResTrackApp, which provides usage data about their local system to the ResMonApp process.
- A server process called ODBQueryApp, which provides information from the operations database to the Operations Console.
- A server process called WLMServer, which provides information about the workload management queues to the Operations Console.
- The Operations Console, which displays information about jobs, job activity, and system resources for each engine system.

To enable monitoring of job runs services, and system resources on an engine system, you must complete these steps after installing the engine tier:

1. Set up a schema and tables in the operations database, or identify an existing operations database which this system can share.
2. Configure the engine system monitor to connect to its operations database.

3. Configure the monitor so it knows what data to collect, and how frequently, if you do not want to use the default.
4. Ensure that the monitor server processes are started on each engine and that monitoring is switched on.

If you do not perform these steps, the engine tier is available for use, but monitoring is not enabled.

To enable monitoring of workload management queues on an engine system, you must complete these steps after installing the engine tier:

1. Configure the workload management configuration properties.
2. Ensure that the workload management server process, called WLMServer, is started on each engine and that workload management monitoring is switched on.

If you do not perform these steps, the engine tier is available for use, but monitoring of workload management queues is not enabled.

Getting started with the Operations Console

Before you use the IBM InfoSphere DataStage and QualityStage Operations Console to monitor your InfoSphere Information Server systems, you must complete these tasks.

About this task

The Operations Console can be used to monitor either your job runs, services, and system resources, or your workload management queues, or to monitor both.

Procedure

1. To monitor your job runs, services, and system resources, you must complete these tasks:
 - a. Create the operations database schema. See “Creating the operations database schema” on page 7.
 - b. Configure the monitoring data that is stored in the operations database. You must set DSODBON=1 to enable the capture of monitoring data. See “Configuring the monitoring data” on page 21.
 - c. Check the configuration of the monitoring system to ensure that the connection to the operations database is correctly configured. See “Checking the configuration of the monitoring system” on page 40.
 - d. Start the DSAppWatcher process that starts and manages the data collection processes. See “Starting and stopping the AppWatcher process” on page 34.
 - e. Check that the data collection processes are running. See “Displaying the status of the AppWatcher process and all its managed processes” on page 38.
2. To monitor your workload management queues, you must complete these tasks:
 - a. Configure the workload management configuration properties. See Configuring workload management.
 - b. Start the workload management process, called WLMServer, that provides information about the workload management queues. See Starting the workload management process.

3. Start the Operations Console. See “Starting the Operations Console” on page 46.
4. Run your InfoSphere DataStage and QualityStage jobs.

Creating and configuring the operations database

You must create and configure the operations database before you can use the IBM InfoSphere DataStage and QualityStage Operations Console to monitor IBM InfoSphere DataStage and QualityStage jobs.

About this task

Note: The operations database is only supported on DB2® in this version of IBM InfoSphere Information Server. Scripts for Oracle and SQL Server are not provided.

Operations database

The operations database stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

The operations database holds information about InfoSphere DataStage jobs that are running, or have been run, on one or more InfoSphere Information Server engine systems. The operations database also holds information about how system resources are used over time on those engine systems.

The Operations Console does not display all the information that is stored in the operations database. In particular, the contents of the JobRunStage, JobRunLink, and DataLocator tables, which are only populated if you set the MonitorLinks=1 configuration option, are not currently used by the Operations Console. Also, more detailed information about system resource usage is captured than is currently displayed in the Operations Console.

You must choose which database that you want to include the operations database:

- Metadata repository database:
 - For performance reasons, use the metadata repository database for production systems only after careful consideration.
 - If the operations database schema does not exist in the metadata repository database, run the scripts that create the operations database schema, and set the configuration connection properties to point to it.
 - If the operations database schema exists in the metadata repository database, configure the engine to point to it.
- Another database that is specific to this InfoSphere Information Server engine:
 - Run the scripts that create the operations database schema.
 - Configure the engine to point to it.
- Another database that is shared between InfoSphere Information Server engines:
 - Collect details of where the existing schema is located, and configure the engine to point to it.

Creating the operations database schema

You need to create the operations database schema that stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

Before you begin

Identify the database system that you want to use for the operations database. See the system requirements for the supported database systems.

Note: The operations database is only supported on DB2 in this version of IBM InfoSphere Information Server. Scripts for Oracle and SQL Server are not provided.

If you created the operations database using DB2 during installation, you do not need to carry out this step.

If you want to use an existing operations database, see “Identifying an existing operations database” on page 17.

Note: All the scripts for creating the operations database schema should be run in the same terminal session. If you want to run the scripts in different terminal sessions, run these commands before running the remainder of the scripts:

1. `db2`
2. connect to *dbname* where *dbname* is the name of the metadata repository database
3. `exit`

Procedure

1. On the computer that hosts the services tier, register the repository to be used for the operations database. See “Registering the repository for the operations database” on page 9. This step can be run by any user who has the DataStage and QualityStage Administrator role.
2. On the computer that hosts the services tier, generate the scripts to be used to create and delete the operations database schema. See “Generating the scripts for the operations database schema” on page 10. This step can be run by any user who has the DataStage and QualityStage Administrator role.
3. Run the generated scripts using the command line for the appropriate database system. This step must be run by a user that has administrative rights to create a schema.

For more information about running the generated scripts, see the topic for your database system and operating system:

- “Creating the operations database schema for a DB2 database on a Windows system” on page 11
 - “Creating the operations database schema for a DB2 database on a UNIX or Linux system” on page 12
 - “Creating the operations database schema for an Oracle database on a Windows system” on page 12
 - “Creating the operations database schema for an Oracle database on a UNIX or Linux system” on page 13
 - “Creating the operations database schema for an SQL Server database” on page 14
4. If the generated scripts issue errors, see “Recovering from errors in the scripts for the operations database schema” on page 14. This step can be run by any user who has the DataStage and QualityStage Administrator role.
 5. On the computer that hosts the engine tier, generate a new repository connection file for the operations database. See “Generating the repository

connection file for the operations database” on page 16. This step can be run by any user who has the DataStage and QualityStage Administrator role.

Registering the repository for the operations database

Register the repository to be used for the operations database.

About this task

Follow the instructions for the operating system that is used for the services tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Registering the repository for the operations database on Linux or Unix: Procedure

1. On the computer that hosts the services tier, switch to the `/InformationServer/ASBServer/bin` directory.
2. Check if the repository for the operations database is already registered:
 - a. Run this command to list the registered repositories:
`RepositoryAdmin.sh -listRepositories`
 - b. Run this command to see the details of any repository in the list:
`RepositoryAdmin.sh -displayRepository -reposName name`

where **name** is the name of the repository.

3. If the repository for the operations database is not already registered, register the repository:
 - a. Copy the `DS0DBrepos_registration.properties` template file from the installation directory on the computer hosting the engine tier into the current directory. The installation directory is `/InformationServer/Server/DS0DB` on Linux or Unix, or `\InformationServer\Server\DS0DB` on Windows.
 - b. Edit the `DS0DBrepos_registration.properties` file, using the instructions contained in the file.

If you plan to use DB2 for the operations database schema, these restrictions apply:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the `db2` command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.
- you cannot create the operations database schema on a system that has only the DB2 client installed.

If you plan to use SQL Server for the operations database schema, this restriction applies:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the `sqlcmd` command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.

- c. Run this command to register the repository for the operations database:

```
RepositoryAdmin.sh -registerRepository -propertyFile
DS0DBrepos_registration.properties
```

Registering the repository for the operations database on Windows: Procedure

1. On the computer that hosts the services tier, switch to the `\InformationServer\ASBServer\bin` directory.
2. Check if the repository for the operations database is already registered:
 - a. Run this command to list the registered repositories:
`RepositoryAdmin.bat -listRepositories`
 - b. Run this command to see the details of any repository in the list:
`RepositoryAdmin.bat -displayRepository -reposName name`

where **name** is the name of the repository.

3. If the repository for the operations database is not already registered, register the repository:
 - a. Copy the `DSODBrepos_registration.properties` template file from the installation directory on the computer hosting the engine tier into the current directory. The installation directory is `/InformationServer/Server/DSODB` on Linux or Unix, or `\InformationServer\Server\DSODB` on Windows.
 - b. Edit the `DSODBrepos_registration.properties` file, using the instructions contained in the file.

If you plan to use DB2 for the operations database schema, these restrictions apply:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the `db2` command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.
- you cannot create the operations database schema on a system that has only the DB2 client installed.

If you plan to use SQL Server for the operations database schema, this restriction applies:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the `sqlcmd` command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.

- c. Run this command to register the repository for the operations database:
`RepositoryAdmin.bat -registerRepository -propertyFile
DSODBrepos_registration.properties`

Generating the scripts for the operations database schema

Generate the scripts that are used to create the operations database.

About this task

Follow the instructions for the operating system that is used for the services tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Generating the scripts for the operations database on Linux or Unix:

Procedure

1. On the computer that hosts the services tier, switch to the /InformationServer/ASBServer/bin directory.
2. Run the **RepositoryAdmin** tool to generate the required scripts:
`RepositoryAdmin.sh -saveSQLScripts -reposName name -scriptLocation location`

where
 - **name** is the name of the repository.
 - **location** is the directory where the scripts are created.
3. If the computer hosting the services tier is not the same as that hosting the operations database, copy the scripts to the computer hosting the operations database.

Generating the scripts for the operations database on Windows: Procedure

1. On the computer that hosts the services tier, switch to the \InformationServer\ASBServer\bin directory.
2. Run the **RepositoryAdmin** tool to generate the required scripts:
`RepositoryAdmin.bat -saveSQLScripts -reposName name -scriptLocation location`

where
 - **name** is the name of the repository.
 - **location** is the directory where the scripts are created.
3. If the computer hosting the services tier is not the same as that hosting the operations database, copy the scripts to the computer hosting the operations database.

Creating the operations database schema for a DB2 database on a Windows system

You need to create the operations database schema that stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

About this task

If you created the operations database during installation, you do not need to carry out this step.

If the setup scripts issue errors, go to step 4 on page 8 in “Creating the operations database schema” on page 7.

Procedure

1. If you want to create the operations database on a path that is not the root where DB2 is installed, you must set the operating system level environment variable `DB2_CREATE_DB_ON_PATH=YES`. After setting the `DB2_CREATE_DB_ON_PATH` environment variable you must stop and then restart IBM WebSphere® Application Server and DB2.
2. Run the setup scripts from a DB2 command line.
 - a. Ensure that you are logged in as a user that can execute the **db2** command and that has access to the scripts that were generated in the previous steps.

- b. If you want to create the operations database on a path that is not the root where DB2 is installed, check that the environment variable `DB2_CREATE_DB_ON_PATH` is set to YES:

```
SET DB2_CREATE_DB_ON_PATH
```

- c. If you are creating the operations database in a new DB2 database, rather than in an existing database such as the metadata repository, run this command to create the database:

```
db2 -l dsodb_setup_db.log -stf dsodb_db_creation.sql
```

- d. To create the database schema and tables, enter the following commands:

```
db2 -l dsodb_setup_tablespace.log -stf dsodb_tablespace_creation.sql
```

```
db2 -l dsodb_setup_table.log -stf dsodb_table_creation.sql
```

- e. To configure user permissions, enter the following command:

```
db2 -l dsodb_setup_user.log -stf dsodb_user_config.sql
```

Creating the operations database schema for a DB2 database on a UNIX or Linux system

You need to create the operations database schema that stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

About this task

If you created the operations database during installation, you do not need to carry out this step.

If the setup scripts issue errors, go to step 4 on page 8 in “Creating the operations database schema” on page 7.

Procedure

Run the setup scripts.

1. Ensure that you are logged in as a user that can execute the **db2** command and that has access to the scripts that were generated in the previous steps.
2. If you are creating the operations database in a new DB2 database, rather than in an existing database such as the metadata repository, run this command to create the database:

```
db2 -l dsodb_setup_db.log -stf dsodb_db_creation.sql
```

3. To create the database schema and tables, enter the following commands:

```
db2 -l dsodb_setup_tablespace.log -stf dsodb_tablespace_creation.sql
```

```
db2 -l dsodb_setup_table.log -stf dsodb_table_creation.sql
```

4. To configure user permissions, enter the following command:

```
db2 -l dsodb_setup_user.log -stf dsodb_user_config.sql
```

Creating the operations database schema for an Oracle database on a Windows system

You need to create the operations database schema that stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

About this task

If the setup scripts issue errors, go to step 4 on page 8 in “Creating the operations database schema” on page 7.

Procedure

Run the setup scripts.

1. Ensure that you are logged in as a user that can execute the **sqlplus** command and that has access to the scripts that were generated in the previous step.
2. To configure user permissions, enter the following command:

```
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_user_config.sql
```

For example, enter this command:

```
sqlplus system/manager@ORCL @dsodb_user_config.sql
```

3. To create the schema and tables, enter the following commands:

```
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_tablespace_creation <dsodbpassword>
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_table_creation <dsodbpassword>
```

For example, enter these commands:

```
sqlplus system/manager@ORCL @dsodb_tablespace_creation.sql test
sqlplus system/manager@ORCL @dsodb_table_creation.sql test
```

Creating the operations database schema for an Oracle database on a UNIX or Linux system

You need to create the operations database schema that stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

About this task

If the setup scripts issue errors, go to step 4 on page 8 in “Creating the operations database schema” on page 7.

Procedure

Run the setup scripts.

1. Ensure that you are logged in as a user that can execute the **sqlplus** command and that has access to the scripts that were generated in the previous step.
2. To configure user permissions, enter the following command:

```
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_user_config.sql
```

For example, enter this command:

```
sqlplus system/manager@ORCL @dsodb_user_config.sql
```

3. To create the schema and tables, enter the following commands:

```
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_tablespace_creation <dsodbpassword>
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_table_creation <dsodbpassword>
```

For example, enter these commands:

```
sqlplus system/manager@ORCL @dsodb_tablespace_creation.sql test
sqlplus system/manager@ORCL @dsodb_table_creation.sql test
```

Creating the operations database schema for an SQL Server database

You need to create the operations database schema that stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

About this task

If the setup scripts issue errors, go to step 4 on page 8 in “Creating the operations database schema” on page 7.

Procedure

1. If the computer hosting the services tier is not the same as that hosting the operations database, copy the scripts to the computer hosting the operations database.
2. Run the setup scripts.
 - a. Ensure that you are logged in as a user that can execute the `sqlcmd` command and that has access to the scripts that were generated in the first step.
 - b. If you are creating the operations database in a new SQL Server database, rather than in an existing database such as the metadata repository, run this command to create the database:

```
sqlcmd -b -i dsodb_db_creation.sql >> logfile
```

For example, enter the command:

```
sqlcmd -b -i dsodb_db_creation.sql >> c:\DSODB_setup.log
```

- c. To create the database schema and tables, enter the following command:

```
sqlcmd -b -i dsodb_table_creation.sql >> logfile
```

For example, enter the command:

```
sqlcmd -b -i dsodb_table_creation.sql >> c:\DSODB_setup.table.log
```

- d. To configure user permissions, enter the following command:

```
sqlcmd -b -i dsodb_user_config.sql >> logfile
```

For example, enter the command:

```
sqlcmd -b -i dsodb_user_config.sql >> c:\DSODB_setup.user.log
```

Recovering from errors in the scripts for the operations database schema

Recover from errors that are issued when the scripts for the operations database schema are run.

About this task

This task can be run by any user who has the DataStage and QualityStage Administrator role.

The properties of the operations database repository and the operations database are specified in the `DSODBrepos_registration.properties` file.

Procedure

1. If any of the scripts encountered a problem because an invalid parameter was specified for the operations database repository, complete one of the following sets of tasks:
 - a. Update the existing definition of the operations database repository. Follow these steps:
 - 1) Update the definition of the operations database repository. Follow the instructions below for the operating system that is used for the services tier.
 - 2) Restart the process from step 2 on page 8 in “Creating the operations database schema” on page 7.
 - b. Unregister the operations database repository, update the definition of the repository, and re-register the repository. Follow these steps:
 - 1) Unregister the operations database repository. Follow the instructions below for the operating system that is used for the services tier.
 - 2) Restart the process from step 1 on page 8 in “Creating the operations database schema” on page 7.
2. If any of the scripts encountered a problem because an invalid parameter was specified for the operations database, complete the following tasks:
 - a. Unregister the operations database repository. Follow the instructions below for the operating system that is used for the services tier.
 - b. Restart the process from step 1 on page 8 in “Creating the operations database schema” on page 7.

Updating the definition of the operations database repository on Linux or UNIX:

Procedure

1. On the computer that hosts the services tier, switch to the /InformationServer/ASBServer/bin directory.
2. Display the details of the operations database repository:

```
RepositoryAdmin.sh -displayRepository -reposName name
```

where **name** is the name of the operations database repository.

3. Update the parameter that was incorrectly specified:

```
RepositoryAdmin.sh -updateRepository -reposName name -parameter newvalue
```

where

- **name** is the name of the operations database repository.
- **parameter** is the name of the parameter to be updated.
- **newvalue** is the new value for the parameter.

You can run this command to see a full list of parameters that can be updated:

```
RepositoryAdmin.sh -help
```

Updating the definition of the operations database repository on Windows:

Procedure

1. On the computer that hosts the services tier, switch to the \InformationServer\ASBServer\bin directory.
2. Display the details of the operations database repository:

```
RepositoryAdmin.bat -displayRepository -reposName name
```

where **name** is the name of the operations database repository.

3. Update the parameter that was incorrectly specified:

```
RepositoryAdmin.bat -updateRepository -reposName name -parameter newvalue
```

where

- **name** is the name of the operations database repository.
- **parameter** is the name of the parameter to be updated.
- **newvalue** is the new value for the parameter.

You can run this command to see a full list of parameters that can be updated:

```
RepositoryAdmin.bat -help
```

Unregistering the operations database repository on Linux or UNIX: Procedure

1. On the computer that hosts the services tier, switch to the /InformationServer/ASBServer/bin directory.
2. Unregister the operations database repository:

```
RepositoryAdmin.sh -unregisterRepository -reposName name
```

where **name** is the name of the operations database repository.

3. If you are creating the operations database in a new database, rather than in an existing database such as the metadata repository, unregister the operations database:

```
RepositoryAdmin.sh -unregisterDatabase -dbName dbname
```

where **dbname** is the name of the operations database as specified in the DSODBrepos_registration.properties file.

Unregistering the operations database repository on Windows: Procedure

1. On the computer that hosts the services tier, switch to the \InformationServer\ASBServer\bin directory.
2. Unregister the operations database repository:

```
RepositoryAdmin.bat -unregisterRepository -reposName name
```

where **name** is the name of the operations database repository.

3. If you are creating the operations database in a new database, rather than in an existing database such as the metadata repository, unregister the operations database:

```
RepositoryAdmin.bat -unregisterDatabase -dbName dbname
```

where **dbname** is the name of the operations database as specified in the DSODBrepos_registration.properties file.

Generating the repository connection file for the operations database

The repository connection file for the operations database is used by the engine tier to connect to the operations database.

About this task

Follow the instructions for the operating system used for the engine tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Generating the repository connection file for the operations database on Linux or Unix:

Procedure

1. On the computer that hosts the engine tier, switch to the `/InformationServer/Server/DSODB` directory.
2. Run the **RegistrationCommand** tool to generate a new repository connection file for the operations database:

```
../../ASBNode/bin/RegistrationCommand.sh -user isadmin -password pswd -gcf  
-repository name -cf DSODBConnect.tpl  
-results DSODBConnect.cfg
```

where

- **isadmin** is the Information Server admin user.
- **pswd** is the password for the **isadmin** user.
- **name** is the repository for the operations database.

The configuration file, `DSODBConnect.cfg`, is created in the `/InformationServer/Server/DSODB` directory.

Generating the repository connection file for the operations database on Windows:

Procedure

1. On the computer that hosts the engine tier, switch to the `/InformationServer/Server/DSODB` directory.
2. Run the **RegistrationCommand** tool to generate a new repository connection file for the operations database:

```
..\..\ASBNode\bin\RegistrationCommand.bat -user isadmin -password pswd -gcf  
-repository name -cf DSODBConnect.tpl  
-results DSODBConnect.cfg
```

where

- **isadmin** is the Information Server admin user.
- **pswd** is the password for the **isadmin** user.
- **name** is the repository for the operations database.

The configuration file, `DSODBConnect.cfg`, is created in the `\InformationServer\Server\DSODB` directory.

Identifying an existing operations database

You can configure an InfoSphere Information Server engine system to connect to an existing operations database.

About this task

This step can be run by any user who has the DataStage and QualityStage Administrator role.

Procedure

1. On the computer that hosts the services tier, register the repository to be used for the operations database. See “Registering the repository for the operations database” on page 9.
2. On the computer that hosts the engine tier, generate a new repository connection file for the operations database. See “Generating the repository connection file for the operations database” on page 16.

Registering the repository for the operations database

Register the repository to be used for the operations database.

About this task

Follow the instructions for the operating system that is used for the services tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Registering the repository for the operations database on Linux or Unix: Procedure

1. On the computer that hosts the services tier, switch to the `/InformationServer/ASBServer/bin` directory.
2. Check if the repository for the operations database is already registered:
 - a. Run this command to list the registered repositories:

```
RepositoryAdmin.sh -listRepositories
```
 - b. Run this command to see the details of any repository in the list:

```
RepositoryAdmin.sh -displayRepository -reposName name
```

where **name** is the name of the repository.
3. If the repository for the operations database is not already registered, register the repository:
 - a. Copy the `DS0DBrepos_registration.properties` template file from the installation directory on the computer hosting the engine tier into the current directory. The installation directory is `/InformationServer/Server/DS0DB` on Linux or Unix, or `\InformationServer\Server\DS0DB` on Windows.
 - b. Edit the `DS0DBrepos_registration.properties` file, using the instructions contained in the file.

If you plan to use DB2 for the operations database schema, these restrictions apply:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the `db2` command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.
- you cannot create the operations database schema on a system that has only the DB2 client installed.

If you plan to use SQL Server for the operations database schema, this restriction applies:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the `sqlcmd` command during the schema creation process. This

user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.

- c. Run this command to register the repository for the operations database:

```
RepositoryAdmin.sh -registerRepository -propertyFile  
DSODBrepos_registration.properties
```

Registering the repository for the operations database on Windows: Procedure

1. On the computer that hosts the services tier, switch to the `\InformationServer\ASBServer\bin` directory.
2. Check if the repository for the operations database is already registered:
 - a. Run this command to list the registered repositories:

```
RepositoryAdmin.bat -listRepositories
```
 - b. Run this command to see the details of any repository in the list:

```
RepositoryAdmin.bat -displayRepository -reposName name
```

where **name** is the name of the repository.

3. If the repository for the operations database is not already registered, register the repository:
 - a. Copy the `DSODBrepos_registration.properties` template file from the installation directory on the computer hosting the engine tier into the current directory. The installation directory is `/InformationServer/Server/DSODB` on Linux or Unix, or `\InformationServer\Server\DSODB` on Windows.
 - b. Edit the `DSODBrepos_registration.properties` file, using the instructions contained in the file.

If you plan to use DB2 for the operations database schema, these restrictions apply:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the **db2** command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.
- you cannot create the operations database schema on a system that has only the DB2 client installed.

If you plan to use SQL Server for the operations database schema, this restriction applies:

- the user that you specify to access the operations database must be an existing operating system user, and cannot be the same user that you use to execute the **sqlcmd** command during the schema creation process. This user must also be different from the those specified for any other DataStage component repository, such as the metadata repository or the Information Analyzer repository.

- c. Run this command to register the repository for the operations database:

```
RepositoryAdmin.bat -registerRepository -propertyFile  
DSODBrepos_registration.properties
```

Generating the repository connection file for the operations database

The repository connection file for the operations database is used by the engine tier to connect to the operations database.

About this task

Follow the instructions for the operating system used for the engine tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Generating the repository connection file for the operations database on Linux or Unix:

Procedure

1. On the computer that hosts the engine tier, switch to the `/InformationServer/Server/DSODB` directory.
2. Run the **RegistrationCommand** tool to generate a new repository connection file for the operations database:

```
../../ASBNode/bin/RegistrationCommand.sh -user isadmin -password pswd -gcf  
-repository name -cf DSODBConnect.tpl  
-results DSODBConnect.cfg
```

where

- **isadmin** is the Information Server admin user.
- **pswd** is the password for the **isadmin** user.
- **name** is the repository for the operations database.

The configuration file, `DSODBConnect.cfg`, is created in the `/InformationServer/Server/DSODB` directory.

Generating the repository connection file for the operations database on Windows:

Procedure

1. On the computer that hosts the engine tier, switch to the `/InformationServer/Server/DSODB` directory.
2. Run the **RegistrationCommand** tool to generate a new repository connection file for the operations database:

```
..\..\ASBNode\bin\RegistrationCommand.bat -user isadmin -password pswd -gcf  
-repository name -cf DSODBConnect.tpl  
-results DSODBConnect.cfg
```

where

- **isadmin** is the Information Server admin user.
- **pswd** is the password for the **isadmin** user.
- **name** is the repository for the operations database.

The configuration file, `DSODBConnect.cfg`, is created in the `\InformationServer\Server\DSODB` directory.

Configuring the operations database

You must configure the operations database before you can use the IBM InfoSphere DataStage and QualityStage Operations Console to monitor InfoSphere DataStage and QualityStage jobs.

About this task

You must configure the connection to the operations database before data can be stored in the database.

Configuring the monitoring data

You can customize the monitoring data that is stored in the operations database.

About this task

The data monitoring options for all jobs in all projects are set in the `DSOdbConfig.cfg` configuration file. This configuration file is stored in the `/Server/DSOdb` directory in the installation directory.

For details of the data monitoring options, see “Configuration options for monitoring data” on page 22.

Procedure

1. Configure the options that control how IBM InfoSphere DataStage job runs are monitored.

Normally, all fatal messages, all control messages, and the first *N* warning messages of a run are captured, where *N* is specified by the **MaxWarnings** option. The capturing of these messages can be modified by the **CaptureLog** and the **IgnoreLog** options. The severity of a message is looked at after any message handlers have been invoked, which might have adjusted the severity of the message or discarded the message.

2. Configure the options that control how InfoSphere DataStage job run usage is monitored.
3. Configure the options that control how InfoSphere DataStage system resource usage is monitored.
4. Configure the options that define the default port numbers for sockets used to communicate between various processes.

Configuring the project monitoring data

Configure the monitoring data that is stored in the operations database for a specific project.

About this task

The options that are used to configure the collection of monitoring data for all jobs in all projects are set in the `DSOdbConfig.cfg` configuration file. This system-level configuration file is stored in the `/Server/DSOdb` directory in the installation directory.

Some of the options in the system-level configuration file can be overridden for a specific project by placing a project-level configuration file with the same name, `DSOdbConfig.cfg`, in the project directory.

Procedure

1. Create a project-level configuration file in the project directory for the project that you want to configure monitoring data for. Either copy the system-level configuration file, or create a file called `DSOdbConfig.cfg` in the project directory.
2. Configure the options that control the monitoring of IBM InfoSphere DataStage job runs for the project.

Configuring for an MPP or grid environment

If you use parallel jobs in an MPP or grid environment, you must configure communications to your remote nodes.

Procedure

1. Configure the Operations Console to use the remote shell (**rsh**) program rather than the secure shell (**ssh**) program.
 - a. Edit the RemoteShell.sh file in the /InformationServer/Server/PXEngine/DSResourceTracker/ directory.
 - b. Configure the Operations Console to use the **rsh** program instead of the **ssh** program.

Initially, the file has these contents:

```
#!/bin/sh
# Set to rsh / ssh as appropriate
ssh "$@"
```

Change the file to have these contents:

```
#!/bin/sh
# Set to rsh / ssh as appropriate
rsh "$@"
```

2. Configure the options that control the monitoring of system resource usage on your remote nodes. Set up the **ResourceNode** and **ResourceRemoteFS** options as required.
3. Specify the port number that is used by the ResTrackApp process on the head node to start and communicate with the ResTrackApp processes on your remote nodes. Set the **ResourceRemotePortNum** to the desired value.

Configuring tuning and troubleshooting options

Configure the options for tuning and troubleshooting the collection of monitoring data.

About this task

The options that are used to configure the tuning and troubleshooting of monitoring data are set in the DSODBTuning.cfg configuration file. This configuration file is stored in the /Server/DSODB directory in the installation directory.

In normal circumstances, there is no need to update this file. You should do so only under guidance from IBM Software Support.

Procedure

Configure the options for tuning and troubleshooting the monitoring of IBM InfoSphere DataStage job runs.

Reference: Configuration options

The reference topics provide details of the options you can use to configure the collection of monitoring data.

Configuration options for monitoring data:

You can configure the options that control the monitoring data that is stored in the operations database.

Configuration file

The data monitoring options for all jobs in all projects are set in the DSODBConfig.cfg configuration file. This configuration file is stored in the

/Server/DSODB directory in the installation directory.

Configuration options for job runs:

You can configure the options that control how IBM InfoSphere DataStage job runs are monitored.

Getting started

As a minimum, you must set DSODBON=1 if you want to capture monitoring data. The default values for the other options are sufficient to enable the capturing of monitoring information in most environments.

Table 3. Options that control the monitoring of InfoSphere DataStage job runs

Option	Values	Purpose
DSODBON	0 or 1	Controls if monitoring data is captured. Set DSODBON=0 if you do not want to capture monitoring data. The default value is 0. Set DSODBON=1 if you want to capture monitoring data. Before you set this option, you must create and configure the operations database. If the connection to the operations database fails, jobs still run but monitoring data is not collected.
MaxWarnings	integer	Specifies how many warning messages are captured for each job run. After this number of warning messages have been converted to events, any further warning messages are not captured, unless they are specified in the CaptureLog option. The default value is 10. The maximum value that you can specify is 1000; any value over 1000 is treated as 1000.

Table 3. Options that control the monitoring of InfoSphere DataStage job runs (continued)

Option	Values	Purpose
CaptureLog	comma-separated strings	<p>Specifies the message ID of a log message that must be captured in the operations database, regardless of their message type or whether the MaxWarnings limit for warning messages has been reached. A comma-separated list of message IDs can also be specified.</p> <p>The message ID can be followed by a <i>ContentType</i> string of up to 20 characters, separated from the ID by a slash (/). If present, these strings are used to tag the log events in the <i>ContentType</i> column in the JobRunLog table.</p> <p>This option can be repeated.</p> <p>For example: CaptureLog=IIS-DSTAGE-RUN-I-0126/ENV_VARS CaptureLog=IIS-DSTAGE-RUN-I-0470/OSH_SCRIPT, IIS-DSEE-TFSU-00016/OSH_SCORE_DUMP</p> <p>A message ID should be specified only once in the CaptureLog or the IgnoreLog options.</p> <p>Entries that are in the IgnoreLog option take precedence over entries in the CaptureLog option.</p> <p>Message IDs of the format DSTAGE_XXX_X_nnnn are stored as IIS-DSTAGE-XXX-X-nnnn in the operations database. You can specify message IDs in either format.</p>
IgnoreLog	comma-separated strings	<p>Specifies the message ID of a log message that should not be captured in the operations database, even if its type is fatal or warning. A comma-separated list of message IDs can also be specified.</p> <p>This option can be repeated.</p> <p>The message ID is specified in the same format as in the CaptureLog option. The strings that follow the slash (/) separator are purely descriptive in this case.</p> <p>Entries that are in the IgnoreLog option take precedence over entries in the CaptureLog option.</p> <p>The configuration file is initially set to ignore these fatal message IDs: IgnoreLog=IIS-DSTAGE-RUN-I-0180/Attempting to Cleanup after ABORT IgnoreLog=IIS-DSEE-TFSR-00019/Could not check all operators because of previous error(s) IgnoreLog=IIS-DSEE-TCOS-00029/Creation of a step finished with status = FAILED</p>

Table 3. Options that control the monitoring of InfoSphere DataStage job runs (continued)

Option	Values	Purpose
UpdateIntSecs	integer	<p>Specifies the minimum time interval in seconds that elapses between successive events that update the overall job run statistics, such as total rows read and written.</p> <p>The default value is 10, which means that such events are written no more frequently than every 10 seconds.</p>
MonitorLinks	0 or 1	<p>Controls if stage-level and link-level statistics, and references to data locators, are captured at the end of each job run.</p> <p>Set MonitorLinks=0 if you do not want to capture this data is not to be captured. The default value is 0.</p> <p>Set MonitorLinks=1 if you want to capture this data.</p> <p>Stage-level and link-level monitoring data is not used by the IBM InfoSphere DataStage and QualityStage Operations Console, so enable this option only if you want to extract this data using SQL statements.</p>
CustomBatchId CustomJobType CustomContact CustomSequence CustomField1 CustomField2	string	<p>Specifies the names of job parameters that you want to use to tag entries in the JobRun table.</p> <p>The values of these options, if set, are compared to the names of the parameters of each job at run time. If the job has a parameter with the given name, the parameter value is saved in the row of the JobRun table associated with the run, in the column with the same name as the option.</p> <p>Use these options to group runs for any desired purpose. The names of the options, and the corresponding columns, have no special significance, other than to suggest common reasons for this tagging.</p> <p>By default these options are empty.</p>
EventsDir	directory path	<p>Specifies the directory where job run event files are created. The directory must exist, it must have write access for users who can run jobs, and it must be specified as an absolute path. Use forward slashes (/) as separators on all platforms, including on Windows. Additionally, on Windows, a drive letter is required.</p> <p>If omitted, the default is the /InformationServer/Server/DSODB/events directory.</p> <p>For example, on Windows: EventsDir=C:/IBM/InformationServer/Server/DSODB/events</p> <p>For example, on Unix and Linux: EventsDir=/opt/InformationServer/Server/DSODB/events</p>

Table 3. Options that control the monitoring of InfoSphere DataStage job runs (continued)

Option	Values	Purpose
TraceMax	integer	<p>Controls if you want to monitor the tracing of job run events, and specifies how many lines are written to the trace file.</p> <p>If TraceMax is set to a number greater than 0, tracing is enabled.</p> <p>Set TraceMax to a number between 1000 and 100000, to limit the maximum number of lines written to the trace file. Any number outside that range enables event tracing monitoring.</p> <p>A file named JobRuntime.log is written to the /Server/DSODB/logs directory in the installation directory.</p> <p>By default TraceMax is set to 0, which means that monitoring the tracing of job run events is disabled.</p>

Configuration options for job run usage:

You can configure the options that control how InfoSphere DataStage job run usage is monitored.

Table 4. Options that control the monitoring of InfoSphere DataStage job run usage

Option	Values	Purpose
JobRunCheckInterval	integer	<p>Specifies, in minutes, how often automatic validation of currently running jobs is performed.</p> <p>At each check interval, every job with a status of running in the operations database is checked to see if it is still running, if the relevant processes still exist. If the job is not still running, the job is marked as finished in the operations database, with a special status denoting that the job was detected as being defunct.</p> <p>The default value is JobRunCheckInterval=60 minutes, with the minimum value being JobRunCheckInterval=10 minutes.</p> <p>If you set JobRunCheckInterval=0 the checking no longer occurs.</p>
JobRunUsage	0 or 1	<p>Controls if job run usage data is collected.</p> <p>Set JobRunUsage=0 if you do not want to capture this data.</p> <p>Set JobRunUsage=1 if you want to capture this data. The default value is 1.</p>

Table 4. Options that control the monitoring of InfoSphere DataStage job run usage (continued)

Option	Values	Purpose
JobRunAggSnaps	integer	<p>Controls the number of snapshot values that are included in a single row of the JobRunUsage table before a new one is started. It is only relevant if JobRunUsage=1.</p> <p>A snapshot value is produced whenever a job reports a change in the number of rows it has processed, provided the number of seconds since the previous report is greater than or equal to the UpdateIntSecs option value. The snapshot is used to update the most recent row of the JobRunUsage table, unless the number of modifications exceeds the JobRunAggSnaps value, in which case a new row is inserted.</p> <p>This option strikes a balance between the number of rows in the table, and therefore how frequently it is written to, and how far the table lags behind the current time.</p> <p>The default is 15 values per row.</p>

Configuration options for system resource usage:

You can configure the options that control how InfoSphere DataStage system resource usage is monitored.

Table 5. Options that control the monitoring of InfoSphere DataStage system resource usage

Option	Values	Purpose
ResourceMonitor	0 or 1	<p>Controls if system resource usage data is collected.</p> <p>Set ResourceMonitor=0 you do not want to capture this data.</p> <p>Set ResourceMonitor=1 you want to capture this data. The default value. is 1</p>
ResourceNode	string	<p>Specifies the name of a remote node whose resources are to be monitored from the current engine system. The local system is always monitored and need not be included here.</p> <p>The name given for each node must match that used in parallel job configuration files.</p> <p>This option can be repeated if several remote nodes are to be monitored.</p>

Table 5. Options that control the monitoring of InfoSphere DataStage system resource usage (continued)

Option	Values	Purpose
ResourceLocalFS	directory path	<p>Specifies a file system path whose disk mount point is to be monitored for used and free space.</p> <p>This option can be repeated to specify multiple disks. You must ensure that paths are not mounted on the same disks, otherwise duplicate information is collected.</p> <p>For example: ResourceLocalFS=/scratch</p>
ResourceRemoteFS	node + directory path	<p>Specifies a file system path on a remote node whose disk mount point is to be monitored for used and free space.</p> <p>Specify the node name and file system path separated by a plus sign (+).</p> <p>The remote node name must match one specified via a corresponding ResourceNode option.</p> <p>This option can be repeated to specify multiple remote disks.</p> <p>For example: ResourceRemoteFS=nodeA+/scratch</p>
ResourcePollPeriod	integer	<p>Specifies, in seconds, how frequently a system resource snapshot is taken.</p> <p>At each poll interval the ResourceSnap table is updated with data for the local node, and for all the remote nodes named in the ResourceNode options that can be contacted.</p> <p>The default value is 10 seconds.</p>

Table 5. Options that control the monitoring of InfoSphere DataStage system resource usage (continued)

Option	Values	Purpose
ResourceSampleSize	integer	<p>Specifies how many system resource snapshots are taken before an aggregated record of those values is stored.</p> <p>A new row for each monitored node is inserted into the ResourceUsage table when this many snapshots have been collected. This pattern is the basis for historical system resource usage information, so at any time the operations database might lag behind real time by up to ResourcePollPeriod multiplied by ResourceSampleSize seconds.</p> <p>The default value is 6. Therefore, using the default values for ResourcePollPeriod and ResourceSampleSize, an aggregated record is inserted every 10*6 seconds, which is every 1 minute.</p>
ResourceAllAggregatedUsage	0 or 1	<p>Specifies whether to always store aggregated resource usage data as described under ResourceSampleSize.</p> <p>If ResourceAllAggregatedUsage=1 then aggregated data is always stored.</p> <p>If ResourceAllAggregatedUsage=0, then aggregated data is only stored while there is any DataStage job activity.</p> <p>If you set ResourceAllAggregatedUsage=0 you avoid storing large numbers of records when there are no jobs running.</p> <p>The default value is 1, which stores aggregated resource usage data continuously.</p>

Table 5. Options that control the monitoring of InfoSphere DataStage system resource usage (continued)

Option	Values	Purpose
ResourceAggRunPollPeriod	integer	<p>Specifies how many aggregated snapshots are stored before and after any job activity has been detected. This option is only relevant if ResourceAllAggregatedUsage=0.</p> <p>This number of aggregated values is stored before a check for any job activity occurs again. So the time period covered is this value multiplied by the aggregated snapshot time.</p> <p>The default value is 10. Using the default values, the minimum time stored around any job activity is 10*(10*6) seconds, which is 10 minutes.</p> <p>The value for ResourceAggNonRunPollPeriod must be less than the value for ResourceAggRunPollPeriod.</p>
ResourceAggNonRunPollPeriod	integer	<p>Specifies how often a check is made for job activity if there has been no recent job activity. This option is only relevant if ResourceAllAggregatedUsage=0.</p> <p>Snapshot data is always stored according to the ResourceAggRunPollPeriod option, but this option specifies how often the check for job activity is made.</p> <p>The time period is this value multiplied by the aggregated snapshot time.</p> <p>The default value is 1. Using the default values, the check for job activity is made every 1*(10*6) seconds, that is every 1 minute.</p> <p>The value for ResourceAggNonRunPollPeriod must be less than the value for ResourceAggRunPollPeriod.</p>

Configuration options for default port numbers:

You can configure the options that define the default port numbers for sockets used to communicate between various processes.

Table 6. Options that define the default port numbers for sockets used to communicate between various processes

Option	Values	Purpose
ResourcePortNum	integer	Specifies the port number that the resource tracking process, ResTrackApp, uses to communicate on the local system. The default is 13450.
ResourceRemotePortNum	integer	Specifies the port number that the resource tracking process, ResTrackApp, uses to communicate between the engine system and remote nodes. The default is 13450.
ODBQAPPNUM	integer	Specifies the port number that the ODBQueryApp process uses to handle requests from the Operations Console. The default is 13451.

Configuration options for project monitoring data:

You can configure the options that control the project monitoring data that is stored in the operations database.

Configuration file

The options that are used to configure the collection of monitoring data for all jobs in all projects are set in the DSODBConfig.cfg configuration file. This system-level configuration file is stored in the /Server/DSODB directory in the installation directory.

Some of the options in the system-level configuration file can be overridden for a specific project by placing a project-level configuration file with the same name, DSODBConfig.cfg, in the project directory.

The options shown in Table 7 can be overridden by using a project-level configuration file. Do not specify any other options in the project-level configuration file as they are ignored.

Table 7. Options that control the monitoring of InfoSphere DataStage job runs for a specific project

Option	Project-level override
DSODBON	If you set DSODBON=1 in the system-level configuration file to enable the capturing of monitoring data, set DSODBON=0 if you do not want to capture monitoring data for this project. If you set DSODBON=0 in the system-level configuration file to disable the capturing of monitoring data, monitoring data is not captured for this projects even if you set DSODBON=1.
MaxWarnings	The setting in the project-level file takes precedence.

Table 7. Options that control the monitoring of InfoSphere DataStage job runs for a specific project (continued)

Option	Project-level override
CaptureLog	The setting in the project-level file takes precedence. If an ID is not present in the project-level file but is in the system-level file, the system-level setting applies.
IgnoreLog	The setting in the project-level file takes precedence. If an ID is not present in the project-level file but is in the system-level file, the system-level setting applies.
UpdateIntSecs	The setting in the project-level file takes precedence.
MonitorLinks	The setting in the project-level file takes precedence.
CustomBatchId	The setting in the project-level file takes precedence.
CustomJobType	If an option is not present in the project-level file but is in the system-level file, the system-level setting applies.
CustomContact	
CustomSequence	
CustomField1	
CustomField2	

Configuration options for tuning and troubleshooting:

You can configure the options for tuning and troubleshooting the collection of monitoring data.

Configuration file

The options that are used to configure the tuning and troubleshooting of monitoring data are set in the DSODBTuning.cfg configuration file. This configuration file is stored in the /Server/DSODB directory in the installation directory.

In normal circumstances, there is no need to update this file. You should do so only under guidance from IBM Software Support.

Table 8. Options for tuning and troubleshooting the monitoring of InfoSphere DataStage job runs

Option	Values	Purpose
EMAAllowSQLErrors	0 or 1	Controls if the EngMonApp process stops if it encounters SQL errors. Set EMAAllowSQLErrors=0 if you want the EngMonApp process to stop if it encounters SQL errors. EMAAllowSQLErrors=0 is the default value. Set EMAAllowSQLErrors=1 if you want the EngMonApp process to not stop if it encounters SQL errors.

Table 8. Options for tuning and troubleshooting the monitoring of InfoSphere DataStage job runs (continued)

Option	Values	Purpose
OpsConsole.MaxInterval	integer	<p>Specifies the maximum time interval, in days, that you can select in the Operations Console. If OpsConsole.MaxInterval is set too high, queries issued by the Operations Console might take a long time to process, or even time out, depending on the number of job runs recorded daily in the operations database.</p> <p>For example, it is suggested that if you process 10,000-15,000 job runs per day, this limit should be set no higher than 10 days. For smaller workloads the limit can be proportionally higher. The default is 90 days, equating to approximately 1000-1500 runs per day.</p>
<i>serviceName</i>	string	<p>Set <i>serviceName</i>=NO to disable the checking of the status of the service by the Operations Console.</p> <p>You can disable the checking of the status of these services:</p> <ul style="list-style-type: none"> • appwatcher • jobmonapp • dsrpcd • odbqueryapp • engmonapp • resmonapp <p>For example, set jobmonapp=NO.</p> <p>Remove this option to re-enable the checking of the status of the service by the Operations Console.</p>

Managing the data collection processes

In normal circumstances, you use the AppWatcher process to ensure that the data collection processes are always running.

About this task

If you have enabled the capturing of monitoring data, the AppWatcher process is automatically started when the engine system is started.

You can manually stop and start the AppWatcher process and the data collection processes.

Data collection processes

The data collection processes populate the operations database.

You can control what data is collected by the data collection processes by using the configuration files. For more information, see “Configuring the monitoring data” on page 21.

EngMonApp process

If monitoring is enabled on an IBM InfoSphere Information Server engine, the runtime system writes selected status and logging events to files in the Events directory. These events form a record of what job runs started on that engine, and what their status was.

The EngMonApp server process transfers the contents of these event files to the operations database, deleting the files after their contents have been committed to the database.

If the EngMonApp process is not running, jobs still continue to run and event files are still written to the Events directory.

If the EngMonApp process is running, but the database is not available for some reason, the events files are retained and periodic attempts are made to transfer them.

ResMonApp process

The ResMonApp process monitors system resource usage on a periodic basis and records the results in the operations database.

The ResMonApp process uses a local ResTrackApp process to collect information about its local system. On the head node of a grid or MPP system, the local ResTrackApp process manages a ResTrackApp process on each remote system to collect information about that system.

If the ResMonApp process is not running, or the database is not available for some reason, the system resource usage information is not recorded for that time period.

ResTrackApp process

The ResTrackApp process queries its local operating system for system resources usage information which it passes to the ResMonApp process that is managing it.

ODBQueryApp

The ODBQueryApp process accesses information from the operations database, in response to requests from the Operations Console. The ASBAgent process acts as the intermediary between the Operations Console and the ODBQueryApp process.

The ODBQueryApp process accesses information about both job run events and system resource usage.

The ODBQueryApp process also supports the maintenance of the operations database that is done using the `istool ODBAdmin` command.

Starting and stopping the AppWatcher process

The AppWatcher process manages the other data collection processes, restarting them if they are not running.

About this task

If the capturing of monitoring data is enabled, the AppWatcher process is automatically started when the engine system is started. If you implemented a method of starting the AppWatcher process for a release of InfoSphere Information Server before Version 11.3, it is no longer needed.

The AppWatcher process creates log files in the `/InformationServer/Server/DSODB/logs` directory. All AppWatcher start and stop requests are recorded in the `AppWatcher_Cmd.log` file.

When AppWatcher is started a log file called `AppWatcher_Svc-YYYYMMDDHHMMSS.log` is created, where `YYYYMMDDHHMMSS` is the date and time when the AppWatcher instance was started. This file is used to log any issues for that instance of AppWatcher.

Starting the AppWatcher process

You can use these commands to start the AppWatcher process.

Procedure

Start the AppWatcher process.

Option	Description
UNIX and Linux	<p>Issue this command from the <code>/InformationServer/Server/DSODB/bin</code> directory:</p> <pre>DSAppWatcher.sh -start</pre> <p>This command must be run as <code>dsadm</code> or an equivalent user. The AppWatcher process is run in the background so the command returns as soon as it has finished starting the other processes. The command does not wait for the other processes to enter the <code>RUNNING</code> state.</p> <p>Add this command to the same place as the <code>uvadmin -start</code> command in the InfoSphere Information Server engine startup scripts.</p>

Option	Description
Windows	<p>Issue these commands to start the Windows service:</p> <pre>net start "DataStage AppWatcher Service"</pre> <p>If the command fails, the following service specific return codes might be shown in the failure message:</p> <ul style="list-style-type: none"> 1 The AppWatcher service was already running. 3 The AppWatcher service was not started because the collection of monitoring data is not enabled in the configuration file. 21 The request to start the AppWatcher process timed out <p>You can set the AppWatcher service to start when the system is rebooted by setting its Startup Type to Automatic on the Windows Services dialog.</p>

Stopping the AppWatcher process

You can use these commands to stop the AppWatcher process.

Procedure

Stop the AppWatcher process.

Option	Description
UNIX and Linux	<p>Issue this command from the /InformationServer/Server/DSODB/bin directory:</p> <pre>DSAppWatcher.sh -stop</pre> <p>The command stops the AppWatcher process and all the data collection processes that it manages.</p>
Windows	<p>Issue these commands to stop the Windows service:</p> <pre>net stop "DataStage AppWatcher Service"</pre> <p>The command stops the AppWatcher process and all the data collection processes that it manages.</p>

Managing the data collection processes using the DSAppWatcher.sh script

Use the **DSAppWatcher.sh** script to manage the data collection processes.

About this task

The **DSAppWatcher.sh** script takes one of these parameters:

- -start [*AppName* | all]
- -stop [*AppName*]
- -status [*AppName*]
- -test

where *AppName* is either EngMonApp, ResMonApp, or ODBQueryApp. The *AppName* parameter is not case-sensitive.

Starting the AppWatcher process and all its managed processes

On Unix and Linux, you can use the **DSAppWatcher.sh** script to start the AppWatcher process and all its managed processes.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, start the AppWatcher process as described in “Starting and stopping the AppWatcher process” on page 34.

Procedure

Issue this command to start the AppWatcher process and all its managed processes.

```
DSAppWatcher.sh -start
```

The command shows the list of processes it attempts to start. For example:

```
AppWatcher:STARTED
EngMonApp:STARTING
ODBQueryApp:STARTING
ResMonApp:STARTING
```

The command returns one of these codes:

- 0 The AppWatcher process started, but the data collection processes might not have started, or might not still be running.
- >0 An error occurred:
 - 1 – the AppWatcher process was already running
 - 3 – the AppWatcher process was not started because monitoring is turned off

Stopping the AppWatcher process and all its managed processes

On Unix and Linux, you can use the **DSAppWatcher.sh** script to stop the AppWatcher process and all its managed processes.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, stop the AppWatcher process as described in “Starting and stopping the AppWatcher process” on page 34.

Procedure

Issue this command to stop the AppWatcher process and all its managed processes.

```
DSAppWatcher.sh -stop
```

The command shows the processes it is attempting to stop, and returns when the stop requests are complete. It stops the processes in the reverse order to the one in which they were started. For example:

```
AppWatcher:STOPPED
ResMonApp:STOPPING
ResMonApp:STOPPED
ODBQueryApp:STOPPING
ODBQueryApp:STOPPED
EngMonApp:STOPPING
EngMonApp:STOPPED
```

The command returns one of these codes:

- 0 The AppWatcher process stopped, or was already stopped. The AppWatcher process attempted to stop the processes, but they might not have stopped yet.
- >0 An error occurred:
 - 0 – the AppWatcher process stopped, or was already stopped. The AppWatcher process attempted to stop the processes, but they might not have stopped.
 - >0 – An error occurred

Displaying the status of the AppWatcher process and all its managed processes

You can use the **DSAppWatcher.sh** script to display the status of the AppWatcher process and all its managed processes.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, use MKS to run **DSAppWatcher.sh** from the `\InformationServer\Server\DSODB\bin` directory.

Procedure

Issue this command to display the status of the AppWatcher process and all its managed processes.

```
DSAppWatcher.sh -status
```

For example:

```
AppWatcher:RUNNING
EngMonApp:RUNNING
ODBQueryApp:STOPPED
ResMonApp:RUNNING
```

Displaying the status of a specified process

You can use the **DSAppWatcher.sh** script to display the status of a specified process.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, use MKS to run **DSAppWatcher.sh** from the `\InformationServer\Server\DSODB\bin` directory.

Procedure

Issue this command to display the status of the specified process.

```
DSAppWatcher.sh -status AppName
```

where *AppName* is either `AppWatcher`, `EngMonApp`, `ResMonApp`, or `ODBQueryApp`. The *AppName* parameter is not case-sensitive. For example:

```
EngMonApp:RUNNING
```

The command returns one of these codes:

- 1 The process is running
- 2 The process is not running
- 3 Monitoring is not enabled

Stopping a specified process

You can use the **DSAppWatcher.sh** script to stop a specified process.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, use MKS to run **DSAppWatcher.sh** from the `\InformationServer\Server\DSODB\bin` directory.

Procedure

Issue this command to stop the specified process, which is not restarted by the `AppWatcher` process until requested to do so.

```
DSAppWatcher.sh -stop AppName
```

where *AppName* is either `EngMonApp`, `ResMonApp`, or `ODBQueryApp`. The *AppName* parameter is not case-sensitive. For example:

```
EngMonApp:STOPPING
```

The command returns one of these codes:

- 0 The process has been stopped
- >0 An error occurred

Starting a specified process

You can use the **DSAppWatcher.sh** script to start a specified process.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, use MKS to run **DSAppWatcher.sh** from the `\InformationServer\Server\DSODB\bin` directory.

Procedure

Issue this command to restart the specified process, which is then monitored as normal by the AppWatcher process.

```
DSAppWatcher.sh -start AppName
```

where *AppName* is either `EngMonApp`, `ResMonApp`, or `ODBQueryApp`. The *AppName* parameter is not case-sensitive. For example:

```
EngMonApp:STARTING
```

The command returns one of these codes:

- 0** The process is already running, or the AppWatcher process attempted to start the process. The process might not still be running
- >0** An error occurred

Starting all the processes that are not running

If the AppWatcher process is running, you can use the **DSAppWatcher.sh** script to start all the processes that are not running.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, use MKS to run **DSAppWatcher.sh** from the `\InformationServer\Server\DSODB\bin` directory.

Procedure

Issue this command to restart all the processes that are not currently running, and then monitor them as normal.

```
DSAppWatcher.sh -start all
```

For example:

```
EngMonApp:ALREADY_STARTED  
ODBQueryApp:STARTING  
ResMonApp:STARTING
```

Checking the configuration of the monitoring system

You can use the **DSAppWatcher.sh** script to check the configuration of the monitoring system.

About this task

On Unix and Linux, run **DSAppWatcher.sh** from the `/InformationServer/Server/DSODB/bin` directory.

On Windows, use MKS to run **DSAppWatcher.sh** from the `\InformationServer\Server\DSODB\bin` directory.

Procedure

Issue this command to check the configuration of the monitoring system.

```
DSAppWatcher.sh -test
```

The command shows whether each type of monitoring is enabled, and then checks if the configuration settings can be used to connect to the operations database. For example, if configuration settings can connect to the operations database:

```
DSODB is ON in the DSODBCfg.config file.  
Link Monitoring is ON.  
Job Run Usage is ON.  
Resource Monitoring is ON.  
Checking Database Connection:  
Successfully loaded the database driver.  
Successfully connected to the database.  
DB Schema version number: 1  
Test Successful.
```

For example, if the username or password settings for the operations database are incorrectly specified:

```
DSODB is ON in the DSODBCfg.config file.  
Link Monitoring is ON.  
Job Run Usage is ON.  
Resource Monitoring is ON.  
Checking Database Connection:  
Successfully loaded the database driver.  
Test Failed: Invalid username or password in the DSODBConnect.config file.
```

The command returns one of these codes:

- 0 - everything OK
- >0 – a test failed. Some failures that have specific codes:
 - 1 – Unexpected error
 - 2 – DSODB is off
 - 3 – JDBC driver failure
 - 4 – Connection failure
 - 5 – Database logon failure
 - 6 – DBTYPE setting in DSODBConnect.config invalid
 - 8 – DBSchema mismatch
 - 9 - Database tables incorrectly set up

Handling data collection process errors

If you receive errors from the data collection processes, you must troubleshoot and correct those errors.

About this task

The EngMonApp and ResMonApp processes write error files and log files in the `/Server/DSODB/logs` directory in the installation directory.

Handling EngMonApp fatal errors

If you receive errors from the EngMonApp process, you must troubleshoot and correct those errors.

About this task

If the EngMonApp process encounters a fatal error that prevents it from processing event files, it creates an error file called EngMonApp.err in the /Server/DSODB/logs directory in the installation directory, and exits. Such errors include an invalid username or password, or other configuration file errors, that require user intervention before the EngMonApp process can be restarted.

If the EngMonApp process encounters a non-fatal error, the error and event file content is logged, the event file is deleted, and the EngMonApp process continues processing. Such errors include data inconsistencies between the event file and the operations database, such as the job run record not being found in the JobRun table.

If the EngMonApp process receives an SQL error from the operations database, the error and event file content is logged, the event file is deleted, and the EngMonApp process stops. Because the event file is deleted, the EngMonApp process continues with the next event file when it is restarted. The *EMAAllowSQLErrors* option in the DSODBTuning.cfg configuration file can be set to cause the EngMonApp process to continue processing if it receives an SQL error.

To check if the EngMonApp process is running, use this command:

```
DSAppWatcher.sh -status EngMonApp
```

If the AppWatcher process is running, it does not restart EngMonApp process while the EngMonApp.err file exists. If the AppWatcher process is restarted, it deletes the EngMonApp.err file and restarts the EngMonApp process.

Procedure

1. If the AppWatcher process is not running, the EngMonApp.err file might exist stating that the EngMonApp process was stopped because the AppWatcher process was stopped.
2. If the AppWatcher process is running but the EngMonApp process is not running, fix the problem that the EngMonApp.err file describes. You might also need to check the latest EngMonApp log file that contains a fuller description of the error condition. The log file is called EngMonApp_YYYYMMDDHHMMSS.log, where YYYYMMDDHHMMSS is the date and time when the EngMonApp process was started.
3. Restart the EngMonApp process by issuing this command:

```
DSAppWatcher.sh -start EngMonApp
```

If the error condition has not been resolved, the EngMonApp process creates a new EngMonApp.err file and exits.

Handling ResMonApp fatal errors

If you receive errors from the ResMonApp process, you must troubleshoot and correct those errors.

About this task

If the ResMonApp process encounters a fatal error that prevents it from processing system resource information, it creates an error file called ResMonApp.err in the /Server/DSODB/logs directory in the installation directory, and exits. Such errors include an invalid username or password, or other configuration file errors, that require user intervention before ResMonApp can be restarted.

To check if the ResMonApp process is running, use this command:

```
DSAppWatcher.sh -status ResMonApp
```

If the AppWatcher process is not running, it does not restart the ResMonApp process while the ResMonApp.err file exists. If the AppWatcher process is restarted, it deletes the ResMonApp.err file and restarts the ResMonApp process.

Procedure

1. If the AppWatcher process is not running, the ResMonApp.err file might exist stating that the ResMonApp process was stopped because the AppWatcher process was stopped.
2. If the AppWatcher process is running but the ResMonApp process is not running, fix the problem that the ResMonApp.err file describes. You might also need to check the latest ResMonApp log file that contains a fuller description of the error condition. The log file is called ResMonApp_YYYYMMDDHHMMSS.log, where YYYYMMDDHHMMSS is the date and time when the ResMonApp process was started.
3. Restart the ResMonApp process by issuing this command:

```
DSAppWatcher.sh -start ResMonApp
```

If the error condition has not been resolved, the ResMonApp process creates a new ResMonApp.err file and exits.

Job runtime errors

The job runtime system that generates event files can encounter errors.

If the job runtime system that generates events encounters an error, it adds the reason to a file named JobRuntime.err in the /InformationServer/Server/DSODB/logs directory. Additionally, the message is added to a file named DSODB_JobRuntime.err in the project directory in which the job is running, provided the user running the job has write access to that directory. To keep the size of these files under control, only the last 100 errors are recorded.

This level of error does not prevent a job from running, but it might lead to incomplete or inconsistent monitoring information being stored in the operations database. In particular, a job run might appear to still be running in the operations database, even though the run has ended. This situation is resolved by EngMonApp, which sets the status of the job run to Finished/Synchronized. For more information, see the **JobRunCheckInterval** option in the DS0BDConfig.cfg configuration file.

Monitoring jobs and job runs by using the Operations Console

Use the IBM InfoSphere DataStage and QualityStage Operations Console to monitor your jobs, job activity, system resources, and workload management queues on your IBM InfoSphere Information Server engines.

Overview of the Operations Console

You can use the Operations Console to access information about your jobs, job activity, system resources, and workload management queues for each of your IBM InfoSphere Information Server engines.

The Operations Console provides engine-wide information about job runs, system resources, workload management queues, and engine status. You can obtain all of

this information yourself by viewing the individual engines and job runs on every system that you own. However, piecing that information together to get an overall view of your environment can be difficult.

The IBM InfoSphere Information Server engine status information and job run information that is displayed in the console is supplied by the operations database. The operations database is populated with information from all of the engines that you have connected to your database. You can view only one IBM InfoSphere Information Server engine at a time in the console, but you can easily switch between the multiple engines that are connected to your operations database and associated with the current services tier machine.

The workload management queue information that is displayed in the console is supplied by the workload management process, called WLMServer. You can view only one IBM InfoSphere Information Server engine at a time in the console, but you can easily switch between the multiple engines that are associated with the current services tier machine.

Each tab in the Operations Console has several panels that display various types of information about the active engine that you have selected. You can use the information in these panels to identify service errors, failed jobs, and performance issues.

Scenarios for using the Operations Console

The Operations Console can help you to monitor your engines in multiple business scenarios.

The Operations Console provides you with all of your job run, system resource, and IBM InfoSphere Information Server engine status information in one location so you can quickly identify the overall status of your environment and be alerted when problems occur.

The Operations Console provides you with the same overall view of your environment from a specific period in history. This historical view allows you to identify conflicts and performance issues that occurred while you were not actively monitoring your environment.

The Operations Console also provides you with information about the current status of your workload management queues.

All of this information helps you to better balance your system workload, troubleshoot engine problems, and schedule job runs more efficiently.

These scenarios show how the Operations Console provides a better view of your InfoSphere Information Server environments.

Troubleshooting failed job runs

Jobs can fail for various reasons. Identifying those reasons without an engine-wide view of your environment can be difficult. The Operations Console alerts you when job runs have failed so you can correct those issues as soon as they occur. When a job run fails, you can quickly view a list of all of the InfoSphere Information Server engine activity that occurred before, during, and after the failure.

Perhaps you identify that your failed job is dependent on other jobs that have not completed due to abnormally long run times. You can then view all of the other job runs and system resource usage information that was gathered when your jobs encountered these problems. With that information, you can resolve the problem so it does not happen then next time that your job run is scheduled to run.

Improving job run performance

Determining the cause of poor job performance is difficult without a view of your entire system. The Operations Console provides you with system resource information as well as information about your job run schedule and job designs. This information in the console can help you to identify when system resources experience heavy workload, or when job runs are using up more than the expected amount of system resources.

The Operations Console also allows you to easily compare job runs to previous job runs so that you can identify the changes that contribute to poor performance or job failures.

Actively monitoring your engines

When a problem occurs in your InfoSphere Information Server environment, you need to know about it immediately so that you can resolve the problem. The dashboard on the **Dashboard tab** notifies you of engine service outages, failed jobs, and low system resource availability. When you want to actively monitor the status of your environment, you can customize the panels in the **Dashboard tab** so that they refresh only as often as you need. You can set the panels to alert you when a specific level of system resources are being used. With all of this information in one location, you can quickly identify and correct problems with your environment.

Your business needs might also require you to monitor multiple systems and multiple InfoSphere Information Server engines within your company. Monitoring the system, job run, and engine service information in multiple locations can be tedious. The Operations Console allows you to track multiple InfoSphere Information Server engines that are registered to an InfoSphere Information Server system with a single interface. Using a single interface to monitor multiple engines can simplify monitoring, troubleshooting, and performance tasks for large InfoSphere DataStage and QualityStage environments. Tracking information between interdependent systems and engines is also easier with a single interface.

To track InfoSphere Information Server engines that are registered to more than one InfoSphere Information Server system, you must use an instance of the Operations Console for each InfoSphere Information Server system.

User access to the Operations Console

Users are able to access the Operations Console and view specific sets of information in the console only when they have the appropriate user role for those tasks. You can manage user roles with the IBM InfoSphere Information Server Web console or the IBM InfoSphere DataStage and QualityStage Administrator client.

To view information in the console, a user must have appropriate access to at least one project or IBM InfoSphere Information Server engine that is on the current services tier machine. InfoSphere DataStage users who have the DataStage and

QualityStage Operations Viewer role, or a superior role, for a project that is included in the operations database can connect to the console.

Users can see information only for the projects for which they have been granted appropriate access. Information about job run workload, engine services, and system resources is available to any user who is authorized to log in to the console. However, details about specific job runs and projects are available only to users who have sufficient access to those individual projects. Users who have the DataStage and QualityStage Administrator role can view the information for all projects and job run details in the console.

Users who have the DataStage and QualityStage Operations Viewer role can only view information in the console. Users with superior roles can also start and stop job runs in projects where they have sufficient access.

Users with the DataStage and QualityStage Administrator role can also define and administer the workload management queues.

Operations Console system requirements

Before you start the Operations Console, ensure that your system meets the minimum requirements.

The minimum screen resolution supported is 1024 x 768. For system requirements see, System requirements.

Starting the Operations Console

Start the IBM InfoSphere DataStage and QualityStage Operations Console.

Procedure

1. To start the Operations Console using HTTPS, enter this URL from a supported browser:

```
https://domain:port/ibm/iis/ds/console/login.html
```

where

- *domain* is the name or IP address of your InfoSphere Information Server system
 - *port* is the port number your InfoSphere Information Server system is configured to use. The default port number for HTTPS is 9443
2. Click **Login** and supply your login credentials. If a message is displayed about a security certificate from the server not being trusted, follow the prompts to accept the certificate. If a tree structure of certificates is displayed, select the root certificate.
 3. If you are using Microsoft Internet Explorer and you get a message that a security certificate from the server is not trusted, accept the certificate:
 - a. Click **Continue to this website (not recommended)**.
 - b. Click **Certification error**.
 - c. Select **View certificates** to view the security certificate.
 - d. Click the **Certification Path** tab, and then select the root certificate.
 - e. Click the **General** tab.
 - f. Click **Install Certificate**, and then click **Next**.
 - g. Select **Place all certificates in the following store**.
 - h. Click **Browse**, and then select **Trusted Root Certification Authorities**.

- i. Click **Next**, and then click **Finish** to import the certificate.
4. If you are using Mozilla Firefox and a message is displayed about a connection from the server not being trusted, follow these steps to confirm the security exception:
 - a. Click **I understand the risks**.
 - b. Click **Add exception**.
 - c. Click **Get certificate**.
 - d. Select **Permanently store this exception**, and click **Confirm Security Exception**.

Viewing log messages in the Operations Console

You can use the Operations Console to view messages from job runs that you have appropriate authorization to view.

Before you begin

Not all of the log messages for a job run will display when you view the message list in the Operations Console. You must configure the operations database to capture only the messages that you want to view. By default, the database captures only job start and stop messages, and no more than 10 messages that are warning messages or fatal messages.

About this task

Messages for a job run are displayed in the Run Details window, which you can access from one of the lists of job runs in the Operations Console. Lists of job runs are available in the following panels:

- The Job Activity panel on the Home tab
- The Job Runs tab within the main Activity tab
- The Job Runs panel that displays when you select a job on the Projects tab

Procedure

1. Select a job run from a job run list by clicking the open space to the left of the job name and job icon.
2. Click the **View Details** button. The **View Details** button is either above the job runs list, or on the same row as the selected job run depending on which job run list you view.
3. In the Run Details window, click the **Log Messages** tab.
4. Optional: Select the **Full Message** check box to see messages from the server rather than from the operations database.
5. To display the details for a message in the list, select a message in the list.

Configuring the display of data in the console

You can configure panels in the Operations Console to fit the needs of your IBM InfoSphere Information Server environment.

About this task

The panels in the Operations Console display information about job runs, engine status, and the use of system resources for the active IBM InfoSphere Information Server engine. You can narrow the scope of information in these panels, and set the refresh rate to update the information only as frequently as you need it. You

can also configure some panels to show information from a certain time period in engine event history.

Selecting the active engines and projects

You can set the scope of the information that is displayed in the Operations Console by selecting a specific IBM InfoSphere Information Server engine that is associated with the current services tier and projects that you want to view.

About this task

The engine that you select and the project scope that you define for that engine affects what information and controls are displayed in all of the panels of the Operations Console. You can use the **Engine Selection** menu and the **View Projects** button at the top of the Operations Console to control the scope of the information that is displayed in the console.

Procedure

1. Click the active engine in the **Engine Selection** menu.
2. Select the engine from the list that you want to view. The console refreshes with data from the engine that you selected. The scope on the **View Projects** button is set to show all projects in the active engine that the user has authorization to view.
3. Click the **View Projects** button.
4. Click the individual projects to select or clear them from the list of projects that you want to view, and click **OK**. The console refreshes and displays data only from the projects that you have selected.

Setting the scope of the event history in a panel

You can narrow or expand the length of the event history that is shown in each pane so that you see only the information that is relevant to you.

About this task

Several panels in the Operations Console can be set to show information and events from a specific length of system history. If a panel can be configured to show event history from a specific length of time, the current event history length is already displayed at the top of that panel.

Procedure

To set the scope of the event history in a panel:

1. Click the link that shows the length of event history that is currently being displayed in the panel.
2. In the event history selection window, select the unit of time measurement that you want to use to set the event history and enter an appropriate value for the length of history that you require.
3. Click the **close button** in the upper right hand corner of the window to close the event history selection window. The panel refreshes and shows data only from the length of event history that you specified.

Setting the refresh rate of information in a panel

You can set the refresh rate of a panel to update information only as often as you need.

About this task

Panels that can be automatically refreshed display the current refresh rate at the top of that panel. You can determine what refresh rate presents the information in a way that is most readable on your system, and does not restrict the performance of the Operations Console on your system.

Procedure

1. Click the link that shows the current refresh rate for the panel.
2. In the refresh rate selection window, select the unit of time measurement that you want to use to set the refresh rate and enter an appropriate value for the refresh rate that you require.
3. Click the **close button** in the upper right hand corner of the window to close the refresh rate selection window. The panel will refresh at the frequency that you specified.

Setting the scope of information in the Activity tab

If you narrow the scope of the information in the Activity tab and the panels within that tab, you can more easily troubleshoot job run and system performance issues that took place during a specific timeframe.

About this task

You can set the scope of information in the Activity tab with the options at the top of the Activity tab.

Procedure

1. Click the **Current Activity** button to specify the range of activity history that is currently occurring. Click the past activity button to specify the range of activity history that has occurred in the past. You are presented with fields that you can use to specify the length or range of activity history that you want to view.
2. Specify the length or range of activity history that you want to view.
3. Click the **Update View** button  to set the current view and refresh the information that is displayed in the panels of the Activity tab.

Configuring charts in the Run Details and Comparison of Run Details windows

You can configure the display of system resource information and row statistics when you view the details of a job run or compare the information from two job runs.

Before you begin

Access the Performance tab in the Run Details window by clicking on **View Details** after selecting a job run from a list of finished job runs. You can access a list of job runs from the following panels:

- The Job Activity panel on the Home tab
- The Job Runs tab within the main Activity tab
- The Job Runs panel that displays when you select a job on the Projects tab

You can also access the Performance tab when you compare job run details by selecting two job runs for a single job in the Projects tab and clicking **Compare**.

About this task

When you view the details of a job run or compare two job runs, you can configure the graph on the Performance tab to display the job run statistics or system resource information that you need to view or compare.

Procedure

1. Remove unnecessary job run statistics from the default chart by clicking the **X** below the job run statistic that you want to remove. You have open slots to add additional statistics to the chart.
2. Click and drag an available chart statistic from the Job Run or Resources sections into one of the open graph slots. The chart updates with information from the statistic that you included

Tabs and panels in the Operations Console

Each tab in the Operations Console contains several panels that display information about job run activity, engine status information, and system resource usage.

You can use panel information to identify job errors and failures, performance issues, and engine service status information. Each panel shows information only from the selected engine and selected projects within that engine.

If job monitoring is not enabled, the Dashboard, Projects, and Activity tabs, and the Engine Status bar are not available.

If workload management is not enabled, the Workload Management tab is not available.

Dashboard tab

The Dashboard tab shows the status of the selected engine. You can use this tab to quickly see the job run workload, recently completed job runs, the status of engine services, and system resource usage.

Activity panel:

The Activity panel provides details about job workload and job status for all of the jobs on the selected IBM InfoSphere Information Server engine or in the selected projects. Use this activity information to help you troubleshoot jobs that failed or completed with warnings.

The Activity panel provides a graph that displays the recent job run workload. By default, the panel displays information from the entire engine, but you can narrow the scope of this information to view only the projects that you have selected with the View Projects button in the upper right corner of the console.

The graph that displays the job run workload shows only the activity that occurs within the specified time period, and refreshes only as often as you have specified. You can set the scope of the information that is shown in the graph and modify the refresh rate of the graph.

The Activity panel also provides a count of the recent job runs that have failed, finished with warnings, or finished without any warnings. You can view a list of recently completed jobs that have finished by clicking the count for each job run result.

The list of job runs provides information about each job run. You can click on the **View Details** link to see detailed properties, parameters, performance, and messages for a specific job run in the list. Additionally, you can view job and project details by clicking on the name of a job or project in the list.

Engine Status panel:

The Engine Status panel provides details about engine services. Use this service status information to help you identify IBM InfoSphere Information Server engine service issues and troubleshoot engine stability problems.

If a service on the active engine is offline or has errors, the service status or service error is shown in this panel. This panel also displays the general properties of the system on which the active engine is running.

Operating System Resources panel:

The Operating System Resources panel provides details about system resources. Use this resource information to help you manage your system resources and identify system performance issues.

In addition to displaying system resource information, you can configure this panel to notify you when a resource has passed a defined threshold of usage. You can set the threshold for CPU usage, virtual memory usage, and physical memory usage by clicking the Configure button in the upper right corner of the panel.

When a system resource exceeds the specified threshold values, you are notified by a red highlight on the system resource value. You can set threshold values to notify you at critical resource usage values that require your attention.

If your engine operates in a grid or MPP environment and you have configured the operations database to associate multiple compute nodes with the current engine system, you can set the Operating System Resources panel to display data from an individual compute node. If your system is configured to monitor individual compute nodes, a drop down menu is present in the upper right corner of the Operating System Resources panel that will allow you to select which node you want to monitor.

Projects tab

The Projects tab gives you an overview of all of the jobs and projects that are associated with the active IBM InfoSphere Information Server engine.

The repository tree shows the projects, folders, and jobs, including sequence jobs but excluding mainframe jobs, for the InfoSphere Information Server engine. Other objects in the repository, such as table definitions, are not shown, which might result in some folders being shown as empty. Use the InfoSphere DataStage and QualityStageDesigner client to see the full contents of a folder.

Details panel:

The Details panel provides details about the content of your projects and their subfolders, and provides details about job activity for individual jobs. Use this information to help you manage the jobs in your connected job repositories and troubleshoot specific jobs. When you select a project or a folder in a project from the repository tree, you can view information about the properties and the contents of that project or folder.

If you select a folder, the Details panel displays the date that the folder was created, and the number of subfolders and jobs that are contained within the selected folder.

If you select a job, you can view detailed information about the properties of that job and information about the latest run activity and status for that job. This information describes the length of time required to execute the job, the number of rows in and out during the job run, and the number of message that occurred during the job run.

If you select a job, the following actions are available:

- Click **Run** to run the job. The Job Run dialog is displayed.
- Click **Stop** to stop the job if it is running.
- Click **Reset** to reset a job that stopped or aborted.
- Click **View Job Design** to see the design of the job.

Sequence jobs and batch jobs

Sequence jobs, batch jobs, and jobs that call other jobs in their job control code have jobs that they depend on. The Details panel displays information about the number of dependencies between the selected job and the other jobs in your system.

Click **Depends on job(s)** to see a list of the jobs that your selected job depends on. Click the name of a job in the list to see details of that job.

Click **Used by job(s)** to see a list of the jobs that use your selected job. Click the name of a job in the list to see details of that job.

Job Run dialog:

Use the Job Run dialog to specify parameters and limits for a server or parallel job.

The Job Run dialog contains only the controls, fields, and tabs that are applicable to the job. Controls, fields, and tabs that are not applicable to the job are not shown.

Invocation ID

Select a previously used invocation ID, or enter a new invocation ID for a multi-instance job.

Queue

Select the workload management queue that the job is submitted to.

Parameters tab

Use the Parameters tab to specify the value that the job parameters take when the job is run.

The list of parameters for the job is shown at the top of the panel. If you select a parameter in the list that is a parameter set, the parameters in the parameter set are shown at the bottom of the panel.

The Parameters tab contains the following controls and fields:

Set to Default

Click to set the selected parameter to its default value.

Set all to Default

Click to set the all the parameters to their default values.

Name Displays the name of the parameter.

Value Displays the value of the parameter. Double-click to edit the value.

Options tab

Use the Options tab to specify limits and general job settings that constrain the job when it runs.

Limits

Stop stages after

Select this option and specify the maximum number of rows of data a stage can process.

Abort job after

Select this option and specify the number of warnings that cause the job to be stopped.

General

Generate operational metadata

Select this option to have the job generate operational metadata. Operational metadata describes the events and processes that occur and the objects that are affected when you run a job.

Disable project-level message handling

Select this option to disable any message handler that is defined for the project.

Disable compiled-in job-level message handling

Select this option to disable any local message handler.

Mode

Run the job in validate mode

Select this option to validate the job only.

OK Click to submit the job.

Cancel

Click to close the Job Run dialog without submitting the job.

Environment Variables panel:

The Environment Variables panel provides information about the environment variables that are associated with the selected project. Use this information to identify missing or incorrect environment variables before you modify those variables in the IBM InfoSphere DataStage and QualityStage Administrator client.

If you select a project from the repository tree in the Navigation panel, the Environment Variables panel displays all of the environment variables that are associated with that project. You can sort environment variables on either of the three columns in the list of environment variables.

Job Designs panel:

The Job Designs panel provides a view of the job design information for all of the jobs in the selected folder. Use this job design information to quickly identify job design problems before you analyze and modify your job designs in the IBM InfoSphere DataStage and QualityStage Designer.

If you select a folder from the repository tree in the Navigation panel, a list of all of the jobs in that folder appears in the Job Designs panel and the design layout for each job is displayed for each job. Click any job in the list to see a larger view of the job design. This panel gives you access to the job design information in the console so you can identify design issues and correct them.

Job Runs panel:

The Job Runs panel provides job run history information for the selected job. Use this run history information to troubleshoot individual job issues and analyze job run performance.

When you select a job from the repository tree in the Navigation panel, the Job Runs panel is displayed, showing a history of job runs for that particular job. You can sort jobs on any column in the list of runs for the selected job.

If the list of job runs has multiple pages, the text Page 1 (more available) is shown above the job run list. You can navigate through these pages with the arrows above the job run list.

If you select an individual run in the list of runs, the following actions are available:

- Click **Stop** to stop the job if it is running.
- Click **View Details** to view more detailed information about that run. The Run Details window is displayed, showing detailed properties, parameters, performance, and messages for the selected job run.
- Click **Investigate Timeframe** to go to the Activity Tab and see a list of job runs that were running during the same timeframe as the selected job run, which can help you to identify the cause of job performance issues.

If you select two job runs in the list of runs and click **Compare**, you can compare the two job runs side by side. This feature helps you to determine the differences between two job runs and analyze job run problems.

Activity tab

You can use the Activity tab to get an overview of all of the job activity on the active engine.

Job Runs tab:

The Job Runs panel provides details about the job run history for all of the jobs that were run on the selected IBM InfoSphere Information Server engine during a set time period. Use this activity information to troubleshoot jobs that failed or completed with warnings during specific time periods.

Before viewing job activity information in the Job Runs tab, you must specify the scope of activity that you want to view within the entire Activity tab.

You can sort job runs on any of the columns in the job run list. If the list of job runs has multiple pages, the text Page 1 (more available) is shown above the job run list. You can navigate through these pages with the arrows above the job run list.

If you select an individual run in the list of runs, the following actions are available:

- Click **Stop** to stop the job if it is running.
- Click **View Details** to view more detailed information about that run. The Run Details window is displayed, showing detailed properties, parameters, performance, and messages for the selected job run.

If you narrow the scope of your information by selecting specific projects to view with the project filter, job runs from projects that are not included in the filter are not displayed. If your view is not narrowed by the project filter and matching job runs exist in projects that you are not authorized to view, the job runs are hidden from display. However, you are notified that some jobs are hidden from your view.

Resources tab:

The Resources tab provides details system resource information for the selected engine during a set time period. Use this resource information to better manage system resources during periods of heavy use, or troubleshoot performance issues that occur during specific time periods.

Before viewing resource activity information in the Resources tab, you must specify the scope of activity that you want to view within the entire Activity tab.

You can add or remove charts from the Resources tab with the chart selection menu.

You can select charts that you want to add by checking them on the list, and remove charts by removing the checks from the charts on the list. Optionally, you can remove charts by clicking the x in the upper right corner of the chart window.

By default, no charts are available on the list. You can add, remove, and configure the charts that are available on the list by configuring the monitoring data in the operations database.

If your engine operates in a grid or MPP environment and you have configured the operations database to associate multiple compute nodes with the current engine system, you can set the Resources tab to display data from an individual compute node. If your system is configured to monitor individual compute nodes, a drop down menu is present in the upper right corner of the Resources tab that will allow you to select which node you want to monitor.

Workload Management tab

Use the Workload Management tab to set system policies, to monitor queued jobs, and to manage the workload management queues.

System Policies panel:

Use the System Policies panel to modify the workload management system policies. The corresponding current system usage is displayed for these policies.

You can set threshold values for these system policies: Job Count, CPU Usage, Memory Usage, and Job Start. Set the required value for the threshold and click **Apply**.

You can set the threshold values for these system policies:

Job Count

Specify the maximum number of concurrent running jobs allowed on the system. The default value is 20 concurrent running jobs.

CPU Usage

Specify the maximum CPU usage that is allowed on the system. If the current CPU usage exceeds this value, the job is not allowed to start. The default value is 80 percent CPU usage.

Memory Usage

Specify the maximum memory usage that is allowed on the system. If the memory usage exceeds this value, the job is not allowed to start. The default value is 80 percent memory usage.

Job Start

Specify the maximum number of jobs that are allowed to start in the specified number of seconds. The default value is 100 jobs in 10 seconds.

The workload management process checks the system resource usage against the four system policies to determine if another job can be started. If all the checks are successful, workload management process checks the queue definitions to determine which job is started. If one of the checks is not successful, a job is not started until more resources are available.

If job monitoring is enabled, the Job Count, CPU Usage, and Memory Usage system policies also show the current system values. You can click the current system values to see further details. If the current system value exceeds the threshold value for a system policy, the current system value is shown with a red background.

Queued Jobs panel:

Use the Queued Jobs panel to view the jobs that are in the workload management queues, and to move jobs to different queues or to the top of their current queue.

For a particular workload management queue, the list shows the order in which the jobs are due to be started. The job at the top of the list is started first.

Users can see information only for the projects for which they have been granted appropriate access. Users who have the DataStage and QualityStage Administrator role can view the information for all projects.

If you have the DataStage and QualityStage Administrator role, you can manage the jobs in the queues. If you select one or more jobs, the following actions are available:

- Click **Move to Queue** to move the jobs to another queue.
- Click **Move to Top** to move the jobs to the top of the queue.
- Click **Remove** to remove the jobs from the queue. The jobs are stopped.

Some of these actions are not available for certain types of queue, such as the predefined system queues.

If job monitoring is enabled, you can click **Running Jobs** to view details of the jobs that are currently running. The **Running Jobs** dialog is displayed, where you can stop a job or see further details of a job.

Queue Management panel:

Use the Queue Management to add or delete workload management queues or to modify the definition of the queues.

Only users with the DataStage and QualityStage Administrator role can configure the workload management queues. Other users of the operations console can view the definition of the queues.

If the system policies allow another job to be started, the top job in each queue that does not have its maximum number of running jobs is examined. The job with the highest priority is started. The Queue priority rule, which is used to determine the priority of a job, ensures that jobs from lower priority queues are eventually started.

Select one of these values for the **Queue priority rule**:

- Select **Job Run Ratio**, and specify the **High to Medium** and **Medium to Low** ratios. The priority of a job is derived from the priority of the queue it was submitted to. The ratios determine how many jobs are started from a high priority queue before a job is started from a medium priority queue, and how many jobs are started from a medium priority queue before a job is started from a low priority queue.
- Select **Elapsed Time**. The priority of a job is derived from the elapsed time since the job was submitted to the queue. This rule gives the highest priority to the job that was submitted first, irrespective of the queue it was submitted to.
- Select **Priority Weight**. The priority of a job is derived from the priority of the queue it was submitted to and from the elapsed time since the job was submitted to the queue. This rule is the default rule.

The list at the bottom of the Queue Management panel shows the definition of all the workload management queues that are defined. The following actions are available:

- Click **New Queue** to add a queue. Specify the definition of the queue.
- Select a queue in the list of queues, and click **Delete** to delete the queue. You cannot delete certain predefined queues, and you cannot delete queues that contain jobs.
- Double-click a field for one of the queues in the list to change the value of the field. The definitions of certain predefined queues cannot be changed.

Click **Save** to save the changes to the definition of the queues.

Engine Status bar

The Engine Status bar is at the bottom of the Operations Console, and provides information about the status of all of the engines that are associated with the services tier machine.

Additionally, the Engine Status bar notifies you when problems occur on an engine that is not specified as the current active engine in the Operations Console. You can view the full list of engines with their current status by clicking on the **All Engines** link on the Engine Status bar in the lower left of the Operations Console.

Extracting monitoring data from the operations database

Run SQL queries against the operations database to extract monitoring data.

Extract the names of all the jobs running on a specific engine

This SQL query extracts the names of all the jobs running on a specific engine.

Procedure

Run this SQL query:

```
SELECT
    H.HostName
    , J.ProjectName
    , J.JobName
    , R.InvocationId
FROM
    DSODB.JobExec AS J
    , DSODB.JobRun as R
    , DSODB.Host as H
WHERE H.HOSTID = J.HOSTID
AND R.JOBID = J.JOBID
AND R.RunMajorStatus = 'RUN' -- i.e. status = "running" --
AND H.HostName = 'HOSTNAME'
;
```

Extract the full status of all the jobs started after a certain time on any host

This SQL query extracts the full status of all the jobs started after a certain time on any host.

Procedure

Run this SQL query:

```
SELECT
    H.HostName
    , J.ProjectName
    , J.JobName
    , R.InvocationId
    , S.MajorStatusName
    , R.RunStartTimeStamp
FROM
    DSODB.JobExec AS J
    , DSODB.JobRun AS R
    , DSODB.Host AS H
    , DSODB.RunMajorStatusRef AS S
WHERE H.HOSTID = J.HOSTID
AND R.JOBID = J.JOBID
AND R.RunMajorStatus = S.MajorStatusCode
AND R.RunStartTimeStamp >= '2011-09-26'
ORDER BY R.RunStartTimeStamp
;
```

Extract the details of all jobs ever run, showing the run type as readable string

This SQL query extracts the details of all jobs ever run, showing the run type as readable string.

Procedure

Run this SQL query:

```
SELECT
    H.HostName
    , J.ProjectName
```

```

        , J.JobName
        , R.InvocationId
        , R.RunStartTimeStamp
        , T.RunTypeName          -- code converted to a readable name --
FROM
    DSODB.JobExec AS J
    , DSODB.JobRun AS R
    , DSODB.Host AS H
    , DSODB.RunTypeRef AS T
WHERE H.HOSTID = J.HOSTID
      AND R.JOBID = J.JOBID
      AND R.RunType = T.RunTypeCode
ORDER BY R.RunStartTimeStamp
;

```

Extract the job run logs for a particular job run

This SQL query extracts the job run logs for a particular job run.

Procedure

Run this SQL query:

```

SELECT
    L.LogTimestamp
    , T.LogTypeName          -- code converted to a readable name --
    , L.MessageId
    , L.MessageText
FROM
    DSODB.JobExec AS J
    , DSODB.JobRun AS R
    , DSODB.JobRunLog AS L
    , DSODB.LogTypeRef AS T
WHERE J.ProjectName = 'projectname'
      AND J.JobName = 'joblname'
      AND R.InvocationId = 'id'
      AND R.JOBID = J.JOBID
      AND L.RUNID = R.RUNID
      AND L.LogType = T.LogTypeCode
      AND R.CreationTimestamp > '2011-09-26'
ORDER BY L.EventId
;

```

Extract the details of all the job runs with a particular parameter set to a given value

This SQL query extracts the details of all the job runs with a particular parameter set to a given value.

Procedure

Run this SQL query:

```

SELECT
    H.HostName
    , J.ProjectName
    , J.JobName
    , R.InvocationId
    , R.RunStartTimeStamp
    , S1.MajorStatusName
    , S2.MinorStatusName
    , R.ElapsedRunSecs
FROM
    DSODB.JobExec AS J
    , DSODB.JobRun AS R
    , DSODB.Host AS H

```

```

        , DSODB.JobRunParamsView AS P
        , DSODB.RunMajorStatusRef AS S1
        , DSODB.RunMinorStatusRef AS S2
WHERE H.HOSTID = J.HOSTID
      AND R.JOBID = J.JOBID
      AND R.RUNID = P.RUNID
      AND P.ParamName = 'paramname' AND P.ParamValue = 'paramvalue'
      AND R.RunMajorStatus = S1.MajorStatusCode
      AND R.RunMinorStatus = S2.MinorStatusCode
ORDER BY R.RunStartTimeStamp
;

```

Extract the details of all job runs that were active after a given time on a particular host

This SQL query extracts the names of all the jobs running on a specific engine.

About this task

There are three sets of job runs that meet these criteria:

1. Job runs that started after the given time
2. Job runs that finished after the given time
3. Job runs that are still running

These three sets of jobs are identified by the final condition in the WHERE clause, which are OR'ed together because they might overlap.

Procedure

Run this SQL query:

```

SELECT
    R.RUNSTARTTIMESTAMP
    , R.INVOCATIONID
    , J.PROJECTNAME
    , J.JOBNAME
FROM
    DSODB.JOBRUN AS R
    , DSODB.JOBEXEC AS J
    , DSODB.HOST AS H
WHERE
    R.JOBID = J.JOBID
    AND J.HOSTID = H.HOSTID
    AND H.HOSTNAME = 'HOSTNAME'
    AND R.CREATIONTIMESTAMP >= '2011-09-26'
    OR R.RUNENDTIMESTAMP >= '2011-09-26'
    OR R.RUNENDTIMESTAMP IS NULL
;

```

Extract the details of all job runs that were active in a given period on a particular host

This SQL query extracts the names of all the jobs running on a specific engine.

About this task

There are three sets of job runs that meet these criteria:

1. job runs that started in the given period
2. job runs that ended in the given period
3. job runs that started before the end of given period and that are still running

These three sets of jobs are identified by the final condition in the WHERE clause, which are OR'ed together because they might overlap.

Procedure

Run this SQL query:

```
SELECT
    R.RUNSTARTTIMESTAMP
    , R.INVOCATIONID
    , J.PROJECTNAME
    , J.JOBNAME
FROM
    DSODB.JOBRUN AS R
    , DSODB.JOBEXEC AS J
    , DSODB.HOST AS H
WHERE
    R.JOBID = J.JOBID
    AND J.HOSTID = H.HOSTID
    AND H.HOSTNAME = 'HOSTNAME'
    AND ( R.CREATIONTIMESTAMP >= '2011-09-26'
        AND R.CREATIONTIMESTAMP <= '2011-09-27' )
    OR ( R.RUNENDTIMESTAMP >= '2011-09-26'
        AND R.RUNENDTIMESTAMP <= '2011-09-27' )
    OR ( R.CREATIONTIMESTAMP <= '2011-09-27'
        AND R.RUNENDTIMESTAMP IS NULL )
;
```

Extract the slowest jobs in a project based on their last runs

This SQL query extracts the slowest jobs in a project based on their last runs.

Procedure

Run this SQL query:

```
SELECT
    J.JOBNAME
    , MAX(R.RUNSTARTTIMESTAMP) AS LATESTRUN
    , MAX(R.ELAPSEDRUNSECS) AS MAXTIME
FROM
    DSODB.JOBRUN AS R
    , DSODB.JOBEXEC AS J
    , DSODB.HOST AS H
WHERE
    R.JOBID = J.JOBID
    AND J.HOSTID = H.HOSTID
    AND J.PROJECTNAME = 'projectname'
    AND H.HOSTNAME = 'HOSTNAME'
GROUP BY J.JOBNAME
ORDER BY MAXTIME DESC
;
```

Maintenance of the operations database

Information is added by the data collection processes to the operations database, but there is no automatic method for deleting or archiving historical data. You must set up a method to delete or archive the historical data to manage the size of the operations database.

You can use the **istool ODBAdmin** command with the **countdb** parameter, or SQL queries, to obtain information about the number of rows in the main tables in the operations database.

You can use the **istool ODBAdmin** command with the **purgedb** parameter, or SQL queries, to delete historical data from the operations database.

Reporting database content

Use the **istool ODBAdmin** command with the **countdb** parameter to obtain information about the number of rows in the main tables in the operations database

About this task

You can use the **istool ODBAdmin** command with the **countdb** parameter to determine how much data would be deleted from the operations database by using the **istool ODBAdmin** command with the **purgedb** parameter.

Procedure

Use a command of this format to produce the report:

```
istool ODBAdmin countdb server selector time_range output
```

where:

- **server** specifies the system that is used to run the command.
For example: `-server servername`.
The **server** parameter defaults to the engine name specified in the **selector** parameter. If the **selector** parameter is not specified, the **server** parameter must be specified.
- (Optional) **selector** restricts the report to a single engine.
For example: `-engine enginename`
- **time_range** restricts the time period that is reported. Only data that was recorded before this time period is reported.
The **time_range** parameter takes one of the following formats:
 - `-upto Number -hours | -days | -months | -years`
The `-upto` parameter treats the current period, which by definition is not yet complete, as one of the *Number* of periods.
For example, to report all data recorded before the last two days:
`-upto 2 days`
 - `-all`
- (Optional) **output** specifies the file to which the report is written.
For example: `-text filename.txt`

Count, from server S, all database content for all engines

This command counts, from server S, all database content for all engines.

Procedure

Run this command:

```
istool ODBAdmin countdb -domain D -authfile af.txt -server S -all
```

Count, from server E, all database content for engine E

This command counts, from server E, all database content for engine E.

Procedure

Run this command:

```
istool ODBAdmin countdb -domain D -authfile af.txt -engine E -all
```

Count, from server S, all database content for engine E

This command counts, from server S, all database content for engine E.

Procedure

Run this command:

```
istool ODBAdmin countdb -domain D -authfile af.txt -server S -engine E -all
```

Count, from server E, the database content for engine E that is more than one year old

This command counts, from server E, the database content for engine E that is more than one year old.

Procedure

Run this command:

```
istool ODBAdmin countdb -domain D -authfile af.txt -engine E -upto 1 -years
```

Deleting historical data from the operations database

Delete historical data from the operations database to regulate the size of the database.

About this task

The operations database schema is defined so that the deletion of a row that is referenced in other associated tables by a foreign key causes the rows that refer to that row to also be deleted. Table 9 shows the tables which are defined in this manner.

Table 9. Primary tables and their associated tables for deletion

Primary table	Associated tables
Host	<ul style="list-style-type: none">• HostDetail• ResourceSnap• ResourceUsage• JobExec• ParallelConfig
JobExec	<ul style="list-style-type: none">• JobRun• JobStage• JobLink
JobRun	<ul style="list-style-type: none">• JobRunStage• JobRunLink• JobRunLog• JobRunParams• JobRunUsage• JobRun for jobs that are part of a sequence started by the primary JobRun job

Deleting a run of a job sequence also deletes all job runs started from that sequence.

Rows in the DataLocator table are not dependent on any other table. Rows in the DataLocator table that are not referenced from any row in the JobRunLink table are no longer used and are candidates for deletion.

Deleting data by using the `istool ODBAdmin` command

Use the `istool ODBAdmin` command with the `purgedb` parameter to delete historical data from the operations database.

About this task

The operations database schema is defined so that the deletion of a row that is referenced in other associated tables by a foreign key causes the rows that refer to that row to also be deleted. For more information, see “Deleting historical data from the operations database” on page 63.

Procedure

Use a command of this format to delete historical data from the operations database:

```
istool ODBAdmin purgedb server area selector time_range output
```

where:

- **server** specifies the system that is used to run the command.

For example: `-server servername`.

The **server** parameter defaults to the engine name specified in the **selector** parameter. If the **selector** parameter is not specified, the **server** parameter must be specified.

- **area** specifies what type of data is to be deleted. Specify one of these values:
 - `-runs` to delete job run data.
 - `-resource` to delete resource usage data.
 - `-full` to delete job run and resource usage data, and to delete redundant locators.
 - `-host hostname` to delete a single host row. This command returns an error if any job run or resource usage data exists for the host.
- (Optional) **selector** restricts the deletion of data. Different values are applicable to each value of the **area** parameter.

If the **area** parameter is `-runs`, you can specify one of these values:

- `-engine enginename`. This option restricts the deletion of data to runs on the named engine. The purge action takes place on this machine, unless the **server** parameter is specified.
- `-project projectname`. This option requires that the `-engine` option is also specified. This option restricts the deletion of data to runs in the named project on the named engine.

If the **area** parameter is `-resources`, you can specify one of these values:

- `-engine enginename`. This option restricts the deletion of data to resource usage where the named engine is the head node. The purge action takes place on this machine, unless the **server** parameter is specified.

- `-node nodename`. This option requires that the `-engine` option is also specified. This option restricts the deletion of data to resource usage associated with the named node as managed by the named engine.

If the **area** parameter is `-full`, you can specify this value:

- `-engine enginename`. This option restricts the deletion of data to resource usage to the named engine.

If the **area** parameter is `-host`, this parameter does not apply.

- **time_range** restricts the time period that is reported. Only data that was recorded before this time period is deleted.

The **time_range** parameter takes one of the following formats:

- `-upto Number -hours | -days | -months | -years`

The `-upto` parameter treats the current period, which by definition is not yet complete, as one of the *Number* of periods.

For example, to delete all data recorded before the last two days:

```
-upto 2 days
```

- `-all`

- (Optional) **output** specifies the file to which the report is written.

For example: `-text filename.txt`

Purge, from server E, all runs within project P on engine E that finished yesterday or earlier:

This command purges, from server E, all runs within project P on engine E that finished yesterday or earlier.

Procedure

Run this command:

```
istool ODBAdmin purgedb -domain D -authfile af.txt
-runs -engine E -project P -upto 1 -days
```

Purge, from server E, all runs on engine E that finished more than 24 hours ago:

This command purges, from server E, all runs on engine E that finished more than 24 hours ago.

Procedure

Run this command:

```
istool ODBAdmin purgedb -domain D -authfile af.txt -runs -engine E -upto 24 -hours
```

Purge, from server E, all runs and resource usage data that are more than a week old, and all redundant locators:

This command purges, from server E, all runs and resource usage data that are more than a week old, and all redundant locators.

Procedure

Run this command:

```
istool ODBAdmin purgedb -domain D -authfile af.txt -full -engine E -upto 7 -days
```

Purge, from server E, all resource usage data for node N on engine E:

This command purges, from server E, all resource usage data for node N on engine E.

Procedure

Run this command:

```
istool ODBAdmin purgedb -domain D -authfile af.txt -resource -node N -engine E -all
```

Purge, from server E, all runs and resource usage data for engine E, and all redundant locators:

This command purges, from server E, all runs and resource usage data for engine E, and all redundant locators.

Procedure

Run this command:

```
istool ODBAdmin purgedb -domain D -authfile af.txt -full -engine E -all
```

Purge, from server E, engine E from the Host table, assuming all its runs and resource usage data have been deleted, as shown in the previous command:

This command purges, from server E, engine E from the Host table, assuming all its runs and resource usage data have been deleted, as shown in the previous command.

Procedure

Run this command:

```
istool ODBAdmin purgedb -domain D -authfile af.txt -host E
```

Deleting data by using SQL Queries

Use SQL queries to delete historical data from the operations database.

About this task

The operations database schema is defined so that the deletion of a row that is referenced in other associated tables by a foreign key causes the rows that refer to that row to also be deleted. For more information, see “Deleting historical data from the operations database” on page 63.

Purge all runs of a specific job:

This command purges all runs of a specific job.

Procedure

Run this command:

```
DELETE
  FROM JobExec
 WHERE JOBID IN
    ( SELECT J.JOBID
      FROM JobExec AS J, Host as H
      WHERE J.HOSTID = H.HOSTID
        AND J.JobName = 'jobname'
```

```

        AND J.ProjectName = 'projectname'
        AND H.HostName = 'hostname'
    )
;

```

Purge all runs of a specific job that had a given invocation ID at run time:

This command purges all runs of a specific job that had a given invocation ID at run time.

Procedure

Run this command:

```

DELETE
  FROM JobRun
 WHERE RUNID IN
   ( SELECT R.RUNID
     FROM JobExec AS J, JobRun AS R, Host as H
    WHERE R.JOBID = J.JOBID
          AND J.HOSTID = H.HOSTID
          AND JobName = 'jobname'
          AND ProjectName = 'projectname'
          AND InvocationId = 'id'
          AND H.HostName = 'hostname'
    )
;

```

Purge all runs within a named DataStage project:

This command purges all runs within a named DataStage project.

Procedure

Run this command:

```

DELETE
  FROM JobExec
 WHERE JOBID IN
   ( SELECT J.JOBID
     FROM JobExec AS J, Host as H
    WHERE J.HOSTID = H.HOSTID
          AND J.ProjectName = 'projectname'
          AND H.HostName = 'hostname'
    )
;

```

Purge all runs that finished before a certain date:

This command purges all runs that finished before a certain date, regardless of host.

Procedure

Run this command:

```

DELETE
  FROM JobRun
 WHERE RunEndTimestamp < '2011-09-26'
;

```

Purge all runs that finished before a certain date within a named project:

This command purges all runs that finished before a certain date within a named project.

Procedure

Run this command:

```
DELETE
  FROM JobRun
  WHERE RUNID IN
    ( SELECT R.RUNID
      FROM JobExec as J, JobRun as R, Host as H
      WHERE J.JOBID = R.JOBID
            AND J.HOSTID = H.HOSTID
            AND R.RunEndTimestamp < '2011-09-26'
            AND J.ProjectName = 'projectname'
            AND H.HostName = 'hostname'
    )
;
```

Purge all jobs that no longer have any runs associated with them:

This command purges all job records, for any host engine, that no longer have any runs associated with them.

Procedure

Run this command:

```
DELETE
  FROM JobExec
  WHERE JOBID IN
    (
      SELECT J.JOBID
      FROM JobExec AS J, JobRun AS R
      WHERE J.JOBID = R.JOBID
      GROUP BY J.JOBID
      HAVING COUNT(*) = 0
    )
;
```

Changing the user password for the operations database

If the user password for the operations database is changed, you must regenerate the repository connection file for the operations database.

About this task

The user password for the operations database was specified when the database was created, and the password was stored in the repository and a repository connection file on each engine tier computer. Use these steps to modify both locations.

Follow the instructions for the operating system that is used for the services tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Changing the user password for the operations database on Linux or Unix

Procedure

1. Stop the AppWatcher process. See “Starting and stopping the AppWatcher process” on page 34.
2. On the computer that hosts the services tier, switch to the /InformationServer/ASBServer/bin directory.
3. Run the **RepositoryAdmin** tool to obtain the name of the operations database repository:

```
RepositoryAdmin.sh -listRepositories
```
4. Review the current registration of the operations database to obtain the name of the operations database repository connection:

```
RepositoryAdmin.sh -displayRepository -reposName name
```

where **name** is the name of the operations database repository. The name of the operations database repository connection is shown in the **RepositoryConnection** property.

5. Update the user password for the operations database:

```
RepositoryAdmin.sh -updateRepositoryConnection  
-connectionName connectionName  
-reposName name  
-connectionPassword newvalue
```

where

- **connectionName** is the name of the operations database repository connection
 - **name** is the name of the operations database repository.
 - **newvalue** is the new value for the password. Encrypt your password by .
6. On the computer that hosts the engine tier, generate a new repository connection file for the operations database. See “Generating the repository connection file for the operations database” on page 16.
 7. Restart the AppWatcher process. See “Starting and stopping the AppWatcher process” on page 34.

Changing the user password for the operations database on Windows

Procedure

1. Stop the AppWatcher process. See “Starting and stopping the AppWatcher process” on page 34.
2. On the computer that hosts the services tier, switch to the \InformationServer\ASBServer\bin director.
3. Run the **RepositoryAdmin** tool to obtain the name of the operations database repository:

```
RepositoryAdmin.bat -listRepositories
```
4. Review the current registration of the operations database to obtain the name of the operations database repository connection:

```
RepositoryAdmin.bat -displayRepository -reposName name
```

where **name** is the name of the operations database repository. The name of the operations database repository connection is shown in the **RepositoryConnection** property.

5. Update the user password for the operations database:

```
RepositoryAdmin.bat -updateRepositoryConnection
                    -connectionName connectionName
                    -reposName name
                    -connectionPassword newvalue
```

where

- **connectionName** is the name of the operations database repository connection
 - **name** is the name of the operations database repository.
 - **newvalue** is the new value for the password. Encrypt your password by .
6. On the computer that hosts the engine tier, generate a new repository connection file for the operations database. See “Generating the repository connection file for the operations database” on page 16.
 7. Restart the AppWatcher process. See “Starting and stopping the AppWatcher process” on page 34.

Generating the repository connection file for the operations database

The repository connection file for the operations database is used by the engine tier to connect to the operations database.

About this task

Follow the instructions for the operating system used for the engine tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Generating the repository connection file for the operations database on Linux or Unix:

Procedure

1. On the computer that hosts the engine tier, switch to the /InformationServer/Server/DSODB directory.
2. Run the **RegistrationCommand** tool to generate a new repository connection file for the operations database:

```
../../ASBNode/bin/RegistrationCommand.sh -user isadmin -password pswd -gcf
                                         -repository name -cf DSODBConnect.tpl
                                         -results DSODBConnect.cfg
```

where

- **isadmin** is the Information Server admin user.
- **pswd** is the password for the **isadmin** user.
- **name** is the repository for the operations database.

The configuration file, DSODBConnect.cfg, is created in the /InformationServer/Server/DSODB directory.

Generating the repository connection file for the operations database on Windows:

Procedure

1. On the computer that hosts the engine tier, switch to the /InformationServer/Server/DSODB directory.
2. Run the **RegistrationCommand** tool to generate a new repository connection file for the operations database:

```
..\..\ASBNode\bin\RegistrationCommand.bat -user isadmin -password pswd -gcf
                                           -repository name -cf DSODBConnect.tpl
                                           -results DSODBConnect.cfg
```

where

- **isadmin** is the Information Server admin user.
- **pswd** is the password for the **isadmin** user.
- **name** is the repository for the operations database.

The configuration file, DSODBConnect.cfg, is created in the \InformationServer\Server\DSODB directory.

Deleting the operations database

Delete the operations database if you no longer plan to collect monitoring data.

Procedure

1. Set DSODBON=0 in the configuration file to stop the collection of monitoring data.
2. Undo any actions you took to ensure the AppWatcher process was started when the system was rebooted.
3. If you do not have the scripts that were generated when you created the operations database, generate the scripts to be used to create and delete the operations database schema. See “Generating the scripts for the operations database schema” on page 10.
4. If you are using DB2 for the operations database, run these commands:

```
db2 -l dsodb_remove_user.log -stf dsodb_remove_user
db2 -l dsodb_remove_functions.log -stf dsodb_remove_functions
db2 -l dsodb_drop_tables.log -stf dsodb_drop_tables
db2 -l dsodb_remove_schema.log -stf dsodb_remove_schema
```

If you created a new database for the operations database, run this command to delete the database:

```
db2 -l dsodb_remove_db.log -stf dsodb_remove_db
```

5. If you are using Oracle for the operations database, run these commands:
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_remove_functions
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_drop_tables
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
@dsodb_remove_all

For example, run these commands:

```
sqlplus system/manager@ORCL @dsodb_remove_functions
sqlplus system/manager@ORCL @dsodb_drop_tables
sqlplus system/manager@ORCL @dsodb_remove_all
```

6. If you are using SQL Server for the operations database, run these commands:

```
sqlcmd -b -i dsodb_remove_functions.sql >> logfile
sqlcmd -b -i dsodb_drop_tables.sql >> logfile
sqlcmd -b -i dsodb_remove_all.sql >> logfile
```

For example, run these commands:

```
sqlcmd -b -i dsodb_remove_functions.sql >> dsodb_remove_functions.log
sqlcmd -b -i dsodb_drop_tables.sql >> dsodb_drop_tables.log
sqlcmd -b -i dsodb_remove_all.sql >> dsodb_remove_all.log
```

Generating the scripts for the operations database schema

Generate the scripts that are used to create the operations database.

About this task

Follow the instructions for the operating system that is used for the services tier. This task can be run by any user who has the DataStage and QualityStage Administrator role.

Generating the scripts for the operations database on Linux or Unix:

Procedure

1. On the computer that hosts the services tier, switch to the /InformationServer/ASBServer/bin directory.
2. Run the **RepositoryAdmin** tool to generate the required scripts:
`RepositoryAdmin.sh -saveSQLScripts -reposName name -scriptLocation location`

where

- **name** is the name of the repository.
 - **location** is the directory where the scripts are created.
3. If the computer hosting the services tier is not the same as that hosting the operations database, copy the scripts to the computer hosting the operations database.

Generating the scripts for the operations database on Windows:

Procedure

1. On the computer that hosts the services tier, switch to the \InformationServer\ASBServer\bin directory.
2. Run the **RepositoryAdmin** tool to generate the required scripts:
`RepositoryAdmin.bat -saveSQLScripts -reposName name -scriptLocation location`

where

- **name** is the name of the repository.
 - **location** is the directory where the scripts are created.
3. If the computer hosting the services tier is not the same as that hosting the operations database, copy the scripts to the computer hosting the operations database.

Chapter 3. Operations database schema reference

The operations database stores the monitoring data that is displayed by the IBM InfoSphere DataStage and QualityStage Operations Console.

Operations database

The operations database stores information about all job runs on the systems being monitored. It also stores information about the system resource usage on those systems.

Column types

Because multiple database systems are supported by IBM InfoSphere Information Server, and because column types vary across database systems, column type codes are used to describe the columns in the tables in the operations database.

The column type codes used the table descriptions are explained in Table 10.

Table 10. Column types

Column type code	Description	DB2	Oracle	Microsoft SQL Server
SK	Surrogate Key	BIGINT	NUMBER (19, 0)	BIGINT
Int	Integer	INTEGER	NUMBER (11, 0)	INT
Big	BigInt	BIGINT	NUMBER (19, 0)	BIGINT
Txt	Large text string ¹	DBCLOB(512M)	CLOB	NTEXT
XML	column held as an XML structure	XML	XMLTYPE	XML
Tim	Timestamp ²	TIMESTAMP	TIMESTAMP	DATETIME
Ref	Reference key ^{2, 3}	CHAR(3)	CHAR(3)	CHAR(3)
<i>n</i>	Variable length string, maximum length <i>n</i> ³	VARGRAPHIC(<i>n</i>)	VARCHAR2(<i>n</i>)	NVARCHAR(<i>n</i>)
<i>p,s</i>	Decimal precision and scale	DECIMAL(<i>p,s</i>)	DECIMAL(<i>p,s</i>)	DECIMAL(<i>p,s</i>)
Bool	Boolean, held as an integer where 1 means TRUE, and 0 or null mean FALSE	SMALLINT	NUMBER(6,0)	SMALLINT

¹ Text strings use the LINEFEED (LF) character (U+001A) for line separators.

² All timestamps are stored in Coordinated Universal Time (UTC) in the operations database. The HostDetail table stores information about the time zone setting for a particular system.

³ If text is longer than the maximum column length, it is shortened by replacing characters in the middle of the string by an ellipsis (...).

Job run audit data tables and views

The audit data tables for job runs contain information about the IBM InfoSphere DataStage and QualityStage jobs that are currently running or that have finished running.

JobRun table

The JobRun table contains details of all job runs.

Table 11. JobRun table

Column name	Column type code	Description
RUNID	SK	A surrogate primary key to identify the job run.
CONTROLLING_RUNID	SK	If the job run was started from another run, such as a sequence job, the foreign key to the RUNID for that other run in this table.
JOBID	SK	The foreign key to the job executable for this job run in the JobExec table.
CONFIGID	SK	For parallel job runs, the foreign key for the details of the configuration file that is used for the run in the ParallelConfig table.
InvocationId	255	For a multi-instanced job, a string that identifies the instance. For a job that is not multi-instanced, this column contains a single hyphen character (-) to indicate that it has no invocation ID.
CreationTimestamp	Tim	Set when the run event is created by the engine. The combination of JOBID, CreationTimestamp, and InvocationId uniquely identifies a run as only one run of a particular job executable with a given invocation ID can start at a particular time.
LastUpdateTimestamp	Tim	The time that this row in the table was last updated.
RunStartTimestamp	Tim	The time that the run started; otherwise, the value is null.
RunEndTimestamp	Tim	The time that the run finished, otherwise null. If the value of RunMajorStatus is FIN and the value of RunMinorStatus is SYN, the time that the run was synchronized as the end time is unknown.
ElapsedRunSecs	Int	If the job has not finished, the length of time in seconds that the job has been running for. If the job has finished, the length of time in seconds that the job ran for. The length of time is calculated from the RunStartTimestamp and the RunEndTimestamp columns. However, if the value of RunMinorStatus is SYN, the value of ElapsedRunSecs is set to the last value reported by the job run.

Table 11. JobRun table (continued)

Column name	Column type code	Description
RunType	Ref	Distinguishes a normal run from a validate or reset run. See the RunTypeRef view for the possible values.
RunMajorStatus	Ref	Distinguishes between queued, running, and finished job runs. See the RunMajorStatusRef view for the possible values.
RunMinorStatus	Ref	Status of the job run. This column is updated when the first warning or fatal message is issued. See RunMinorStatusRef for the possible values.
UserStatus	40	The value set by the DSSetUserStatus() function. This value is recorded at the end of the run.
MasterPid	Big	The process ID of the DSD.RUN process.
ConductorPid	Big	For parallel jobs only, the process ID of the OSH conductor process, after it is known.
NumMessagesTotal	Int	The number of messages that are logged by the job run. If the run is not complete, the total is updated periodically rather than after every log event.
NumMessagesWarning	Int	The number of warnings that are logged by the job run. A subset of these warnings are logged in the JobRunLog table, depending on the configuration. If the run is not complete, the total is updated at intervals rather than after every log event.
NumMessagesFatal	Int	The number of fatal messages that are logged by the run. If the run is not complete, the total is updated at intervals rather than after every log event. All fatal messages are logged in the JobRunLog table, unless configured differently.
TotalRowsConsumed	Big	The total of all the row counts from source stage links. If the run is not complete, the total is updated periodically.
TotalRowsProduced	Big	The total of all the row counts from target stage links. If the run is not complete, the total is updated periodically.
TotalCPU	Big	The total of the CPU used, in milliseconds, for all stages and control processes.
ConfigFileName	255	For parallel jobs only, the path to the APT_CONFIG_FILE that is used for the run.
TotalPhysicalNodes	Int	For parallel jobs only, the number of different physical node names (fastnames) found in the configuration file. The details of the nodes are in the ParallelConfig table.
TotalLogicalNodes	Int	For parallel jobs only, the number of different logical node names found in the configuration file. The details of the nodes are in the ParallelConfig table.
ISUserName	40	The Information Server user that started the run.
DSUserName	40	The credential-mapped InfoSphere DataStage user.

Table 11. JobRun table (continued)

Column name	Column type code	Description
CustomBatchId	40	The string that is optionally supplied with each job run that you can use to group runs in whatever way is required. A configuration file is used to associate named job parameters with this column.
CustomJobType, CustomContact, CustomSequence, CustomField1, CustomField2	40	Similar to the CustomBatchId column, custom values for named job parameters. You can configure these columns to pick up the values from any named job parameters, or you can leave them null.
QueueName	80	The name of the workload management queue that the job run was submitted to.

JobExec table

The JobExec table contains details of the job versions used by the job runs in the JobRun table.

Table 12. JobExec table

Column name	Column type code	Description
JOBID	SK	A surrogate primary key to identify each specific executable version of a job that has been run.
HOSTID	SK	The foreign key, in the Host table, to the host system that holds the project for this job.
ProjectName	80	The project that the executable was run from.
JobName	255	The name of the job, which is unique within the project.
CompilationTimestamp	Tim	The compilation time for the version of the job executable that is running. Runs for the same job with executables at different compilation times will use different JOBIDs. The combination of HOSTID, ProjectName, JobName, and CompilationTimestamp is unique.
FolderPath	255	The folder that the job is in within the project. If a job is moved without recompilation, this column shows the path of the last run of the job.
JobType	Ref	The type of the job. See the JobTypeRef view for possible values.

Table 12. JobExec table (continued)

Column name	Column type code	Description
JobShortDescription	Txt	The description of the job executable. Changing a job description means that the job must be recompiled, so the CompilationTimestamp also changes.
JobLongDescription	Txt	The long description of the job executable. Changing a job long description means that the job must be recompiled, so the CompilationTimestamp also changes.
DesignProjectName	80	The project that the job was designed in.
DesignHostName	80	The host system that the job was designed on.
IsMultiInstance	Bool	Set to True if the job can be a multi-instance job.
WebServicesEnabled	Bool	Set to True if the job is a WebServices job, that is a job that has RTI/WISD stages.

Host table

The Host table contains details of engine systems where jobs are started, and of remote nodes used by an engine system to distribute parallel jobs.

The HostDetail table contains information about the available system resources of every engine system and compute node in the Host table.

Table 13. Host table

Column name	Column type code	Description
HostName	80	The name of the DataStage engine or remote system. The name is obtained by executing the hostname command or from the DS_HOSTNAME_ALIAS environment variable if it is set. The name is normalized to uppercase.

Table 13. Host table (continued)

Column name	Column type code	Description
InstallationDir	255	The location of the DS0DB directory associated with an installation of the DataStage engine on the host, which can be used to distinguish between multiple installations of IBM InfoSphere Information Server on the same system. If this column is a string consisting of a single hyphen (-), the host represents a remote node rather than an engine node. The combination of HostName and InstallationDir is unique.
CreatedTimestamp	Tim	The time that this row was inserted into the table.
MonStartTimestamp	Tim	The latest time that an EngMonApp process started and identified this row as the one representing its own installation.

JobRunParams table

The JobRunParams table contains the parameters used for each job runs in the JobRun table, stored as an XML fragment.

Table 14. JobRunParams table

Column name	Column type code	Description
RUNID	SK	The foreign key of the job run in the JobRun table that this set of parameter values belongs to.
ParamList	XML	An XML fragment containing the parameter values. Use the JobRunParamsView view to extract this information.

JobRunParamsView view

The JobRunParamsView view expands the ParamList column of the JobRunParams table. The JobRunParamsView view expands the parameter names and values individually, for all job runs in the JobRun table.

Table 15. JobRunParamsView view

Column name	Column type code	Description
RUNID	SK	The foreign key of the job run in the JobRun table that this set of parameter values belongs to.
ParamName	255	The name of the parameter.
ParamValue	255	The value of the parameter.

JobRunLog table

The JobRunLog table contains all the events that are emitted by the job runs in the JobRun table that match the monitoring configuration settings.

Table 16. JobRunLog table

Column name	Column type code	Description
RUNID	SK	The foreign key to the job run in the JobRun table for the run that emitted this log.
EventId	Int	A number that can be used to sort log events for a run. The EventId is not necessarily the event number found in the RT_LOG file. The combination of RUNID and EventID is unique.
LogTimestamp	Tim	The time, to the nearest second, when the log event was emitted by the run.
LogType	Ref	The message type of the log event. See the LogTypeRef view for possible values.
MessageID	24	The ID string for the message. For example, IIS-DSEE-TFSC-00010 or IIS-DSTAGE-RUN-I-0158.
ContentType	24	Used to indicate that certain messages have special significance. The values that appear in this column are set up in the configuration file using the CaptureLog option.
MessageText	Txt	The full text of the logged message.

ParallelConfig table

The ParallelConfig table contains details of every engine in each parallel job configuration file that has been used in a job run and that contains a unique set of physical and logical nodes, stored as an XML fragment.

Table 17. ParallelConfig table

Column name	Column type code	Description
CONFIGID	SK	A surrogate primary key.
HOSTID	SK	The foreign key to the Host in the Host table for the engine that hosted the job runs which used this definition.
CreationTimestamp	Tim	The time that this row was inserted into the table.
NodeListHash	Int	The hash key of the Nodelist column, which together with the HOSTID column forms a unique key for this table.
NodeList	XML	An XML fragment containing the physical and logical nodes. Use the ParallelConfigNodes view to extract this information.

ParallelConfigNodes view

The ParallelConfigNodes view expands the NodeList column of the ParallelConfig table. The ParallelConfigNodes view shows details of every engine in each parallel job configuration file that has been used in a job run and that contains a unique set of physical and logical nodes, shown as individual physical nodes.

Table 18. ParallelConfigNodes view

Column name	Column type code	Description
CONFIGID	SK	A surrogate primary key.
HOSTID	SK	The foreign key to the Host in the Host table for the engine that hosted the job runs which used this definition.
PhysicalName	255	The name of the physical node, known as the fastname.
NumLogicalNodes	Int	The number of logical nodes that are assigned to the physical node.

Job run metrics tables and views

The job run metrics tables contain information about the resources used by IBM InfoSphere DataStage and QualityStage job runs.

JobStage table

The JobStage table contains the static details, that is the details of stages that are set at compilation time, for all job stages that emit events.

The JobRunStage table contains details of the processing of every stage in all job runs.

Table 19. JobStage table

Column name	Column type code	Description
STAGEID	SK	A surrogate primary key for the job stage.
JOBID	SK	The foreign key to the job in the JobExec table that this stage was compiled from.
StageName	80	The name of the stage.
ContainerPath	255	If the stage is contained within a container, the path of the container. If the stage is at the top level of the job, this column contains a hyphen (-). The combination of JOBID, StageName, and ContainerPath is unique.
StageDescription	Txt	The stage description.
StageTypeName	40	The stage type for the stage, as shown in the Designer client.

JobRunStage table

The JobRunStage table contains details of the processing of every stage in all job runs.

Table 20. JobRunStage table

Column name	Column type code	Description
RUNID	SK	The foreign key to the job run in the JobRun table.
STAGEID	SK	The foreign key to the stage in the JobStage table. The combination of RUNID and STAGEID is unique.
StageStartTimeStamp	Tim	The time that the stage started.
StageEndTimeStamp	Tim	The time that the stage finished. If the stage is still running, this column is set to null.
LastUpdateTimeStamp	Tim	The time that this row in the table was last updated.
ElapsedRunSecs	Int	The length of time in seconds that the stage ran for. The length of time is calculated from the StageStartTimeStamp and the StageEndTimeStamp columns.
StageStatus	Ref	The status of the stage. See the StageStatusRef view for possible values.

Table 20. JobRunStage table (continued)

Column name	Column type code	Description
NumInstances	Int	For parallel jobs, the number of instances of the stage that were used. For server jobs the number of instances is always 1. The NumInstances column shows how many entries there are in the InstancePidList and InstanceCPUList columns.
InstancePidList	512	A comma-separated list of process IDs.
InstanceCPUList	512	A comma-separated list of the CPU time used, in milliseconds, by the instances of the stage in each of the processes in the InstancePidList column.
TotalCPU	Int	The total of the CPU time, in milliseconds, for all entries in the InstanceCPUList column.

JobLink table

The JobLink table contains the static details, that is the details of links that are set at compilation time, for all links for which events are emitted.

The JobRunLink table contains details of the use of every link in all job runs.

Table 21. JobLink table

Column name	Column type code	Description
LINKID	SK	A surrogate primary key for the link.
FROMSTAGEID	SK	The foreign key to the stage in the JobRunStage table for the stage that this link outputs from.
TOSTAGEID	SK	The foreign key to the stage in the JobRunStage table for the stage that this link inputs to.
LinkName	80	The name of the link, which is unique within the pair of stages it connects. The combination of FROMSTAGEID, TOSTAGEID, and LinkName is unique.
LinkType	Ref	The type of the link. See the LinkTypeRef view for possible values.
LinkDescription	Txt	The description of the link.
IsSource	Bool	Set to True if the link is reading from an external data source.

Table 21. JobLink table (continued)

Column name	Column type code	Description
IsTarget	Bool	Set to True if the link is writing to an external target.

JobRunLink table

The JobRunLink table contains details of the use of every link in all job runs.

Table 22. JobRunLink table

Column name	Column type code	Description
RUNID	SK	The foreign key to the job run in the JobRun table.
LINKID	SK	The foreign key to the link in the JobLink table. The combination of RUNID and LINKID is unique.
LOCATORID	SK	The foreign key to a locator in the DataLocator table, if the link reports that information.
LastUpdateTimeStamp	Tim	The time that this row in the table was last updated.
InstanceRowsList	512	The count of the rows handled by the link. This column is a comma-separated list if more than one instance of the link was used, for example by a parallel job running on multiple nodes.
TotalRows	Big	The total of all the rows handled by all the instances of the links in the InstanceRowsList column.

DataLocator table

The DataLocator table contains details of every source or target for every link in all job runs.

Use the DataLocator table to connect job runs that read or write to the same data source.

Table 23. DataLocator table

Column name	Column type code	Description
LOCATORID	SK	A surrogate primary key to the locator.
CreationTimeStamp	Tim	The time that this row was inserted into the table.

Table 23. DataLocator table (continued)

Column name	Column type code	Description
ComputerName	80	The name of the system that is hosting the data source/target.
SoftwareProductName	60	The name of the software system that contains the resource. For example, DB2 or FileSystem.
DataStoreSubclass	20	The type of the data store. For example, DATABASE or DIRECTORY.
DataStoreName	80	The name of the data store.
DataSchemaSubclass	20	The type of the data schema. For example, SCHEMA or FILE.
DataSchemaName	80	The name of the data schema, if any.
DataCollectionSubclass	20	The subclass of the data collection. Usually, the subclass is TABLE or FILE.
DataCollectionName	80	The name of the data collection. Usually, the name of the table, or the path to the file.

Examples of Locators

The locator for a parallel job writing to a local sequential file has values like the following:

```

ComputerName      = NAME1
SoftwareProductName = FileSystem
DataStoreSubclass = DIRECTORY
DataStoreName     = C:\IBM\InformationServer\Server\Projects\dstage0
DataSchemaSubClass = FILE
DataSchemaName    = PXFILE1.txt
DataCollectionSubClass = FILE
DataCollectionName = PXFILE1.txt
    
```

The locator for a parallel job reading from an ODBC source on Microsoft Windows has values like the following:

```

ComputerName      = NAME1
SoftwareProductName = PX-ODBC
DataStoreSubClass = DATABASE
DataStoreName     = XMETADSN
DataSchemaSubClass = SCHEMA
DataSchemaName    = db2admin
DataCollectionSubClass = TABLE
DataCollectionName = DB2_TEST1
    
```

Reference tables and views

The reference tables provide readable strings, in English, for the codes contained in the job run audit data tables and the job run metrics tables.

MasterRef table

The MasterRef table contains readable descriptions for the enumerations used in the Job run audit tables and the Job run metrics tables.

Use the JobTypeRef, RunTypeRef, RunMajorStatus, RunMinorStatus, LogTypeRef, StageStatusRef, and LinkTypeRef views to access the meanings of the codes that are stored in the tables that use the enumerations.

Table 24. MasterRef table

Column name	Column type code	Description
Enumeration	20	The name of the enumeration.
Code	3	The 3-character code for the enumeration. These strings are stored in the tables that use the enumerations.
Name	32	The long name of the code.
Description	255	The description of the code.

Schema version

The MasterRef table also contains a record showing the version of the operations database schema that has been installed:

Table 25. Error codes

Enumeration name	Code	Name	Description
Version	SCH	1	Schema version from YYYY-MM-DD

You can extract the schema version by executing the following SQL query:

```
SELECT Name
  FROM MasterRef
 WHERE Enumeration = 'Version'
    AND Code = 'SCH';
```

Error codes

The enumerations also return two error codes if unexpected values are generated by job run events:

Table 26. Error codes

Code	Name	Description
?!?	Invalid value	A non-numeric value was found
!!!	Unknown value	An out-of-range numeric value was found

JobTypeRef view

The JobTypeRef view enumerates the JobType values used in the JobExec table.

Table 27. JobTypeRef view

JobTypeCode	JobTypeName	JobTypeDescription
SRV	Server	Server job
PAR	Parallel	Parallel job
SEQ	Sequence	Job sequence

RunTypeRef view

The RunTypeRef view enumerates the RunType values used in the JobRun table.

Table 28. RunTypeRef view

RunTypeCode	RunTypeName	RunTypeDescription
RUN	Run	Normal run mode
VAL	Validate	Validation-only run
RES	Reset	Reset run

RunMajorStatus view

The RunMajorStatus view enumerates the RunMajorStatus values used in the JobRun table.

Table 29. RunMajorStatus view

MajorStatusCode	MajorStatusName	MajorStatusDescription
SUB	Submitted	Run submitted but not yet started
STA	Started	Run has started
FIN	Finished	Run has finished

RunMinorStatus view

The RunMinorStatus view enumerates the RunMinorStatus values used in the JobRun table.

Table 30. RunMinorStatus view

MinorStatusCode	MinorStatusName	MinorStatusDescription
QUE	Submitted - queued	Run is waiting in a queue (MajorStatusCode = SUB)
STR	Submitted - starting	Run is about to start (MajorStatusCode = SUB)
RUN	Running - no warnings	Running, and has not yet logged any warning or fatal messages (MajorStatusCode = RUN)
RNW	Running - with warnings	Running, and has logged at least one warning message but no fatals (MajorStatusCode = RUN)

Table 30. RunMinorStatus view (continued)

MinorStatusCode	MinorStatusName	MinorStatusDescription
RNF	Running - with fatals	Running, and has logged at least one fatal message (MajorStatusCode = RUN)
RNS	Running - stop requested	Running, but has received an external stop request (MajorStatusCode = RUN)
FOK	Finished - OK	Run has finished without logging any warning or fatal messages (MajorStatusCode = FIN)
FWW	Finished - with warnings	Run has finished and logged at least one warning message but no fatals (MajorStatusCode = FIN)
FWF	Finished - aborted	Run has finished and logged at least one fatal message (MajorStatusCode = FIN)
CRA	Finished - crashed	Run finished unexpectedly (MajorStatusCode = FIN)
STP	Finished - stopped by request	Run terminated due to external stop request (MajorStatusCode = FIN)
SYN	Finished - synchronized	Run forcibly set to finished state by maintenance check (MajorStatusCode = FIN)

LogTypeRef view

The LogTypeRef view enumerates the LogType values used in the JobRunLog table.

Table 31. LogTypeRef view

LogTypeCode	LogTypeName	LogTypeDescription
INF	Info	Informational message only
WAR	Warning	Warning message
FAT	Fatal	Fatal message - job will be aborted
REJ	Reject	Stage reject message
CTL	Control	Job started/finished
PUR	Purge	Previous messages in the log file have been purged
RUN	RunJob	Job run started under control of sequence, or returning to caller

StageStatusRef view

The StageStatusRef view enumerates the StageStatus values used in the JobRunStage table.

Table 32. StageStatusRef view

StageStatusCode	StageStatusName	StageStatusDescription
RDY	Ready	Stage is ready to run but has not yet started, or is a Server job passive stage (source/target) which is always in this state
RUN	Running	Stage is running
FIN	Finished	Stage has finished

LinkTypeRef view

The LinkTypeRef view enumerates the LinkType values used in the JobLink table.

Table 33. LinkTypeRef view

LinkTypeCode	LinkTypeName	LinkTypeDescription
PRI	Primary	Primary link between two stages
REF	Reference	Reference input to a stage
REJ	Reject	Reject output from a stage

Job run time-series data tables and views

The job run time-series data tables contain information about the progress of IBM InfoSphere DataStage and QualityStage job runs over the period of their execution.

JobRunUsage table

The JobRunUsage table contains snapshots, stored as XML fragments, of the number of rows consumed and produced by job runs over the period of their execution.

The number of snapshots in each row is limited so that the size of the row does not exceed a fixed maximum. This limit can be specified in the configuration files. When a snapshot is taken, it is either added to the current row, or, if the maximum size of the row would be exceeded, it is inserted as a new row.

Table 34. JobRunUsage table

Column name	Column type code	Description
StartTimestamp	Tim	The time when the first snapshot in the row is recorded.
RUNID	SK	The foreign key to the job run in the JobRun table that this set of snapshots belongs to. The combination of the RUNID and the StartTimestamp columns are a unique key for the table.

Table 34. JobRunUsage table (continued)

Column name	Column type code	Description
EndTimeStamp	Tim	The time that the last snapshot in the row was recorded.
ResourceInfo	Txt	An XML fragment containing the snapshot data. Use the JobRunTotalRowUsage view to extract this information.

JobRunTotalRowUsage view

The JobRunTotalRowUsage view expands the ResourceInfo column of the JobRunUsage table. The JobRunTotalRowUsage view shows the number of rows consumed and produced by each snapshot in the JobRunUsage table.

Table 35. JobRunTotalRowUsage view

Column name	Column type code	Description
RUNID	SK	The foreign key to the job run in the JobRun table that this set of snapshots belongs to. The combination of the RUNID and the StartTimeStamp columns are a unique key for the table.
StartTimeStamp	Tim	The time when the first snapshot in the row was recorded.
EndTimeStamp	Tim	The time when the last snapshot in the row was recorded.
RunElapsedSecs	Int	The time, in seconds, since the start of the run.
TotalRowsConsumed	Big	The total of the row counts from all the source stage links at the time of the snapshot.
TotalRowsProduced	Big	Total of the row counts from all the target stage links at the time of the snapshot.

System resource usage tables and views

The system resource usage tables contain information about the system resource usage for all engine systems and the compute nodes they use to deploy parallel jobs.

HostDetail table

The HostDetail table contains information about the available system resources of every engine system and compute node in the Host table.

Table 36. HostDetail table

Column name	Column type code	Description
CreatedTimestamp	Tim	The time that this row was created.
HOSTED	SK	The foreign key to the host system in the Host table.
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
LastCheckedTimestamp	Tim	The time that the host system last reported this information and it was found to be unchanged. The LastCheckedTimestamp column is initially set to the same value as the CreatedTimestamp column.
UTCOffsetMins	Int	The offset, in minutes, between the current timezone and Coordinated Universal Time (UTC).
TimezoneName	80	The name of the timezone at the time of the LastCheckedTimestamp.
PlatformName	80	The descriptive name of the operating system .
PlatformVersion	80	The version string of the operating system.
NumCPUs	Int	The number of CPUs on the host system.
CPUModel	80	The descriptive name of the hardware.
PhysicalMemoryKB	Big	The size, in kilobytes, of the physical memory of the host system.
VirtualMemoryKB	Big	The size, in kilobytes, of the virtual memory of the host system.

The state of a host at a specific time can be found by extracting the row where the CreatedTimestamp immediately precedes that time.

Note: If the LastCheckedTimestamp is earlier than the Host.MonStartTimestamp for the host pointed to by the HEAD_HOSTID column, the host has not reported its details since the EngMonApp process was started. In this case, the host details are unreliable.

Current system resource usage tables and views

The current system resource usage tables contain information about the current system resource usage for all engine systems and the compute nodes they use to deploy parallel jobs.

ResourceSnap table

The ResourceSnap table contains information about the current system resource usage of every system being monitored.

Use the ResourceSnapSystem view and the ResourceSnapDisks view to extract information from the ResourceSnap table.

Table 37. ResourceSnap table

Column name	Column type code	Description
HOSTID	SK	The foreign key to the host system in the Host table.
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
LastUpdateTimestamp	Tim	The time when this row was last updated. Note: If the LastUpdatedTimestamp is earlier than the Host.MonStartTimestamp for the host pointed to by the HEAD_HOSTID column, this host is no longer being monitored.
CPUPctUser	4,1	The percentage of CPU time spent in user mode.
CPUPctPrivileged	4,1	The percentage of CPU time spent in system mode.
CPUPctIdle	4,1	The percentage of CPU time spent idle.
CPUPctStolen	4,1	The percentage of CPU time stolen (Linux only).
CPUPctIOWait	4,1	The percentage of CPU time spent waiting for I/O (not on Windows).
ProcNumRunning	Int	The number of running processes (not on Windows).
ProcNumWaiting	Int	The number of processes waiting on a resource (not on Windows).

Table 37. ResourceSnap table (continued)

Column name	Column type code	Description
ProcNumSleeping	Int	The number of processes sleeping or waiting on an event (not on Windows).
ProcNumZombied	Int	The number of zombie processes (not on Windows).
ProcNumBlocked	Int	The number of processes that are blocked or stopped due to a signal (not on Windows).
MemFreeKBPhysical	Int	The size of the unused physical RAM in kilobytes.
MemFreeKBVirtual	Int	The size of the unused swap memory in kilobytes.
PageNumIn	Int	The number of pages paged in per second.
PageNumOut	Int	The number of pages paged out per second.
PageNumInterrupts	Int	The number of interrupts serviced per second.
PageNumSwitches	Int	The number of context switches per second.
DiskSnap	XML	An XML fragment containing the disk usage snapshot data. Use the ResourceSnapDisks view to extract this information.

Note: Not all the columns have values set for host systems on all platforms; columns that are not set contain nulls.

On Windows, these columns are not set: CPUPctStolen, CPUPctIOWait, ProcNumWaiting, ProcNumSleeping, ProcNumZombied, and ProcNumBlocked.

The CPUPctStolen column is only set on Linux.

ResourceSnapSystem view

The ResourceSnapSystem view shows information about the current system resource usage of every system in the ResourceSnap table.

Table 38. ResourceSnapSystem view

Column name	Column type code	Description
HOSTID	SK	The foreign key to the host system in the Host table.

Table 38. ResourceSnapSystem view (continued)

Column name	Column type code	Description
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
LastUpdateTimestamp	Tim	The time when this row was last updated. Note: If the LastUpdatedTimestamp is earlier than the Host.MonStartTimestamp for the host pointed to by the HEAD_HOSTID column, this host is no longer being monitored.
CPUPctUser	4,1	The percentage of CPU time spent in user mode.
CPUPctPrivileged	4,1	The percentage of CPU time spent in system mode.
CPUPctIdle	4,1	The percentage of CPU time spent idle.
CPUPctStolen	4,1	The percentage of CPU time stolen (Linux only).
CPUPctIOWait	4,1	The percentage of CPU time spent waiting for I/O (not on Windows).
ProcNumRunning	Int	The number of running processes (not on Windows).
ProcNumWaiting	Int	The number of processes waiting on a resource (not on Windows).
ProcNumSleeping	Int	The number of processes sleeping or waiting on an event (not on Windows).
ProcNumZombied	Int	The number of zombie processes (not on Windows).
ProcNumBlocked	Int	The number of processes that are blocked or stopped due to a signal (not on Windows).
MemFreeKBPhysical	Int	The size of the unused physical RAM in kilobytes.
MemFreeKBVirtual	Int	The size of the unused swap memory in kilobytes.
PageNumIn	Int	The number of pages paged in per second.

Table 38. ResourceSnapSystem view (continued)

Column name	Column type code	Description
PageNumOut	Int	The number of pages paged out per second.
PageNumInterrupts	Int	The number of interrupts serviced per second.
PageNumSwitches	Int	The number of context switches per second.

Note: Not all the columns have values set for host systems on all platforms; columns that are not set contain nulls.

On Windows, these columns are not set: CPUPctStolen, CPUPctIOWait, ProcNumWaiting, ProcNumSleeping, ProcNumZombied, and ProcNumBlocked.

The CPUPctStolen column is only set on Linux.

ResourceSnapDisks view

The ResourceSnapDisks view shows information about the current file system usage of every system in the ResourceSnap table.

Table 39. ResourceSnapDisks view

Column name	Column type code	Description
HOSTID	SK	The foreign key to the host system in the Host table.
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
LastUpdateTimestamp	Tim	The time when this row was last updated. Note: If the LastUpdatedTimestamp is earlier than the Host.MonStartTimestamp for the host pointed to by the HEAD_HOSTID column, this host is no longer being monitored.
DiskPathMonitored	255	File system path of disk being monitored, as specified in the configuration file.
DiskTotalKB	Big	The number of kilobytes on the disk.
DiskFreeKB	Big	The number of free kilobytes on the disk.

Past system resource usage tables and views

The past system resource usage tables contain information about the past system resource usage for all engine systems and the compute nodes they use to deploy parallel jobs.

ResourceUsage table

The ResourceUsage table contains information about the past system resource usage of every system being monitored.

The values held in the rows of the ResourceUsage table are derived from successive updates to the ResourceSnap table for the same combination of the HOSTID and HEAD_HOSTID columns. As the ResourceSnap table is updated, the values involved are aggregated for a configured number of updates and then a row is inserted in the ResourceUsage table for the time period covered by the set of samples. For each data point collected in the ResourceSnap table, a maximum, a minimum, and an average value are calculated for the set of samples, and these aggregate values are stored in the ResourceUsage table.

Use the ResourceUsageSystem view and the ResourceUsageDisks view to extract information from the ResourceUsage table.

Table 40. ResourceUsage table

Column name	Column type code	Description
HOSTID	SK	The foreign key to the host system in the Host table.
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
StartTimestamp	Tim	The start of the sample time period this row relates to.
EndTimestamp	Tim	The end of the sample time period this row relates to. Note: If the EndTimestamp is earlier than the MonStartTimestamp for the host pointed to by the HEAD_HOSTID column, this host is no longer being monitored.
NumSamples	Int	The number of samples included in this aggregate row.
CPUPctUserAvg	4,1	The average percentage of CPU time spent in user mode.
CPUPctUserMax	4,1	The maximum percentage of CPU time spent in user mode.

Table 40. ResourceUsage table (continued)

Column name	Column type code	Description
CPUPctUserMin	4,1	The minimum percentage of CPU time spent in user mode.
CPUPctPrivilegedAvg	4,1	The average percentage of CPU time spent in system mode.
CPUPctPrivilegedMax	4,1	The maximum percentage of CPU time spent in system mode.
CPUPctPrivilegedMin	4,1	The minimum percentage of CPU time spent in system mode.
CPUPctIdleAvg	4,1	The average percentage of CPU time spent idle.
CPUPctIdleMax	4,1	The maximum percentage of CPU time spent idle.
CPUPctIdleMin	4,1	The minimum percentage of CPU time spent idle.
CPUPctStolenAvg	4,1	The average percentage of CPU time stolen (Linux only).
CPUPctStolenMax	4,1	The maximum percentage of CPU time stolen (Linux only).
CPUPctStolenMin	4,1	The minimum percentage of CPU time stolen (Linux only).
CPUPctIOWaitAvg	4,1	The average percentage of CPU time spent waiting for I/O (not on Windows).
CPUPctIOWaitMax	4,1	The maximum percentage of CPU time spent waiting for I/O (not on Windows).
CPUPctIOWaitMin	4,1	The minimum percentage of CPU time spent waiting for I/O (not on Windows).
ProcNumRunningAvg	Int	The average number of running processes (not on Windows).
ProcNumRunningMax	Int	The maximum number of running processes (not on Windows).
ProcNumRunningMin	Int	The minimum number of running processes (not on Windows).
ProcNumWaitingAvg	Int	The average number of processes waiting on a resource (not on Windows).

Table 40. ResourceUsage table (continued)

Column name	Column type code	Description
ProcNumWaitingMax	Int	The maximum number of processes waiting on a resource (not on Windows).
ProcNumWaitingMin	Int	The minimum number of processes waiting on a resource (not on Windows).
ProcNumSleepingAvg	Int	The average number of processes sleeping or waiting on an event (not on Windows).
ProcNumSleepingMax	Int	The maximum number of processes sleeping or waiting on an event (not on Windows).
ProcNumSleepingMin	Int	The minimum number of processes sleeping or waiting on an event (not on Windows).
ProcNumZombiedAvg	Int	The average number of zombie processes (not on Windows).
ProcNumZombiedMax	Int	The maximum number of zombie processes (not on Windows).
ProcNumZombiedMin	Int	The minimum number of zombie processes (not on Windows).
ProcNumBlockedAvg	Int	The average number of processes that are blocked or stopped due to a signal (not on Windows).
ProcNumBlockedMax	Int	The maximum number of processes that are blocked or stopped due to a signal (not on Windows).
ProcNumBlockedMin	Int	The minimum number of processes that are blocked or stopped due to a signal (not on Windows).
MemFreeKBPhysicalAvg	Big	The average number of kilobytes of unused physical RAM.
MemFreeKBPhysicalMax	Big	The maximum number of kilobytes of unused physical RAM.
MemFreeKBPhysicalMin	Big	The minimum number of kilobytes of unused physical RAM.
MemFreeKBVirtualAvg	Big	The average number of kilobytes of unused swap memory.

Table 40. ResourceUsage table (continued)

Column name	Column type code	Description
MemFreeKBVirtualMax	Big	The maximum number of kilobytes of unused swap memory.
MemFreeKBVirtualMin	Big	The minimum number of kilobytes of unused swap memory.
PageNumInAvg	Int	The average number of pages paged in per second.
PageNumInMax	Int	The maximum number of pages paged in per second.
PageNumInMin	Int	The minimum number of pages paged in per second.
PageNumOutAvg	Int	The average number of pages paged out per second.
PageNumOutMax	Int	The maximum number of pages paged out per second.
PageNumOutMin	Int	The minimum number of pages paged out per second.
PageNumInterruptsAvg	Int	The average number of interrupts serviced per second.
PageNumInterruptsMax	Int	The maximum number of interrupts serviced per second.
PageNumInterruptsMin	Int	The minimum number of interrupts serviced per second.
PageNumSwitchesAvg	Int	The average number of context switches per second.
PageNumSwitchesMax	Int	The maximum number of context switches per second.
PageNumSwitchesMin	Int	The minimum number of context switches per second.
DiskUsage	XML	An XML fragment containing the disk usage data. Use the ResourceUsageDisks view to extract this information.

Note: Not all the columns have values set for host systems on all platforms; columns that are not set contain nulls.

On Windows, these columns are not set: CPUPctStolen, CPUPctIOWait, ProcNumWaiting, ProcNumSleeping, ProcNumZombied, and ProcNumBlocked.

The CPUPctStolen columns are only set on Linux.

ResourceUsageSystem view

The ResourceUsageSystem view shows information about the past system resource usage of every system in the ResourceUsage table.

Table 41. ResourceUsageSystem view

Column name	Column type code	Description
HOSTID	SK	The foreign key to the host system in the Host table.
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
StartTimestamp	Tim	The start of the sample time period this row relates to.
EndTimestamp	Tim	The end of the sample time period this row relates to. Note: If the EndTimestamp is earlier than the MonStartTimestamp for the host pointed to by the HEAD_HOSTID column, this host is no longer being monitored.
NumSamples	Int	The number of samples included in this aggregate row.
CPUPctUserAvg	4,1	The average percentage of CPU time spent in user mode.
CPUPctUserMax	4,1	The maximum percentage of CPU time spent in user mode.
CPUPctUserMin	4,1	The minimum percentage of CPU time spent in user mode.
CPUPctPrivilegedAvg	4,1	The average percentage of CPU time spent in system mode.
CPUPctPrivilegedMax	4,1	The maximum percentage of CPU time spent in system mode.
CPUPctPrivilegedMin	4,1	The minimum percentage of CPU time spent in system mode.
CPUPctIdleAvg	4,1	The average percentage of CPU time spent idle.
CPUPctIdleMax	4,1	The maximum percentage of CPU time spent idle.
CPUPctIdleMin	4,1	The minimum percentage of CPU time spent idle.

Table 41. ResourceUsageSystem view (continued)

Column name	Column type code	Description
CPUPctStolenAvg	4,1	The average percentage of CPU time stolen (Linux only).
CPUPctStolenMax	4,1	The maximum percentage of CPU time stolen (Linux only).
CPUPctStolenMin	4,1	The minimum percentage of CPU time stolen (Linux only).
CPUPctIOWaitAvg	4,1	The average percentage of CPU time spent waiting for I/O (not on Windows).
CPUPctIOWaitMax	4,1	The maximum percentage of CPU time spent waiting for I/O (not on Windows).
CPUPctIOWaitMin	4,1	The minimum percentage of CPU time spent waiting for I/O (not on Windows).
ProcNumRunningAvg	Int	The average number of running processes (not on Windows).
ProcNumRunningMax	Int	The maximum number of running processes (not on Windows).
ProcNumRunningMin	Int	The minimum number of running processes (not on Windows).
ProcNumWaitingAvg	Int	The average number of processes waiting on a resource (not on Windows).
ProcNumWaitingMax	Int	The maximum number of processes waiting on a resource (not on Windows).
ProcNumWaitingMin	Int	The minimum number of processes waiting on a resource (not on Windows).
ProcNumSleepingAvg	Int	The average number of processes sleeping or waiting on an event (not on Windows).
ProcNumSleepingMax	Int	The maximum number of processes sleeping or waiting on an event (not on Windows).
ProcNumSleepingMin	Int	The minimum number of processes sleeping or waiting on an event (not on Windows).
ProcNumZombiedAvg	Int	The average number of zombie processes (not on Windows).

Table 41. ResourceUsageSystem view (continued)

Column name	Column type code	Description
ProcNumZombiedMax	Int	The maximum number of zombie processes (not on Windows).
ProcNumZombiedMin	Int	The minimum number of zombie processes (not on Windows).
ProcNumBlockedAvg	Int	The average number of processes that are blocked or stopped due to a signal (not on Windows).
ProcNumBlockedMax	Int	The maximum number of processes that are blocked or stopped due to a signal (not on Windows).
ProcNumBlockedMin	Int	The minimum number of processes that are blocked or stopped due to a signal (not on Windows).
MemFreeKBPhysicalAvg	Big	The average number of kilobytes of unused physical RAM.
MemFreeKBPhysicalMax	Big	The maximum number of kilobytes of unused physical RAM.
MemFreeKBPhysicalMin	Big	The minimum number of kilobytes of unused physical RAM.
MemFreeKBVirtualAvg	Big	The average number of kilobytes of unused swap memory.
MemFreeKBVirtualMax	Big	The maximum number of kilobytes of unused swap memory.
MemFreeKBVirtualMin	Big	The minimum number of kilobytes of unused swap memory.
PageNumInAvg	Int	The average number of pages paged in per second.
PageNumInMax	Int	The maximum number of pages paged in per second.
PageNumInMin	Int	The minimum number of pages paged in per second.
PageNumOutAvg	Int	The average number of pages paged out per second.
PageNumOutMax	Int	The maximum number of pages paged out per second.
PageNumOutMin	Int	The minimum number of pages paged out per second.

Table 41. ResourceUsageSystem view (continued)

Column name	Column type code	Description
PageNumInterruptsAvg	Int	The average number of interrupts serviced per second.
PageNumInterruptsMax	Int	The maximum number of interrupts serviced per second.
PageNumInterruptsMin	Int	The minimum number of interrupts serviced per second.
PageNumSwitchesAvg	Int	The average number of context switches per second.
PageNumSwitchesMax	Int	The maximum number of context switches per second.
PageNumSwitchesMin	Int	The minimum number of context switches per second.

Note: Not all the columns have values set for host systems on all platforms; columns that are not set contain nulls.

On Windows, these columns are not set: CPUPctStolen, CPUPctIOWait, ProcNumWaiting, ProcNumSleeping, ProcNumZombied, and ProcNumBlocked.

The CPUPctStolen columns are only set on Linux.

ResourceUsageDisks view

The ResourceUsageDisks view shows information about the past file system usage of every system in the ResourceSnap table.

Table 42. ResourceUsageDisks view

Column name	Column type code	Description
HOSTID	SK	The foreign key to the host system in the Host table.
HEAD_HOSTID	SK	The foreign key to the host system in the Host table that generated this row. For an engine system the HOSTID and the HEAD_HOSTID are the same. For a compute node, HEAD_HOSTID points to the conductor node.
StartTimestamp	Tim	The start of the sample time period this row relates to.

Table 42. ResourceUsageDisks view (continued)

Column name	Column type code	Description
EndTimeStamp	Tim	The end of the sample time period this row relates to. Note: If the EndTimeStamp is earlier than the MonStartTimeStamp for the host pointed to by the HEAD_HOSTID column, this host is no longer being monitored.
NumSamples	Int	The number of samples included in this aggregate row.
DiskPathMonitored	255	The file system path of disk being monitored.
DiskTotalKB	Big	The number of kilobytes on the disk.
DiskFreeKBAvg	Big	The average number of free kilobytes on the disk.
DiskFreeKBMax	Big	The maximum number of free kilobytes on the disk.
DiskFreeKBMin	Big	The minimum number of free kilobytes on the disk.

Appendix A. Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere Information Server product modules and user interfaces are not fully accessible.

For information about the accessibility status of IBM products, see the IBM product accessibility information at http://www.ibm.com/able/product_accessibility/index.html.

Accessible documentation

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most web browsers. Because the information center uses XHTML, you can set display preferences in your browser. This also allows you to use screen readers and other assistive technologies to access the documentation.

The documentation that is in the information center is also provided in PDF files, which are not fully accessible.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Appendix B. Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

Table 43. IBM resources

Resource	Description and location
IBM Support Portal	You can customize support information by choosing the products and the topics that interest you at www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server
Software services	You can find information about software, IT, and business consulting services, on the solutions site at www.ibm.com/businesssolutions/
My IBM	You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at www.ibm.com/account/
Training and certification	You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at http://www.ibm.com/training
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/

Appendix C. Accessing the product documentation

Documentation is provided in a variety of formats: in the online IBM Knowledge Center, in an optional locally installed information center, and as PDF books. You can access the online or locally installed help directly from the product client interfaces.

IBM Knowledge Center is the best place to find the most up-to-date information for InfoSphere Information Server. IBM Knowledge Center contains help for most of the product interfaces, as well as complete documentation for all the product modules in the suite. You can open IBM Knowledge Center from the installed product or from a web browser.

Accessing IBM Knowledge Center

There are various ways to access the online documentation:

- Click the **Help** link in the upper right of the client interface.
- Press the F1 key. The F1 key typically opens the topic that describes the current context of the client interface.

Note: The F1 key does not work in web clients.

- Type the address in a web browser, for example, when you are not logged in to the product.

Enter the following address to access all versions of InfoSphere Information Server documentation:

```
http://www.ibm.com/support/knowledgecenter/SSZJPZ/
```

If you want to access a particular topic, specify the version number with the product identifier, the documentation plug-in name, and the topic path in the URL. For example, the URL for the 11.3 version of this topic is as follows. (The ⇒ symbol indicates a line continuation):

```
http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/⇒  
com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html
```

Tip:

The knowledge center has a short URL as well:

```
http://ibm.biz/knowctr
```

To specify a short URL to a specific product page, version, or topic, use a hash character (#) between the short URL and the product identifier. For example, the short URL to all the InfoSphere Information Server documentation is the following URL:

```
http://ibm.biz/knowctr#SSZJPZ/
```

And, the short URL to the topic above to create a slightly shorter URL is the following URL (The ⇒ symbol indicates a line continuation):

```
http://ibm.biz/knowctr#SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/⇒  
common/accessingiidoc.html
```

Changing help links to refer to locally installed documentation

IBM Knowledge Center contains the most up-to-date version of the documentation. However, you can install a local version of the documentation as an information center and configure your help links to point to it. A local information center is useful if your enterprise does not provide access to the internet.

Use the installation instructions that come with the information center installation package to install it on the computer of your choice. After you install and start the information center, you can use the **iisAdmin** command on the services tier computer to change the documentation location that the product F1 and help links refer to. (The `⇒` symbol indicates a line continuation):

Windows

```
IS_install_path\ASBServer\bin\iisAdmin.bat -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

AIX® Linux

```
IS_install_path/ASBServer/bin/iisAdmin.sh -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

Where `<host>` is the name of the computer where the information center is installed and `<port>` is the port number for the information center. The default port number is 8888. For example, on a computer named `server1.example.com` that uses the default port, the URL value would be `http://server1.example.com:8888/help/topic/`.

Obtaining PDF and hardcopy documentation

- The PDF file books are available online and can be accessed from this support document: <https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1>.
- You can also order IBM publications in hardcopy format online or through your local IBM representative. To order publications online, go to the IBM Publications Center at <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.

Appendix D. Providing feedback on the product documentation

You can provide helpful feedback regarding IBM documentation.

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- To provide a comment about a topic in IBM Knowledge Center that is hosted on the IBM website, sign in and add a comment by clicking **Add Comment** button at the bottom of the topic. Comments submitted this way are viewable by the public.
- To send a comment about the topic in IBM Knowledge Center to IBM that is not viewable by anyone else, sign in and click the **Feedback** link at the bottom of IBM Knowledge Center.
- Send your comments by using the online readers' comment form at www.ibm.com/software/awdtools/rcf/.
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, include the location of the text (for example, a title, a table number, or a page number).

Notices and trademarks

This information was developed for products and services offered in the U.S.A. This material may 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.

Notices

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
Armonk, NY 10504-1785 U.S.A.

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

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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 Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 the names and addresses used by an actual business enterprise 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session or persistent cookies. If a product or component is not listed, that product or component does not use cookies.

Table 44. Use of cookies by InfoSphere Information Server products and components

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
Any (part of InfoSphere Information Server installation)	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
Any (part of InfoSphere Information Server installation)	InfoSphere Metadata Asset Manager	<ul style="list-style-type: none"> • Session • Persistent 	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication • Enhanced user usability • Single sign-on configuration 	Cannot be disabled

Table 44. Use of cookies by InfoSphere Information Server products and components (continued)

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
InfoSphere DataStage	Big Data File stage	<ul style="list-style-type: none"> • Session • Persistent 	<ul style="list-style-type: none"> • User name • Digital signature • Session ID 	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere DataStage	XML stage	Session	Internal identifiers	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere DataStage	IBM InfoSphere DataStage and QualityStage Operations Console	Session	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Data Click	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Data Quality Console		Session	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere QualityStage Standardization Rules Designer	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Information Governance Catalog		<ul style="list-style-type: none"> • Session • Persistent 	<ul style="list-style-type: none"> • User name • Internal identifiers • State of the tree 	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere Information Analyzer	Data Rules stage in the InfoSphere DataStage and QualityStage Designer client	Session	Session ID	Session management	Cannot be disabled

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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 www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

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

The United States Postal Service owns the following trademarks: CASS, CASS Certified, DPV, LACS^{Link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS^{Link} licensee of the United States Postal Service.

Other company, product or service names may be trademarks or service marks of others.

Index

A

- administering
 - workload management 1
- AppWatcher
 - starting 35
 - starting Linux 37
 - starting UNIX 37
 - stopping 35
 - Linux 37
 - UNIX 37
- AppWatcher process
 - displaying status 38
 - starting 33
 - stopping 33

C

- configuration file 22
- configuration options 22
- configuring
 - InfoSphere Information Server 1
 - workload management 2
- configuring options 3
- customer support
 - contacting 107

D

- data collection process
 - DSAppWatcher.sch 37
 - errors
 - troubleshooting 41
 - managing 37
- data collection processes
 - EngMonApp 34
 - managing 33
 - ODBQueryApp 34
 - ResMonApp 34
 - ResTrackApp 34
- data monitoring options 21
- database content
 - reporting 62
- default port numbers
 - configuration options 31
- Details panel
 - batch jobs 52
 - sequence jobs 52
- DSAppWatch.sch script
 - parameters 37
- DSAppWatcher.sh script 37
- DSAppWatcher.sh script 37, 38, 39, 40
- DSOBCfg.cfg 21
- DSOBCfg.cfg 21, 22, 31
- DSODTuning.cfg 22

E

- enabling
 - job monitoring 5

- enabling (*continued*)
 - monitoring
 - engine system 5
- engine system
 - configuring 17
- engines
 - monitoring 44
- EngMonApp 41
- EngMonApp fatal errors
 - troubleshooting 42
- EngMonApp.err file 42

F

- failed job runs
 - troubleshooting 44

G

- grid environment
 - configuring 22

H

- historical data
 - archiving 61
 - deleting 61, 63
 - deleting using istool ODBAdmin 64
 - deleting using SQL queries 66

I

- istool ODBAdmin command 61
 - countdb parameter 62
 - purgedb 64

J

- job run performance
 - improving 44
- job run usage
 - configuration options 26
- job runs
 - configuration options 23
 - monitoring 43
- job runtime errors 43
 - event files 43
- JobRuntime.err file 43

L

- legal notices 113

M

- monitoring
 - job runs 5
 - services 5

- monitoring (*continued*)
 - system resources 5
 - workload management 5
- monitoring data
 - configuration options 22
 - customizing 21
 - extracting 58
- monitoring jobs 5
 - tasks 6
- monitoring system configuration
 - checking 40
- MPP
 - configuring 22

N

- not running processes
 - staring 40

O

- operating database schema
 - creating
 - Linux 13
 - Oracle database 13
 - UNIX 13
- Operations Console 2, 50
 - accessing information 43
 - Activity panel 50
 - Activity tab 54
 - configuring panels 47
 - Dashboard tab 50
 - Details panel 52
 - Engine Status bar 57
 - Engine Status panel 51
 - Environment Variables panel 53
 - event history 48
 - Job Activity tab
 - Resources tab 55
 - Job Run dialog 52
 - Job Runs tab 54
 - Jobs Designs panel 54
 - Jobs Runs panel 54
 - monitoring jobs 43
 - Operating System Resources
 - panel 51
 - overview 43
 - project scope 48
 - Projects tab 51
 - Queue Management panel 57
 - Queued Jobs panel 56
 - scenarios 44
 - selecting engines 48
 - setting refresh rate 49
 - setting scope 48
 - Activity tab 49
 - starting 46
 - System Policies panel 55
 - system requirements 46
 - user access 45

- Operations Console (*continued*)
 - viewing log messages 47
 - Workload Management tab 55
 - operations database
 - configuring 7
 - configuring connection 20
 - creating 7
 - deleting 71
 - generating scripts 10, 72
 - identifying 17
 - Linux 10, 72
 - registering repository
 - Linux 9, 18
 - Unix 9, 18
 - Windows 9, 18
 - Unix 10, 72
 - Windows 10, 72
 - Operations database
 - Current system resource usage tables and views 91
 - ResourceSnap table 91
 - ResourceSnapDisks view 94
 - ResourceSnapSystem view 92
 - Job run audit data tables and views 74
 - Host table 77
 - JobExec table 76
 - JobRun table 74
 - JobRunLog table 79
 - JobRunParams table 78
 - JobRunParamsView view 79
 - ParallelConfig table 80
 - ParallelConfigNodes view 80
 - Job run metrics tables and views 80
 - DataLocator table 83
 - JobLink table 82
 - JobRunLink table 83
 - JobRunStage table 81
 - JobStage table 81
 - Job run time-series data tables and views 88
 - JobRunTotalRowUsage view 89
 - JobRunUsage table 88
 - Past system resource usage tables and views 95
 - ResourceUsage table 95
 - ResourceUsageDisks view 102
 - ResourceUsageSystem view 99
 - Reference tables and views 85
 - JobTypeRef view 86
 - LinkTypeRef view 88
 - LogTypeRef view 87
 - MasterRef table 85
 - RunMajorStatus view 86
 - RunMinorStatus view 86
 - RunTypeRef view 86
 - StageStatusRef view 88
 - Schema 73
 - System resource usage tables and views 89
 - HostDetail table 90
 - operations database schema
 - creating 8, 14
 - Linux 12
 - Oracle database 12
 - UNIX 12
 - Windows 11, 12
 - operations database schema (*continued*)
 - SQL Server database 14
 - options
 - configuring 22
- P**
- panels 50
 - product accessibility
 - accessibility 105
 - product documentation
 - accessing 109
 - project monitoring data
 - configuration options 31
 - configuring 21
- R**
- remote nodes 22
 - repository connection file
 - generating 17, 20, 70
 - Linux 17, 20, 70
 - UNIX 17, 20, 70
 - Windows 17, 20, 70
 - ResMonApp 41
 - ResMonApp fatal errors
 - troubleshooting 42
 - ResMonApp.err file 42
 - Run Details window
 - configuring charts 49
- S**
- script errors
 - recovering from 14
 - software services
 - contacting 107
 - specified process
 - displaying status 39
 - starting 40
 - stopping 39
 - support
 - customer 107
 - system resource usage
 - configuration options 27
- T**
- tabs 50
 - trademarks
 - list of 113
 - troubleshooting options
 - configuring 22, 32
 - tuning options
 - configuring 22, 32
- U**
- user password
 - changing 68
- W**
- WLMServer 4
- workload management
 - configuration options 3
 - predefined queues 1
 - queues 1
 - system policies 1
 - workload management process
 - starting 4
 - stopping 4
 - workload management queues 1
 - workload management system
 - defining 2
 - monitoring 2



Printed in USA

SC19-4287-00

