stryker

Vocera Report Server Database Schema Guide

Version 5.3.3

Notice

Stryker Corporation or its divisions or other corporate affiliated entities own, use or have applied for the following trademarks or service marks: Stryker, Vocera. All other trademarks are trademarks of their respective owners or holders. The absence of a product or service name or logo from this list does not constitute a waiver of Stryker's trademark or other intellectual property rights concerning that name or logo. Copyright © 2023 Stryker.

Last modified: 2023-02-21 11:26

VRS-533-Docs build 220

Contents

Overview 6
How Report Server Works 6
Writing Database Queries 7
Connecting to the Vocera Report Server Database 7
Java Example 8
Installing Crystal Reports 2008 and MySQL Connector/ODBC 3.51 9
Tables at a Glance 11
addressbookentries 11
audit 11
broadcasts 12
calls 12
deviceinfo 13
devicestatus 13
devicestatuschange 13
dnd 14
failures 14
groupactivity 14
groupconference 14
groups 15
incompletedata 15
inventory 15
locations 15
loginlout 16
lostconnectiondetail 16
lostconnectionlog 16
messageactivity 17
posted 17
radiodetail 17
radiolog 18
recresults 18
sites 18
speechports 19
systeminfo 19
unansweredcalls 19
users 19
vmiactions 20
vmievent 20

vmimessages 20
Creating Custom Reports 22
Adding Custom Reports to the Vocera Report Console 22
Updating the customconfig.xml File 22
Configuring Report Parameters 24
Adding a Hidden Watermark Parameter 24
Adding a Text Field Parameter 25
Adding a Date Range Parameter 25
Adding a Drop-Down List Parameter 25
Adding Cascading Drop-Down List Parameters 25
Adding a Static List Parameter 26
Adding a Multilist Parameter 27
Writing SQL Queries to Define Report Parameters 27
Step-by-Step Guide to Custom Reports 28
Creating a Custom Report from a Standard Report 28
Creating a Custom Report from a SQL Query 30
Troubleshooting Problems with Custom Reports 37
Database Details 39
Vocera Usage Data 39
addressbookentries 40
audit 40
broadcasts 41
calls 42
deviceinfo 43
devicestatus 44
devicestatuschange 44
dnd 45
failures 45
groupactivity 46
groupconference 47
groups 48
incompletedata 48
inventory 49
locations 49
loginlogout 50
lostconnectiondetail 50
lostconnectionlog 51
messageactivity 52
posted 53
radiodetail 53
radiolog 54
recresults 55
sites 56
speechports 56
systeminfo 57

unansweredcalls..... 58
users..... 58
vmiactions..... 59
vmievent..... 60
vmimessages..... 61
Report Server Application Data..... 61

Overview

This chapter provides an overview to the Vocera Report Server database schema. It contains the following sections:

- How Report Server Works
- Writing Database Queries
- Connecting to the Vocera Report Server Database
- Installing Crystal Reports 2008 and MySQL Connector/ODBC 3.51
- Tables at a Glance

How Report Server Works

This document describes the tables used by the Vocera Report Server to generate reports based on data gathered as people use the Vocera system. The Vocera server records events (for example, the start of a badge call, Genie prompts, the end of a call) in log files. The Report server parses these files into database tables and queries the tables to generate data for reports.

The following figure illustrates Report Server functionality. It shows how data is loaded from the Vocera Server into the Report Server database. Report definition files (including custom reports) are created using the Crystal Reports Designer. The Report Console generates reports.

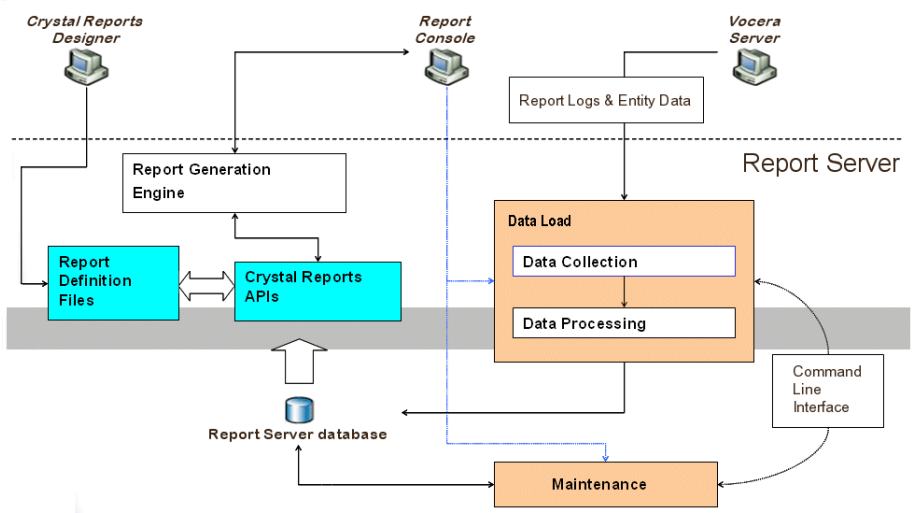


Figure 1: Report Server diagram

The files that define Vocera reports (for example, `DailySystemCallVolume.rpt`) are installed by default in the `\vocera\reports\Reports` directory when you install the Vocera Report Server. To read and study these files as a basis for your own custom reports, you must also install Crystal Reports. See Installing Crystal Reports 2008 and MySQL Connector/ODBC 3.51.

Writing Database Queries

You can use the schema information to write queries and create custom reports based on the same data. For example, the following query returns a list of the groups on a Vocera server by extracting values from the GroupName column of the `groups` table:

```
select GroupName from groups;
```

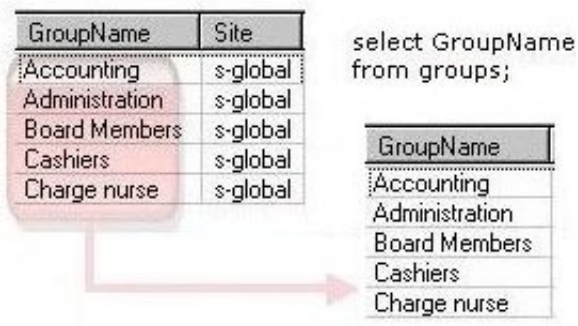


Figure 2: Querying groups

Here's an example of how to get related data from two tables. The following query returns speech recognition scores for a specified user:

```
select r.score, u.FirstName, u.LastName
from recresults r, users u
where r.UserID='rhall' and u.UserID='rhall';
```

This query returns rows from both tables where the value of the UserID field is rhall.

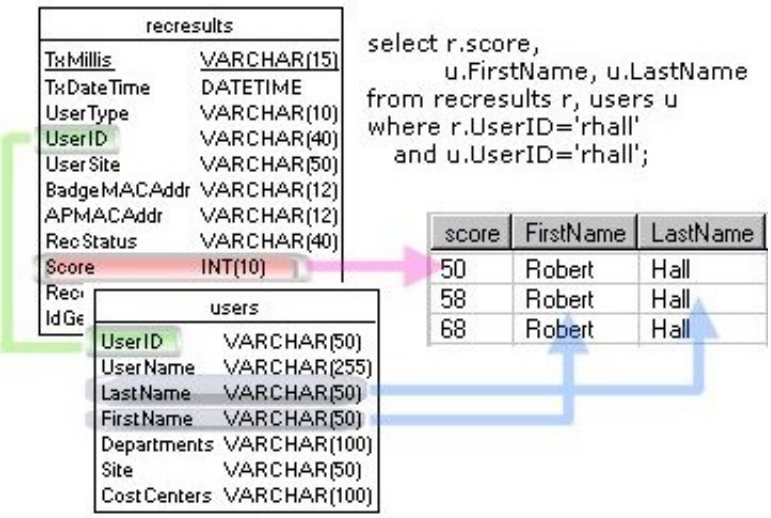


Figure 3: Querying related data from two tables

Connecting to the Vocera Report Server Database

The following table lists connection parameters for the Vocera Report Server Database.

Table 1: Database connection parameters

Parameter	Value
Database name	vocera_reports

Parameter	Value
User name	vocera
Password	vocera

Java Example

The following code example shows how to connect to the database and execute a query from a Java application.

```
import java.sql.*;
//import java.util.*;

public class VRSConnection {
  // Replace "vrsHost" with the host name or IP address
  // of the Vocera Report Server.
  static String sHost = "vrsHost";
  static String sUsername = "vocera";
  static String sPassword = "vocera";
  static String sThinConn = "jdbc:mysql://" + sHost +
                            "/vocera_reports";
  static String driverClass = "com.mysql.jdbc.Driver";

  public static Connection getConnection()
                                  throws SQLException {
    Connection c = null;
    try {
      Class.forName(driverClass).newInstance();
      c = DriverManager.getConnection(sThinConn,
                                      sUsername,
                                      sPassword);
    }
      catch (Exception e) {
      e.printStackTrace(System.out);
    }
    return c;
  }

  public static void main(String argv[]) {
    try {
      Connection conn = getConnection();
      if (conn == null) {
        System.out.println("No connection."); System.exit(0);
      }
      else {
        System.out.println("Connected.");
        Statement stmt = conn.createStatement();
        String query = "SELECT SITENAME FROM SITES";
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            System.out.println(rs.getString("SiteName"));
        }
        conn.close();
        System.out.println("Connection closed.");
      }
    }
    catch (SQLException sqle) {
      sqle.printStackTrace();
    }
```

```
  }
}
```

Figure 4: Connecting to the database in a Java program

Installing Crystal Reports 2008 and MySQL Connector/ODBC 3.51

Install Crystal Reports 2008 Release 2 on a different computer from the Vocera Report Server. To connect from Crystal Reports to your Vocera Report Server, you need to install the MySQL Connector/ODBC 3.51 driver and set up an ODBC data source.

Important:
DO NOT install Crystal Reports Server on the same machine as Vocera Report Server. Vocera Report Server has not been certified to work on computers where Crystal Reports Server is also installed.

To install Crystal Reports 2008 Release 2 and MySQL Connector/ODBC 3.51:

1. Download the MySQL Connector/ODBC 3.51 driver from the following location:
   http://dev.mysql.com/downloads/connector/odbc/3.51.html

   Note:
   Vocera Report Server requires MySQL Connector/ODBC 3.51.27 or later.

2. Install the MySQL Connector/ODBC 3.51 driver.
3. Set up an ODBC data source to connect to your Vocera Report Server:
   a. Open the ODBC Data Source Administrator, which is usually found under **Start > Programs > Administrative Tools > Data Sources (ODBC),** or **Start > Settings > Control Panel > Data Sources (ODBC).** The ODBC Data Source Administrator dialog box opens.
   b. Click the System DSN tab.
   c. Click **Add**. The Create New Data Source dialog box opens.
   d. Select the MySQL 0DBC 3.51 driver from the list.
   e. Click **Finish.**
   f. The Connector/ODBC 3.51.x dialog box opens with the Login tab selected. Enter the following information:

Field	Value
Data Source Name	VoceraReports
Description	Vocera Report Server connection
Server	[IP address of the Vocera Report Server machine]
User	vocera
Password	vocera
Database	vocera_reports

   Note:
   The User and Password values above are the default values for Vocera Report Server. If you changed the User and Password on the Vocera Report Server, use those values instead.

   The Connector/ODBC 3.51.x dialog box should look like this:

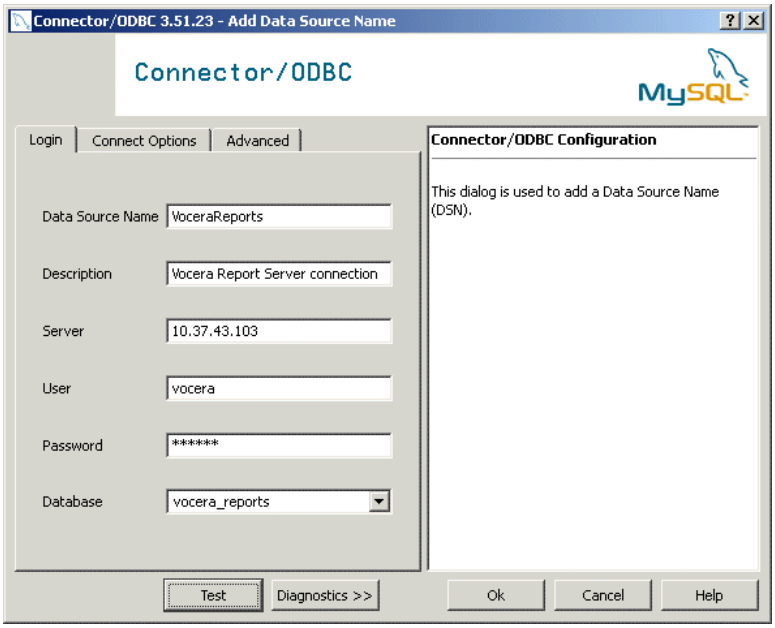


Figure 5: Connector/ODBC 3.51.x dialog box

g. Click **OK** to save the connection and close the Connector/ODBC 3.51.x dialog box.
h. In the ODBC Data Source Administrator dialog box, click **OK.**

4. Install Crystal Reports 2008 Release 2. For detailed instructions, see the separate Crystal Reports documentation.
5. When prompted, restart the computer.
6. After the computer restarts, run Crystal Reports.
7. Log on to the VoceraReports data source:
   a. Choose **File > Log On or Off Server.** The Data Explorer dialog box opens.
   b. Expand the Create New Connection folder, and then expand the ODBC (RDO) folder. The ODBC (RDO) dialog box opens.
   c. Make sure the **Select Data Source** radio button is selected. In the **Data Source Name** list, select VoceraReports. Click **Next.**

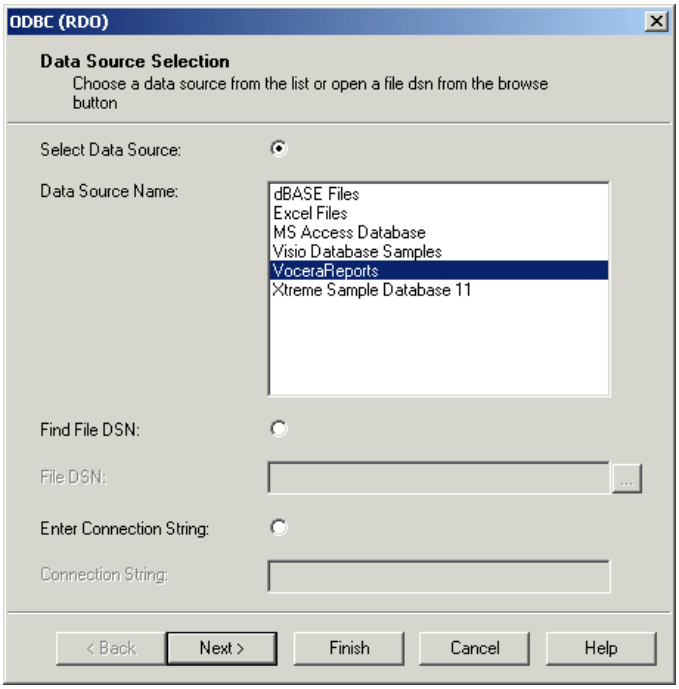


Figure 6: ODBC (RDO) dialog box

d. Enter the **UserID** and **Password.** By default, both values are "vocera". If you changed the **UserID** and **Password** on the Vocera Report Server, use those values instead.

e. Click **Finish.**

Tables at a Glance

The Report server generates reports from logs and user data acquired from the Vocera server. This section provides an overview of of tables of Vocera usage data maintained in the Vocera Report Server database. Tables containing Report Server application data are excluded. Underlined columns indicate primary keys for the table.

addressbookentries

Column	Type
LastName	VARCHAR(50)
FirstName	VARCHAR(50)
Type	VARCHAR(50)
Site	VARCHAR(50)
Phone	VARCHAR(50)
Pager	VARCHAR(50)
DeletedFlag	INT(1)

audit

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserId	VARCHAR(70)

Column	Type
UserSite	VARCHAR(50)
LoginId	VARCHAR(50)
Operation	VARCHAR(30)
ModifiedEntityType	VARCHAR(20)
ModifiedEntityId	VARCHAR(70)

broadcasts

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserId	VARCHAR(70)
UserSite	VARCHAR(50)
GroupName	VARCHAR(50)
GroupSite	VARCHAR(50)
Duration	BIGINT(15)
IdBroadcast	BIGINT(15)
IdGenie	BIGINT(15)
ModifiedEntityId	VARCHAR(70)

calls

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
CalledType	VARCHAR(10)
Called	VARCHAR(40)
CalledName	VARCHAR(100)
CalledSite	VARCHAR(50)
Accepted	INT(1)
AcceptedByType	VARCHAR(10)
AcceptedBy	VARCHAR(40)
AcceptedByName	VARCHAR(100)
AcceptedBySite	VARCHAR(50)
DeviceType	VARCHAR(64)
Duration	BIGINT(15)

Column	Type
IdCall	BIGINT(20)
IdGenie	BIGINT(10)

deviceinfo

Column	Type
DeviceID	VARCHAR(20)
BadgeMACAddr	VARCHAR(12)
SerialNo	VARCHAR(20)
Label	VARCHAR(30)
Type	VARCHAR(15)
Color	VARCHAR(20)
Owner	VARCHAR(50)
Site	VARCHAR(50)
TrackingDate	DATE
BadgeUsage	TINYINT(1)
Notes	TEXT
DeviceCode	INT(5)
FirmwareVersion	INT(5)
LastStatus	VARCHAR(50)
LastUser	VARCHAR(70)
LasLastLocation	VARCHAR(50)
LastAccessed	TIMESTAMP
InternalSerialNo	VARCHAR(20)

devicestatus

Column	Type
Status	VARCHAR(100)
Description	TEXT
DeletedFlag	INT(1)

devicestatuschange

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
DeviceID	VARCHAR(20)
DeviceSite	VARCHAR(50)
Status	VARCHAR(30)
Notes	TEST

dnd

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
DND	INT(1)

failures

Column	Type
TxMillis	BIGINT(15)
TxDate	DATE
TxDateTime	DATETIME
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
BadgeMACAddr	VARCHAR(12)
Failuretext	VARCHAR(50)

groupactivity

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
Operation	VARCHAR(64)
GroupName	VARCHAR(50)
GroupSite	VARCHAR(50)
PartyID	VARCHAR(70)
PartySite	VARCHAR(50)
IdGenie	BIGINT(15)

groupconference

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserID	VARCHAR(70)

Column	Type
UserSite	VARCHAR(50)
GroupName	VARCHAR(50)
GroupSite	VARCHAR(50)
Duration	BIGINT(15)
IdGroupConf	BIGINT(15)
IdGenie	BIGINT(15)

groups

Column	Type
GroupID	VARCHAR(50)
GroupName	VARCHAR(50)
Site	VARCHAR(50)
CostCenter	VARCHAR(100)
DeletedFlag	INT(1)
UserMembers	MEDIUMTEXT
Department	VARCHAR(50)

incompletedata

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
Type	VARCHAR(20)
ReportData	TEXT

inventory

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
BadgeMACAddr	VARCHAR(12)
APMACAddr	VARCHAR(12)
Voltage	DECIMAL(7,2)
UIState	VARCHAR(15)
SerialNo	VARCHAR(30)

locations

Column	Type
MACAddress	VARCHAR(20)
LocationName	VARCHAR(50)

Column	Type
Site	VARCHAR(50)
DeletedFlag	INT(1)

loginlout

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
UserSite	VARCHAR(50)
Operation	VARCHAR(64)
DeviceMACAddr	VARCHAR(12)
APMACAddr	VARCHAR(12)
LoginUserCount	INT(11)
TotalLic	INT(11)

lostconnectiondetail

Column	Type
id	INT(11)
TxParentMillis	VARCHAR(15)
TxMillis	BIGINT(15)
TxDateTime	DATETIME
APMacAddress	VARCHAR(12)
OtherParty	VARCHAR(20)
Type	VARCHAR(10)

lostconnectionlog

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
BadgeMacAddress	VARCHAR(12)
BadgeBinary	TEXT
SFAPCount	SMALLINT(6)
CAPCount	SMALLINT(6)
LSCount	SMALLINT(6)
SFSCount	SMALLINT(6)

Column	Type
SFDHCPCount	SMALLINT(6)
Duration	SMALLINT(6)

messageactivity

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
Operation	VARCHAR(64)
Parties	VARCHAR(512)
Parties	VARCHAR(512)
Duration	BIGINT(15)
IdMsgActivity	BIGINT(15)
IdGenie	BIGINT(15)

posted

Column	Type
reportName	VARCHAR(50)
reportTime	BIGINT(15)

radiodetail

Column	Type
id	INT(11)
TxParentMillis	VARCHAR(15)
TxMillis	BIGINT(15)
TxDateTime	DATETIME
APMacAddress	VARCHAR(12)
OtherParty	VARCHAR(20)
PacketsReceived	SMALLINT(6)
PacketsMissed	SMALLINT(6)
CQ	SMALLINT(6)
RoamTime	MEDIUMINT(9)
Type	SMALLINT(1)
Reserved1	MEDIUMINT(9)
Reserved2	MEDIUMINT(9)
Reserved3	MEDIUMINT(9)

radiolog

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
BadgeMacAddress	VARCHAR(12)
BadgeBinary	TEXT
PacketsReceived	SMALLINT(6)
PacketsMissed	SMALLINT(6)
AvgCQ	SMALLINT(6)
JBhit	MEDIUMINT(9)
JBMissed	MEDIUMINT(9)
Type	SMALLINT(1)
Reserved1	MEDIUMINT(9)
Reserved2	MEDIUMINT(9)
Reserved3	MEDIUMINT(9)

recresults

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
BadgeMACAddr	VARCHAR(12)
APMACAddr	VARCHAR(12)
RecStatus	VARCHAR(40)
Score	INT(10)
Recognized	VARCHAR(50)
Grammar	VARCHAR(64)
DeviceType	VARCHAR(64)
IdGenie	BIGINT(15)

sites

Column	Type
SiteId	VARCHAR(50)
SiteName	VARCHAR(50)
DeletedFlag	INT(1)

speechports

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
UserType	VARCHAR(20)
UserID	VARCHAR(70)
UserSite	VARCHAR(50)
Duration	MEDIUMINT(8)
SpeechPort	INT(5)
IdGenie	BIGINT(15)

systeminfo

Column	Type
CompanyName	VARCHAR(150)
ServerIp	VARCHAR(120)
BackupInfo	TEXT
DataLoadInfo	TEXT
Password	VARCHAR(100)
MailInfo	TEXT
SweepInfo	TEXT

unansweredcalls

Column	Type
TxMillis	BIGINT(15)
Called	VARCHAR(40)
CallID	BIGINT(20)
GenieID	BIGINT(10)
ReasonID	INT(2)
Reason	VARCHAR(25)

users

Column	Type
UserID	VARCHAR(70)
UserName	VARCHAR(255)
LastName	VARCHAR(50)
FirstName	VARCHAR(50)
Departments	VARCHAR(100)
Site	VARCHAR(50)
CostCenters	VARCHAR(100)
Email	VARCHAR(100)

Column	Type
DeletedFlag	INT(1)

vmiactions

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
ClientID	VARCHAR(50)
MessageID	BIGINT(20)
RecipientType	VARCHAR(20)
RecipientSite	VARCHAR(50)
RecipientID	VARCHAR(70)
RecipientName	VARCHAR(100)
ActionType	VARCHAR(50)
ActionDetail	TEXT

vmievent

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE
ClientID	VARCHAR(50)
EventID	BIGINT(20)
EventType	VARCHAR(30)
Item1	VARCHAR(64)
Item2	VARCHAR(64)
Item3	VARCHAR(64)
Item4	VARCHAR(64)
Item5	VARCHAR(64)
Item6	VARCHAR(64)
Item7	VARCHAR(64)
Item8	VARCHAR(64)
Item9	VARCHAR(64)
Item10	VARCHAR(64)

vmimessages

Column	Type
TxMillis	BIGINT(15)
TxDateTime	DATETIME
TxDate	DATE

Column	Type
ClientID	VARCHAR(50)
DestinationType	VARCHAR(20)
DestinationSite	VARCHAR(50)
DestinationID	VARCHAR(70)
DestinationID	VARCHAR(70)
DestinationName	VARCHAR(100)
MessageID	BIGINT(20)
Priority	VARCHAR(20)
MessageText	TEXT

Creating Custom Reports

This chapter describes how to use Crystal Reports to create custom report definition files (*.RPT) add then add them to the Report Console. It contains the following sections:

- Adding Custom Reports to the Vocera Report Console
- Updating the customconfig.xml File
- Configuring Report Parameters
- Step-by-Step Guide to Custom Reports
- Troubleshooting Problems with Custom Reports

Adding Custom Reports to the Vocera Report Console

Use the Crystal Reports designer to define a report's data source, select and group the data records to use, and format the report. After you define a report, you can add it to the Vocera Report Console to include with other Vocera Report Server reports.

Attention: Custom Report Server database tables are not supported. Custom database tables are not backed up or restored after an upgrade. Vocera supports only the database tables that are installed with Report Server.

For information about using Crystal Reports, see the separate Crystal Reports documentation.

To add a custom report to the Vocera Report Console:

1. Use Crystal Reports to define the report definition file (*.rpt).
2. Place the report definition file in the `\vocera\reports\Reports\custom` directory.
3. Update the `customconfig.xml` file to include the new report in the Vocera Report Console. See Updating the customconfig.xml File.
4. From the Windows Start menu, select **Settings > Control Panel > Administrative Tools > Services**. The Services window appears.
5. Stop the Tomcat service and then start it again.
6. Close the Services window.

Updating the customconfig.xml File

The `customconfig.xml` file, which is located in the `\vocera\reports\Reports\custom` folder, specifies the custom reports that appear in the Report Console. To add a custom report to the Report Console, you must update `customconfig.xml`. You can use any XML editor to modify the file.

When you specify a custom report configuration in `customconfig.xml`, you must select one of the valid report types. If you specify a report type other than “CUSTOM” (for example, “SYSTEM” or “EXPORT”), the report is added to end of the corresponding list of standard reports in the Report Console. You cannot create new report types or subtypes.

The `customconfig.xml` file has the following XML structure:

```
<REPORTCONFIG>
  <REPORT ID="reportId">
    <NAME></NAME>
    <DESCRIPTION></DESCRIPTION>
    <FILENAME>filename.rpt</FILENAME>
    <TYPE></TYPE>
    <PARAMETERS>
      <PARAMETER>
        <DISPLAYNAME></DISPLAYNAME>
        <BINDNAME></BINDNAME>
        <TYPE></TYPE>
        <DEFAULT></DEFAULT>
      </PARAMETER>
    </PARAMETERS>
  </REPORT>
</REPORTCONFIG>
```

Figure 7: XML structure for customconfig.xml

The `customconfig.xml` file uses the same XML schema as the `reportconfig.xml` located in the `\vocera\reports` folder. The best way to become familiar with the `reportconfig.xml` schema is to open the file and see how existing reports have been configured. You can copy and paste an existing report configuration from `reportconfig.xml` into `customconfig.xml`, and then modify the configuration to add a new report.

The following table describes the XML elements of `customconfig.xml`.

Table 2: customconfig.xml schema elements

Element	Description
REPORTCONFIG	Root element.
REPORT	A report definition. Parameters: • ID = report ID. The ID can be identical to the value of the NAME element. Its purpose is merely to make the XML more readable.
NAME	Name of the report. The NAME value appears in the Vocera Report Console.
DESCRIPTION	Description of the report. The DESCRIPTION value appears below the name of the report in the Vocera Report Console.
FILENAME	Report definition file. The file must be placed in the \vocera\reports\Reports directory.
TYPE	Type of report. TYPE can be any of the following values: • SYSTEM = Summary reports • CALL = Call reports • DIAG = Diagnostics reports • ASSET = Asset Tracking reports • EXPORT = Export Data reports • DEVICEMANAGEMENT = Device reports • SCHEDULERDIAGNOSTICS = Scheduler Diagnostics reports • CUSTOM = Custom reports that you create with Crystal Reports Designer. **Note:** You cannot define new report types.
PARAMETERS	Parent element containing the report's parameters.

Element	Description
PARAMETER	A report parameter, which appears in the Report Parameters page when you generate the report from Vocera Report Console. Subelements include: DISPLAYNAME, BINDNAME, TYPE, DEFAULT, DBLOOKUP, and LOOKUP.
DISPLAYNAME	The name of the parameter as it appears in Vocera Report Console.
BINDNAME	The name of the parameter in the RPT file to which this parameter is bound.
TYPE	Type of parameter (for example, TEXT or DATE).
DEFAULT	Default value of the parameter.
DBLOOKUP	Database lookup query for the parameter's values.
LOOKUP	Defines a static lookup table for the parameter's values.

Configuring Report Parameters

This section provides several examples showing how to define report parameters in the `customconfig.xml` file. These parameters appear on the Report Parameters page when you generate the report in the Vocera Report Console.

The following sections describe different types of report parameters:

- Adding a Hidden Watermark Parameter
- Adding a Text Field Parameter
- Adding a Date Range Parameter
- Adding a Drop-Down List Parameter
- Adding Cascading Drop-Down List Parameters
- Adding a Static List Parameter
- Adding a Multilist Parameter
- Writing SQL Queries to Define Report Parameters

Adding a Hidden Watermark Parameter

The following example shows how to add a hidden watermark parameter to a report. The text of the watermark is specified in the `db.properties` file using the following convention:

`Watermark=`*WatermarkText*

where WatermarkText is a string.

```
<PARAMETER>
  <DISPLAYNAME>WatermarkText</DISPLAYNAME>
  <BINDNAME>WatermarkText</BINDNAME>
  <TYPE>HIDDEN</TYPE>
  <DEFAULT>$OptText</DEFAULT>
</PARAMETER>
```

Figure 8: Defining a hidden watermark parameter

Adding a Text Field Parameter

The following example shows how to display a free form Text field on the Report Parameters page:

```
<PARAMETER>
  <DISPLAYNAME>Name</DISPLAYNAME>
  <BINDNAME>UserName</BINDNAME> 1
  <TYPE>TEXT</TYPE>
</PARAMETER>
```

Figure 9: Defining a parameter for a Text field

1 UserName is the name of the parameter in the RPT file.

Adding a Date Range Parameter

The following example shows how to display a date range field on the Report Parameters page:

```
<PARAMETER>
  <DISPLAYNAME>Report Date</DISPLAYNAME>
  <BINDNAME>ReportDate</BINDNAME> 1
  <TYPE>DATE</TYPE>
</PARAMETER>
```

Figure 10: Defining a parameter for a date range field

1 ReportDate is the name of the parameter in the RPT file.

Adding a Drop-Down List Parameter

The following example shows how to display a drop-down list field on the Report Parameters page. The list is populated from the database.

```
<PARAMETER>
  <DISPLAYNAME>Called Type</DISPLAYNAME>
  <BINDNAME>CalledType</BINDNAME> 1
  <TYPE>DROPLIST</TYPE>
  <DBLOOKUP>select SiteID as code,
  SiteName as description from sites</DBLOOKUP> 2
</PARAMETER>
```

Figure 11: Defining a drop-down list parameter

1 CalledType is the name of the parameter in the RPT file.

2 Description will be displayed in the list. Code is the corresponding value for the parameter.

Adding Cascading Drop-Down List Parameters

A cascading drop-down list is one with choices that change based on the value a user selects in another list field. In the following example, the Owners Site drop-down list parameter is a parent of the Device Owner drop-down list parameter.

```
<PARAMETER>
  <DISPLAYNAME>Owners Site</DISPLAYNAME>
  <BINDNAME>ownerssite</BINDNAME>1
  <TYPE>DROPLIST</TYPE>
  <DBLOOKUP>(select SiteName code, SiteName description
    from sites) union (select '||||', 'All Sites' description )
    Order by description</DBLOOKUP>
</PARAMETER>
<PARAMETER>
  <DISPLAYNAME>Device Owner</DISPLAYNAME>
```

```
    <BINDNAME>owning</BINDNAME>2
    <TYPE>DROPLIST</TYPE>
    <DBLOOKUP>(select DeviceInfo.Owner as code, Groups.GroupName
      as description from DeviceInfo, Groups, Sites
      Where Groups.GroupID = DeviceInfo.Owner and
      Groups.Site = Sites.SiteID and DeviceInfo.Owner <> ''
      and Sites.SiteName like replace(?,'||||','%%'))union
      (select '||||', ' All Owners' description )
      Order by description
    </DBLOOKUP>
    <PARENT>ownerssite</PARENT>3
  </PARAMETER>
```

Figure 12: Defining parameters for cascading drop-down lists

1 `ownerssite` is the name of the parameter in the RPT file.

2 `owning` is the name of the parameter in the RPT file.

3 Specifies that the parent list of `owning` is `ownerssite`.

Adding a Static List Parameter

The following example shows how to display a static drop-down list field on the Report Parameters page:

```
  <PARAMETER>
    <DISPLAYNAME>Completed</DISPLAYNAME>
    <BINDNAME>Accepted</BINDNAME> 1
    <TYPE>TEXT</TYPE>
    <LOOKUP>
      <OPTION>
        <CODE>1</CODE> 2
        <DESCRIPTION>True</DESCRIPTION> 3
      </OPTION>
      <OPTION>
        <CODE>0</CODE>
        <DESCRIPTION>False</DESCRIPTION>
      </OPTION>
    </LOOKUP>
  </PARAMETER>
```

Figure 13: Defining a parameter for a static list

1 Accepted is the name of the parameter in the RPT file.

2 The CODE element is the corresponding value for the parameter.

3 The DESCRIPTION element will be displayed in the list.

Adding a Multilist Parameter

The following example shows how to display a multilist field on the Report Parameters page:

```
<PARAMETER>
  <DISPLAYNAME>Device Status</DISPLAYNAME>
  <BINDNAME>status</BINDNAME>
  <TYPE>MULTILIST</TYPE>
  <DBLOOKUP>(select Status code,1
    Status description from devicestatus )
    union (select '||||', ' All Device Status'
    description ) Order by
    description</DBLOOKUP>
</PARAMETER>
```

Figure 14: Defining a parameter for a multilist field

1 Database lookup query that returns status descriptions from the devicestatus table.

Writing SQL Queries to Define Report Parameters

More advanced Crystal Reports users can write custom SQL queries to define reports, with `WHERE` clauses that are evaluated and populated with data based on parameters that you specify. Many of the Report Server reports are defined this way. Open the `reportconfig.xml` file located in the `\vocera\reports` folder to see how Report Server parameters are specified.

For example, the following SQL command has a `WHERE` clause that references a report parameter named `DateRange_Clause`:

```
SELECT TxDate, BadgeMacAddr, RecStatus, Score, Recognized
from RecResults WHERE {DateRange_Clause}
```

Figure 15: SQL query with a WHERE clause

In this example, `{DateRange_Clause}` is populated by creating two report parameters, which must also be specified in the `customconfig.xml` file:

- The first parameter inputs the report date as a `daterange` parameter (the From and Through dates).

  ```
  <PARAMETER>
    <DISPLAYNAME>Report Date</DISPLAYNAME>
    <BINDNAME>ReportDate</BINDNAME> 1
    <TYPE>DATE</TYPE>
  </PARAMETER>
  ```

  1 ReportDate is the name of the parameter in the RPT file.
- The second parameter is a hidden parameter that uses Vocera's `$BuildClause()` macro to evaluate the expression for `{DateRange_Clause}` as `(RecResults.TxDate >= ReportDate.Lowerbound and RecResults.TxDate <= ReportDate.Upperbound)`.

  The `$BuildClause()` macro is used to build a `WHERE` clause for a SQL query. It takes the following arguments:
  - Column name in the SQL to bind
  - Name of the input parameter
  - Operation. Specify any of the following operations:
    - `daterange`: Used to generate Date range expressions. [column > lowerbound and column < upperbound]
    - `stringequals`: Used to generate an expression for Text parameters [column = parameter value]
    - `list-in`: Used to generate expressions for MultiList parameters [column in (P1, P2, P3)]

  Here is an example of a hidden parameter with the `$BuildClause()` macro:

```
<PARAMETER>
  <DISPLAYNAME>Hidden Where Clause</DISPLAYNAME>
  <BINDNAME>DateRange_Clause</BINDNAME>
  <TYPE>HIDDEN</TYPE>
  <DEFAULT> $BuildClause(RecResults.TxDate, ReportDate,
    daterange)</DEFAULT>
</PARAMETER>
```

Step-by-Step Guide to Custom Reports

This section provides step-by-step tutorials on how to create custom reports for Report Server. You can follow these basic steps to create your own reports or modify the standard Vocera reports.

Crystal Reports allows you to create reports using several different methods. You can build reports from scratch, or you can use the four Report Creation wizards, which provide a series of screens to help you define a report.

This section assumes you are familiar with Crystal Reports. For more in depth coverage of Crystal Reports features, see the Crystal Reports 2008 User's Guide.

Before starting these tutorials, prepare your Crystal Reports computer:

- Make sure you have copied the Vocera report files (`*`.RPT) from the Report Server computer to your Crystal Reports designer computer. The files are located in the `\vocera\reports\Reports` folder.
- Install the MySQL Connector/ODBC 3.51 driver, and set up an ODBC data source to connect to your Vocera Report Server. See Installing Crystal Reports 2008 and MySQL Connector/ODBC 3.51.

This section has the following step-by-step tutorials:

- Creating a Custom Report from a Standard Report
- Creating a Custom Report from a SQL Query

Creating a Custom Report from a Standard Report

This tutorial shows how to save one of the standard Vocera reports, the Broadcasts report, as a new report and then modify it. This is the quickest way to use Crystal Reports designer to create a custom report.

To modify a standard Vocera report:

1. In Crystal Reports, choose **File > Open**. The Open dialog box appears.
2. Navigate to the folder where Vocera reports are saved, and select `Broadcasts.rpt`. Click **Open**.
   The Broadcasts to Groups report opens in the Crystal Reports designer:

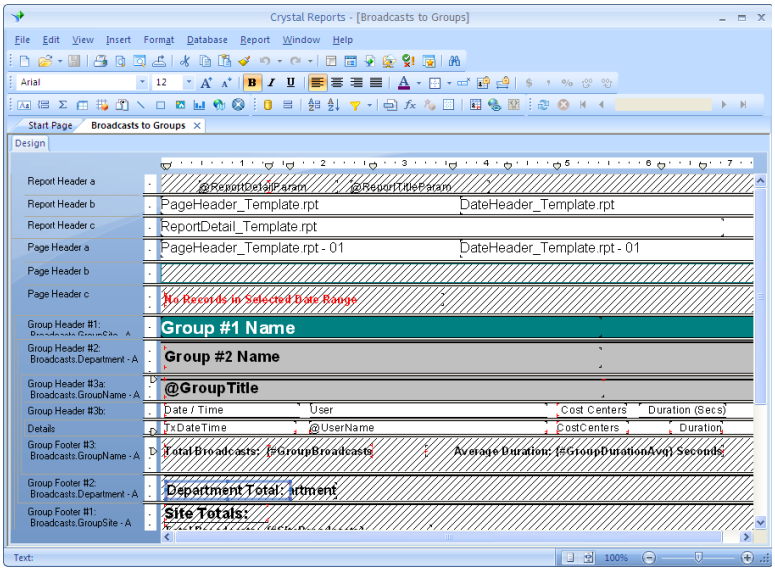


Figure 16: Broadcasts to Groups report

3. Choose **File > Save As**. Save the file as `BroadcastsCustom.rpt`.
4. Choose **View > Field Expolorer**.In the Field Explorer, expand the **Database Fields** to show the **Broadcasts** table.
5. Expand the **Broadcasts** table. Click and Drag the **UserType** field into the **Details** section of the report between **UserName** and **CostCenters**. Resize the fields to fit them on the report.
6. Choose **File > Save** to save the report.
7. Copy the `BroadcastsCustom.rpt` file to the `\vocera\reports\Reports\custom` folder on the Report Server computer.

   Now you are ready to add the report to the `customconfig.xml` file.
8. On the Report Server computer, navigate to the `\vocera\reports\Reports\custom` folder and open the `customconfig.xml` file in a text editor.
9. Add the following new report configuration to `customconfig.xml`. This configuration is nearly the same as the existing one for the existing Broadcasts report. You can copy it from `reportconfig.xml`, which is located in the `\vocera\reports` folder. You should make changes only to to the following elements in the configuration and save.

```
<REPORT ID="BroadcastsCustom">
  <NAME>Custom Report - Broadcasts</NAME>
  <DESCRIPTION>This report shows which groups are
    receiving Broadcasts.</DESCRIPTION>
  <FILENAME>custom/BroadcastsCustom.rpt</FILENAME>
  <TYPE>CUSTOM</TYPE>
```

Figure 17: Updates to the customconfig.xml

There are many different parameters in this report configuration. For information about each type of parameter, see Configuring Report Parameters.

10. From the Windows Start menu, select **Settings > Control Panel > Administrative Tools > Services**. The Services window appears.
11. Stop the Tomcat service and then start it again.
12. Close the Services window.
13. Log into the Report Console.
14. Click **Custom Reports**.

15. Select **Custom Report - Broadcasts** and then click **Generate**.
    The Report Parameters page appears.
16. Specify the report date range and site, and then click **Generate**.
    The report should look like the following screen.

vocera COMMUNICATIONS

4/1/06 - 5/23/06

Broadcasts to Groups

Site: North Little Rock

Case Coordination

Date / Time	User	UserType	Cost Centers	Duration (Secs)
4/6/06 6:43:59AM	Danner, Tamara	User		6
5/17/06 10:57:25AM	Hart, Bill	User	2008323	25
Total Broadcasts: 2				**Average Duration: 16 Seconds**

Housekeeping

Date / Time	User	UserType	Cost Centers	Duration (Secs)
4/7/06 5:38:34AM	Services Two, Linen	User	42	17
4/7/06 6:01:34AM	Weiland, Liz	User		12
4/7/06 7:24:55AM	Services, Trash	User		15
4/17/06 11:31:58AM	Services Two, Linen	User	42	16
4/19/06 4:55:53AM	Services Two, Linen	User	42	9
4/21/06 4:54:45AM	Services Two, Linen	User	42	25
4/21/06 5:08:05AM	Services Two, Linen	User	42	7
4/21/06 7:05:07AM	Services Two, Linen	User	42	49
4/21/06 7:12:39AM	Services Two, Linen	User	42	12
4/21/06 9:49:21AM	Services Two, Linen	User	42	10
4/24/06 8:39:27AM	Weiland, Liz	User		33
4/24/06 11:22:21AM	Weiland, Liz	User		19
4/24/06 11:35:50AM	Lawrence, Shirley	User		14
4/24/06 1:07:11PM	Lawrence, Shirley	User		9
4/26/06 6:45:04AM	Weiland, Liz	User		40
4/27/06 7:31:34AM	Services Two, Linen	User	42	5
4/27/06 12:58:55PM	Lawrence, Shirley	User		23
4/30/06 9:48:40AM	Weiland, Liz	User		12
4/30/06 10:49:13AM	Weiland, Liz	User		11
4/30/06 11:14:39AM	Housekeeper, Second Floor	User		8
5/6/06 6:54:17PM	Housekeeper, Second Floor	User		5
5/16/06 7:02:19AM	Lawrence, Shirley	User		17
Total Broadcasts: 22				**Average Duration: 17 Seconds**

Linen Services

Date / Time	User	UserType	Cost Centers	Duration (Secs)
4/24/06 8:48:46AM	Weiland, Liz	User		10
Total Broadcasts: 1				**Average Duration: 10 Seconds**

Figure 18: Broadcasts report

Creating a Custom Report from a SQL Query

This tutorial shows how to create a report from scratch using a SQL query. It is for more sophisticated users who are both familiar with the Crystal Reports designer and experienced with SQL. The tutorial explains how one of the standard Report Server reports, the Inactive Users report, was created. You can use the steps from this tutorial to create your own reports.

To create a custom report from scratch using a SQL query:

1. In Crystal Reports, choose **File > New > Blank Report**. The Database Expert dialog box appears.

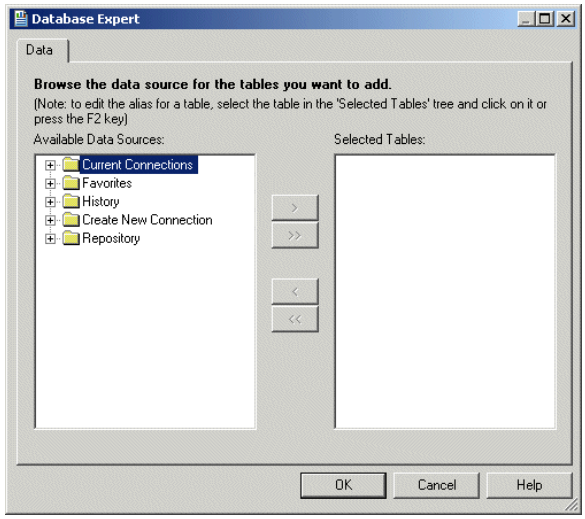


Figure 19: Database Expert dialog box

2. In the **Available Data Sources** box, open the **Create New Connection** folder, and then open the **ODBC (RDO)** folder.
   The Data Source Selection dialog box appears.

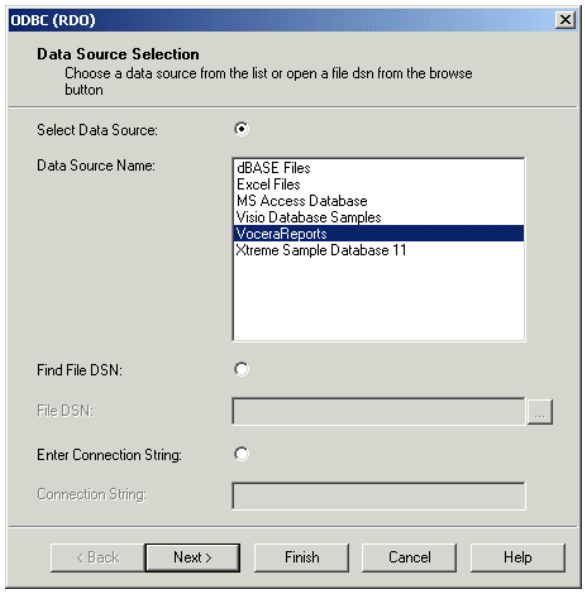


Figure 20: Data Source Selection dialog box

3. Select **VoceraReports**, and then click **Next**.
   The Connection Information dialog box appears.

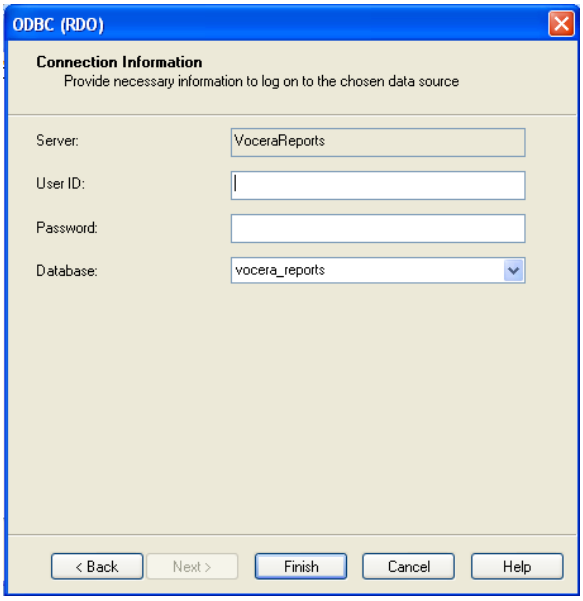


Figure 21: Connection Info dialog box

4. For both the **User ID** and **Password** fields, enter "vocera". Click **Finish**.
   The Connection Information dialog box closes, leaving the Database Expert.
5. In the **VoceraReports** folder, double-click **Add Command**. The Add Command To Report dialog box appears.

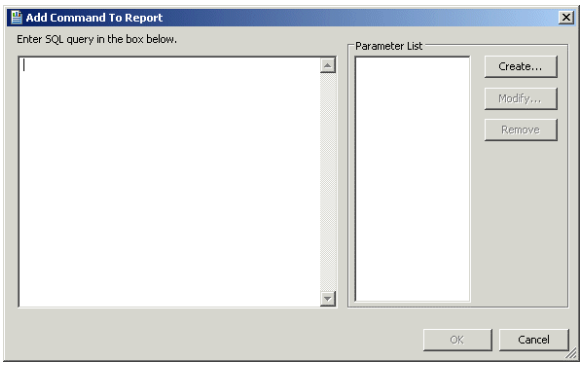


Figure 22: Add Command To Report dialog box

6. Enter the following SQL command in the box. This is the SQL command for the Inactive Users report:

```
select * from users, sites where users.site=sites.siteID
and users.deletedflag = 0 and userid not in (select
Distinct UserID from Inventory force index(idx_TxDate)
where  {?DateRange_Clause}) and `UserID` <> '__NLI__'
and `UserID` <> '__PHONE__'  and
 {?Site_Clause}  and
 {?Dept_Clause}
```

In the **Parameter List** box, click **Create**. The Command Parameter dialog box appears.

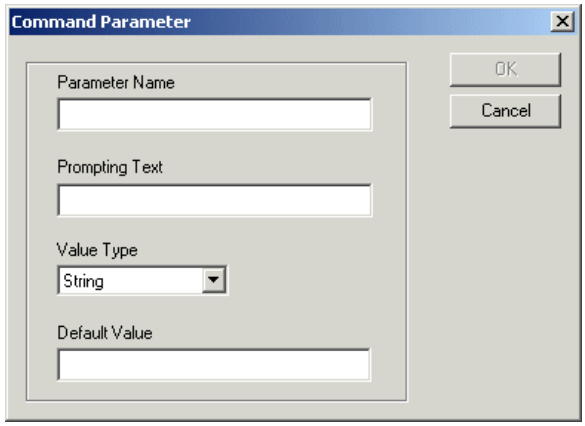


Figure 23: Command Parameter dialog box

7. In the **Parameter Name** field, enter `DateRange_Clause`. In the **Value Type** field, select `String`. For **Default Value**, enter `1=1`.
   Click **OK**.
8. Create parameters for Dept_Clause and Site_Clause. They are also String parameters whose default value is `1=1`.
9. Click **OK** in the Add Command to Report dialog box.
   The Enter Values page appears.

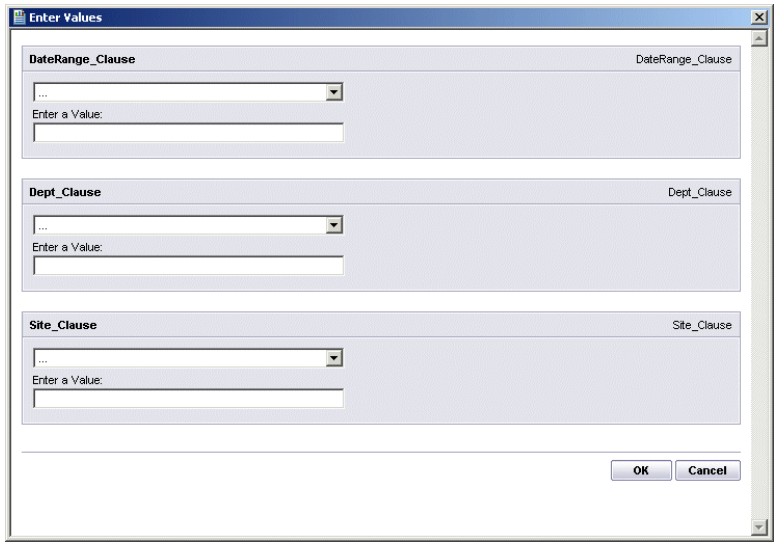


Figure 24: Enter Values page

10. Select `1=1` for each value, and then click **OK**.
    The Enter Values page closes, returning to the Database Expert. In the **Selected Tables** box, a new Command table is listed under VoceraReports. This is the SQL command you just created.
11. Select the Command table, and press F2. Change the name of the table to `Inactive`.

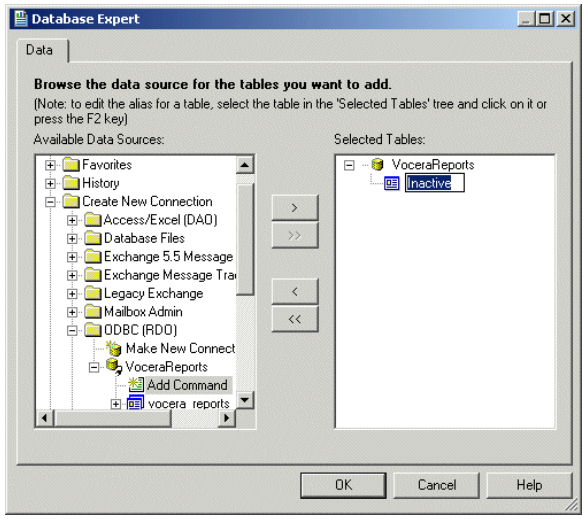


Figure 25: Inactive command

Click **OK**.

12. Now you need to specify how the fields should be grouped in the report. Choose **Report > Group Expert**.

    The Group Expert dialog box appears.

13. In the **Available Fields** box, select the Inactive.Site field, and then click the > button to move it to the **Group By** box.

    Select the Inactive.Departments field, and click the > button again.

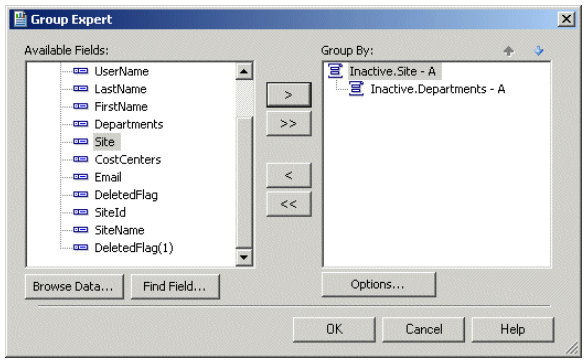


Figure 26: Group Expert dialog box

Click **OK**.

14. In the Field Explorer, expand Inactive table and insert the UserID, LastName, and FirstName fields into the report.

    Afterward, the designer should look like this:

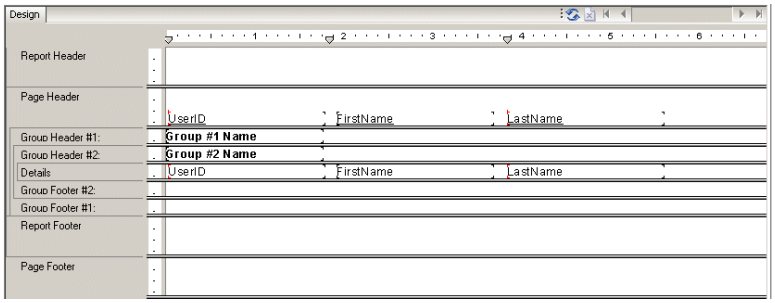


Figure 27: Fields inserted onto the report

15. Now let's also apply a Vocera template to the report. Choose **Report > Template Expert**. The Template Expert dialog box appears.

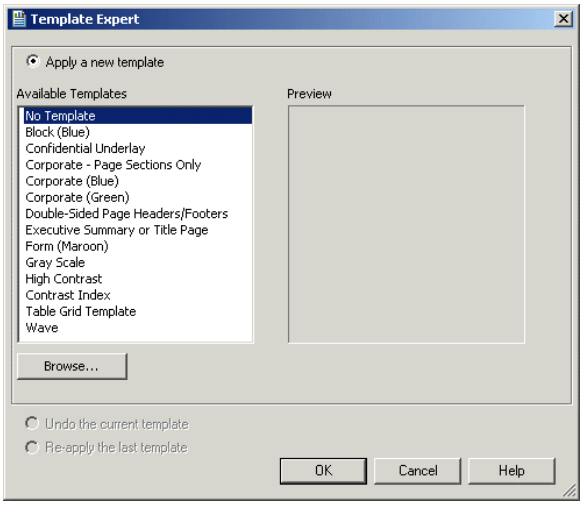


Figure 28: Template Expert dialog box

16. Click **Browse**. In the Open dialog box, navigate to the folder where Vocera standard reports are located. On the Report Server, the reports are installed in `\vocera\reports\Reports`.
17. Select the report named `VoceraReportTemplatePortrait.rpt`, and then click **OK**.

Note:
Vocera also provides a template for landscape format named `VoceraReportTemplateLandscape.rpt`.

18. Insert text objects in the Group Footer #2, Group Footer #1, and Report Footer to provide a count of inactive users for groups and for the report. Afterward, the designer should look like this:

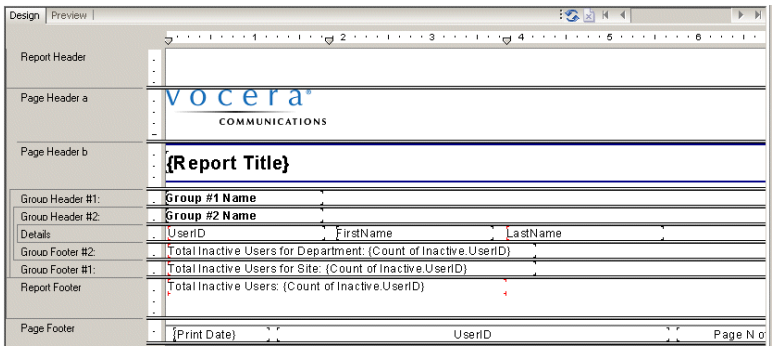


Figure 29: Inserted text objects to summarize groups

19. In the Field Explorer, right-click **Parameter Fields** and choose **New** from the popup menu. The Create New Parameter dialog box appears.

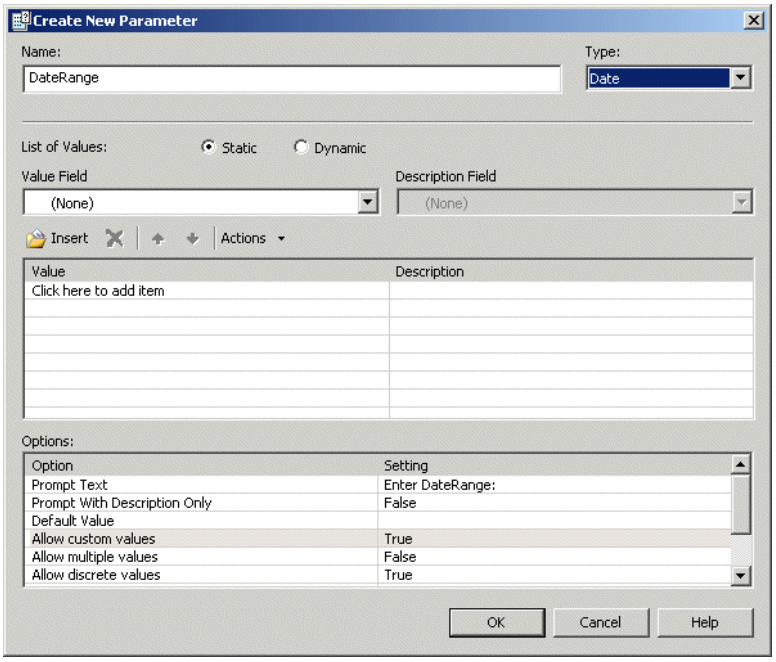


Figure 30: Create New Parameter dialog box

20. In the **Name** field, type `DateRange`. For the **Type** field, select `Date`. In the **Options** list, scroll down until you see the **Allow Range Values** option. Change its setting to **True**. Click **OK**.

    Note: If you fail to change the setting of **Allow Range Values** to **True**, you will be unable to run the report. See Troubleshooting Problems with Custom Reports

21. Add another parameter called `CompanyName` whose type is String.
22. Choose **File > Summary Info**. The Document Properties dialog box appears.
23. In the **Title** field, enter `Inactive Users Report`.

    Enter values in other fields as you choose, such as **Author** and **Subject**, and then click **OK**.

24. Format the report to improve its look. Afterward, it should look like this:

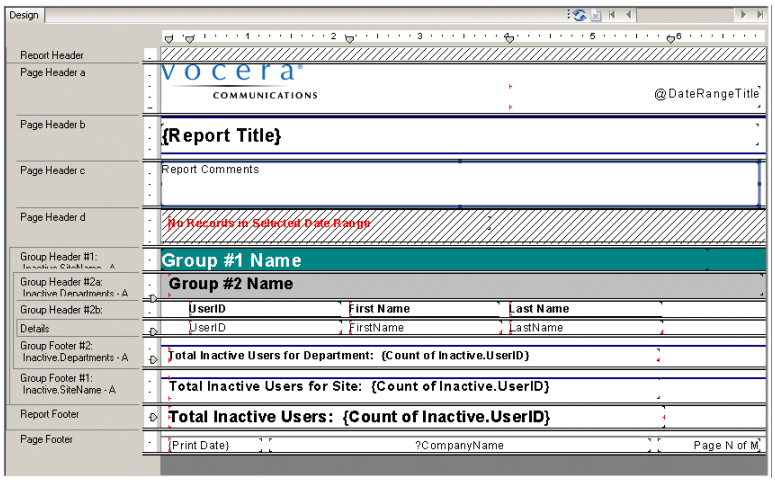


Figure 31: Finished report

25. Choose **File > Save** to save the report. Save the report as `CustomInactiveUsers.rpt`.
26. Copy the `CustomInactiveUsers.rpt` file to the `\vocera\reports\Reports\custom` folder on the Report Server computer.

    Now you are ready to add the report to the `customconfig.xml` file.

27. On the Report Server computer, navigate to the `\vocera\reports\Reports\custom` folder and open the `customconfig.xml` file in a text editor.

28. Add the following new report configuration to `customconfig.xml`. You do not need to type this text. Instead, you can copy the **Inactive Users** report configuration from `reportconfig.xml`, which is located in the `\vocera\reports` folder. You should make changes only to to the following elements in the configuration and save.

    There are many different parameters in this report configuration. For information about each type of parameter, see Configuring Report Parameters.

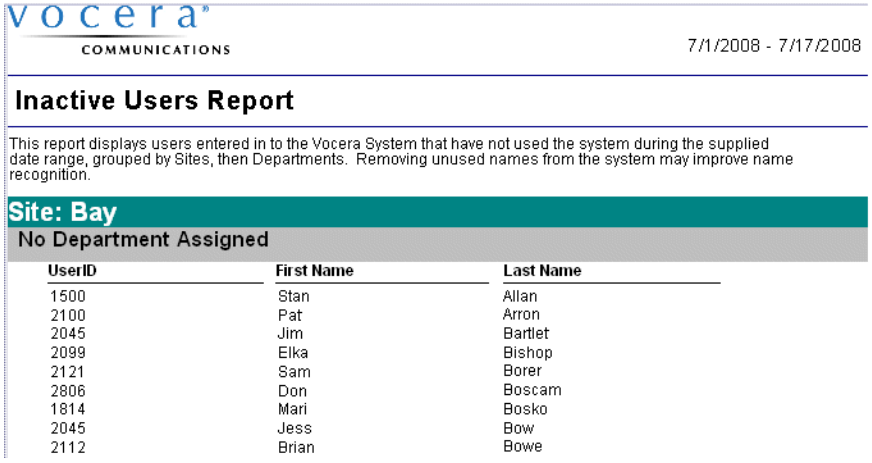
```
<REPORT ID="Custom Inactive Users">
  <NAME>Custom Report - Inactive Users</NAME>
  <DESCRIPTION>This report shows the users who have not
    used the Vocera System during the specified date
    range.</DESCRIPTION>
  <FILENAME>custom/CustomInactiveUsers.rpt</FILENAME>
  <TYPE>CUSTOM</TYPE>
```

Figure 32: New report configurations to customconfig.xml

29. From the Windows Start menu, select **Settings > Control Panel > Administrative Tools > Services**. The Services window appears.
30. Stop the Tomcat service and then start it again.
31. Close the Services window.
32. Log into the Report Console.
33. Click **Custom Reports**.
34. Select **Custom Report - Inactive Users** and then click **Generate**.

    The Report Parameters page appears.
35. Specify the report date range, site, and department(s), and then click **Generate**.

    The report should look like the following screen.



vocera COMMUNICATIONS

7/1/2008 - 7/17/2008

Inactive Users Report

This report displays users entered in to the Vocera System that have not used the system during the supplied date range, grouped by Sites, then Departments. Removing unused names from the system may improve name recognition.

Site: Bay

No Department Assigned

UserID	First Name	Last Name
1500	Stan	Allan
2100	Pat	Arron
2045	Jim	Bartlet
2099	Elka	Bishop
2121	Sam	Borer
2806	Don	Boscam
1814	Mari	Bosko
2045	Jess	Bow
2112	Brian	Bowe

Figure 33: Inactive Users report

Troubleshooting Problems with Custom Reports

For custom reports to run successfully, you should have executed the following tasks correctly:

- Define the report correctly, which involves selecting the right data source, defining the parameters, and applying the report template.
- Copy the ***.rpt** file to the correct location on the report server.
- Add the new report configuration to the customconfig.xml file and configure all the root elements and parameters correctly.

The following are common problems encountered while running custom reports:

Table 3: Troubleshooting Problems with Custom Reports

Problem	Solution
The report fails to generate, or you are prompted to install Adobe Flash player.	You have misconfigured a report parameter in the RPT file. To fix this problem, open the rpt file in the Crystal Reports, confirm the values defined for each parameter are correct. For example, if the report has a DateRange parameter, make sure the 'Allow Range Values' option for the parameter is set to TRUE.
When you generate a custom report and specify the report parameters, the report fails to generate and an HTML dialog box prompts for missing report parameters.	You have omitted parameters for the report in the customconfig.xml file. To fix the problem, edit the customconfig.xml file and configure all the parameters defined in the report.

Database Details

Vocera Usage Data

The following topics provide details about the structure and contents of tables of Vocera usage data maintained in the Vocera Report Server database. The tables are listed in alphabetical order; columns are listed according to their position in each table.

- addressbookentries on page 40
- audit on page 40
- broadcasts on page 41
- calls on page 42
- deviceinfo on page 43
- devicestatus on page 44
- devicestatuschange on page 44
- dnd on page 45
- failures on page 45
- groupactivity on page 46
- groupconference on page 47
- groups on page 48
- incompletedata on page 48
- inventory on page 49
- locations on page 49
- loginlogout on page 50
- lostconnectiondetail on page 50
- lostconnectionlog on page 51
- messageactivity on page 52
- posted on page 53
- radiodetail on page 53
- radiolog on page 54
- recresults on page 55
- sites on page 56
- speechports on page 56
- systeminfo on page 57
- unansweredcalls on page 58
- users on page 58
- vmiactions on page 59
- vmievent on page 60
- vmimessages on page 61

Note: For instructions on how to set up an ODBC data source to connect to your Report Server database, see Installing Crystal Reports 2008 and MySQL Connector/ODBC 3.51.

addressbookentries

The *addressbookentries* table stores data about Address Book Entries that can be entered and edited via the Address Book screen in the Vocera server Administration Console. This table has a primary key consisting of the following columns: FirstName, LastName, Site and Type.

Table 4: addressbookentries table

Column	Description
LastName	`varchar(50) NOT NULL default '0'` The LastName column specifies the last name for the entry, if the entry type is Person. If the entry type is Place, the value is an empty string.
FirstName	`varchar(50) NOT NULL default '0'` The FirstName column specifies the first name for the entry, if the entry type is Person. If the entry type is Place, the value is the name of the Place.
Type	`varchar(50) NOT NULL default ''` The Type column specifies the entry type: Person or Place.
Site	`varchar(50) NOT NULL default ''` The Site column specifies the Vocera site for which the entry is defined.
Phone	`varchar(50) NOT NULL default ''` Specifies the phone number.
Pager	`varchar(50) NOT NULL default ''` Specifies the pager number.
DeletedFlag	`int(1) unsigned zerofill default '0'` The DeletedFlag column indicates whether an entry has been deleted (0 = false). Although the entity may be flagged as deleted, it is not purged from the database because historical data (that is, older report logs) may continue to reference it.

audit

The *audit* table stores data about changes made to the Vocera system. This table has a primary key based on the TxMillis column.

Table 5: audit table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person who made the change. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server, unless the user logged in with the built-in login ID `Administrator`, in which case this value is __NONE__. Or, if the operation was performed automatically by the Vocera system, for example, an automated restore of the database, this value is null. See LoginId.
UserSite	`varchar(50) NOT NULL default ''` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.

Column	Description
LoginId	`varchar(50) default NULL` The LoginId column specifies the login ID of the person who made the change. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server, unless the user logged in with the built-in login ID `Administrator`, in which case this value is Administrator. See UserID.
Operation	`varchar(30) NOT NULL default ''` The Operation column specifies the operation that changed the Vocera database. Sample values include: Backup, Update, Create, and Update Conference Group.
ModifiedEntityType	`varchar(20) NOT NULL default ''` The ModifiedEntityType column specifies the type of the entity that was changed. Sample values include: AddrBook, Group, and User.
ModifiedEntityID	`varchar(70) NOT NULL default ''` The ModifiedEntityID column identifies the entity that was changed. For a user, the value is the user ID. For a group, the value is the group name. For an address book entry, the value is the address book entry name. If the Vocera database was not changed, for example, when a Backup operation occurs, this value is __NONE__.

broadcasts

The *broadcasts* table stores data about broadcasts made to Vocera groups. This table has a primary key based on the TxMillis column.

Table 6: broadcasts table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default ''` The UserType column specifies which type of user initiated the broadcast. Example values: User, Phone
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person who initiated the broadcast. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
GroupName	`varchar(50) default NULL` The GroupName column specifies the Vocera group to which the broadcast was sent.
GroupSite	`varchar(50) default NULL` In a multi-site installation, the GroupSite column specifies the Vocera site associated with the group to which the broadcast was sent. In a single-site installation, this value is null.
Duration	`bigint(15) unsigned NOT NULL default '0'` The Duration column specifies the duration of the broadcast in seconds.
IdBroadcast	`bigint(15) unsigned NOT NULL default '0'` The IdBroadcast column identifies a Vocera broadcast. However, this value is not guaranteed to be unique. Any given broadcast could have the same IdBroadcast value as another broadcast that took place earlier or later. One way to distinguish broadcasts is to compare values in the TxMillis column.

Column	Description
IdGenie	`bigint(15) unsigned NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-to-badge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to ..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

calls

The *calls* table stores data about calls placed and received through the Vocera system. This table has a primary key based on the TxMillis column.

Table 7: calls table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of the event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default ''` The UserType column specifies which type of user initiated the call. Example values: User, Phone
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person who initiated the call. When the UserType value is User, this value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server. When the UserType value is Phone, this value is the literal string __PHONE__.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the calling party is associated. In a single-site installation, this value is null.
CalledType	`varchar(10) NOT NULL default ''` The CalledType column specifies which type of user was the called party. Example values: User, Group, Phone
Called	`varchar(40) NOT NULL default ''` The Called column specifies the User ID, Group Name, or Phone Number of the called party, depending on the CalledType value.
CalledName	`varchar(100) default NULL` The CalledName column specifies the name of the called party, if the CalledType is User. If the CalledType is Group or Phone, the value is null.
CalledSite	`varchar(50) default NULL` In a multi-site installation, the CalledSite column specifies the Vocera site with which the called party is associated. In a single-site installation, this value is null.
Accepted	`int(1) NOT NULL default '0'` The Accepted column specifies whether a call was accepted by the called party. The value is 1 if the call was accepted; otherwise, the value is 0.
AcceptedByType	`varchar(10) default NULL` The AcceptedByType column specifies which type of user accepted a call. If a call was not accepted, this value is null. Example values: User, Phone
AcceptedBy	`varchar(40) default NULL` The AcceptedBy column specifies the User ID or Phone Number of the party who accepted a call, depending on the value of the AcceptedByType column.

Column	Description
AcceptedByName	`varchar(100) default NULL` The AcceptedByName column specifies the User Name of the party who accepted a call, if the AcceptedBy value is a User ID. If the value of AcceptedBy is a Phone Number or the literal string value __NONE__, AcceptedByName is null.
AcceptedBySite	`varchar(50) default NULL` In a multi-site installation, the AcceptedeBySite column specifies the Vocera site with which the accepting party is associated. In a single-site installation, this value is null.
DeviceType	`varchar(64) default NULL` Device type that initiates a call. Possible values: B1000, B1000A, B2000, B3000, MC70, and Smartphone. If you have deployed Vocera Connect, other possible values are Cisco, Apple, and Android.
Duration	`bigint(15) unsigned NOT NULL default '0'` The Duration column specifies the duration of the call in seconds.
IdCall	`bigint(20) NOT NULL default '0'` The IdCall column identifies a call placed or received through the Vocera system. However, this value is not guaranteed to be unique. Any given call could have the same IdCall value as another call that took place earlier or later. One way to distinguish calls is to compare values in the TXDateTime column.
IdGenie	`bigint(10) NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-to-badge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to ..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

deviceinfo

The *deviceinfo* table stores data about Vocera devices, such as badges. This table has a primary key based on the BadgeMacAddr column.

Table 8: deviceinfo table

Column	Description
DeviceID	`varchar(20) default NULL` Specifies the device ID from the Vocera Server. Example value: D-0009ef050686
BadgeMACAddr	`varchar(12) NOT NULL` Specifies the badge MAC address. Example value: 0009ef006e9a
SerialNo	`varchar(20) default NULL` Specifies the badge serial number.
Label	`varchar(30) default NULL` Specifies the device label.
Type	`varchar(15) default NULL` Specifies the device type. Possible values: B1000A, B2000, B3000, MC70, or Smartphone.
Color	`varchar(20) default NULL` Specifies the device color.
Owner	`varchar(50) default NULL` Specifies the device Owner group.
Site	`varchar(50) default NULL` Specifies the device Site.
TrackingDate	`date default NULL` The tracking date of the device. **Note:** This field can be any date used to track a device, for example, the date it was sent for repair or RMA'ed.

Column	Description
BadgeUsage	`tinyint(1) default NULL` Indicates whether the badge is shared or nonshared. 1 = shared, 0 = nonshared.
Notes	`text` Notes describing the device.
DeviceCode	`int(5) default '0'` Specifies the badge device code. Possible values: -1 = B1000 badge, 0 = B1000A badge, 2 = B2000 badge **Note:** This is a legacy data field that is no longer used by standard Report Server reports.
FirmwareVersion	`int(5) default '0'` Specifies the badge firmware version.
LastStatus	`varchar(50) default NULL` Specifies the last status of the device.
LastUser	`varchar(70) default NULL` Specifies the last user of the device.
LastLocation	`varchar(50) default NULL` Specifies the last location of the device based on the access point it connected with.
LastAccessed	`timestamp NULL default NULL` The timestamp when the device last access the Vocera Server.
InternalSerialNo	`varchar(20) default NULL` The serial number provided by the device. This value can be used to validate the serial number entered by a user.

devicestatus

The *devicestatus* table stores data for device statuses. This table has a primary key based on the Status column.

Table 9: devicestatus table

Column	Description
Status	`varchar(100) NOT NULL` The device status.
Description	`text` A description of the device status.
DeletedFlag	`int(1) unsigned zerofill default '0'` Indicates whether a device status has been deleted (0 = false, 1 = true). Although the entity may be flagged as deleted, it is not purged from the database because historical data (that is, older report logs) may continue to reference it.

devicestatuschange

The *devicestatuschange* table stores data for device statuses that have been changed by users. This table has a primary key based on the TxMillis column.

Table 10: devicestatuschange table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.

Column	Description
TxDate	`date default NULL` The TxDate column stores the date of an event, accurate to the day.
DeviceID	`varchar(20) NOT NULL` Specifies the device ID from the Vocera Server. Example value: D-0009ef050686
DeviceSite	`varchar(50) NOT NULL` Specifies the site of the device when the status changed.
Status	`varchar(30) NOT NULL` Specifies the device status.
Notes	`text` Notes describing the device status.

dnd

The *dnd* table stores data about when users turn Do Not Disturb (DND) mode on or off. This table has a primary key based on the TxMillis column.

Table 11: dnd table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date default NULL` The TxDate column stores the date of an event, accurate to the day.
UserID	`varchar(70) default NULL` Specifies the user ID of the person who turned DND mode on or off. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera Server.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
DND	`int(1) default null` Whether the user turned DND mode off or on; 0 means off and 1 means on.

failures

The *failures* table stores data about badge software failures. This table has a primary key based on the TxMillis column.

Table 12: failures table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDate	`date default NULL` The TxDate column stores the date of an event, accurate to the day.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of the event, accurate to the second.

Column	Description
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the badge user, if a user was logged in to the badge. If no user was logged in to the badge at the time of the failure, this value is the literal string __NLI__.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the badge is associated. In a single-site installation, this value is null.
BadgeMACAddr	`varchar(12) default NULL` The BadgeMACAddr column specifies the badge's MAC address. Example value: 0009ef0078a3
Failuretext	`varchar(50) NOT NULL default ''` The Failuretext column stores text describing the failure. Example value: Failure in pre.c at line 166. Code: 0009

groupactivity

The *groupactivity* table stores data about changes to Vocera groups when members are added or removed. This table has a primary key based on the TxMillis column.

Table 13: groupactivity table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxMillis	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default ''` Specifies which type of user who initiated AddToGroup and RemoveFromGroup. Example values: User, Phone
UserID	`varchar(70) default NULL` Specifies the user ID of the person who initiated AddToGroup and RemoveFromGroup. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera Server.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
Operation	`varchar(64) NOT NULL default ''` Specifies the operation, either addtogroup or removefromgroup.
GroupName	`varchar(50) default NULL` Specifies the Vocera group to which the party was added ot from which the party was removed.
GroupSite	`varchar(50) default NULL` In a multi-site installation, the GroupSite column specifies the Vocera site associated with the group whose membership was modified. In a single-site installation, this value is null.
PartyID	`varchar(70) default NULL` Specifies the user ID of the person who was added to or removed from the group. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera Server.
PartySite	`varchar(50) default NULL` In a multi-site installation, the PartySite column specifies the Vocera site with which the user who was added to or removed from the group is associated. In a single-site installation, this value is null.

Column	Description
IdGenie	`bigint(15) unsigned NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-tobadge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

groupconference

The *groupconference* table stores data about group conferences, which are also called instant conferences. This table has a primary key based on the TxMillis column.

Table 14: groupconference table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxMillis	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default ''` Specifies which type of user initiated the group conference. Example values: User, Phone
UserID	`varchar(70) default NULL` Specifies the user ID of the person who initiated the group conference. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera Server.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
GroupName	`varchar(50) default NULL` Specifies the Vocera group to which the group conference was sent.
GroupSite	`varchar(50) default NULL` In a multi-site installation, the GroupSite column specifies the Vocera site associated with the group to which the group conference was sent. In a single-site installation, this value is null.
Duration	`bigint(15) unsigned NOT NULL default '0'` Specifies the duration of the group conference in seconds.
IdGroupConf	`bigint(15) unsigned NOT NULL default '0'` Identifies a Vocera group conference. However, this value is not guaranteed to be unique. Any given group conference could have the same IdGroupConf value as another group conference that took place earlier or later.
IdGenie	`bigint(15) unsigned NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-tobadge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

groups

The *groups* table stores data about Vocera groups. This data can be entered and edited via the Groups screen in the Vocera server Administration Console. This table has a primary key based on the GroupID column.

Table 15: groups table

Column	Description
GroupID	`varchar(70) default NULL` Specifies the internal ID of a group.
GroupName	`varchar(50) NOT NULL default '0'` The GroupName column specifies the name of a group. Example values: Code Blue Group, Facilities
Site	`varchar(50) NOT NULL default ''` The Site column specifies the Vocera site with which a group is associated. In a single-site installation, this value is the literal string s-global.
CostCenter	`varchar(100) default NULL` The CostCenter column stores a value that identifies the group's cost center (typically used for accounting purposes).
DeletedFlag	`int(1) unsigned zerofill default '0'` The DeletedFlag column indicates whether an entry has been deleted (0 = false). Although the entity may be flagged as deleted, it is not purged from the database because historical data (that is, older report logs) may continue to reference it.
UserMembers	`mediumtext default NULL` Stores the user members of the group. The values are separated by a vertical bar (|).
Department	`varchar(50) default NULL` Stores the group's department. A group can have only one department or none.

incompletedata

The *incompletedata* table stores data generated when a log file does not contain all the events of a Vocera transaction. For example, suppose a badge user initiates a call just before midnight and doesn't end the call until after midnight. Because Vocera automatically creates a new log file at midnight, the events for this call would be spread over two log files, and would end up in the *incompletedata* table when the Report server parses the Vocera server logs.

Incomplete data may also be generated if the log file stores more than 10,000 entries, or if the server is restarted while a transaction is in progress. This table has a primary key based on the TxMillis column.

Table 16: incompletedata table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of the event, accurate to the second.
Type	`varchar(20) default '0'` The Type column specifies the type of the badge transaction. A value of 0 specifies a call, a value of 1 specifies a broadcast, and a value of 2 specifies a Genie prompt.
ReportData	`text NOT NULL` The ReportData column stores the generated data.

inventory

The *inventory* table stores data about the status of badges, useful for asset tracking reports. This table has a primary key based on the TxMillis column.

Table 17: inventory table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of the event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person logged in to the badge. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server. If no user is logged in, this value is the literal string __NLI__.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the badge user is associated. In a single-site installation, or if no user is logged in to the badge, this value is null.
BadgeMACAddr	`varchar(12) default NULL` The BadgeMACAddr specifies the badge MAC address. Example value: 0009ef006e9a
APMACAddr	`varchar(12) default NULL` The APMACAddr column specifies the MAC address of the wireless access point with which the badge was most recently associated. Example value: 000f24e9dce0
Voltage	`decimal(7,2) default '0.00'` The Voltage column specifies the badge battery voltage. Example value: 3.90
UIState	`varchar(15) default NULL` The UIState column specifies whether a badge is in use (for example, on a call). Example values: Active, Standby
SerialNo	`varchar(30) NOT NULL` The SerialNo column stores the badge's serial number.

locations

The *locations* table stores data about access points and named locations. This data can be defined using the Locations screen of the Vocera server Administration Console. This table has a primary key based on the MACAddress column.

Table 18: locations table

Column	Description
MACAddress	`varchar(20) NOT NULL default '0'` The MACAddress column specifies the MAC address of the wireless access point with which the badge was most recently associated. Example value: 000f24e9dce0
LocationName	`varchar(50) NOT NULL default ''` The Location name column specifies the name of the location. Example value: The Lobby
Site	`varchar(50) default NULL` In a multi-site installation, the Site column specifies which Vocera site the location is associated with. In a single-site installation, the value is the literal string s-global.

Column	Description
DeletedFlag	`int(1) unsigned zerofill default '0'` The DeletedFlag column indicates whether an entry has been deleted (0 = false). Although the entity may be flagged as deleted, it is not purged from the database because historical data (that is, older report logs) may continue to reference it.

loginlogout

The *loginlogout* table stores information about when Vocera devices logged in and logged out. This table has a primary key based on the TxMillis column.

Table 19: loginlogout table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person who logged in or logged out. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera Server.
UserSite	`varchar(50) NOT NULL default ''` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
Operation	`varchar(64) NOT NULL default ''` Specifies the operation, either login or logout.
DeviceMACAddr	`varchar(12) NOT NULL default ''` Specifies the device MAC address. Example value: 0009ef006e9a
APMACAddr	`varchar(12) default ''` Specifies the MAC address of the wireless access point with which the device was most recently associated. Example value: 000f24e9dce0
LoginUserCount	`int(11) default null` The current number of users that are logged in immediately following the Login or Logout operation.
TotalLic	`int(11) default null` The total number of users login licenses.

lostconnectiondetail

The *lostconnectiondetail* table stores detailed information about each time a Vocera badge loses its connection with the Vocera server, connects to an access point, or searches for a DHCP server. This table has a primary key based on the id column.

Table 20: lostconnectiondetail table

Column	Description
id	`int(11) NOT NULL auto_increment` An incremental ID assigned to the lost connection instance.
TxParentMillis	`varchar(15) default NULL` The TxParentMillis column is the unique identifier for the parent record in the lost connection log.

Column	Description
TxMillis	`bigint(15) unsigned default NULL` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL` The TxDateTime column stores the date and time of an event, accurate to the second.
APMacAddress	`varchar(12) default NULL` The APMacAddress column specifies the MAC address of the wireless access point with which the badge was most recently associated. Example value: 000f24e9dce0
OtherParty	`varchar(20) default NULL` Source IP address that the badge received while in a call.
Type	`varchar(10) default NULL` Type of lost connection. Possible values are: • SFAP—Searching for AP. • SFS—Searching for Server. • LCS—Lost Connection to Server. • SFDHCP—Searching for DHCP. • CAP—Connecting to Access Point.

lostconnectionlog

The *lostconnectionlog* table stores summary information about badges that have lost their connection with the Vocera server, connected to an access point, or searched for a DHCP server. This table has a primary key based on the TxMillis column. For details about each record, go to the lostConnectiondetail table with lostConnectiondetail.txParentMillis = lostconnectionlog.TxMillis.

Table 21: lostconnectionlog table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the last person logged into the badge. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server, unless the user logged in with the built-in login ID `Administrator`, in which case this value is __NONE__. See LoginId.
UserSite	`varchar(50) NOT NULL default ''` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
BadgeMacAddress	`varchar(12) default NULL` The BadgeMACAddr specifies the badge MAC address. Example value: 0009ef006e9a
BadgeBinary	`text` Raw binary log sent by the badge.
SFAPCount	`smallint(6) NOT NULL default '0'` Number of times the user's badge went searching for an access point.
CAPCount	`smallint(6) default NULL` Number of times the user's badge attempted to connect and authenticate at the access point.

Column	Description
LSCount	`smallint(6) default NULL` Number of times the user's badge lost the Vocera server connection.
SFSCount	`smallint(6) NOT NULL default '0'` Number of times the user's badge was searching for a Vocera server.
SFDHCPCount	`smallint(6) NOT NULL default '0'` Number of times the user's badge was searching for a DCHP server.
Duration	`smallint(6) NOT NULL` The Duration column specifies the duration of the lost connection, in seconds.

messageactivity

The *messageactivity* table stores information about voice messages, not text messages. This table has a primary key based on the TxMillis column.

Table 22: messageactivity table

Column	Type
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default ''` Specifies which type of user initiated the message activity. Example values: User, Phone
UserID	`varchar(70) default NULL` Specifies the user ID of the person who initiated the message activity. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera Server.
UserSite	`varchar(70) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
Operation	`varchar(64) NOT NULL default ''` Specifies the operation; leavemessage is the only operation supported currently.
Parties	`varchar(512) default NULL` Specifies a comma-delimited list of Vocera groups or users to which the message was sent.
Duration	`bigint(15) unsigned NOT NULL default '0'` Specifies the duration of the message activity in seconds.
IdMsgActivity	`bigint(15) unsigned NOT NULL default '0'` Identifies a Vocera message activity, which may not be unique. Any given message activity could have the same IdMsgActivity value as another Message Activity that took place earlier or later.

Column	Type
IdGenie	`bigint(15) unsigned NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-to-badge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

posted

The *posted* table stores data about log files used by the Report server. The files themselves are stored on the Vocera server in the `vocera\logs` directory. This table has a primary key based on the reportName column.

Table 23: posted table

Column	Description
reportName	`varchar(50) NOT NULL default '0'` The reportName column specifies the name of a log file. Example value: report-jul01-0001.txt
reportTime	`bigint(15) NOT NULL default '0'` The reportTime column specifies, in milliseconds since January 1, 1970, the date and time a report log file was posted to the server. Example value: 1120287620317

radiodetail

The *radiodetail* table stores detailed information regarding the audio quality experienced by Vocera badges during a call. This table has a primary key based on the id column.

Table 24: radiodetail table

Column	Description
id	`int(11) NOT NULL auto_increment` An incremental ID assigned to the lost connection instance.
TxParentMillis	`varchar(15) default NULL` The TxParentMillis column is the unique identifier for the parent record in the radio log.
TxMillis	`bigint(15) unsigned default NULL` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL` The TxDateTime column stores the date and time of an event, accurate to the second.
APMacAddress	`varchar(12) default NULL` The APMacAddress column specifies the MAC address of the wireless access point with which the badge was most recently associated. Example value: 000f24e9dce0
OtherParty	`varchar(20) default NULL` Source IP address that the badge received while in a call.
PacketsReceived	`smallint(6) default NULL` Total packets received before a new log event occurred. PacketsReceived and PacketsMissed are used to compute the Packet Error Rate.
PacketsMissed	`smallint(6) default NULL` Total packets missed before a new log event occurred.

Column	Description
CQ	`smallint(6) default NULL` Communication Quality (CQ) value, which is an even number value on a logarithmic scale ranging from 0 to 92, where 0 represents no signal and 92 is the strongest possible signal with essentially no background noise.
RoamTime	`mediumint(9) default NULL` Specifies the Roam Time.
Type	`smallint(1) default '1'` 1 if frames are unicast; 0 if frames are multicast.
Reserved1	`mediumint(9) default NULL` Reserved for future use.
Reserved2	`mediumint(9) default NULL` Reserved for future use.
Reserved3	`mediumint(9) default NULL` Reserved for future use.

radiolog

The *radiolog* table stores summary information regarding the audio quality experienced by Vocera badges during a call. This table has a primary key based on the TxMillis column. For details about each record, go to the radiodetail table with radiodetail.txParentMillis = radiolog.TxMillis.

Table 25: radiolog table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the last person logged into the badge. This value corresponds to a User ID specified via the Administration Console or the User Console on the Vocera server, unless the user logged in with the built-in login ID `Administrator`, in which case this value is __NONE__. See LoginId.
UserSite	`varchar(50) NOT NULL default ''` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
BadgeMacAddress	`varchar(12) default NULL` The BadgeMACAddr specifies the badge MAC address. Example value: 0009ef006e9a
BadgeBinary	`text` Raw binary log sent by the badge.
PacketsReceived	`smallint(6) default NULL` Total packets received before a new log event occurred. PacketsReceived and PacketsMissed are used to compute the Packet Error Rate.
PacketsMissed	`smallint(6) default NULL` Total packets missed before a new log event occurred.
AvgCQ	`smallint(6) NOT NULL default '0'` Average Communication Quality (CQ) value. A CQ value is an even number value on a logarithmic scale ranging from 0 to 92, where 0 represents no signal and 92 is the strongest possible signal with essentially no background noise.

Column	Description
JBhit	`mediumint(9) default NULL` Number of Jitter Buffer hits.
JBMissed	`mediumint(9) default NULL` Number of Jitter Buffer misses.
Type	`smallint(1) default '1'` 1 = Without Jitter Buffer data; 2 = New format with Jitter Buffer data.
Reserved1	`mediumint(9) default NULL` Reserved for future use.
Reserved2	`mediumint(9) default NULL` Reserved for future use.
Reserved3	`mediumint(9) default NULL` Reserved for future use.

recresults

The *recresults* table stores speech recognition results. This table has a primary key based on the TxMillis column.

Table 26: recresults table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of the event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default '0'` The UserType column specifies which type of user uttered the speech. Example values: User, Phone
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person who uttered the speech. When the UserType value is User, this value is either a User ID specified via the Administration Console or the User Console on the Vocera server, or the literal string value __NLI__, which stands for Not Logged In. When the UserType value is Phone, this value is the literal string __PHONE__.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.
BadgeMACAddr	`varchar(12) default NULL` The BadgeMACAddr specifies the badge MAC address. Example value: 0009ef006e9a
APMACAddr	`varchar(12) default NULL` The APMACAddr column specifies the MAC address of the wireless access point with which the badge was most recently associated. Example value: 000f24e9dce0
RecStatus	`varchar(40) NOT NULL default ''` The RecStatus column specifies the recognition status of the utterance. Example values: Recognized, Rejected, NoSpeech
Score	`int(10) unsigned NOT NULL default '0'` The Score column specifies the recognition score for the utterance. Values range from 0 to 100.
Recognized	`varchar(50) default NULL` The Recognized column specifies the speech that was recognized by the Vocera system. Example values: WhereIs Randy Floren, Call Chris Long, Yes, LogOut

Column	Description
Grammar	`varchar(64) default NULL` One of the Vocera speech recognition grammar categories, such as mainmenu, yesorno, forwardwhen, singlename, messageresponse, and so on.
DeviceType	`varchar(64) default NULL` Device type that initiates a call. Possible values: B1000, B1000A, B2000, B3000, MC70, and Smartphone. If you have deployed Vocera Connect, other possible values are Cisco, Apple, and Android.
IdGenie	`bigint(15) NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-to-badge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to ..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

sites

The *sites* table stores data about Vocera sites. This data can be entered and edited via the Sites screen in the Vocera server Administration Console. The table has a primary key based on the SiteId column.

Table 27: sites table

Column	Description
SiteId	`varchar(50) NOT NULL default ''` The SiteId column uniquely identifies a site. Example value: s-global
SiteName	`varchar(50) NOT NULL default '0'` The SiteName column specifies the name of a site. Example value: Global
DeletedFlag	`int(1) unsigned zerofill default '0'` The DeletedFlag column indicates whether an entry has been deleted (0 = false). Although the entity may be flagged as deleted, it is not purged from the database because historical data (that is, older report logs) may continue to reference it.

speechports

The *speechports* table stores data about speech port usage.

Table 28: speechports table

Column	Description
TxMillis	`bigint(15) unsinged NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of the event, accurate to the second.
TxDate	`date NOT NULL default '0000-00-00'` The TxDate column stores the date of an event, accurate to the day.
UserType	`varchar(20) NOT NULL default ''` The UserType column specifies which type of user uttered the speech. Example values: User, Phone
UserID	`varchar(70) default NULL` The UserID column specifies the user ID of the person who uttered the speech. When the UserType value is User, this value is either a User ID specified via the Administration Console or the User Console on the Vocera server, or the literal string value __NLI__, which stands for Not Logged In. When the UserType value is Phone, this value is the literal string __PHONE__.
UserSite	`varchar(50) default NULL` In a multi-site installation, the UserSite column specifies the Vocera site with which the user is associated. In a single-site installation, this value is null.

Column	Description
Duration	`mediumint(8) unsigned default '0'` The Duration column specifies the duration of the utterance, in seconds.
SpeechPort	`int(5) unsigned NOT NULL default '0'` The SpeechPort column specifies which speech port processed the utterance. Example value: 3
IdGenie	`bigint(15) unsigned NOT NULL default '0'` The IdGenie column identifies a Genie session. A single Genie session may comprise several events. For example, a Genie session for a badge-to-badge call may include the "Vocera" prompt played on the calling party's badge and the "Can you talk to ..." prompt played on the called party's badge. In the table, rows for events in the same Genie session have the same value in the IdGenie column. However, this value is not guaranteed to be unique. Any given Genie session could have the same IdGenie value as another Genie session that took place earlier or later. One way to distinguish Genie sessions is to compare values in the TxMillis column.

systeminfo

The *systeminfo* table stores data about Report server. This data is entered via the Administration screen of the Report server Administration Console.

Note:
Although the *systeminfo* table does not store Vocera usage data, it is commonly used by the standard Report Server reports to display system information.

Table 29: systeminfo table

Column	Description
CompanyName	`varchar(150) default '0'` The CompanyName column stores the name of the company where Vocera is installed.
ServerIp	`varchar(120) default NULL` The ServerIp column stores the comma-separated list of IP addresses or DNS names of the Vocera server. If the Vocera server is not clustered, this column contains only one IP address or DNS name. The report server uses this information to fetch data from the Vocera server. Example values: 12.123.11.1, 127.0.0.1, myserver

BackupInfo	`text`
	The BackupInfo column specifies how often Report server data is backed up. The value represents information entered in the following fields of the Backup page of the Administration screen: Enable Automatic Scheduled Backups, Backup Frequency, Backup Time, and Maximum Number of Backup Files to Save.
	The field values are separated by a vertical bar (|). Backup times are presented in milliseconds from midnight. For example, the following value specifies automatic backups every 2 days at 2:00 a.m., retaining a maximum of 5 backup files:
	`true|2|7200000|5`

DataLoadInfo	`text`
	The DataLoadInfo column specifies how often the Report server loads data from the Vocera server. The value represents information entered in the following fields of the Data Load page of the Administration screen: Enable Automatic Scheduled Data Load, Load Data Frequency, and Data Load Time.
	The field values are separated by a vertical bar (|). For example, the following value specifies automatic data loading every day at 4:00 a.m.
	`true|1|14400000|`
	Data load times are presented in milliseconds from midnight. Note the trailing vertical bar after the time.

Password	`varchar(100) default NULL` The Password column stores the Administrator password for the Report server. This value is encrypted.
MailInfo	`text` Specifies mail server information. The values are separated by a vertical bar (|).
SweepInfo	`text` Specifies server sweep information. The values are separated by a vertical bar (|).

unansweredcalls

The *unansweredcalls* table stores data about calls that were not answered.

Table 30: unansweredcalls table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
Called	`varchar(40) NOT NULL` The Called column stores the user ID of the called party
CallID	`bigint(20) NOT NULL default '20'` The CallID column stores the ID of the call as entered in the Vocera server log.
GenieID	`bigint(10) NOT NULL default '0'` The GenieID column stores the ID of the Genie session as entered in the Vocera server log.
ReasonID	`int(2) NOT NULL default '0'` The ReasonID column stores an integer that maps to a Reason string.
Reason	`VARCHAR(25) NOT NULL` The Reason column stores a string that explains why the call was unanswered. Possible values: • Busy • Call rejected • Call wait rejected • Caller blocked • Not logged in • Not online • Phone not answered • Unavailable

users

The *users* table stores data about badge users. This data can be entered and edited via the Users screen in the Vocera server Administration Console or the Basic Information screen of the Vocera server User Console.

Other tables in the Vocera Report Server schema contain a UserID column that corresponds to the UserID column in this table. Therefore, you can write queries that return data about badge users in other contexts.

For example, the following query returns data about badge users in the context of a report about broadcasts:

```
select * from broadcasts, users
where broadcasts.UserID = users.UserID
```

This table has a unique key based on the UserID column.

Table 31: users table

Column	Description
UserID	`varchar(70) NOT NULL default ''` The UserID column specifies a user ID entered via the Administration Console or the User Console on the Vocera server. Example value: ALINCOLN
UserName	`varchar(255) NOT NULL default ''` The UserName column stores the value of the FirstName column, a space, and the value of the LastName column. Example value: ABRAHAM LINCOLN

Column	Description
LastName	`varchar(50) NOT NULL default ''` The LastName column stores the user's last name. Example value: LINCOLN.
FirstName	`varchar(50) NOT NULL default ''` The FirstName column stores the user's first name. Example value: ABRAHAM
Departments	`varchar(100) default NULL` The Departments column lists department groups that the user belongs to. Multiple department names are separated by a plus sign (+). Example values: Engineering, Admin+Marketing+Sales
Site	`varchar(50) default NULL` The Site column lists the site ID (not the site name) of each site the user is associated with. Example value: s-global
CostCenters	`varchar(100) default NULL` The CostCenters column specifies the user's cost centers. Multiple cost centers are separated by a plus sign (+). Example values: Engineering, Admin+Marketing+Sales
DeletedFlag	`int(1) unsigned zerofill NOT NULL default '0'` The DeletedFlag column indicates whether a user has been deleted (0 = false). Although the entity may be flagged as deleted, it is not purged from the database because historical data (that is, older report logs) may continue to reference it.
Email	`varchar(100) default NULL` Specifies the user's email address.

vmiactions

The *vmiactions* table stores data about actions performed by recipients in response to messages sent from Vocera Messaging Interface clients.

This table has a unique key based on the TxMillis column.

Table 32: vmiactions table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date default NULL` The TxDate column stores the date of an event, accurate to the day.
ClientID	`varchar(50) NOT NULL default ''` The unique ID of the Vocera Messaging Interface client.
MessageID	`bigint(20) unsigned NOT NULL default '0'` The Vocera Messaging Interface message ID, which must be unique for the client.
RecipientType	`varchar(20) NOT NULL default ''` Specifies which type of Vocera entity (for example, user or group) received the message.
RecipientSite	`varchar(50) default NULL` In a multi-site installation, the RecipientSite column specifies the Vocera site with which the recipient is associated. In a single-site installation, this value is null.
RecipientID	`varchar(70) NOT NULL default ''` Specifies the ID of the message recipient.
RecipientName	`varchar(100) default NULL` Specifies the name of the message recipient.
ActionType	`varchar(50) NOT NULL default ''` Specifies the type of action. Sample values include: Accepted, Rejected, Delivered, Read, and Response.

Column	Description
ActionDetail	`text NOT NULL` Specifies detailed information about the action. For example, if the action type is Response, the response could be OK, Busy, or Cancel, which would appear in the ActionDetail column. If the action type is Rejected, the ActionDetail column might show "User is not logged in" to explain why the message was rejected.

vmievent

The *vmievent* table stores data about events logged by Vocera Messaging Interface clients. This table has a unique key based on the TxMillis column.

For information about how to log information about external events—such as those from a middleware system—to Vocera Report Server logs, see the `LogEvent()` method in the Vocera Messaging Interface Guide.

Table 33: vmievent table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date default NULL` The TxDate column stores the date of an event, accurate to the day.
ClientID	`varchar(50) NOT NULL default ''` The unique ID of the Vocera Messaging Interface client.
EventID	`bigint(20) unsigned NOT NULL default '0'` VMI event ID.
EventType	`varchar(30) default NULL` Specifies VMI Event type, supported even type are, MessageEvent, CancelEvent, SystemEvent
Item1	`varchar(64) default NULL` Stores the first item in the event info specified by the logged VMI event.
Item2	`varchar(64) default NULL` Stores the second item in the event info specified by the logged VMI event.
Item3	`varchar(64) default NULL` Stores the third item in the event info specified by the logged VMI event.
Item4	`varchar(64) default NULL` Stores the fourth item in the event info specified by the logged VMI event.
Item5	`varchar(64) default NULL` Stores the fifth item in the event info specified by the logged VMI event.
Item6	`varchar(64) default NULL` Stores the sixth item in the event info specified by the logged VMI event.
Item7	`varchar(64) default NULL` Stores the seventh item in the event info specified by the logged VMI event.
Item8	`varchar(64) default NULL` Stores the eighth item in the event info specified by the logged VMI event.
Item9	`varchar(64) default NULL` Stores the ninth item in the event info specified by the logged VMI event.
Item10	`varchar(64) default NULL` Stores the tenth item in the event info specified by the logged VMI event.

vmimessages

The *vmimessages* table stores data about messages sent by Vocera Messaging Interface clients.

This table has a unique key based on the TxMillis column.

Table 34: vmimessages table

Column	Description
TxMillis	`bigint(15) unsigned NOT NULL default '0'` The TxMillis column gives each row in the table a unique identifier. The value represents the number of milliseconds since January 1, 1970 (a date specified for internal use by the report system). Because of the uniqueness constraint, it may not reflect the exact actual date and time of the event. See TxDateTime.
TxDateTime	`datetime NOT NULL default '0000-00-00 00:00:00'` The TxDateTime column stores the date and time of an event, accurate to the second.
TxDate	`date default NULL` The TxDate column stores the date of an event, accurate to the day.
ClientID	`varchar(50) NOT NULL default ''` The unique ID of the Vocera Messaging Interface client.
DestinationType	`varchar(20) NOT NULL default ''` Specifies which type of Vocera entity (for example, user or group) was sent the message.
DestinationSite	`varchar(50) default NULL` In a multi-site installation, the DestinationSite column specifies the Vocera site with which the destination is associated. In a single-site installation, this value is null.
DestinationID	`varchar(70) NOT NULL default ''` Specifies the ID of the message destination.
DestinationName	`varchar(100) default NULL` Specifies the name of the message recipient.
MessageID	`bigint(20) unsigned NOT NULL default '0'` The Vocera Messaging Interface message ID, which must be unique for the client.
Priority	`varchar(20) default NULL` Specifies the Vocera Messaging Interface priority (Normal, High, or Urgent).
MessageText	`text NOT NULL` Specifies the content of the message. The length of the message is limited to 256 characters.

Report Server Application Data

The following tables contain Report Server application data rather than Vocera usage data. These tables should not be used to create custom reports.

- diskdestination
- jobexceptions
- jobupdates
- mailinglist
- package
- packagedestination
- packagereports
- packageschedule
- packageschedulehistory
- reportconfiginfo
- reportparameters
- scheduledreporthistory

- task
- taskconfiginfo
- taskexceptions
- taskexecutables
- taskjobupdates
- taskparameters
- taskschedule
- taskschedulehistory