IBM InfoSphere DataStage
Version 8 Release 7

Operations Console Guide and Reference

SC19-3609-01

IBM InfoSphere DataStage
Version 8 Release 7

Operations Console Guide and Reference

SC19-3609-01

Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 99.

Contents

Chapter 1. Monitoring jobs

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

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 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 provide information from the operations database to the Operations Console.
- The Operations Console, which displays information about jobs, job activity, and system resources for each engine system.

To enable monitoring 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 these steps have not been performed, the engine tier is available for use, but monitoring is not enabled.

Getting started with the Operations Console

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

Procedure

1. Create the operations database schema. See "Creating the operations database schema" on page 3.
2. Configure the connection to the operations database. See "Configuring the connection" on page 10.
3. 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 13.
4. 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 33.
5. Start the DSAppWatcher process that starts and manages the data collection processes. See "Starting and stopping the AppWatcher process" on page 26.
6. Check that the data collection processes are running. See "Displaying the status of the AppWatcher process and all its managed processes" on page 31.
7. Start the Operations Console. See "Starting the Operations Console" on page 38.
8. 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 Operations Console to monitor IBM InfoSphere DataStage and QualityStage jobs.

Operations database

The operations database stores the monitoring data that is displayed by the IBM InfoSphere DataStage 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 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.

Procedure

1. Run the command that generates the scripts that you use to create the database and database tables and to grant a user the appropriate permissions. The command also generates a script that you can use to undo those operations, if required. This step can be run by any user with access rights to the current directory.
2. 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. You might need to copy the generated scripts to a directory that the user has access to, and from which the user can execute the command shell for the database system.

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 Operations Console.

About this task

You can choose to create the operations database schema within an existing database instance, or you can create a new database instance to hold the schema and tables that are required. In either case, you must select a user name that will be granted read and write access to the tables. This user must be an existing operating system user, and cannot be the same one that you use to execute the **db2** command during the schema creation process.

Procedure

1. Switch to the appropriate directory, either `\InformationServer\Server\DSODB\scripts\DB2_LUW_9_5` or `\InformationServer\Server\DSODB\scripts\DB2_LUW_9_7`.
2. Run the **generate_ddl.cmd** command to generate the required DB2 scripts.

   To create the schema in a new database instance, enter this command:

   ```
   generate_ddl.cmd DatabaseName DatafilePath UserName new
   ```

   To create the schema within an existing database instance, enter this command:

```
generate_ddl.cmd DatabaseName UserName existing
```

where

- **DatabaseName** is the name of the database to create or add to.
- **DatafilePath** is the location where the table space data files are created. This location is normally the path to the root where DB2 is installed. Do not add a trailing slash or backslash to the path name. For example, if DB2 is installed in `C:\DB2`, specify `C:\` as the **DatafilePath**.

  If you want to specify a path that is not the root where DB2 is installed, you must enable the DB2_CREATE_DB_ON_PATH environment variable by issuing this command:

  ```
  db2set DB2_CREATE_DB_ON_PATH=YES
  ```

  If you do not issue this command, this error message might be issued: `The database path "DatafilePath" does not exist.`
- **UserName** is the name of an Windows user that you intend to configure to access the operations database. This user must be a different user than the one used to execute the setup scripts in the next step.

For example, to create the schema in a new database that is called `dsodbdb` that is configured by the user `dsodbuser`, enter this command:

```
generate_ddl.cmd dsodbdb c:\datapath dsodbuser new
```

For example, to create the schema in the metadata repository database that is configured by the user `dsodbuser`, enter this command:

```
generate_ddl.cmd xmeta dsodbuser existing
```

The **generate_ddl.cmd** command generates three DB2 scripts:

DSODB_setup.sql
Use to create a database and table space as necessary.

DSODB_setup_user.sql
Use to grant privileges to the database user you specified

DSODB_setup_remove.sql
Use to revoke privileges and to drop tables, schema, and the database, as necessary.

3. Run the setup scripts.
   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 first step.
   b. To create the database, schema, and tables, enter the command:

   ```
   db2 -l DSODB_setup.log -stf DSODB_setup.sql
   ```

   If this command encounters a problem because an invalid parameter was specified when the DB2 scripts were generated, regenerate the DB2 scripts using new values and try again.
   c. To set up the user permissions, enter the command:

   ```
   db2 -l DSODB_setup_user.log -stf DSODB_setup_user.sql
   ```

   If this command encounters a problem because an invalid user name was specified when the DB2 scripts were generated, regenerate the scripts with a different user name and try again.
   d. Enter the database name and user name in the `\DSODB\DSODBConnect.cfg` configuration file to define the access path.

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 Operations Console.

About this task

You can choose to create the operations database schema within an existing database instance, or you can create a new database instance to hold the schema and tables that are required. In either case, you must select a user name that will be granted read and write access to the tables. This user must be an existing operating system user, and cannot be the same one that you use to execute the **db2** command during the schema creation process.

Procedure

1. Switch to the appropriate directory, either /InformationServer/Server/DSODB/scripts/DB2_LUW_9_5 or /InformationServer/Server/DSODB/scripts/DB2_LUW_9_7.
2. Run the **generate_ddl.sh** command to generate the required DB2 scripts.

   To create the schema in a new database instance, enter this command:

   ```
   generate_ddl.sh DatabaseName DatafilePath UserName new
   ```

   To create the schema within an existing database instance, enter this command:

   ```
   generate_ddl.sh DatabaseName UserName existing
   ```

   where

   - **DatabaseName** is the name of the database to create or add to.
   - **DatafilePath** is the location where the table space data files are created. This location must be an existing path to the root where DB2 is installed. Do not add a trailing slash or backslash to the path name. For example, /var/lib.
   - **UserName** is the name of an operating system user that you intend to configure to access the operations database. This user must be a different user than the one used to execute the setup scripts in the next step.

   For example, to create the schema in a new database that is called dsodbdb that is configured by the user dsodbuser, enter this command:

   ```
   generate_ddl.sh dsodbdb /var/lib dsodbuser new
   ```

   For example, to create the schema in the metadata repository database that is configured by the user dsodbuser, enter this command:

   ```
   generate_ddl.sh xmeta dsodbuser existing
   ```

   The **generate_ddl.sh** command generates three DB2 scripts:

   DSODB_setup.sql
   Use to create a database and table space as necessary.

   DSODB_setup_user.sql
   Use to grant privileges to the database user you specified

   DSODB_setup_remove.sql
   Use to revoke privileges and to drop tables, schema, and the database, as necessary.
3. Run the setup scripts.
   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 first step.

b. To create the database, schema, and tables, enter the command:

```
db2 -l DSODB_setup.log -stf DSODB_setup.sql
```

If this command encounters a problem because an invalid parameter was specified when the DB2 scripts were generated, regenerate the DB2 scripts using new values and try again.

c. To set up the user permissions, enter the command:

```
db2 -l DSODB_setup_user.log -stf DSODB_setup_user.sql
```

If this command encounters a problem because an invalid user name was specified when the DB2 scripts were generated, regenerate the scripts with a different user name and try again.

d. Enter the database name and user name in the `/DSODB/DSODBConnect.cfg` configuration file to define the access path.

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 Operations Console.

About this task

Procedure

1. Switch to the appropriate directory, either `\InformationServer\Server\DSODB\scripts\Oracle10g` or `\InformationServer\Server\DSODB\scripts\Oracle11g`.
2. Run the **generate_ddl.cmd** command to generate the required Oracle scripts.

   To create a tablespace within an existing Oracle database, enter this command:

   ```
   generate_ddl.cmd DSODBTableSpaceName DSODBDatafilePath
   ```

   where
   - **DSODBTableSpaceName** is the name of the operations database table space to create.
   - **DSODBDatafilePath** is the location where the table space data files are created. This location must be an existing path. Do not add a trailing slash or backslash to the path name. For example, `c:\datapath`.

   For example, enter this command:

   ```
   generate_ddl.cmd dsodbspace c:\datapath
   ```

   The **generate_ddl.cmd** command generates two Oracle scripts:

   DSODB_setup.sql
   Use to create a database schema and tablespace, and to grant required privileges.

   DSODB_setup_remove.sql
   Use to revoke privileges and to drop tables and database schema as necessary.
3. Run the setup scripts.

   a. 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 first step.

   b. To create the database schema and tables, and to grant the required privileges, enter the command:

```
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
    @DSODB_setup.sql <dsodbpassword>
```

For example, enter this command:

```
sqlplus system/manager@ORCL @DSODB_setup.sql test
```

An Oracle user named DSODB is created to manage the operations database table space and tables. You must set the associated password for the DSODB user by specifying the **dsodbpassword** parameter.

If this command encounters a problem because an invalid parameter was specified when the Oracle scripts were generated, regenerate the Oracle scripts using new values and try again.

c. Enter the database name and user name in the \DSODB\DSODBConnect.cfg configuration file to define the access path.

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 Operations Console.

About this task

Procedure

1. Switch to the appropriate directory, either /InformationServer/Server/DSODB/scripts/Oracle10g or /InformationServer/Server/DSODB/scripts/Oracle11g.
2. Run the **generate_ddl.sh** command to generate the required Oracle scripts.

   To create a tablespace within an existing Oracle database, enter this command:

   ```
   generate_ddl.sh DSODBTableSpaceName DSODBDatafilePath
   ```

   where

   - **DSODBTableSpaceName** is the name of the operations database table space to create.
   - **DSODBDatafilePath** is the location where the table space data files are created. This location must be an existing path. Do not add a trailing slash or backslash to the path name. For example, /u01/app/oracle/oradata/dbname.

   For example, enter this command:

   ```
   generate_ddl.cmd dsodbspace /u01/app/oracle/oradata/dbname
   ```

   The **generate_ddl.cmd** command generates two Oracle scripts:

   DSODB_setup.sql
   Use to create a database schema and tablespace, and to grant required privileges.

   DSODB_setup_remove.sql
   Use to revoke privileges and to drop tables and database schema as necessary.
3. Run the setup scripts.

   a. 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 first step.

   b. To create the database schema and tables, and to grant the required privileges, enter the command:

```
sqlplus <oracle_system_user/oracle_system_password@database_service_name>
    @DSODB_setup.sql <dsodbpassword>
```

For example, enter this command:

```
sqlplus system/manager@ORCL @DSODB_setup.sql test
```

An Oracle user named DSODB is created to manage the operations database tables pace and tables. You must set the associated password for the DSODB user by specifying the **dsodbpassword** parameter.

If this command encounters a problem because an invalid parameter was specified when the Oracle scripts were generated, regenerate the Oracle scripts using new values and try again.

c. Enter the database name and user name in the /DSODB/DSODBConnect.cfg configuration file to define the access path.

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 Operations Console.

About this task

You can choose to create the operations database schema within an existing database instance, or you can create a new database instance to hold the schema and tables that are required. In either case, you must select a user name that will be granted read and write access to the tables. This user must be an existing operating system user, and cannot be the same one that you use to execute the **sqlcmd** command during the schema creation process.

Procedure

1. Switch to the appropriate directory, either \InformationServer\Server\DSODB\scripts\SQLServer2005 or \InformationServer\Server\DSODB\scripts\SQLServer2008.
2. Run the **generate_ddl.cmd** command to generate the required Windows scripts.

   To create the schema in a new database instance, enter this command:

   ```
   generate_ddl.cmd HostServerName DatabaseName DatafilePath UserName new
   ```

   To create the schema within an existing database instance, enter this command:

   ```
   generate_ddl.cmd HostServerName DatabaseName UserName existing
   ```

   where

   - **HostServerName** is the name of the server to create the database in.
   - **DatabaseName** is the name of the database to create or add to.
   - **DatafilePath** is the location where the data and log files are created. This location must be an existing path. Do not add a trailing slash or backslash to the path name. For example, c:\datapath.
   - **UserName** is the name of an Windows user that you intend to configure to access the operations database.

   For example, to create the schema in a new database that is called dsodbdb that is configured by the user dsodbuser, enter this command:

   ```
   generate_ddl.cmd dbhostname dsodbdb c:\datapath dsodbuser new
   ```

   For example, to create the schema in the metadata repository database that is configured by the user dsodbuser, enter this command:

```
generate_ddl.cmd dbhostname dsodbdb dsodbuser existing
```

The **generate_ddl.cmd** command generates three DB2 scripts:

DSODB_setup.sql
Use to create a database and table space as necessary.

DSODB_setup_user.sql
Use to grant privileges to the database user you specified

DSODB_setup_remove.sql
Use to revoke privileges and to drop tables, schema, and the database, as necessary.

3. Run the setup scripts.
   a. Ensure that you are logged in as a user that can execute the **sqalcmd** command and that has access to the scripts that were generated in the first step.
   b. To create the database, schema, and tables, enter the command:

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

      For example, enter the command:

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

      If this command encounters a problem due to an invalid parameter being specified when the Windows scripts were generated, regenerate the Windows scripts using new values and try again.
   c. To set up the user permissions, enter the command:

      ```
      sqlcmd -b -i DSODB_setup_user.sql -v PASSWORD=<dsodbpassword> >> logfile
      ```

      For example, enter the command:

      ```
      sqlcmd -b -i DSODB_setup_user.sql -v PASSWORD=test >> c:\DSODB_setup_user.log
      ```

      If this command encounters a problem due to an invalid user name, regenerate the scripts with a different user name and try again. A new version of the removal script is also generated.
   d. Enter the database name and user name in the `\DSODB\DSODBConnect.cfg` configuration file to define the access path.

Identifying an existing operations database

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

Procedure

1. Collect this information about the existing operations database instance:

Table 1. Information to collect about the existing operations database instance

Information	Description
Database management system type	The database system for the existing operations database (DB2®, Oracle, or SQL Server). Also note which version is being used.
JDBC connection info	The host name and port number on which the database accepts JDBC connections.

Table 1. Information to collect about the existing operations database instance (continued)

Information	Description
Database user name and password	Name of the database user that must be used by the services on the engine to access the operations database tables. Also note the password for this user.

2. Configure the connection to the operations database.

Configuring the operations database

You must configure the operations database before you can use the IBM InfoSphere DataStage 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 connection

Before you can collect monitoring data, you must configure the connection to the operations database.

About this task

The connection parameters are set in the `DSODBConnect.cfg` configuration file which is stored in the `/Server/DSODB` directory in the installation directory.

Template files are provided for each supported database system. The contents of the template files are similar to this DB2 example:

```
DataStage Operations Database Connection Definition for DB2
# ============================================================
# The following parameters define the connection to the Operations Database.

# "DBTYPE" must be set to identify which database manager is in use.
# This file is for DB2. For Oracle and SQL Server databases, copy the file
# ./scripts/OracleXXX/DSODBConnect.cfg or ./scripts/SQLServerXXXX/DSODBConnect.cfg
# into the DSODB directory, then edit it as appropriate.
DBTYPE=DB2

# "Driver" is the Java class name for the JDBC driver to be used.
# (The following example illustrates a standard DB2 connection)
Driver=com.ibm.db2.jcc.DB2Driver

# "JAR" specifies a jar file that is needed to run the driver.
# The file must be placed in the ASBNode/lib/java subdirectory of the
# InformationServer installation directory on this system unless a full path is
# specified.
# NOTE that a full path MUST use forward-slashes as separators EVEN ON A WINDOWS
# SYSTEM.
# The property may be repeated if more than one jar needs to be loaded.
JAR=db2jcc.jar
JAR=db2jcc_license_cu.jar

# "URL" is used to identify the database concerned.
# (The following example is for the standard DB2 database that holds the
# Information Server repository, as shipped, unless the port number has been changed
# from the default of 50000 during installation.)
URL=jdbc:db2://localhost:50000/xmeta
```

```
# The following specify the username and password to connect to the DSODB schema.
Username=xxxx
Password=xxxx
```

Procedure

1. Copy the template file for your chosen database system to the DSODB directory. The template files are installed in these directories:

Table 2. Directories containing template files

DBMS	Directory

DB2 on Windows	`\InformationServer\Server\DSODB\scripts\DB2_LUW_9_5`
	or
	`\InformationServer\Server\DSODB\scripts\DB2_LUW_9_7`

DB2 on UNIX and Linux	`/InformationServer/Server/DSODB/scripts/DB2_LUW_9_5`
	or
	`/InformationServer/Server/DSODB/scripts/DB2_LUW_9_7`

Oracle on Windows	`\InformationServer\Server\DSODB\scripts\Oracle10g`
	or
	`\InformationServer\Server\DSODB\scripts\Oracle11g`

Oracle on UNIX and Linux	`/InformationServer/Server/DSODB/scripts/Oracle10g`
	or
	`/InformationServer/Server/DSODB/scripts/Oracle11g`

Microsoft SQL Server on Windows	`\InformationServer\Server\DSODB\scripts\SQLServer2005`
	or
	`\InformationServer\Server\DSODB\scripts\SQLServer2008`

2. Set the **URL** parameter to point to the database which contains the operations database schema. By default, the configuration files assume the operations database schema is contained in the metadata repository database.
3. Set the database system connection parameters.

Database system	Connection parameters
DB2	1. Ensure a JDBC driver for DB2 exists in the right location. Typically, these two JAR files that are installed with DB2 in the `/IBM/SQLLIB/java` directory are required: • db2jcc.jar • db2jcc_license_cu.jar These JAR files are normally installed in the `InformationServer/ASBNode/lib/java` directory. If these JAR files are not in that directory, copy them from the `/IBM/SQLLIB/java` directory, or specify the full path name. If different JAR files are required, they must either be copied to the `InformationServer/ASBNode/lib/java` directory, or the full path must be supplied. You must specify the names of the JAR files in the JARS property in the `DSODBConnect.cfg` file. 2. Define the port number for the JDBC connection. The URL key contains a port number as part of its string, which is set to 50000 in the template files. Change the port number for your JDBC connection if you use a different one.

Oracle	These parameters are set in the template file for an Oracle 10g or an Oracle 11g connection:
	`Driver=com.ibm.isf.jdbc.oracle.OracleDriver`
	`JAR=ISoracle.jar`
	`URL=jdbc:oracle:thin:@localhost:1521:xmeta`
	The default settings should only be changed under advice from IBM. If so, the JAR file must either be copied to the `InformationServer/ASBNode/lib/java` directory, or specify the full path must be supplied.

Microsoft SQL Server	These parameters are set in the template file for an SQL Server 2005 or SQL Server 2008 connection:
	`Driver=com.ibm.isf.jdbc.sqlserver.SQLServerDriver`
	`JAR=ISsqlserver.jar`
	`URL=jdbc:sqlserver://localhost:1433;databaseName=xmeta`
	The default settings should only be changed under advice from IBM. If so, the JAR file must either be copied to the `InformationServer/ASBNode/lib/java` directory, or the full path must be supplied.

4. Define the **Username** and **Password** parameters.
5. Optionally, encrypt the **Username**, **Password**, or both parameters The values of one or both of these parameters can be stored in a strongly encrypted form. The monitoring services detect that the values of these fields are encrypted by the presence of a {iisenc} prefix on the values of the parameters.
   a. Run the encrypt script that is in the /ASBNode/bin directory in the installation directory, and specify the value to be encrypted.

      For example, on Windows, run this command to encrypt a value of test to be used for the password for the user ODBADMin

      ```
      C:\IBM\InformationServer\ASBNode\bin\encrypt.bat test
      ```

      The command returns this value:

      ```
      {iisenc}xvybkFWZNPlfJmdZa3WpMQ==
      ```

      Copy this value to the relevant lines in the configuration file:

```
# The following specify the username and password
# to connect to the DSODB schema.
Username=ODBAdmin
Password={iisenc}xvybkFWZNPlfJmdZa3WpMQ==
```

6. Check the configuration of the monitoring system.

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 14.

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 a 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 **ResouceNode** 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 ofIBM 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	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.
		The message ID can be followed by a *ContentType* 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 *ContentType* column in the JobRunLog table.
		This option can be repeated.
		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`
		A message ID should be specified only once in the **CaptureLog** or the **IgnoreLog** options.
		Entries that are in the **IgnoreLog** option take precedence over entries in the **CaptureLog** option.
		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.

IgnoreLog	comma-separated strings	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.
		This option can be repeated.
		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.
		Entries that are in the **IgnoreLog** option take precedence over entries in the **CaptureLog** option.
		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`

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

Option	Values	Purpose
UpdateIntSecs	integer	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. The default value is 10, which means that such events are written no more frequently than every 10 seconds.
MonitorLinks	0 or 1	Controls if stage-level and link-level statistics, and references to data locators, are captured at the end of each job run. Set `MonitorLinks=0` if you do not want to capture this data is not to be captured. The default value is 0. Set `MonitorLinks=1` if you want to capture this data. Stage-level and link-level monitoring data is not used by the IBM InfoSphere DataStage Operations Console, so enable this option only if you want to extract this data using SQL statements.
CustomBatchId CustomJobType CustomContact CustomSequence CustomField1 CustomField2	string	Specifies the names of job parameters that you want to use to tag entries in the JobRun table. 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. 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. By default these options are empty.

EventsDir	directory path	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.
		If omitted, the default is the `/InformationServer/Server/DSODB/events` directory.
		For example, on Windows:
		`EventsDir=C:/IBM/InformationServer/Server/DSODB/events`
		For example, on Unix and Linux:
		`EventsDir=/opt/InformationServer/Server/DSODB/events`

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

Option	Values	Purpose
TraceMax	integer	Controls if you want to monitor the tracing of job run events, and specifies how many lines are written to the trace file. If TraceMax is set to a number greater than 0, tracing is enabled. 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. A file named `JobRuntime.log` is written to the `/Server/DSODB/logs` directory in the installation directory. By default TraceMax is set to 0, which means that monitoring the tracing of job run events is disabled.

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	Specifies, in minutes, how often automatic validation of currently running jobs is performed. 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. The default value is JobRunCheckInterval=60 minutes, with the minimum value being JobRunCheckInterval=10 minutes. If you set JobRunCheckInterval=0 the checking no longer occurs.
JobRunUsage	0 or 1	Controls if job run usage data is collected. Set `JobRunUsage=0` if you do not want to capture this data. Set `JobRunUsage=1` if you want to capture this data. The default value is 1.

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

Option	Values	Purpose
JobRunAggSnaps	integer	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. 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. 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. The default is 15 values per row.

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	Controls if system resource usage data is collected. Set `ResourceMonitor=0` you do not want to capture this data. Set `ResourceMonitor=1` you want to capture this data. The default value. is 1
ResourceNode	string	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. The name given for each node must match that used in parallel job configuration files. This option can be repeated if several remote nodes are to be monitored.

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

Option	Values	Purpose
ResourceLocalFS	directory path	Specifies a file system path whose disk mount point is to be monitored for used and free space. 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. For example: `ResourceLocalFS=/scratch`
ResourceRemoteFS	node + directory path	Specifies a file system path on a remote node whose disk mount point is to be monitored for used and free space. Specify the node name and file system path separated by a plus sign (+). The remote node name must match one specified via a corresponding **ResourceNode** option. This option can be repeated to specify multiple remote disks. For example: `ResourceRemoteFS=nodeA+/scratch`
ResourcePollPeriod	integer	Specifies, in seconds, how frequently a system resource snapshot is taken. 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. The default value is 10 seconds.

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

Option	Values	Purpose
ResourceSampleSize	integer	Specifies how many system resource snapshots are taken before an aggregated record of those values is stored. 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. 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.
ResourceAllAggregatedUsage	0 or 1	Specifies whether to always store aggregated resource usage data as described under **ResourceSampleSize**. If ResourceAllAggregatedUsage=1 then aggregated data is always stored. If ResourceAllAggregatedUsage=0, then aggregated data is only stored while there is any DataStage job activity. If you set ResourceAllAggregatedUsage=0 you avoid storing large numbers of records when there are no jobs running. The default value is 1, which stores aggregated resource usage data continuously.

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

Option	Values	Purpose
ResourceAggRunPollPeriod	integer	Specifies how many aggregated snapshots are stored before and after any job activity has been detected. This option is only relevant if ResourceAllAggregatedUsage=0. 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. 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. The value for **`ResourceAggNonRunPollPeriod`** must be less than the value for **`ResourceAggRunPollPeriod`**.
ResourceAggNonRunPollPeriod	integer	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. Snapshot data is always stored according to the **`ResourceAggRunPollPeriod`** option, but this option specifies how often the check for job activity is made. The time period is this value multiplied by the aggregated snapshot time. 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. The value for **`ResourceAggNonRunPollPeriod`** must be less than the value for **`ResourceAggRunPollPeriod`**.

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.
ODBQAPPPORTNUM	integer	Specifies the port number that the ODBQueryApp process uses to handlerequests 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 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 CustomJobType CustomContact CustomSequence CustomField1 CustomField2	The setting in the project-level file takes precedence. If an option is not present in the project-level file but is in the system-level file, the system-level setting applies.

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	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. 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.
serviceName	string	Set `*serviceName*=NO` to disable the checking of the status of the service by the Operations Console. You can disable the checking of the status of these services: • appwatcher • jobmonapp • dsrpcd • odbqueryapp • engmonapp • resmonapp For example, set `jobmonapp=NO`. Remove this option to re-enable the checking of the status of the service by the Operations Console.

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

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 13.

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

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	Issue this command from the `/InformationServer/Server/DSODB/bin` directory: `DSAppWatcher.sh –start` This command must be run as dsadm 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 RUNNING state. Add this command to the same place as the `uvadmin –start` command in the InfoSphere Information Server engine startup scripts.
Windows	Issue these commands to start the Windows service: `net start "DataStage AppWatcher Service"` If the command fails, the following service specific return codes might be shown in the failure message: **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 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.

Stopping the AppWatcher process

You can use these commands to stop the AppWatcher process.

Procedure

Stop the AppWatcher process.

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

Automatically starting and stopping the AppWatcher process

Configure the AppWatcher process to automatically start after restarting the engine system.

About this task

On Windows, the AppWatcher process is implemented by a service called DataStage AppWatcher Service. By default, this service is configured to automatically start when the system is started. You can configure this service by modifying the settings for the service in the control panel.

On UNIX and Linux, you can update the DataStage Engine startup script to automatically start and stop the AppWatcher process.

Procedure

1. As the root user, edit the file `InformationServerRoot/Server/DSEngine/sample/ds.rc`.
2. Update the DataStage Engine startup script to automatically start the AppWatcher process.
   a. Locate the block of the script that is commented `#start PX JobMonApp`.

      The block looks like this example:

      ```
      #start px JobMonApp:
      if [ "${PXHOME}" != "" ]
      then
         ...
      fi
      ```

   b. Add these two lines after that block:

      ```
      echo "Starting DSAppWatcher"
      su dsadm "${DSHOME}/../DSODB/bin/DSAppWatcher.sh -start"
      ```

If your system is configured with an administrator user name other than dsadm, change the user specified accordingly.

3. Optional: Update the DataStage Engine startup script to automatically stop the AppWatcher process.
   a. Locate the block of the script that is commented `#stop PX JobMonApp`.

      The block looks like this example:

      ```
      #stop px JobMonApp:
      if [ -n "${PXHOME}" ]
      then
         ...
      fi
      ```

   b. Add these two lines after that block:

      ```
      echo "Stopping DSAppWatcher"
      su dsadm "${DSHOME}/../DSODB/bin/DSAppWatcher.sh -stop"
      ```

      If your system is configured with an administrator user name other than dsadm, change the user specified accordingly.

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 26.

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 26.

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 DSODBConfig.cfg 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 DSODBConfig.cfg 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.cfg 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.cfg 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 EngMonApp 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 EngMonApp 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 DSOBDConfig.cfg configuration file.

Monitoring jobs and job runs by using the Operations Console

Use the IBM InfoSphere DataStage Operations Console to monitor your jobs, job activity, and system resources 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, and system resources for each of your IBM InfoSphere Information Server engines.

The Operations Console provides engine-wide information about job runs, system resources, 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.

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. Additionally, 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. 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 IBM 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 IBM 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 IBM InfoSphere Information Server environment, you need to know about it immediately so that you can resolve the problem. The dashboard on the **Home 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 **Home 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 IBM 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 IBM InfoSphere Information Server engines 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.

Configuring user access to the Operations Console

Users are able to access the 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 InfoSphere DataStage and QualityStage Administrator.

To view information in the Operations 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. You can connect to the Operations Console if you are a InfoSphere DataStage user who has the Operations Viewer role, or a superior role, for a project that is included in the operations database.

Once you have logged in to the console, you can see information only for the projects for which you 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 Operations 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.

Operations Console system requirements

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

- A minimum screen resolution of 1024x768
- One of the following browsers
  - Firefox 3.5
  - Firefox 3.6
  - Internet Explorer 7
  - Internet Explorer 8

Starting the Operations Console

Start the IBM InfoSphere DataStage Operations Console.

Procedure

1. If you are using HTTP, enter this URL from a supported browser:
   ```
   http://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 HTTP is 9080
2. If you are 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

Viewing log messages in the Operations Console

You can use the Operations Console to view messages from job runs 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 on the open space to the left of the job name and job icon. A highlighted background appears for the job that you selected from the job run list.
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. The Run Details window displays.
3. Click on the **Log Messages** tab in the Run Details window. A list of messages displays for the job run that you selected.
4. Click on a message on the list of messages. The details for that message displays below 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

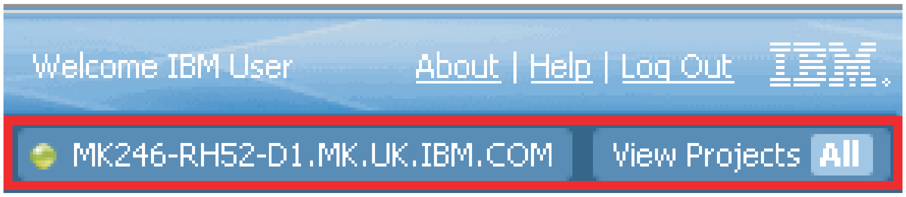
The panels in the Operations Console display information about job runs, engine status, and the use of system resources for the active IBM InfoSphere Information Server engine. You can narrow the scope of information in these panels, and set the refresh rate to update the information only as frequently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

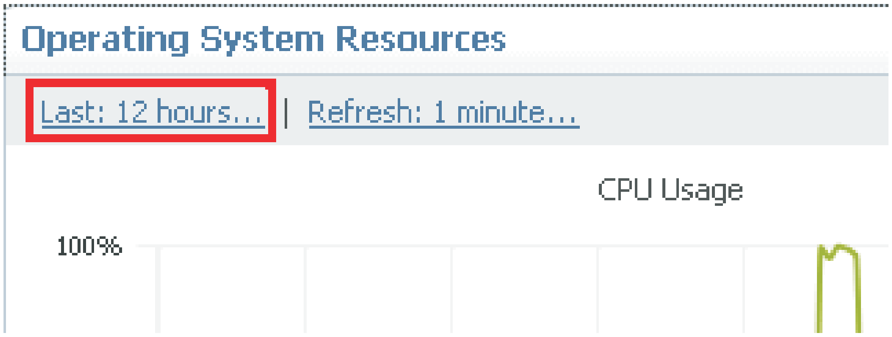
1. Click the active engine in the **Engine Selection** menu.
2. Select the engine from the list that you want to view. The console refreshes with data from the engine that you selected. The scope on the **View Projects** button is set to show all projects in the active engine that the user has authorization 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.


Operating System Resources
Last: 12 hours... | Refresh: 1 minute...
CPU Usage
100%

Procedure

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

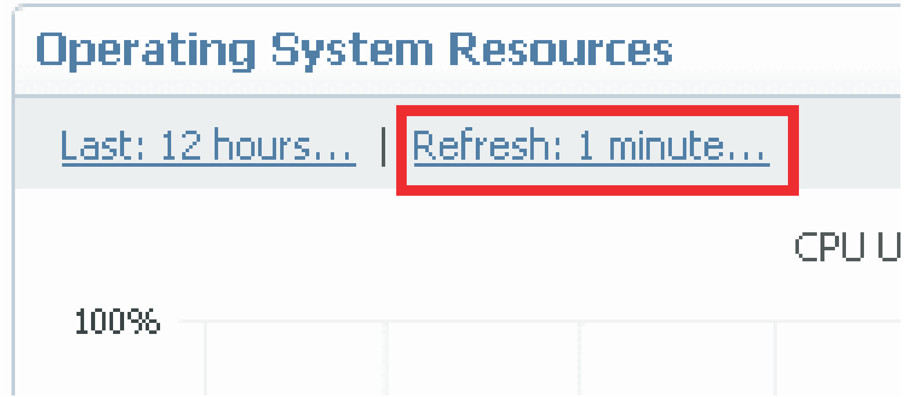
1. Click the link that shows the length of event history that is currently being displayed in the panel.
2. In the event history selection window, select the unit of time measurement that you want to use to set the event history and enter an appropriate value for the length of history that you require.
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 doesn't restrict the performance of the Operations Console on your system.


Operating System Resources

Procedure

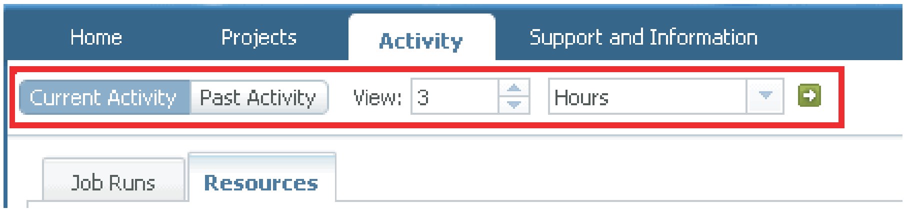
1. Click the link that shows the current refresh rate for the panel.
2. In the refresh rate selection window, select the unit of time measurement that you want to use to set the refresh rate and enter an appropriate value for the refresh rate that you require.
3. Click the **close button** in the upper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



View: 3 Hours

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 Job 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 Job Runs 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 **View Details**.

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.

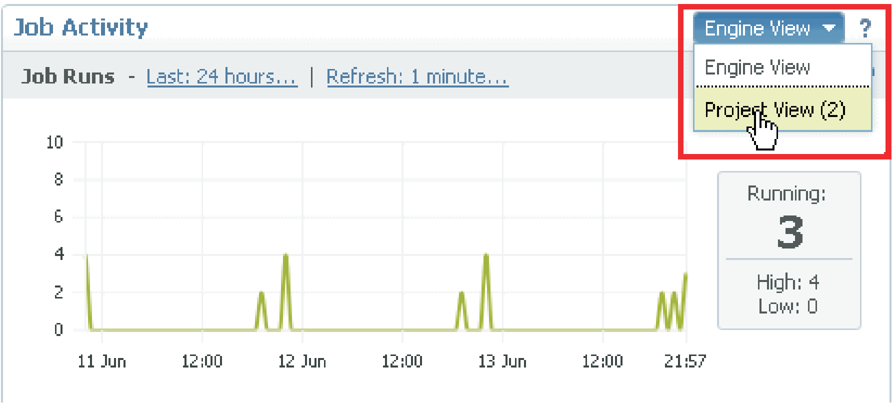
Home tab

The home tab serves as a dashboard for 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.

Job Activity panel:

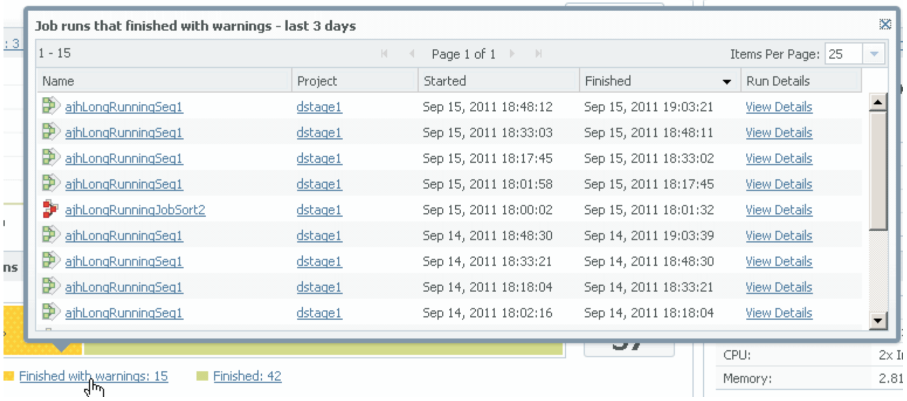
The Job Activity panel provides details about job workload and job status for all of the jobs on the selected IBM InfoSphere Information Server engine or in the selected projects. Use this activity information to help you troubleshoot jobs that failed or completed with warnings.

The Job 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 Job 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.



Job runs that finished with warnings - last 3 days

1 - 15 Page 1 of 1 Items Per Page: 25

Name	Project	Started	Finished	Run Details
ajhLongRunningSeq1	dstage1	Sep 15, 2011 18:48:12	Sep 15, 2011 19:03:21	View Details
ajhLongRunningSeq1	dstage1	Sep 15, 2011 18:33:03	Sep 15, 2011 18:48:11	View Details
ajhLongRunningSeq1	dstage1	Sep 15, 2011 18:17:45	Sep 15, 2011 18:33:02	View Details
ajhLongRunningSeq1	dstage1	Sep 15, 2011 18:01:58	Sep 15, 2011 18:17:45	View Details
ajhLongRunningJobSort2	dstage1	Sep 15, 2011 18:00:02	Sep 15, 2011 18:01:32	View Details
ajhLongRunningSeq1	dstage1	Sep 14, 2011 18:48:30	Sep 14, 2011 19:03:39	View Details
ajhLongRunningSeq1	dstage1	Sep 14, 2011 18:33:21	Sep 14, 2011 18:48:30	View Details
ajhLongRunningSeq1	dstage1	Sep 14, 2011 18:18:04	Sep 14, 2011 18:33:21	View Details
ajhLongRunningSeq1	dstage1	Sep 14, 2011 18:02:16	Sep 14, 2011 18:18:04	View Details

🟨 Finished with warnings: 15 🟩 Finished: 42

CPU: 2x I

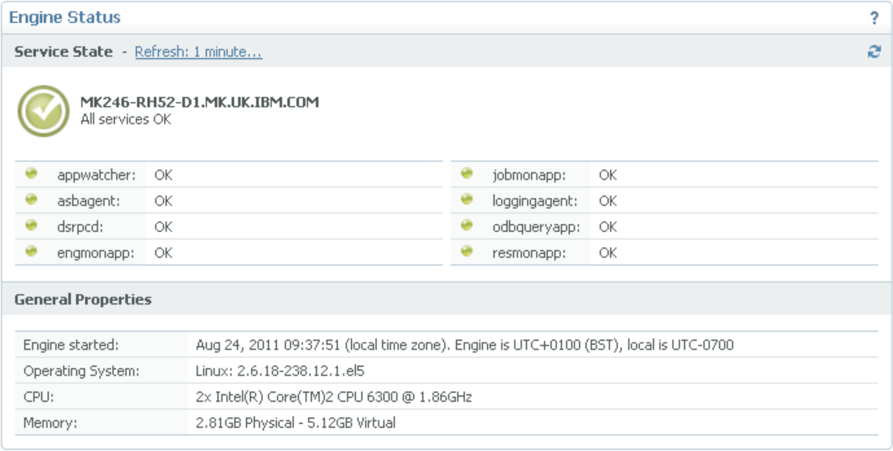
Memory: 2.81

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.



Engine Status ?

Service State - Refresh: 1 minute...

MK246-RH52-D1.MK.UK.IBM.COM
All services OK

● appwatcher:	OK	● jobmonapp:	OK
● asbagent:	OK	● loggingagent:	OK
● dsrpcd:	OK	● odbqueryapp:	OK
● engmonapp:	OK	● resmonapp:	OK

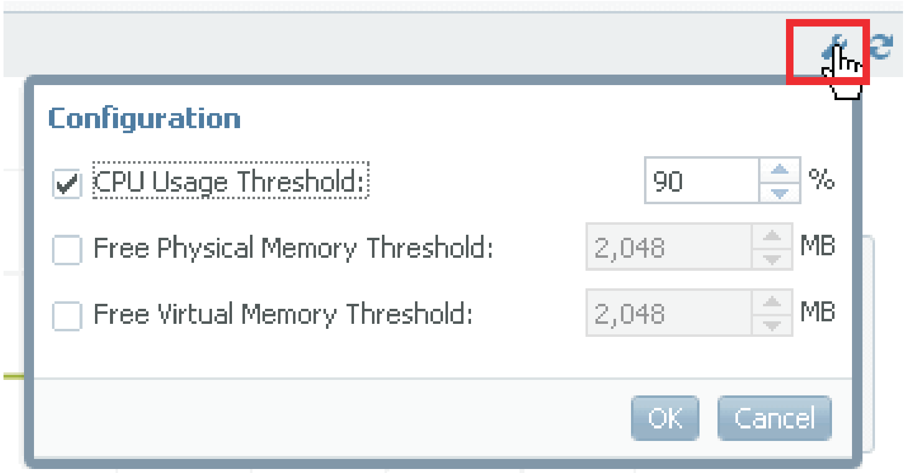
General Properties

Engine started:	Aug 24, 2011 09:37:51 (local time zone). Engine is UTC+0100 (BST), local is UTC-0700
Operating System:	Linux: 2.6.18-238.12.1.el5
CPU:	2x Intel(R) Core(TM)2 CPU 6300 @ 1.86GHz
Memory:	2.81GB Physical - 5.12GB Virtual

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.



Configuration

☑ CPU Usage Threshold: 90 %

□ Free Physical Memory Threshold: 2,048 MB

□ Free Virtual Memory Threshold: 2,048 MB

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 sub-folders 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.

Additionally, the Details panel displays information about the number of dependencies between the selected job and the other jobs in your system. You can click on the number of dependent jobs or jobs that the selected job depends upon to navigate through all of the jobs in the dependency stack.

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 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:

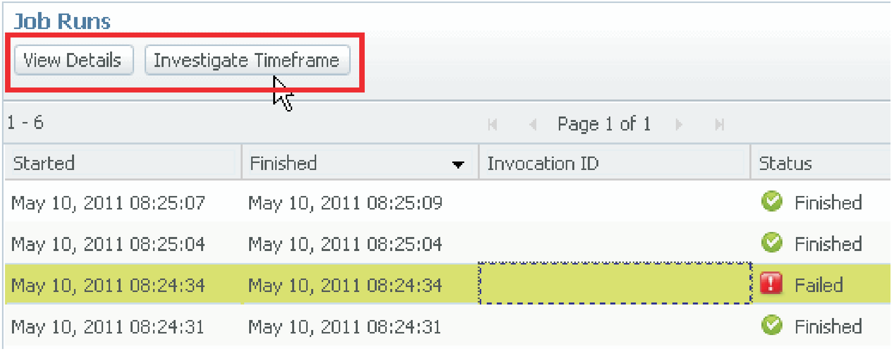
The Job Designs panel provides a view of the job design information for all of the jobs in the selected folder. Use this job design information to quickly identify job design problems before you analyze and modify your job designs in the IBM InfoSphere DataStage and QualityStage Designer.

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.

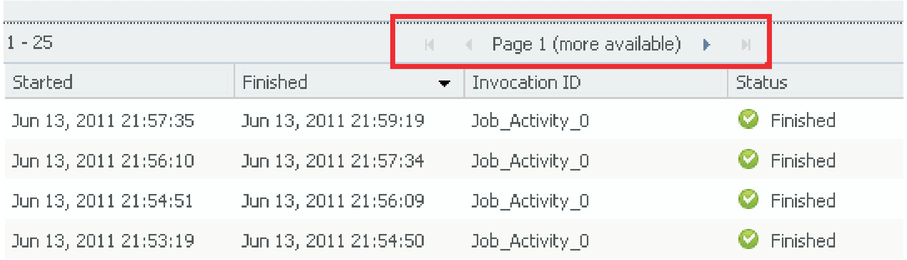
When you select a job from the repository tree in the Navigation panel, the Job Runs panel appears and displays a history of job runs for that particular job. You can sort jobs on any column in the list of runs for the selected job. You can also select an individual run in the list of runs and click the **View Details** button to view more detailed information about that run. You can select a job run and click the **Investigate Timeframe** button to go to the Activity Tab and see a list of job runs that executed during the same timeframe as the selected job run, which can help you to identify the cause of job performance issues.



Started	Finished	Invocation ID	Status
May 10, 2011 08:25:07	May 10, 2011 08:25:09		✓ Finished
May 10, 2011 08:25:04	May 10, 2011 08:25:04		✓ Finished
May 10, 2011 08:24:34	May 10, 2011 08:24:34		! Failed
May 10, 2011 08:24:31	May 10, 2011 08:24:31		✓ Finished

If you select two job runs and click **View Details**, 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.

The list of job runs might have multiple pages. You can navigate through these pages with the arrows at the top of the job run list.



1 - 25

Page 1 (more available)

Started	Finished	Invocation ID	Status
Jun 13, 2011 21:57:35	Jun 13, 2011 21:59:19	Job_Activity_0	✓ Finished
Jun 13, 2011 21:56:10	Jun 13, 2011 21:57:34	Job_Activity_0	✓ Finished
Jun 13, 2011 21:54:51	Jun 13, 2011 21:56:09	Job_Activity_0	✓ Finished
Jun 13, 2011 21:53:19	Jun 13, 2011 21:54:50	Job_Activity_0	✓ Finished

Job Activity tab

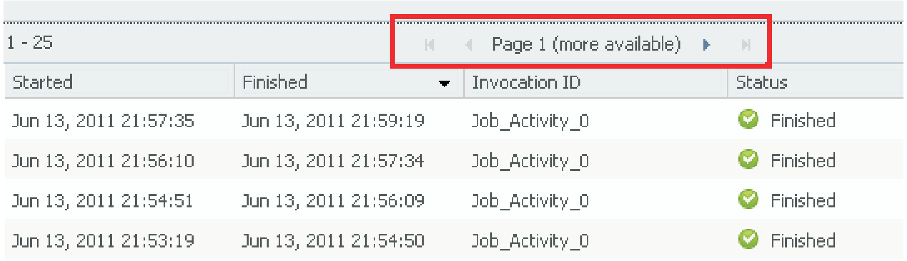
You can use the Job Activity tab to get an overview of all of the job activity on the active engine.

Job Runs tab:

The Job Runs panel provides details about the job run history for all of the jobs that have 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, and view detailed information about a selected job by clicking the View Details button. The list of job runs might have multiple pages. You can navigate through these pages with the arrows at the top of the job run list.



1 - 25

Page 1 (more available)

Started	Finished	Invocation ID	Status
Jun 13, 2011 21:57:35	Jun 13, 2011 21:59:19	Job_Activity_0	✓ Finished
Jun 13, 2011 21:56:10	Jun 13, 2011 21:57:34	Job_Activity_0	✓ Finished
Jun 13, 2011 21:54:51	Jun 13, 2011 21:56:09	Job_Activity_0	✓ Finished
Jun 13, 2011 21:53:19	Jun 13, 2011 21:54:50	Job_Activity_0	✓ Finished

If you select a job from the job run list and click on the **View Details** button, the Run Details window will appear. This window displays detailed properties, parameters, performance, and messages for the selected job run.

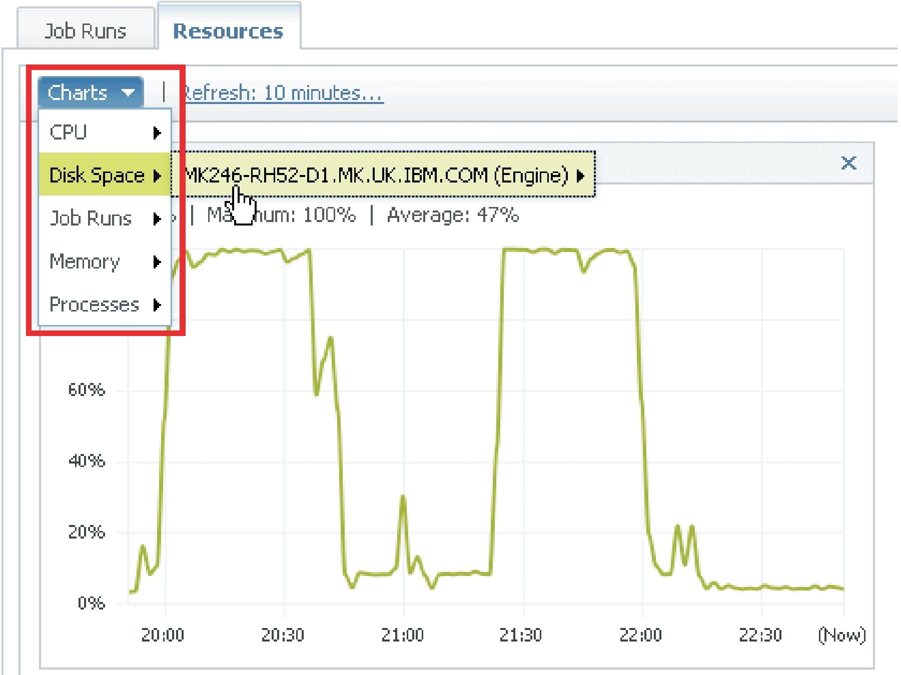
If you have narrowed the scope of your information by selecting specific projects to view with the project filter, job runs from projects that are not included in the filter are not displayed. If your view is not narrowed by the project filter and matching job runs exist in projects that you are not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

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 = 'job1name'
  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	• HostDetail • ResourceSnap • ResourceUsage • JobExec • ParallelConfig
JobExec	• JobRun • JobStage • JobLink

Table 9. Primary tables and their associated tables for deletion (continued)

Primary table	Associated tables
JobRun	• 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 54.

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 54.

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 change the password in the configuration file.

About this task

The user password for the operations database is stored in the DSODBConnect.cfg configuration file which is stored in the /InformationServer/Server/DSODB directory on Unix and Linux, and in the\InformationServer\Server\DSODB directory on Windows.

It is not necessary to stop running jobs while changing the password, but no system resource usage information is captured while data collection processes are stopped.

Procedure

1. Stop the AppWatcher process.
2. Update the password in the `DSODBConnect.cfg` configuration file.
3. Restart the AppWatcher process.

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. Use the remove script, called `DSODB_setup_remove.sql`, that was created when you created the operations database schema.

Chapter 2. Operations database schema reference

The operations database stores the monitoring data that is displayed by the IBM InfoSphere DataStage 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)
n	Variable length string, maximum length *n*[3]	VARGRAPHIC(n)	VARCHAR2(n)	NVARCHAR(n)
p,s	Decimal precision and scale	DECIMAL(p,s)	DECIMAL(p,s)	DECIMAL(p,s)
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.

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 DSODB 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 ParellelConfig 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
DataStoreSublass       = 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 ResouceInfo 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.

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/software/sw-training/
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/

Providing feedback

The following table describes how to provide feedback to IBM about products and product documentation.

Table 44. Providing feedback to IBM

Type of feedback	Action
Product feedback	You can provide general product feedback through the Consumability Survey at www.ibm.com/software/data/info/consumability-survey

Table 44. Providing feedback to IBM (continued)

Type of feedback	Action
Documentation feedback	To comment on the information center, click the Feedback link on the top right side of any topic in the information center. You can also send comments about PDF file books, the information center, or any other documentation in the following ways: • Online reader comment form: www.ibm.com/software/data/rcf/ • E-mail: comments@us.ibm.com

Accessing product documentation

Documentation is provided in a variety of locations and formats, including in help that is opened directly from the product client interfaces, in a suite-wide information center, and in PDF file books.

The information center is installed as a common service with IBM InfoSphere Information Server. The information 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 the information center from the installed product or from a Web browser.

Accessing the information center

You can use the following methods to open the installed information center.

- Click the **Help** link in the upper right of the client interface.

  Note: From IBM InfoSphere FastTrack and IBM InfoSphere Information Server Manager, the main Help item opens a local help system. Choose **Help > Open Info Center** to open the full suite information center.
- 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.
- Use a Web browser to access the installed information center even when you are not logged in to the product. Enter the following address in a Web browser: http://host_name:port_number/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.home.doc/ic-homepage.html. The host_name is the name of the services tier computer where the information center is installed, and port_number is the port number for InfoSphere Information Server. The default port number is 9080. For example, on a Microsoft® Windows® Server computer named iisdocs2, the Web address is in the following format: http://iisdocs2:9080/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.nav.doc/dochome/iisinfsrv_home.html.

A subset of the information center is also available on the IBM Web site and periodically refreshed at http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r7/index.jsp.

Obtaining PDF and hardcopy documentation

- A subset of the PDF file books are available through the InfoSphere Information Server software installer and the distribution media. The other 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.

Providing feedback about the documentation

You can send your comments about documentation in the following ways:

- Online reader comment form: www.ibm.com/software/data/rcf/
- E-mail: comments@us.ibm.com

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. The installation program installs the following product modules and components:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- IBM InfoSphere DataStage
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere QualityStage

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. XHTML allows you to set display preferences in your browser. It also allows you to use screen readers and other assistive technologies to access the documentation.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Notices and trademarks

This information was developed for products and services offered in the U.S.A.

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.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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.

Trademarks

IBM, the IBM logo, and ibm.com are 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.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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

Printed in USA