



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

iWD Deployment Guide

Stat Server Extensions

12/16/2025

Contents

- 1 Stat Server Extensions
 - 1.1 Installing Stat Server Extensions
 - 1.2 Stat Server Configuration Options
 - 1.3 Creating iWD Virtual Queues automatically
 - 1.4 Report Stats for each Dimension on its own Virtual Queue
 - 1.5 Configure Stat Extensions for separate iWD Solutions using one Stat Server

Stat Server Extensions

After you have installed iWD, you can install the iWD Stat Extensions, which provide access to the aggregated data in the Data Mart.

Important

iWD Setup Utility is not supported in release 9.0.

Installing Stat Server Extensions

Important

If the Data Mart is not used, Stat Extensions are not required and the solution will work fine.

Purpose

To provide access to the aggregated data in the Data Mart.

Prerequisites

- An instance of Stat Server is installed, dedicated for use with iWD. Refer to the [Stat Server 8.5 Deployment Guide](#) for more information.
- ORS 8.1.400.48 is installed if you wish to use the Stat Server Extensions features implemented in iWD release 8.5.106.03.

Procedure

1. From the server that is running Stat Server, navigate to the **iWD Stat Extensions** folder of the iWD CD. Locate and double-click **Setup.exe**.
2. Click **Next** on the **Welcome** screen.
3. Select the appropriate Stat Server instance from the list that is displayed and click **Next**.
4. Click **Install** to install iWD Stat Extensions. Click **Finish** when the installation has been completed.

Stat Server Configuration Options

[java-extensions] section

During installation, a new option—**BPR_iWD_Extension.jar**—is added to the [java-extensions] section, with a default value of `true`.

[java-extensions-bpr-iwd] section

The **[java-extensions-bpr-iwd]** section contains options which specify the JDBC connection driver and parameters for access to the iWD Data Mart database. Most options are managed by the Stat Adapter job of iWD Data Mart and are rewritten each time the Stat Adapter is run. For reference, those options are listed below:

- **jdbc-driver-jar**—The **.jar** file with the JDBC driver. The path is relative to the directory specified as **java-libraries-dir** in the Stat Server configuration (which defaults to **./java/lib** in the Stat Server installation directory). Valid values include:
 - `mssql-jdbc-6.1.0.jre8.jar` (for MS SQL)
 - `ojdbc8.jar` (for Oracle)
 - `postgresql-9.4.1212.jar` (for PostgreSQL)
- **jdbc-driver**—The class name for the corresponding JDBC driver. Valid values include:
 - `com.microsoft.sqlserver.jdbc.SQLServerDriver` (for MS SQL)
 - `oracle.jdbc.OracleDriver` (for Oracle)
 - `org.postgresql.Driver` (for PostgreSQL)
- **jdbc-url**—The JDBC URL, which describes RDBMS-specific access parameters. Below are some sample values:
 - `jdbc:sqlserver://hostname:1433;databaseName=databasename` (for MS SQL)
 - `jdbc:oracle:thin:@//hostname:1521/databasename` (for Oracle)
 - `jdbc:postgresql://hostname:5432/databasename` (for PostgreSQL)
- **user**—The user name for database access.
- **password**—The password for database access.
- **verbose**—The level to control debug information, provided in the Stat Server log file. Possible values are `debug`, `trace`, or `standard`.
- **refresh-interval**—The interval (in minutes) for data updates from the database.
- **service-id-1**—The runtime ID of the Statistics Adapter service in the iWD configuration.
- **service-tenant-1**—The name of the configuration tenant—this is linked to the iWD managed tenant where the Statistics Adapter service is configured.

- **virtual-queue-name-1**—

- If Dimension Mapping is set to Filter, it is the name of the single Genesys virtual queue to which statistics are distributed.
- If Dimension Mapping is set to Virtual Queue, it is the prefix to be added to the names of Genesys virtual queues to which statistics are distributed.

- **dimension-mapping-1**—The type of the dimension mapping between iWD Data Mart and Stat Server. Valid values are Filter and Virtual Queue.

You can set a small subset of options manually:

- **java-extension-jar**—Name of the BPR iWD extension file. Defaults to `BPR_iWD_Extension.jar`. This must match the option in **java-extensions** section in the Stat Server configuration.
- **jdbc-properties-file**—Path to JDBC driver properties file relative to directory specified as `java-libraries-dir` in Stat Server configuration (defaults to `./java/lib` in Stat Server installation directory). Can be used to specify optional driver-specific JDBC options. Please consult driver documentation for further information.
- **reconnection-timeout**—Delay in milliseconds between database reconnection attempts performed by the BPR iWD extension. Defaults to 10000.
- **tenant-ids**—A list of Configuration Server/Genesys Administrator tenant names mapped to iWD tenant IDs, separated by a comma. For a system with one tenant with name TenantA and ID T2, this option should be set to value: `Environment=1,TenantA=2`.

Creating iWD Virtual Queues automatically

The Statistics Adapter job can automatically create Virtual Queues for statistics to be reported. This feature is disabled by default.

For every unique value stored for the tenant in the **gtl_stat** table in the **dimensionId** column, the Statistics Adapter job does the following:

1. Checks whether the Virtual Queue with the defined name exists.
2. If the Virtual Queue does not exist, the Statistics Adapter job creates the Virtual Queue.
3. If the Virtual Queue does exist, the Statistics Adapter skips Virtual Queue creation.

These steps are continuously performed on each run of the Statistic Adapter job. No automatic update or delete is implemented.

Configuring Virtual Queues creation

To enable this feature, the following options in iWD Runtime Node application must be set:

- **stat-server \ dimension-mapping** = Virtual Queue
- **stat-server \ virtual-queue-name** = `<VQ_name_prefix>` (iWD_ by default)
- **virtual-queues \ create-vqs** = true

- **virtual-queues \ switch** = *<switch_name>* (name of the switch where VQs will be created)

You can configure these options in a more convenient way by using Data Mart settings in the iWD Plug-in for GAX, as described in [this article](#).

Virtual Queues naming convention

Virtual Queues are created according to the following naming convention:

- *<VQ prefix>* + *<dimensionId value>*

where:

- *<VQ prefix>* is taken from iWD Runtime Node options > **stat-server\virtual-queue-name** option.
- *<dimensionId value>* is taken from **dimensionId** column of **gtl_stat** table.

Report Stats for each Dimension on its own Virtual Queue

The iWD Stat Server Java Extension can be configured to report statistics in two different ways:

- All statistics for all dimensions can be reported on one Virtual Queue, or;
- Each dimension can have its statistics reported on its own Virtual Queue.

Manual setup of IWD configures the Stat Server to use the iWD Stat Server Java Extension to report each statistic on its own Virtual Queue. If you want to change this, you must make the change manually.

Procedure

To report each dimension on its own Virtual Queue:

1. Set the the option **dimension-mapping-1** to the value `Virtual Queue`.
2. Instead of using the Virtual Queue name as a value for the option **virtual-queue-name-1**, indicate the prefix that will be used for Virtual Queue names. For example, if **virtual-queue-name-1**=`dim-`, then Virtual Queues with the names `dim-CNT_T2_C106`, `dim-CNT_T2_C107` and so on, have to be created.

To report all statistics on one Virtual Queue:

1. Set the option **dimension-mapping-1** to the value `Filter`.
2. Set the Virtual Queue name as the value for the option **virtual-queue-name-1**.
3. Create the Virtual Queue using the name value from step 2.

Configure Stat Extensions for separate iWD Solutions using one

Stat Server

1. Copy the iWD SSJE binary on the back-end for each solution to provide statistics. In this example they are put into separate directories:

```
$ mkdir statserver/java/ext/iwd1 statserver/java/ext/iwd2
$ mv statserver/java/ext/BPR_iWD_Extension.jar statserver/java/ext/iwd1
$ cp statserver/java/ext/iwd1/BPR_iWD_Extension.jar statserver/java/ext/iwd2/
```

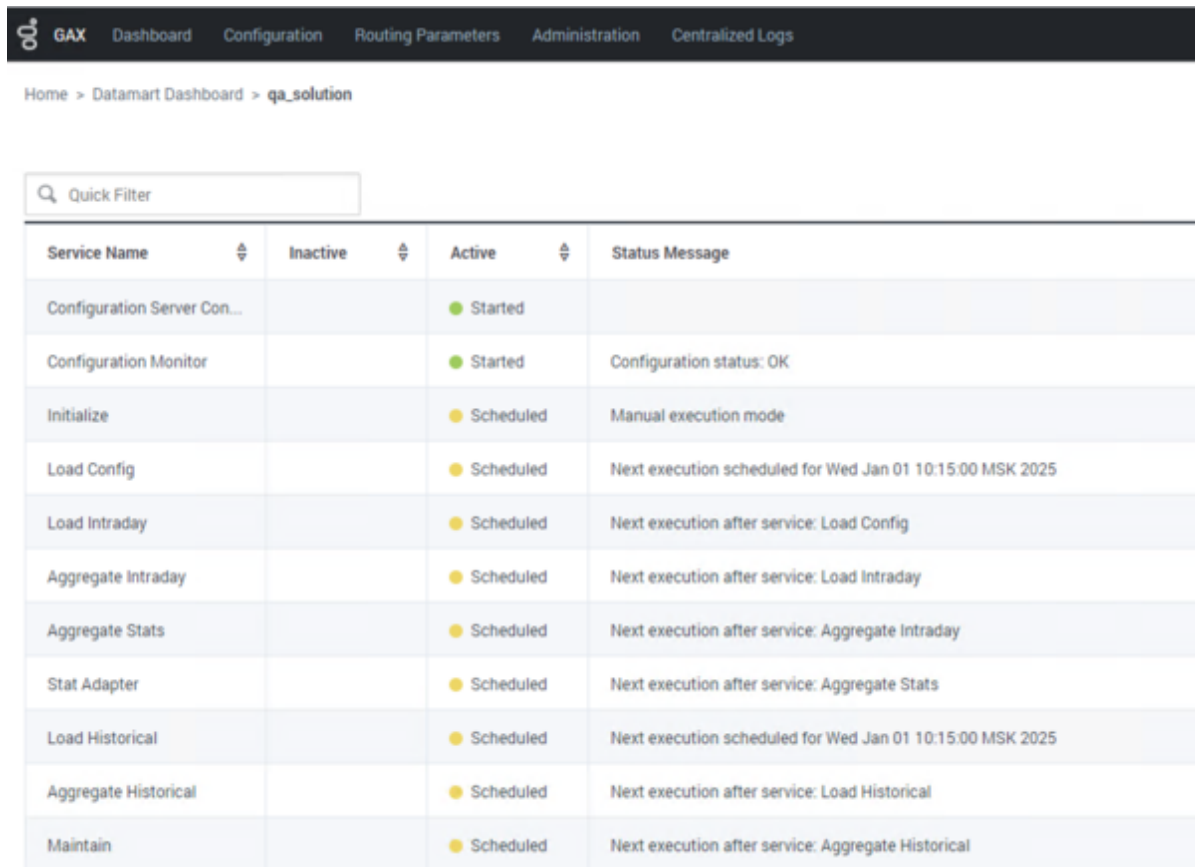
```
$ ls statserver/java/ext/*
statserver/java/ext/iwd1:
BPR_iWD_Extension.jar
```

```
statserver/java/ext/iwd2:
BPR_iWD_Extension.jar
```

2. Reflect these changes in each iWD Solution by using iWD Plug-in for GAX (**Configuration > iWD > Datamart > [tenant] > [solution] > Stat Server**). Ensure that the **Extension File Name** is unique for each solution:

The screenshot displays two side-by-side configuration windows from the GAX interface. Both windows show the 'Stat Server' configuration page for a specific solution. The left window is for 'qa_solution' and the right window is for 'IWD-8705_slt'. Both configurations show the 'Extension File Name' field, which is highlighted with arrows and the text 'unique extension file names'. The 'Extension File Name' field in the left window is 'iwd1/BPR_iWD_Extension.jar' and in the right window is 'iwd2/BPR_iWD_Extension.jar'. The 'Application' field is 'statserver' in both. The 'JDBC Driver Class' is 'The same as for Datamart' in both. The 'JDBC URL' is 'The same as for Datamart' in both. The 'Create Virtual Queues' checkbox is checked in both. The 'Dimension Mapping' field is 'Virtual queue (postfix)' in the left window and 'Switch_102' in the right window. The 'Service Index' field is '1' in the right window. The 'Extension Section Name' field is 'java-extensions-bpr-iwd-second' in the right window. The 'JDBC Driver JAR File' field is 'The same as for Datamart' in the right window.

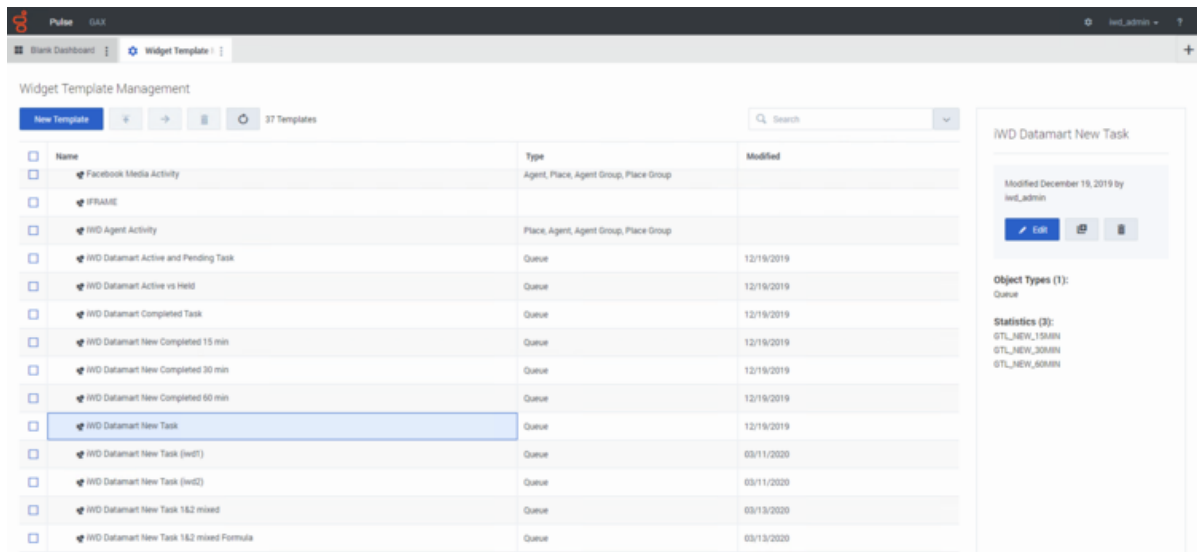
- Run the **Stat Adapter** job for each Data Mart you have configured (**Configuration > iWD > Datamart Dashboard > [tenant] > [solution] > Stat Adapter > Start**):



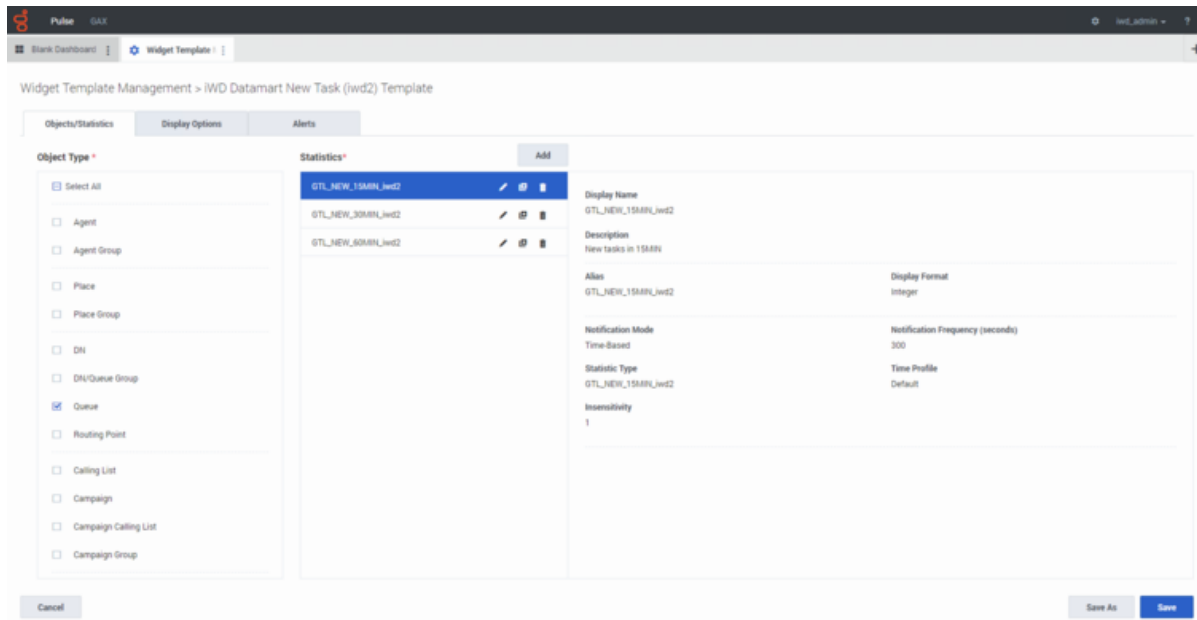
The screenshot shows the GAX interface with the breadcrumb path: Home > Datamart Dashboard > qa_solution. Below the breadcrumb is a 'Quick Filter' search bar. The main content is a table with columns: Service Name, Inactive, Active, and Status Message. The table lists several services, most of which are in a 'Scheduled' state.

Service Name	Inactive	Active	Status Message
Configuration Server Con...		Started	
Configuration Monitor		Started	Configuration status: OK
Initialize		Scheduled	Manual execution mode
Load Config		Scheduled	Next execution scheduled for Wed Jan 01 10:15:00 MSK 2025
Load Intraday		Scheduled	Next execution after service: Load Config
Aggregate Intraday		Scheduled	Next execution after service: Load Intraday
Aggregate Stats		Scheduled	Next execution after service: Aggregate Intraday
Stat Adapter		Scheduled	Next execution after service: Aggregate Stats
Load Historical		Scheduled	Next execution scheduled for Wed Jan 01 10:15:00 MSK 2025
Aggregate Historical		Scheduled	Next execution after service: Load Historical
Maintain		Scheduled	Next execution after service: Aggregate Historical

- Open your Stat Server application object, open **Application Options** and export them to either a cfg or csv file.
- Within these options there are Stat Types (options sections) with names starting with GTL_; such as GTL_ACTIVE, GTL_NEW_15MIN, and more. Clone these sections while providing some unique affix to them, like _iwd1 and _iwd2, for each solution. You should end up with sections GTL_ACTIVE_iwd1, GTL_ACTIVE_iwd2, and so on.
- Within each such section amend option **JavaSubCategory** with the name of the Java Extension name. Please refer to the [sample configuration file for two IWD solutions](#) below.
- Import the amended configurations to the Stat Server application object.
- Restart Stat Server.
- Open Pulse and clone the default iWD widget templates for each solution:



- In the templates, change the **Statistics Type** for each Statistic to match with the section names given in Stat Server:



- With new templates created, you can now add widgets and assign proper Virtual Queues to them, depending on the solution. Here's the result of a sample configuration:

Name	GTL_NEW_15MIN_iwd2	GTL_NEW_30MIN_iwd2	GTL_NEW_60MIN_iwd2
IWD_Second_DPT_T1_C4g@Switch_102	10	10	10
IWD_Second_DPT_T1_C4g@Switch_102	0	0	0
IWD_Second_SLT_SLT3g@Switch_102	10	10	10

Name	GTL_NEW_15MIN_iwd1	GTL_NEW_30MIN_iwd1	GTL_NEW_60MIN_iwd1
IWD_Test_DPT_T1_C4g@Switch_101	0	0	0
IWD_Test_DPT_T1_C4g@Switch_101	0	0	0
IWD_Test_DPT_T1_C4g@Switch_101	0	0	0
IWD_Test_DPT_T1_C7g@Switch_101	0	0	0
IWD_Test_DPT_T1_C8g@Switch_101	11	11	11
IWD_Test_DPT_Unclassified@Switch...	0	0	0
IWD_Test_SLT_SLT3g@Switch_101	11	11	11

Sample configuration file for two IWD solutions

```
[java-config]
java-extensions-dir=./java/ext
jvm-path=/usr/java/jre-1.8/lib/amd64/server/libjvm.so

[java-extensions-bpr-iwd-test]
service-id-1=STAT_1
service-tenant-1=selenium
virtual-queue-name-1=iWD_Test_
dimension-mapping-1=Virtual Queue
java-extension=iwd1/BPR_iWD_Extension.jar
java-extension-jar=iwd1/BPR_iWD_Extension.jar
jdbc-driver-jar=mssql-jdbc-6.1.0.jre8.jar
jdbc-driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc-url=jdbc:sqlserver://<db_host>:1433;databaseName=<iwd_dm_1_db_name>
user=<iwd_dm_1_db_user>
refresh-interval=15
verbose=debug

[java-extensions-bpr-iwd-second]
service-id-1=STAT_1
service-tenant-1=selenium
virtual-queue-name-1=iWD_Second_
dimension-mapping-1=Virtual Queue
java-extension=iwd2/BPR_iWD_Extension.jar
java-extension-jar=iwd2/BPR_iWD_Extension.jar
jdbc-driver-jar=mssql-jdbc-6.1.0.jre8.jar
jdbc-driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc-url=jdbc:sqlserver://<db_host>:1433;databaseName=<iwd_dm_2_db_name>
user=<iwd_dm_2_db_user>
refresh-interval=15
verbose=debug

[GTL_NEW_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_15MIN
```

```
[GTL_COMPLETED_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_15MIN
```

```
[GTL_NEW_30MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_30MIN
```

```
[GTL_COMPLETED_30MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_30MIN
```

```
[GTL_NEW_60MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_60MIN
```

```
[GTL_COMPLETED_60MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_60MIN
```

```
[GTL_OVERDUE_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=OVERDUE_15MIN
```

```
[GTL_PENDING_15MIN_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=PENDING_15MIN
```

```
[GTL_ACTIVE_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=ACTIVE
```

```
[GTL_HELD_iwd1]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd1/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=HELD
```

```
[GTL_NEW_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_15MIN
```

```
[GTL_COMPLETED_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_15MIN
```

```
[GTL_NEW_30MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_30MIN
```

```
[GTL_COMPLETED_30MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_30MIN
```

```
[GTL_NEW_60MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=NEW_60MIN
```

```
[GTL_COMPLETED_60MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=COMPLETED_60MIN
```

```
[GTL_OVERDUE_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=OVERDUE_15MIN
```

```
[GTL_PENDING_15MIN_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=PENDING_15MIN
```

```
[GTL_ACTIVE_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=ACTIVE
```

```
[GTL_HELD_iwd2]
Objects=Queue
Category=JavaCategory
JavaSubCategory=iwd2/BPR_iWD_Extension.jar:BPR iWD Statistics
Description=BPR iWD Statistics
measure-id=HELD
```

```
[java-extensions]
iwd1/BPR_iWD_Extension.jar=true
iwd2/BPR_iWD_Extension.jar=true
```