



Configure Custom Reports for Parallels RAS

- Parallels Remote Application Server 17.0

Target Audience

- IT professionals such as a server, network, and system administrators familiar with Microsoft SQL Server environments
- Existing Parallels RAS administrators

Prerequisites

- Experience operating Parallels RAS farm, including RAS Reporting Services
- Experience querying databases using Transact-SQL or Microsoft SQL Server Management Studio

Note: Parallels RAS should be installed and configured with RAS Reporting component in order to utilize the steps described below.

In this article, we are going to create a report that displays applications used by each user and a number of application launches.

In order to create reports on your own, first download Microsoft SQL Server Report Builder:

[Report Builder 2016](#)

[Report Builder 2014](#)

[Report Builder 2012](#)

Note: In order to avoid compatibility issues, we suggest using a version that corresponds to the version of Microsoft SQL Server edition.

[Install Report Builder](#)

Once it is installed, we can run it:

And create a new blank report:

Each report requires to have a data source, which can be configured using one of two approaches below.
(Download illustrated steps by clicking on the links below.)

- [Using embedded connection](#)
- [Using shared connection or report model](#)

After configuring the data source we should also create a data set. (Scroll to the end of the article for an example of SQL Query and its explanation.)

[Download the steps for Data Set configuration](#)

Once a data set is configured, we may start populating the actual report.

Populate the table according to the fields that will be provided by Data Set SQL Statement:

Run it to see the report output.

Open **Remote Application Server Console > Administration > Reporting > Tracking Settings**

Enable **Custom Reports** and specify the folder name (or keep default name) for saving .rdl reports.

Click **OK** on Advanced Setting and **Apply setting changes to RAS Console**.

Once done, click **Refresh** and you should see a **Custom Reports** folder:

RAS Reporting Database tables

Table	Description
ApplicationConnections	Contains names and IDs of applications that were used in sessions. The table is linked to the RDSessions table that contains session information.
ComponentsConnections	This table is used in conjunction with the PAC table (PublishingAgentConnections). When a machine is connected, it is connected to a Publishing Agent. This table links a machine to a Publishing Agent.
DBUpgradeHistory	Contains all upgrade versions of the database.
Devices	Contains information about devices that were used to connect to Farm.
Disconnections	Contains the disconnected time of every connection.
Farms	Contains the farm information.
Gateways	This table is linked to the GatewayTypes table and shows all gateways and their types.
GatewayTunnelledConnections	

	This table is linked to the Gateways table and shows the number of connections going through the gateway.
GatewayTypes	Contains gateway types (Normal, Forwarding, Unknown).
Groups	This table contains all groups.
Guests	Keeps a record of the guest VMs that are connected and to which VDI they belong. This table is NOT used in any queries.
HostTypes	Contains all host Types used in the VDIHost table
IdleConnections	Contains the idle time of every connection.
Machines	Stores information about all servers that are connected to the farm. This includes Terminal Servers (TS), Publishing Agents (PA), Gateways (GW), RemotePCs (RPC), VDI hosts (VDIH) and VDI Guests (VDIG).
MachineStateLogs	Keeps a record of various states the machine has been in. For the complete list of states, check the MachineStates table.
MachineStates	This table contains all Machine states used in MachineStateLogs Table.
MachineType	Contains machine types used in the Machines table.
Members	This table is used in conjunction with tables Groups and Users. It allows you to see which users belong to which groups.
MemoryHealth	Stores information about the current server memory usage, which is marked by a timestamp. For detailed information on how to query this table, see the queries section.
Notifications	Contains notifications that were displayed in the RAS Console.
ProcessorHealth	Stores information about the current server processors usage, which is marked by a timestamp. For detailed information on how to query this table, see the queries section.
Protocols	Contains the protocols used in the RDSessions table.
PublishingAgentConnections	This table is used to log how long a PA has been active for. When a PA is still active, the Started and Ended fields are the same. When a disconnected state is received, the engine will set the ended field to the new timestamp.
RDCConnections	This table links with the RDSessions table. A session can have multiple states. To have other two states (Disconnected and Idle), a connection must first be established. Therefore, the other two tables that show these states are in a relationship with RDCConnections table.
RDSessions	Stores a list of established sessions by clients, along with the protocol used (Console or RDP), session type (Desktop, Published Apps, VDI Apps), the user that made the session (all users are pre-populated in the table), and start and ends time.
RemotePC	This table is a placeholder and currently has no columns, except the PK column
SessionTypes	This table contains the Session types for RDSessions table.
Sites	This table contains each site from Farm.
TerminalServers	This table is a placeholder and currently has no columns, except the PK column
TimeZones	Contains all time zones.
Users	Contains all the users logged in.
VDIHost	Keeps a record of the VDI hosts and their types. This table is not used in any queries.Keeps a record of the VDI hosts and their types. This table is not used in any queries.

[See the RAS Reporting database schema.](#)

Please check Navigation Properties for each table in order to clarify which tables it is linked to.

SQL Query example:

```
SELECT us.Username,  
ac.ApplicationName AS [Application Name],  
COUNT(ac.ApplicationName) AS [NumOf Times Used]  
  
FROM  
dbo.Users us INNER JOIN  
dbo.RDSessions rd ON us.SID = rd.User_fk INNER JOIN  
dbo.ApplicationConnections ac ON ac.Session_fk = rd.Session_ID  
  
GROUP BY  
us.Username,  
ac.ApplicationName
```

Referring to the database schema and SQL query itself, you may find that the following is being performed:

- Select usernames from table dbo.Users and gives "us" alias to the table.
- Since dbo.Users table does not have link to dbo.ApplicationConnections, it should first join table dbo.RDSessions using matches on field SID from dbo.Users and User_fk from dbo.RDSessions
- After we have joined dbo.RDSessions (alias "rd") which has a link to dbo.ApplicationConnections, we may also join ApplicationConnections itself with alias "ac" and select Application names and number of launches from it.