

**GBASE<sup>®</sup>**

**GBase 8s更改数据捕获(CDC)API程序员指南**



**GBase 8s更改数据捕获(CDC)API程序员指南**，南大通用数据技术股份有限公司**GBase** 版权所有©2004-2021，保留所有权利

## 版权声明

本文档所涉及的软件著作权及其他知识产权已依法进行了相关注册、登记，由南大通用数据技术股份有限公司合法拥有，受《中华人民共和国著作权法》、《计算机软件保护条例》、《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的南大通用数据技术股份有限公司的版权信息由南大通用数据技术股份有限公司合法拥有，受法律的保护，南大通用数据技术股份有限公司对本文档可能涉及到的非南大通用数据技术股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅，并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经南大通用数据技术股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将视为侵权，南大通用数据技术股份有限公司具有依法追究其责任的权利。

本文档中包含的信息如有更新，恕不另行通知。您对本文档的任何问题，可直接向南大通用数据技术股份有限公司告知或查询。

## 通讯方式

南大通用数据技术股份有限公司

天津市高新区开华道22号普天创新产业园东塔20-23层

电话：400-013-9696

邮箱：[info@gbase.cn](mailto:info@gbase.cn)

## 商标声明

**GBASE<sup>®</sup>**是南大通用数据技术股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由南大通用数据技术股份有限公司合法拥有，受法律保护。未经南大通用数据技术股份有限公司书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯南大通用数据技术股份有限公司商标权的，南大通用数据技术股份有限公司将依法追究其法律责任。

## 目 录

介绍 .....	1
1 开始使用 .....	2
1.1 变更数据捕获API .....	2
1.2 更改数据捕获API组件 .....	3
1.2.1 函数 .....	3
1.2.2 系统数据库 .....	3
1.2.3 错误代码 .....	3
1.2.4 智能大对象读取函数 .....	3
1.2.5 CDC记录 .....	3
1.3 智能大对象读取函数 .....	4
1.4 读取超时 .....	5
1.5 读取缓冲区大小 .....	5
1.6 智能大对象文件描述符 .....	5
1.7 CDC记录序列号 .....	5
1.8 捕获数据 .....	6
1.9 结束表的捕获 .....	6
1.10 准备使用变更数据捕获API .....	6
1.11 编写应用程序以捕获数据更改 .....	7
1.12 错误处理 .....	8
1.13 处理智能大对象 .....	9
1.14 重新启动数据捕获 .....	9
1.15 监控数据捕获 .....	10
2 更改数据捕获功能 .....	10
2.1 cdc_activatesess() 函数 .....	11
2.2 cdc_closesess() 函数 .....	12
2.3 cdc_deactivatesess() 函数 .....	12
2.4 cdc_endcapture() 函数 .....	13
2.5 cdc_errortext() 函数 .....	14
2.6 cdc_opensess() 函数 .....	16
2.7 cdc_recboundary() 函数 .....	18
2.8 cdc_set_fullrowlogging() 函数 .....	18
2.9 cdc_startcapture() 函数 .....	19
3 更改数据捕获记录 .....	21

3.1 CDC_REC_BEGINTX记录 .....	22
3.2 CDC_REC_COMMTX记录 .....	23
3.3 CDC_REC_DELETE记录 .....	23
3.4 CDC_REC_DISCARD记录 .....	25
3.5 CDC_REC_ERROR记录 .....	25
3.6 CDC_REC_INSERT记录 .....	26
3.7 CDC_REC_RBTX记录 .....	27
3.8 CDC_REC_TABSCHEMA记录 .....	28
3.9 CDC_REC_TIMEOUT记录 .....	29
3.10 CDC_REC_TRUNCATE记录 .....	30
3.11 CDC_REC_UPDAFT记录 .....	30
3.12 CDC_REC_UPDBEF记录 .....	32
4 syscdc系统数据库 .....	33
4.1 syscdcerrcodes表 .....	33
4.2 syscdrectypes表 .....	34
5 更改数据捕获错误代码 .....	34
6 onstat -g cdc .....	36

# 介绍

## 关于本文档

本文档描述了如何使用更改数据捕获API编写捕获更改数据以供外部处理的应用程序。

## 用户类型

本文档面向数据库应用程序的程序员。

## 示例代码约定

SQL代码示例贯穿本出版物。除此之外，代码并不特定于任何单个GBase 8s应用程序开发工具。

如果示例中只列出了SQL语句，则它们不会用分号分隔。例如，您可能会看到以下示例中的代码：

```
CONNECT TO stores_demo
...

DELETE FROM customer
      WHERE customer_num = 121
...

COMMIT WORK

DISCONNECT CURRENT
```

要将此SQL代码用于特定产品，必须应用该产品的语法规则。例如，如果使用的是SQL API，则必须在每条语句的开头使用EXEC SQL，在语句的结尾使用分号（或其他适当的分隔符）。如果使用DB - Access，则必须用分号分隔多个语句。

提示：代码示例中的省略号表示将在完整的应用程序中添加更多代码，但不必显示它来描述所讨论的概念。

有关为特定应用程序开发工具或SQL API使用SQL语句的详细说明，请参阅产品的文档。

# 1 开始使用

本章节描述了变更数据捕获 API，以及如何使用它。

## 1.1 变更数据捕获API

变更数据捕获 API 允许外部客户机应用程序从 GBase 8s 数据库捕获事务数据。

变更数据捕获(CDC) API 提供了捕获事务数据的功能。您可以使用各种客户端来运行这些功能，例如 JDBC、ODBC、ESQL/C 和 DB 访问。数据由标准 GBase 8s 智能大对象读取函数作为 CDC 记录返回。如何处理捕获的数据取决于您的应用程序。例如，您可以编写一个应用程序，将数据从 GBase 8s 数据库复制到另一个异构数据库。

可以捕获以下类型的操作：

- INSERT
- DELETE
- UPDATE
- TRUNCATE

变更数据捕获 API 开始从当前逻辑日志捕获事务，并顺序处理所有事务。第一次开始捕获特定表的数据时，数据捕获从当前日志位置开始。如果稍后停止捕获并重新启动它，则可以在逻辑日志中停止数据捕获的位置重新启动。您不能在逻辑日志中按时间倒转以捕获表的历史记录，也不能在逻辑记录中执行随机搜索。

在表的数据捕获开始时，更改数据捕获 API 提供了表架构信息，您可以在应用程序中使用这些信息来创建目标表。但是，数据捕获开始后对表模式的任何更改都不会被更改数据捕获 API 捕获。

更改数据捕获 API 只能在数据正在更改时提供数据；它不提供表内容的初始快照。如果需要填充的目标表，可以从外部将现有数据加载到目标表。或者，您可以为每一行创建表的虚拟更新，以便变更数据捕获 API 可以捕获这些更新并填充目标表。

更改数据捕获 API 不捕获对表架构的更改或任何其他数据库更改。

更改数据捕获 API 只能从启用了日志记录的数据库中捕获数据。

## 1.2 更改数据捕获API组件

变更数据捕获 API 由函数、系统数据库、CDC 记录和错误代码组成。

### 1.2.1 函数

更改数据捕获函数是通过使用 EXECUTE FUNCTION 语句运行的内置 SQL 函数。您可以使用这些函数来控制数据捕获。`cdc_opensess()` 函数的作用是：返回 CDC 会话 ID，这是一个用于检索捕获数据的智能大对象文件描述符。

`cdc_startcapture()` 函数的作用是：指定要从中捕获数据的表。其他函数指定开始或结束数据捕获。

您必须从客户端应用程序调用更改数据捕获函数。不能从数据库服务器中运行的用户定义例程调用此函数。

### 1.2.2 系统数据库

`syscdc` 系统数据库包含变更数据捕获函数和系统表。系统表存储有关更改数据捕获 API 错误代码和记录类型的信息。

### 1.2.3 错误代码

更改数据捕获 API 函数返回错误代码。如果成功或失败，大多数函数都会返回错误代码。更改数据捕获 API 错误代码列在 `syscdc` 数据库的 `syscdcerrcodes` 表中。您可以查询 `syscdcerrcodes` 表以确定函数执行是否失败。若失败，则可查看失败原因。

### 1.2.4 智能大对象读取函数

通过 `cdc_opensess()` 函数提供的智能大对象文件描述符，可以使用智能大对象读取函数读取捕获的数据。智能大对象读取函数不是变更数据捕获 API 的一部分。您可以使用智能大对象读取函数，例如 `mi_lo_read()` 或 `ifx_lo_read()`。

### 1.2.5 CDC 记录

更改数据捕获（CDC）记录由智能大对象读取功能返回，并提供当前正在捕获的事务的信息以及实际捕获的数据。

### 1.3 智能大对象读取函数

使用智能大对象读取函数将捕获的数据传输到应用程序可以访问的缓冲区。

您可以使用下表中列出的任何智能大对象读取函数，具体取决于您的应用程序语言。在特定会话期间，必须对所有读取调用使用相同的智能大对象读取函数。在同一会话中使用不同的函数可能会导致捕获数据的不完整传递。

表 1-1. 智能大对象读取函数

函数	参数	GBase 8s API	应用程序语言
mi_lo_read()	A pointer to a connection descriptor  A smart large object file descriptor  A data buffer The maximum number of bytes to read	DataBlade API	Use in a C language application.
ifx_lo_read()	A smart large object file descriptor  A data buffer	ODBC	Use in an ODBC application.
ifx_lo_read()	A smart large object file descriptor  A data buffer The maximum number of bytes to read  A pointer to an error code	ESQL/C	Use in a C language application.
IfxLoRead()	A smart large object file descriptor  A data buffer The maximum number of bytes to read	JDBC	Use in a Java application.



lfxBlob.Read()	A data buffer	.NET	Use in a .NET application.
----------------	---------------	------	----------------------------

## 1.4 读取超时

如果没有可检索的捕获数据，则 `read()` 类调用将在 `cdc_opensess()` 函数指定的超时时间内等待数据。如果超过了超时时间段，则向读取调用返回 `CDC_REC_TIMEOUT` 记录。`read` 类调用将 `CDC_REC_TIMEOUT` 记录传递到数据缓冲区并成功返回。

## 1.5 读取缓冲区大小

读取函数调用中指定的缓冲区大小应至少为 128 字节。读取缓冲区的最大大小为 2 GB。您可以通过计算最大可能的 CDC 记录大小（例如：`CDC_REC_INSERT` 记录），并将该值乘以在 `cdc_opensess()` 函数中指定的每次读取调用返回的最大记录数，来计算应用程序缓冲区的近似最小的 `buffer` 大小。

读取调用返回的数据量是读取调用中指定的缓冲区大小和要返回的最大记录数中的较小值。一次读取调用返回的记录数不超过最大值，即使这些记录中包含的字节数小于读取调用允许的最大字节数。但是，即使返回的记录数小于允许的最大数，也不会返回超过读取调用允许的最大字节数。如果记录不适合缓冲区，则返回尽可能多的记录，随后的读取调用将返回其余的记录。

## 1.6 智能大对象文件描述符

读取函数中的智能大对象文件描述符参数的值是 `cdc_opensess()` 函数返回的 CDC 会话 ID。

## 1.7 CDC记录序列号

返回给客户端的大多数更改数据捕获（CDC）记录都包含序列号。

与 CDC 记录关联的序列号是 BIGINT 数据类型。

CDC 记录序列号不一定与正在捕获的 GBase 8s 逻辑日志的 LSN 相同。

您可以比较为同一事务返回的 CDC 记录的序列号。在一个事务中，CDC 记录返回的序列号随着时间的推移而增加。较低的序列号表示 CDC 记录比序列号较高的 CDC 记录更早返回。

您可以比较不同事务的 CDC\_REC\_BEGINTX 记录的序号或 CDC\_REC\_COMMTX 记录的序号。每个提交的事务都有一个 CDC\_REC\_BEGINTX 记录和一个 CDC\_REC\_COMMTX 记录。CDC\_REC\_BEGINTX 和 CDC\_REC\_COMMTX 记录的序列号按单调顺序排列。较低的序列号表示关联的事务比与较高序列号关联的事务更早开始或提交。

## 1.8 捕获数据

您可以捕获大多数 GBase 8s 数据类型。您可以指定要在列级别捕获的数据。

数据捕获不支持以下数据类型：

- 简单大型对象（TEXT、BYTE 数据类型）
- 用户定义的数据类型
- 采集数据类型（SET、MULTISET、LIST、ROW 数据类型）

指定要捕获的数据

使用 `cdc_startcapture()` 函数指定一个表以及该表中要捕获的列。对于要捕获的每个表，必须运行一次 `cdc_startcapture()` 函数。有关当前正在捕获哪些表和列的信息，请查看 `syscdctab` 表。

## 1.9 结束表的捕获

要停止特定表的数据捕获，请运行 `cdc_endcapture()` 函数。运行 `cdc_endcapture()` 函数后，将从 `syscdctab` 表中删除该表的相关信息。

## 1.10 准备使用变更数据捕获API

在开始使用变更数据捕获 API 之前，必须准备好数据库和数据库服务器。

执行以下任务以准备使用变更数据捕获 API:

1、打开要从中捕获数据更改的所有数据库的日志记录。有关日志记录的信息，请参阅 GBase 8s SQL 指南：语法。

2、从 \$GBASEBTDIR/etc 目录以用户 gbasedbt 身份运行以下脚本：

```
syscdcv1.sql
```

3、通过以用户 gbasedbt 的身份创建到 syscdcv1 数据库的连接，验证该数据库是否存在。例如，您可以使用 DB Access 连接到 syscdcv1 数据库。

4、将 DB\_LOCALE 环境变量设置为与要从中捕获数据的数据库的区域设置相同。

## 1.11 编写应用程序以捕获数据更改

使用变更数据捕获功能控制数据捕获过程。处理 CDC 记录以提取数据。查询 syscdc 表以检索 CDC 记录和错误的符号名称和描述。

完成准备使用变更数据捕获 API 的先决条件任务。

应用程序应包含以下结构和功能：

- 存储表架构信息的结构。您可以使用表模式来分析列数据。
- 用于解释表模式信息并填充表模式结构的函数。您可以从 CDC\_REC\_TABSCHEMA 记录中获取表模式信息。
- 从数据缓冲区检索和分析列值的函数。
- 一个处理错误的函数。您可以查询 syscdcerrcodes 表以确定错误代码的符号名称和描述。

在应用程序中包含以下任务以捕获数据更改：

1、作为用户 gbasedbt，连接到客户端当前连接到的数据库服务器上的 syscdcv1 数据库。

2、通过运行 cdc\_opensess() 函数打开捕获会话。函数的作用是：返回会话 ID。

3、通过运行 `cdc_set_fullrowlogging()` 函数，为要从中捕获数据的每个表启用全行日志记录。

4、通过运行 `cdc_startcapture()` 函数指定要捕获的数据。对要从中捕获数据的每个表运行此函数。

5、通过运行 `cdc_activatesess()` 函数启动捕获过程。CDC 记录（包括包含捕获数据的记录）将返回给应用程序。

6、通过传递会话 ID 作为大对象文件描述符，使用智能大对象读取函数（如 `mi_lo_Read()`）读取包含捕获数据的 CDC 记录。对所有读取调用使用相同的智能大对象读取函数。

7、按列值分析数据。如果您正在用 Java 编写应用程序，则可以使用 `IfxToJavaType` 类将数据类型的 Informix 表示的字节流转换为适当的 Java 数据类型和值。

8、通过为每个表运行 `cdc_endcapture()` 函数来停止捕获数据。

9、通过为每个表运行 `cdc_set_fullrowlogging()` 函数来禁用全行日志记录。在禁用全行日志记录之前，请确保没有其他应用程序或进程依赖于全行日志。

10、通过运行 `cdc_closess()` 函数关闭捕获会话。

## 1.12 错误处理

要处理更改数据捕获函数返回的错误，请通过在 `syscdcerrcodes` 表中查找错误号的符号名来引用错误号。

向应用程序中添加代码以处理可能的错误情况。

1、为要单独处理的错误类型声明错误代码变量。

2、查询 `syscdcerrcodes` 表以查找与变更数据捕获错误代码的每个符号名称对应的错误号。

3、将错误代码变量设置为变更数据捕获错误编号。

4、添加代码以处理每个错误条件。

您可以使用 `cdc_errortext()` 函数返回指定符号名称。

## 1.13 处理智能大对象

变更数据捕获 API 不直接支持从捕获的 BLOB 或 CLOB 行检索智能大对象列数据。您必须使用 DataBlade API 或客户端 API 智能大对象读取函数来检索智能大对象。

要检索智能大对象列中的数据，请执行以下常规步骤：

- 1、使用变更数据捕获 API 检索包含智能大对象的数据行。
- 2、提取唯一标识数据行的列的值，如主键或唯一约束。
- 3、使用标识值运行 SQL SELECT 语句以检索数据行。
- 4、从数据行的列中打开智能大对象。
- 5、使用以下类型之一检索智能大对象数据：

智能大对象函数：

- DataBlade API 函数，如 `mi_lo_read()`、`mi_lo_to_buffer()` 或 `mi_lo_to_file()`

- SQL 函数，如 `LOTOFILE()`

- ESQL/C 函数，如 `ifx_lo_read()`

- 6、关闭智能大对象。

## 1.14 重新启动数据捕获

您可以在上次数据捕获会话结束的位置重新启动数据捕获。

重新启动位置是在上一个数据捕获会话中返回的 CDC 记录的序列号。您可以使用上一个数据捕获会话中处理的最后一个 CDC 记录的序列号。但是，为了保持事务完整性，您应该确定未处理提交或回滚操作的最后一个事务，并在该事务开始时重新启动捕获。在这种情况下，重新启动位置是不完整事务的 CDC\_REC\_BEGINTX 记录的最低序列号。为避免重新处理已提交的事务，还应确定在上一个数据捕获会话中已处理的 CDC\_REC\_COMMTX 记录的最大序号。

要重新启动数据捕获：

- 1、确定重启位置。要保持事务完整性：
  - a. 查找未返回 CDC\_REC\_COMMTX 或 CDC\_REC\_RBTX 记录的所有捕获事务。
  - b. 比较未完成事务的 CDC\_REC\_BEGINTX 记录的序列号。最低序列号是重新启动位置。
- 2、通过运行 `cdc_opensess()` 函数打开新的捕获会话。
- 3、对要重新启动的表运行 `cdc_startcapture()` 函数捕获数据。
- 4、通过运行 `cdc_activatesess()` 函数激活会话。传递适当的序列号作为位置参数。在处理的最后一个事务中，表的数据捕获重新启动。
- 5、丢弃 CDC\_REC\_COMMTX 序列号小于您在上一次数据捕获会话中处理的具有最大序列号的 CDC\_REC\_COMMTX 记录的任何事务。

## 1.15 监控数据捕获

您可以通过运行 `onstat -g cdc` 命令来监视数据捕获的状态。

要查看数据捕获会话的当前状态，请运行 `onstat -g cdc` 命令。对于此命令和所有其他 `onstat -g cdc` 命令选项，您可以指定单个会话或查看有关所有当前会话的信息。

要查看会话缓冲区的状态，请运行 `onstat -g cdc bufm` 命令。要查看有关会话配置的信息，请运行 `onstat -g cdc config` 命令。

要查看有关当前捕获的表的信息，请运行 `onstat -g cdc table` 命令。您可以为所有表提供单个表名或视图信息。

## 2 更改数据捕获功能

本章节描述了变更数据捕获功能的相关函数。

## 2.1 cdc\_activatesess() 函数

对于打开的捕获会话，开始从指定的日志和日志位置捕获数据。

会话激活后，syscdcses 表将更新。

语法：

```
cdc_activatesess ( session_ID , position )
```

参数：

表 2-1.cdc\_activatesess() 参数

参数	数据类型	描述
session_ID	INTEGER	The session ID of the open capture session for which to start capturing data.
position	BIGINT	Must be 0 or the restart position.

用法：

在使用 cdc\_opensess() 函数打开会话后，可以使用 cdc\_activatesess() 函数在指定的日志位置开始捕获数据。如果您是第一次在表上开始数据捕获，则位置必须为 0。如果您以前执行过数据捕获，您可以通过指定上一次捕获会话中返回的 CDC 记录的序列号来重新开始数据捕获。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值：

如果成功，则返回 0。

如果不成功，则返回与错误代码相对应的整数，并更新包含错误信息的 syscdcses 表。

## 2.2 cdc\_closesess() 函数

关闭与指定会话 ID 关联的捕获会话。

捕获会话使用的任何资源都将被释放。删除 syscdctab 中包含指定会话 ID 的行及其相关的 sysdcscses 表。

语法：

```
cdc_closesess ( session_ID )
```

参数：

表 2-2. cdc\_closesess() 参数

参数	数据类型	描述
session_ID	INTEGER	The session ID of the capture session that you want to close.

用法：

使用 cdc\_closesess() 函数关闭不再需要的捕获会话。如果捕获会话处于活动状态，则在会话关闭时立即停止所有数据捕获。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值：

如果成功，则返回 0。

如果失败，则返回与错误代码相对应的整数，并更新包含错误信息的 sysdcscses 表。

## 2.3 cdc\_deactivatesess() 函数

停止捕获活动捕获会话的数据。

更新 sysdcscses 表以显示捕获会话未处于活动状态。

语法：

```
cdc_deactivatesess ( session_ID )
```

参数：

表 2-3. cdc\_deactivatesess() 参数



参数	数据类型	描述
session_ID	INTEGER	The session ID of the capture session that you want to deactivate.

用法:

使用 `cdc_deactivatesess()` 函数停止捕获特定捕获会话的数据。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值:

如果成功，则返回 0。

如果不成功，则返回与错误代码相对应的整数，并更新包含错误信息的 `syscdcses` 表。

## 2.4 cdc\_endcapture() 函数

结束指定表的捕获。

将删除 `syscdctabs` 表中与指定会话 ID 和表关联的行。

意图:

```
cdc_endcapture ( session_ID , MBZ ,
"database:owner.table_name" )
```

参数:

表 2-4. cdc\_endcapture() 参数

参数	数据类型	描述
session_ID	INTEGER	The session ID of an open capture session.
MBZ	BIGINT	Must be 0. Reserved.

database:owner.table_name	LVARCHAR	<p>The qualified name of the table from which to capture data. The qualified name includes the following elements:</p> <p>database The name of the database in which the table exists.</p> <p>owner The name of the owner of the table.</p> <p>table The name of the table</p>
---------------------------	----------	--

用法:

使用 `cdc_endcapture()` 函数停止从特定表捕获数据。此功能不影响会话状态；会话保持打开和活动状态。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值:

如果成功，则返回 0。

如果不成功，则返回与错误代码相对应的整数，并更新包含错误信息的 `syscdcses` 表。

## 2.5 cdc\_errortext() 函数

返回与指定的符号错误名称对应的错误消息文本。符号错误名称列在 `syscdc` 数据库的 `syscdcerrcodes` 表中。

语法:

```
cdc_errortext ( ' error_name ' , ' locale_name ' )
```

参数

表 2-5. `cdc_errortext()` 参数

参数	数据类型	描述
----	------	----

error_name	LVARCHAR	The symbolic name of the error.
locale_name	LVARCHAR	The name of the locale in which to display the error text. If locale name parameter is SQL NULL or a string of 0 length ("") the default locale is used.

#### 用法:

使用 `cdc_errortext()` 函数返回从另一个 `cdc` 函数收到的错误的错误文本。并非所有区域设置中都有错误文本。如果 `cdc_errortext()` 函数未返回指定区域设置中的文本，请尝试使用其他区域设置（例如：“en\_us.819” 或 “en\_us.033”）再次运行该函数。

#### 返回值:

如果成功，则返回 `SQLCODE 0` 和错误消息文本。

如果不成功，则返回非零的 `SQLCODE`:

- 23109: Invalid locale specification.

区域设置名称不正确或找不到指定的区域设置。

- 1824: Message cannot be found.

区域设置有效，但在该区域设置的消息文件中未找到该消息。指定其他区域设置，如 “en\_us.033”。

- 其他 `SQLCODE` 表示内部错误。

#### 示例:

以下示例返回区域设置 `en_us.033` 中捕获的错误 `CDC_E_TABCAPTURED` 的错误文本:

```
> select cdc_errortext(' CDC_E_TABCAPTURED' , ' en_us.033' ) from
syscdcvrs;
```

(expression) The specified table is already being captured by the CDC session.

1 row(s) retrieved.

## 2.6 cdc\_opensess() 函数

打开捕获会话并创建会话 ID。将在会话的 syscdcses 表中插入一行。

语法:

```
cdc_opensess ( " server_name " , session_ID ,
timeout , max_recs , major_version , minor_version )
```

参数:

表 2-6. cdc\_opensess() 参数

参数	数据类型	描述
server_name	LVARCHAR	The name of the server. Must be the server to which the client application that is calling the <b>cdc_opensess()</b> function is connected.
session_ID	INTEGER	Must be 0.
timeout	INTEGER	Specifies the timeout behavior of a read call on the captured data:  <b>&lt;0</b> Do not timeout.  <b>1.</b> Return immediately if no data is available.  <b>2.</b> <b>or more</b> The number of seconds to wait for data before timing out.

max_recs	INTEGER	The maximum number of CDC records to return per read function call. This value takes precedence over the maximum number of bytes to return that is specified in the smart large object read function.
major_version	INTEGER	The major version number of the Change Data Capture API. Must be 1.
minor_version	INTEGER	The minor version number of the Change Data Capture API.  Must be 1 for new applications.  Can be 0 for existing applications.

#### 用法:

使用 `cdc_opensess()` 函数打开客户端应用程序和数据库服务器之间的通信会话。`cdc_opensess()` 返回的会话 ID 是提供给智能大对象读取函数的智能大对象文件描述符。要开始捕获数据，必须使用 `cdc_activatesess()` 函数和 `cdc_startcapture()` 函数。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

**注意:**如果您有多个应用程序使用变更数据捕获 API 并连接到同一 GBase 8s 服务器，则所有应用程序必须使用相同的 `major_version` 和 `minor_version` 参数值。

#### 返回值:

如果成功，则返回作为会话 ID 的整数。

如果不成功，则返回与错误代码对应的整数。

## 2.7 cdc\_recboundary() 函数

从当前返回的 CDC 记录开始重新启动数据捕获。

语法：

```
cdc_recboundary ( session_ID )
```

参数：

表 2-7. cdc\_recboundary() 参数

参数	数据类型	描述
session_ID	INTEGER	The session ID of the open capture session.

用法：

如果需从当前日志记录的开头重新启动捕获，请使用 cdc\_recboundary() 函数。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值：

如果成功，则返回一个正整数，表示在当前会话期间捕获但跳过的完整或部分日志记录数。

如果不成功，则返回与错误代码相对应的整数，并更新包含错误信息的 syscdcses 表。

## 2.8 cdc\_set\_fullrowlogging() 函数

启用或禁用表的 full-row 日志记录。

意图：

在开始从表中捕获数据之前，必须运行此函数以启用表的全行日志记录。

运行此函数时，DB\_LOCALE 环境变量必须设置为与数据库区域设置相同的区域设置。

```
cdc_set_fullrowlogging ( "database:owner.table_name" ,
logging )
```

参数:

表 2-8. cdc\_set\_fullrowlogging() 参数

参数	数据类型	描述
database:owner.table_name	LVARCHAR	The qualified name of the table. The qualified name includes the following elements:  database The name of the database in which the table exists.  owner The name of the owner of the table.  table The name of the table.
logging	INTEGER	<ul style="list-style-type: none"> <li>•0 Disable full-row logging</li> <li>•1 Enable full-row logging</li> </ul>

用法:

使用 cdc\_set\_fullrowlogging() 函数在要从中执行数据捕获的表上启用 full-row 日志记录。此函数必须使用用户 gbasedbt 运行。停止从表中捕获数据后，可以禁用 full-row 日志记录。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值

如果成功，则返回 0。

如果不成功，则返回与错误代码相对应的整数，并更新包含错误信息的 sysdcscses 表。

## 2.9 cdc\_startcapture() 函数

指定要从表中开始捕获的数据。

如果捕获会话处于打开状态且处于活动状态（您已经运行了 `cdc_activatesess()` 函数），则数据捕获将立即在指定表中的指定列上开始。否则，数据捕获将在激活打开捕获会话时开始。

运行此函数时，`DB_LOCALE` 环境变量必须设置为与数据库区域设置相同的区域设置。

将在与指定会话 ID 和表关联的 `syscdctab` 表中添加一行。

语法：

```
cdc_startcapture ( session_ID , MBZ ,
“database:owner.table_name” , “ column_name[, column_name] ” ,
user_data )
```

参数：

表 2-9. `cdc_startcapture()` 参数

参数	数据类型	描述
<code>session_ID</code>	INTEGER	The session ID of an open capture session.
<code>MBZ</code>	BIGNIT	Must be 0. Reserved.
<code>database:owner.table_name</code>	LVARCHAR	The qualified name of the table from which to capture data. The qualified name includes the following elements:  database The name of the database in which the table exists.  owner The name of the owner of the table.  table The name of the table.
<code>column_name</code>	LVARCHAR	A comma-separated list of column names in the specified



		table, from which to capture data.
user_data	INTEGER	The table identifier.

用法:

使用 `cdc_startcapture()` 函数指定一个表和该表中的列,从中开始捕获数据。不能包含具有简单大型对象、用户定义数据类型或集合数据类型的列。

表标识符是应用程序中用于唯一标识将参与数据捕获的每个表的数字。

必须从客户端应用程序调用此函数。不能从数据库服务器中运行的用户定义例程调用此函数。

返回值:

如果成功,则返回 0。

如果不成功,则返回与错误代码相对应的整数,并使用错误信息更新 `syscdcess` 表。

### 3 更改数据捕获记录

本章节描述从打开的捕获会话读取函数的调用返回的 CDC 记录。

CDC 记录格式

变更数据捕获 (CDC) 记录包含所有记录共有的标题,之后是 CDC 记录类型的特定标题。

`CDC_REC_INSERT`、`CDC_REC_DELETE`、`CDC_REC_UPDBEF` 和 `CDC_REC_UPDAFT` 记录也包含列数据。

所有 CDC 记录共用的标题描述了 CDC 记录的大小和类型。

表 3-1. 所有 CDC 记录公共消息头信息

区分	长度	描述
Header size	4 bytes	The number of bytes in the common and CDC record-specific headers.

Payload size	4 bytes	The number of bytes of data in the record after the common and CDC record-specific headers.
Packet scheme	4 bytes	The packetization scheme number of one of the packetization schemes contained in the <b>syscdcpacketschemes</b> table. The only packetization scheme is 66, CDC_PKTScheme_LRECBINARY.
Record number	4 bytes	The record number of one of the CDC records contained in the <b>syscdcrectypes</b> table.

### 3.1 CDC\_REC\_BEGINTX记录

指示事务的开始。

CDC\_REC\_BEGINTX 记录的报头位于公共报头之后。

报文头后面没有数据；公共报头中的有效负载大小为 0。

表 3-2. CDC\_REC\_BEGINTX 记录的格式

区分	长度	描述
Sequence number	8 bytes	The sequence number of the record.
Transaction ID	4 bytes	The transaction ID.
Start time	8 bytes	The UTC time at which the transaction began, in time_t format.
User ID	4 bytes	The operating system user ID of the user who started the transaction.

## 3.2 CDC\_REC\_COMMTX记录

指示事务已提交。

CDC\_REC\_COMMTX 记录的报头位于公共报头之后。

报文头后面没有数据；公共报头中的有效负载大小为 0。

表 3-3. CDC\_REC\_COMMTX 记录的格式

区分	长度	描述
Sequence number	8 bytes	The sequence number of the record.
Transaction ID	4 bytes	The transaction ID.
Commit time	8 bytes	The UTC time at which the transaction was committed, in time_t format.

## 3.3 CDC\_REC\_DELETE记录

提供由于删除操作而删除的行。

CDC\_REC\_DELETE 记录由以下字段组成：

- 公共标题。
- 记录特定标题。
- 字段，列出行中每个可变长度列的大小（如果有）。
- 每个固定长度列的 v 列数据（如果有）。
- 每个可变长度列的 v 列数据（如果有）。

公共标头的标头大小字段中的值表示公共标头、记录特定标头和列出可变长度列大小的字段的组合所占用的字节数。

公共标头中有效负载大小字段中的值表示固定长度列和可变长度列的列数据字节数。

特定于记录的标题:

特定于 CDC\_REC\_DELETE 记录的报头位于公共报头之后。

表 3-4. CDC\_REC\_DELETE 记录头

区分	长度	描述
Sequence number	8 bytes	The sequence number associated with the DELETE operation.
Transaction ID	4 bytes	The transaction ID.
User data	4 bytes	The table identifier passed to the <code>cdc_startcapture()</code> function and stored in the <code>syscdtabs</code> table.
Flags	4 bytes	Reserved.

可变长度列大小字段:

如果要删除的行中有可变长度的列，则会显示每个列的 4 字节字段，其中包含列大小。列大小字段的顺序与 CDC\_REC\_TABSCHEMA 记录中列的顺序相同。

固定长度列数据:

来自固定长度列的数据（如果有的话）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

可变长度列数据:

来自可变长度列的数据（如果有）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

### 3.4 CDC\_REC\_DISCARD记录

指示应放弃事务的某些操作。

应丢弃此记录之后的同一事务的 CDC 记录。

特定于 CDC\_REC\_DISCARD 记录的报头在公共报头之后。标题后面没有数据；公共报头中的有效负载大小为 0。

表 3-5. CDC\_REC\_DISCARD 记录的格式

区分	长度	描述
Sequence number	8 bytes	The sequence number of the record. Any CDC records that have the same transaction ID value and that have a sequence number greater than or equal to this sequence number should be discarded.
Transaction ID	4 bytes	The transaction ID.

### 3.5 CDC\_REC\_ERROR记录

指示发生错误，会话不再有效。

特定于 CDC\_REC\_ERROR 记录的报头在公共报头之后。标题后面没有数据；公共报头中的有效负载大小为 0。

表 3-6. CDC\_REC\_ERROR 记录格式

区分	长度	描述
----	----	----

Flags	4 bytes	Hexadecimal flag: <ul style="list-style-type: none"> <li>▪ 0x1 indicates that the capture session is no longer valid and the only valid operation is to run the <b>cdc_closesess()</b> function to close the session.</li> <li>▪ any other value indicates that the session is still valid.</li> </ul>
Error code	4 bytes	The error code.

### 3.6 CDC\_REC\_INSERT记录

提供插入操作产生的行。

CDC\_REC\_INSERT 记录由以下字段组成：

- 公共标题。
- 记录特定标题。
- 字段，列出行中每个可变长度列的大小（如果有）。
- 每个固定长度列的 v 列数据（如果有）。
- 每个可变长度列的 v 列数据（如果有）。

公共标头的标头大小字段中的值表示公共标头、记录特定标头和列出可变长度列大小的字段的组合所占用的字节数。

公共标头中有效负载大小字段中的值表示固定长度列和可变长度列的列数据字节数。

特定于记录的标题：

特定于 CDC\_REC\_INSERT 记录的报头位于公共报头之后。

表 3-7. CDC\_REC\_INSERT 记录头

区分	长度	描述
Sequence number	8 bytes	The sequence number associated with the INSERT operation.
Transaction ID	4 bytes	The transaction ID.
User data	4 bytes	The table identifier passed to the <code>cdc_startcapture()</code> function and stored in the <code>syscdtabs</code> table.
Flags	4 bytes	Reserved.

可变长度列大小字段：

如果要插入的行中有可变长度的列，则会显示每个列的 4 字节字段，其中包含列大小。列大小字段的顺序与 CDC\_REC\_TABSCHEMA 记录中列的顺序相同。

固定长度列数据：

来自固定长度列的数据（如果有的话）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

可变长度列数据：

来自可变长度列的数据（如果有）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

### 3.7 CDC\_REC\_RBTX记录

指示事务已回滚。

特定于 CDC\_REC\_RBTX 记录的报头在公共报头之后。标题后面没有数据；公共报头中的有效负载大小为 0。

表 3-8. CDC\_REC\_RBTX 记录格式

区分	长度	描述
Sequence number	8 bytes	The sequence number associated with the ROLLBACK operation.
Transaction ID	4 bytes	The transaction ID.

### 3.8 CDC\_REC\_TABSCHEMA记录

描述从中捕获数据的表。

公共标头中有效负载大小字段中的值表示列名和数据类型列表占用的字节数。

特定于 CDC\_REC\_TABSCHEMA 记录的报头位于公共报头之后。

表 3-9. CDC\_REC\_TABSCHEMA 记录的格式

区分	长度	描述
User data	4 bytes	The table identifier that was specified in the <b>cdc_startcapture()</b> function for the table being captured.
Flags	4 bytes	Must be 0.
Fixed-length size	4 bytes	The number of bytes of data in fixed-length columns in the table.



Fixed-length columns	4 bytes	<p>The number of fixed-length columns in the table being captured.</p> <p>A 0 indicates that there are no fixed-length columns.</p>
Variable-length columns	4 bytes	<p>The number of variable-length columns in the table being captured.</p> <p>A 0 indicates that there are no variable-length columns.</p>
Column names and data types	variable byte length	<p>A comma-separated list of column names and data types in UTF-8 format. The column list conforms to the syntax of the column list in a CREATE TABLE statement.</p> <p>Names of any fixed-length columns appear before names of any variable-length columns.</p> <p>The number of columns equals the number of fixed-length columns plus the number of variable-length columns.</p>

### 3.9 CDC\_REC\_TIMEOUT记录

指示读取调用在 `cdc_opensess()` 函数中指定的超时时间之前未返回数据。

特定于 CDC\_REC\_TIMEOUT 记录的报头在公共报头之后。标题后面没有数据；公共报头中的有效负载大小为 0。

表 3-10. CDC\_REC\_TIMEOUT 记录的格式

区分	长度	描述
Sequence number	8 bytes	The sequence number of the last data retrieved from the source database.

### 3.10 CDC\_REC\_TRUNCATE记录

指示对表执行了截断操作。

特定于 CDC\_REC\_TRUNCATE 记录的报头位于公共报头之后。标题后面没有数据；公共报头中的有效负载大小为 0。

表 3-11. CDC\_REC\_TRUNCATE 记录的格式

区分	长度	描述
Sequence number	8 bytes	The sequence number associated with the TRUNCATE operation.
Transaction ID	4 bytes	The transaction ID.
User data	4 bytes	The table identifier passed to the <code>cdc_startcapture()</code> function and stored in the <code>syscdtabs</code> table.

### 3.11 CDC\_REC\_UPDAFT记录

提供更新操作后的行图像。

CDC\_REC\_UPDAFT 记录由以下字段组成：

- 公共标题。
- 记录特定标题。

- 字段，列出行中每个可变长度列的大小（如果有）。
- 每个固定长度列的 v 列数据（如有）。
- 每个可变长度列的 v 列数据（如果有）。

公共标头的标头大小字段中的值表示公共标头、记录特定标头和列出可变长度列大小的字段的组合所占用的字节数。

公共标头中有效负载大小字段中的值表示固定长度列和可变长度列的列数据字节数。

记录特定标题：

特定于 CDC\_REC\_UPDAFT 记录的报头遵循公共报头。

表 3-12. CDC\_REC\_UPDAFT 记录头

区分	长度	描述
Sequence number	8 bytes	The sequence number associated with the UPDATE operation.
Transaction ID	4 bytes	The transaction ID.
User data	4 bytes	The table identifier passed to the <code>cdc_startcapture()</code> function and stored in the <code>syscdtabs</code> table.
Flags	4 bytes	Reserved.

可变长度列大小字段：

如果正在更新的行中有可变长度的列，则会显示每个列的 4 字节字段，其中包含列大小。列大小字段的顺序与 CDC\_REC\_TABSCHEMA 记录中列的顺序相同。

固定长度列数据：

来自固定长度列的数据（如果有的话）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

可变长度列数据：

来自可变长度列的数据（如果有）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

### 3.12 CDC\_REC\_UPDBEF记录

在更新操作之前提供行的图像。

CDC\_REC\_UPDBEF 记录由以下字段组成：

- 公共标题。
- 记录特定标题。
- 字段，列出行中每个可变长度列的大小（如果有）。
- 每个固定长度列的 v 列数据（如有）。
- 每个可变长度列的 v 列数据（如果有）。

公共标头的标头大小字段中的值表示公共标头、记录特定标头和列出可变长度列大小的字段的组合所占用的字节数。

公共报头中有效负载大小字段中的值表示固定长度列和可变长度列的列数据字节数。

特定于记录的标题：

特定于 CDC\_REC\_UPDBEF 记录的报头在公共报头之后。

表 3-13. CDC\_REC\_UPDBEF 记录头

区分	长度	描述
Sequence number	8 bytes	The sequence number associated with the UPDATE operation.

Transaction ID	4 bytes	The transaction ID.
User data	4 bytes	The table identifier passed to the <code>cdc_startcapture()</code> function and stored in the <code>syscdtabs</code> table.
Flags	4 bytes	Reserved.

可变长度列大小字段：

如果正在更新的行中有可变长度的列，则会显示每个列的 4 字节字段，其中包含列大小。列大小字段的顺序与 CDC\_REC\_TABSCHEMA 记录中列的顺序相同。

固定长度列数据：

来自固定长度列的数据（如果有的话）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

可变长度列数据：

来自可变长度列的数据（如果有）按照相应列在 CDC\_REC\_TABSCHEMA 记录中列出的顺序显示。

## 4 syscdc系统数据库

syscdc 系统数据库包含存储有关变更数据捕获 API 的信息的表。

syscdc 数据库只能由用户 `gbasedbt` 访问或连接。它使用 UTF-8 区域语言。不能更改 syscdc 数据库中的表；您只能查询进行查询。

### 4.1 syscdcerrcodes表

包含更改数据捕获 API 使用的错误代码。

使用此表查找与错误代码对应的符号名称和描述。

表 4-1. syscdcerrcodes 表

参数	数据类型	描述
errcode	INTEGER	Numeric value of the error.
errname	VARCHAR(16)	Symbolic name of the error.
errdesc	VARCHAR(127)	Error description.

## 4.2 syscdrectypes表

使用此表查找与记录代码相对应的符号名称和描述。

表 4-2. syscdrectypes 表

参数	数据类型	描述
recnum	INTEGER	Numeric value of the record type.
recname	VARCHAR(16)	Symbolic name of the record type.
recdesc	VARCHAR(127)	Record type description.

## 5 更改数据捕获错误代码

如果变更数据捕获函数遇到问题，它将返回错误代码。如果成功，大多数函数返回 0。

不保证后续版本中的错误号保持不变。在应用程序代码中始终使用符号名称。您可以使用 `cdc_errortext()` 函数查看与符号错误名称对应的错误消息文本。

表 5-1.更改数据捕获错误代码

符号名称	描述
CDC_E_OK	Operation succeeded.
CDC_E_NOCDCDB	The <b>syscdc</b> database does not exist.
CDC_E_APIVERS	The requested CDC API behavior version is not valid or is unsupported.
CDC_E_NODB	The specified database does not exist.
CDC_E_DBNOTLOGGED	The specified database is not logged.
CDC_E_NOTAB	The specified table does not exist.
CDC_E_TABPROPERTIES	The table properties do not support capture: it is a temporary table, a view, or otherwise not logged.
CDC_E_NOCOL	The specified column does not exist.
CDC_E_NOSES	The specified CDC session does not exist.
CDC_E_NOREOPEN	The CDC session cannot be reopened.
CDC_E_TABCAPTURED	The specified table is already being captured by the CDC session.
CDC_E_TABNOTCAPTURED	The specified table is not being captured by the CDC session.
CDC_E_ARGNULL	An argument to the function has the SQL NULL value, which is not allowed.
CDC_E_LSN	Data at the requested log sequence number is unavailable for capture.
CDC_E_DUPLSESS	A CDC session is already active.
CDC_E_ARG	A parameter passed to the function is not valid.
CDC_E_ARG1	The first parameter passed to the function is not valid.
CDC_E_ARG2	The second parameter passed to the function is not valid.

CDC_E_ARG3	The third parameter passed to the function is not valid.
CDC_E_ARG4	The fourth parameter passed to the function is not valid.
CDC_E_ARG5	The fifth parameter passed to the function is not valid.
CDC_E_ARG6	The sixth parameter passed to the function is not valid.
CDC_E_INTERNAL	Internal error. Contact IBM Support.
CDC_E_NOMEM	Memory allocation failed.
CDC_E_MUSTCLOSE	The CDC capture session cannot continue and must be closed.
CDC_E_BADSTATE	The resource state does not allow the attempted operation.
CDC_E_BADCHAR	A byte sequence that is not a valid character in the character code set was encountered.
CDC_E_INTERRUPT	The CDC session was interrupted.
CDC_E_UNIMPL	Unimplemented feature.
CDC_E_LOCALEMISMATCH	The locale setting in the environment does not match the locale of the database.

## 6 onstat -g cdc

监控变更数据捕获中涉及的会话。

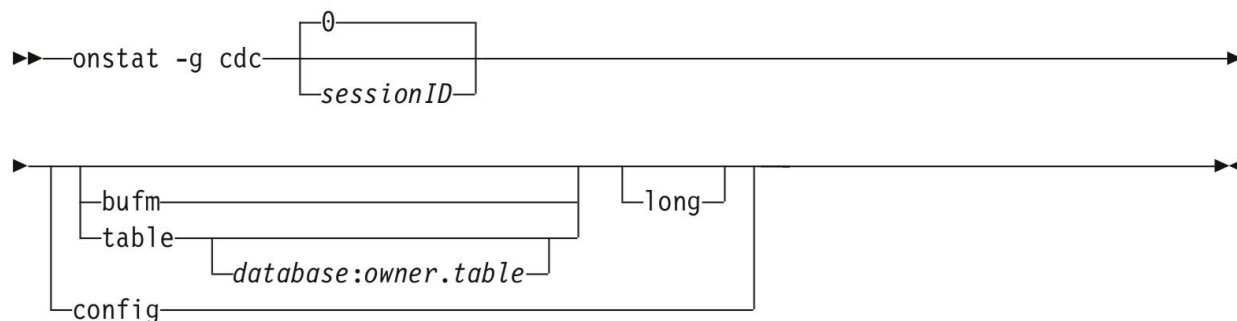




表 6-1. onstat -g cdc 语法元素

元素	内容
<b>bufm</b>	<p>Displays information about the buffers being used by the session, including:</p> <ul style="list-style-type: none"> <li>•The highest number of buffers used by the session.</li> <li>•The number of buffers currently being used by the session.</li> <li>•With the <b>long</b> option, the address of each allocated buffer.</li> </ul>
<b>config</b>	<p>Displays information about the session configuration, including:</p> <ul style="list-style-type: none"> <li>•The read timeout setting for the session, in seconds.</li> <li>•The maximum number of records returned by a read call.</li> </ul>
database:owner.table	<p>The fully-qualified name of the table for which to display information. The qualified name includes the following elements:</p> <ul style="list-style-type: none"> <li>•database: The name of the database in which the table exists.</li> <li>•owner: The name of the owner of the table.</li> <li>•table: the name of the table.</li> </ul>
<b>long</b>	<p>Provides additional detail for sessions, the <b>bufm</b> option, or the <b>table</b> option.</p>
sessionID	<p>Displays information for the specified session ID:</p> <ul style="list-style-type: none"> <li>•The associated SQL session ID.</li> <li>•The number of tables being captured by the session.</li> <li>•With the <b>long</b> option, information about the number of records processed by the session.</li> </ul> <p>If you do not specify a session ID, or if you specify a session ID of 0, information for all sessions is displayed.</p>
<b>table</b>	<p>Displays information about the tables being captured, including:</p> <ul style="list-style-type: none"> <li>•The number of tables being captured in a session.</li> <li>•The full name of each table being captured.</li> <li>•The time when data capture on each table started.</li> <li>•With the <b>long</b> option, information about the captured columns for each table.</li> </ul>

	<p>If you specify a fully-qualified table name, only the information for that table is displayed. If you do not specify a table name, information for all tables is displayed.</p>
--	--