SYBASE®

用户指南

Sybase ETL

4.8

文档 ID：DC00961-01-0480-01
最后修订日期：2009 年 2 月

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目录

关于本手册

读者

本指南面向 Sybase ETL Development 用户。

如何使用本手册

本手册包含以下章节：

- 第 1 章“Sybase ETL”概述了 Sybase ETL 体系结构以及 Sybase ETL Development 和 Sybase ETL Server 的功能集。
- 第 2 章“快速入门”介绍如何开始使用 Sybase ETL。它可使您熟悉 Sybase ETL Development 界面，并说明了使用此界面可以执行的功能。
- 第 3 章“项目和作业”指导您创建、模拟和执行项目及作业。本章讨论如何使用模拟模式，以及如何使用模板创建项目和作业。
- 第 4 章“高级概念和工具”介绍了一些可以简化设计工作的内置工具。
- 第 5 章“组件”介绍用于创建项目和作业的 Sybase ETL 组件。
- 第 6 章“Sybase ETL Server”介绍如何使用 Sybase ETL Server。
- 附录 A“函数参考”说明了在 Sybase ETL 中可用的内置函数。
- 附录 B“连接参数”说明了数据库配置选项。此外，还提供了有关部分所支持接口的其它信息。
- 附录 C“将 ETL 用于渐变维度”介绍渐变维度 (SCD)，包括某些常见 SCD 方案，并介绍如何使用 Sybase ETL 实现这些方案。
- 附录 D“最佳实践”介绍使用 Sybase ETL 的建议和原则。

相关文档

有关详细信息，请参见以下文档：

- *Sybase ETL 新增功能指南* – 介绍 Sybase ETL 4.8 中的新增功能。
- *Sybase ETL 发行公告* – 包含未能及时写入手册的最新信息。

其它信息来源

使用 Sybase Getting Started CD、 SyBooks CD 及 Sybase Product Manuals Web 站点可以了解有关产品的详细信息：

- Getting Started CD 包含 PDF 格式的发行公告和安装指南，也可能包含 SyBooks CD 中未收纳的其它文档或更新信息。Getting Started CD 随软件一起提供。要阅读或打印 Getting Started CD 上的文档，需要使用 Adobe Acrobat Reader，该软件可以通过 CD 上提供的链接从 Adobe Web 站点免费下载。
- SyBooks CD 含有产品手册，随软件提供。基于 Eclipse 的 SyBooks 浏览器使您能够阅读以基于 HTML 的简单易用格式编写的手册。

  有些文档可能是以 PDF 格式提供的，您可以通过 SyBooks CD 上的 PDF 目录访问这些文档。要阅读或打印 PDF 文件，需要使用 Adobe Acrobat Reader。

  有关安装和启动 SyBooks 的说明，请参见 Getting Started CD 上的 *SyBooks 安装指南* 或 SyBooks CD 上的 *README.txt* 文件。
- Sybase Product Manuals Web 站点是 SyBooks CD 的联机版本，您可以使用标准 Web 浏览器进行访问。除了产品手册之外，还可以找到“EBFs/Maintenance”、“Technical Documents”、“Case Management”、“Solved Cases”、“Newsgroups”和 “Sybase Developer Network”的链接。

  若要访问 Sybase Product Manuals Web 站点，请转至位于 http://www.sybase.com/support/manuals/ 的 “Product Manuals”。

Web 上的 Sybase 认证

Sybase Web 站点上的技术文档经常更新。

❖ **查找有关产品认证的最新信息**

1. 将 Web 浏览器定位到位于 http://www.sybase.com/support/techdocs/ 的 “Technical Documents”。
2. 单击 “Certification Report”。
3. 在 “Certification Report”过滤器中选择相应的产品、平台和时间范围，然后单击 “Go”。
4. 单击 “Certification Report”标题以显示此报告。

❖ **查找有关组件认证的最新信息**

1. 将 Web 浏览器定位到位于 http://certification.sybase.com/ 的 “Availability and Certification Reports”。
2. 在 “Search by Base Product”（按基本产品搜索）下面选择产品系统和产品，或在 “Search by Platform”（按平台搜索）下面选择平台和产品。

3 选择 “Search”以显示所选项目的可用性和认证报告。

❖ **创建 Sybase Web 站点 （包括支持页）的个性化视图**

设置 MySybase 配置文件。MySybase 是一项免费服务，它允许您创建 Sybase Web 页的个性化视图。

1 将 Web 浏览器定位到位于 http://www.sybase.com/support/techdocs/ 的 “Technical Documents”。

2 单击 “MySybase”并创建 MySybase 配置文件。

Sybase EBF 和软件维护

❖ **查找有关 EBF 和软件维护的最新信息**

1 将 Web 浏览器定位到位于 http://www.sybase.com/support 的 “Sybase Support”页面。

2 选择 “EBFs/Maintenance”。如果出现提示，请输入您的 MySybase 用户名和口令。

3 选择产品。

4 指定时间范围并单击 “Go”。随即显示一个 EBF/ 维护版本的列表。

挂锁图标表示您没有注册为 “Technical Support Contact”，因此您没有某些 EBF/ 维护版本的下载授权。如果您尚未注册，但拥有 Sybase 代表提供的或通过支持合同获得的有效信息，请单击 “Edit Roles”将 “Technical Support Contact”角色添加到 MySybase 配置文件中。

5 单击 “Info”图标显示 “EBF/Maintenance”报告，或者单击产品说明下载软件。

约定

本手册中使用的语法约定如下：

关键字	定义
commands 和 methods	命令名、命令选项名、实用程序名、实用程序标记、 Java 方法 / 类 / 软件包及其它关键字用小写 Arial 字体显示。
variable	斜体表示： • 程序变量，例如 *myServer* • 必须被替换的输入文本部分；例如： `Server.log` • 文件名

关键字	定义
“File”	“Save”	菜单名称和菜单项以纯文本格式显示。竖线展示如何浏览菜单选项。例如，“File”	“Save”表示从“File”菜单选择“Save”。
`package 1`	“Monospace”字体表示： • 您在 GUI 界面、命令行中或作为程序文本输入的信息 • 示例程序片段 • 输出样本片段

易用性特点

本文档提供专门针对易用性编写的 HTML 版本。可以利用适应性技术（如屏幕阅读器）浏览 HTML，也可以用屏幕放大器查看。

Sybase ETL 和 HTML 文档已经过了测试，符合美国政府 “第 508 条 易用性”的要求。符合 “第 508 节”的文档一般也符合非美国的易用性原则，如针对 Web 站点的 World Wide Web 协会 (W3C) 原则。

注释 您可能需要对辅助工具进行配置以实现最优化。某些屏幕阅读器按照大小写来辨别文本，例如将 “ALL UPPERCASE TEXT”看作首字母的缩写，而将 “MixedCase Text”看作单词。您可能会发现按语法约定来配置工具更为方便实用。有关工具的信息，请查阅文档。

有关 Sybase 如何支持易用性的信息，请参见位于 http://www.sybase.com/accessibility 的 “Sybase Accessibility”。Sybase Accessibility 站点包括有关 “第 508 节”和 W3C 标准的信息的链接。

如果需要帮助

对于购买了支持合同的每项 Sybase 安装，均指定了一位或多位人员负责与 Sybase 技术支持部门联系。如果您通过手册或联机帮助不能解决问题，请让指定的人员与您所在区域的 Sybase 技术支持部门或 Sybase 子公司联系。

第 1 章

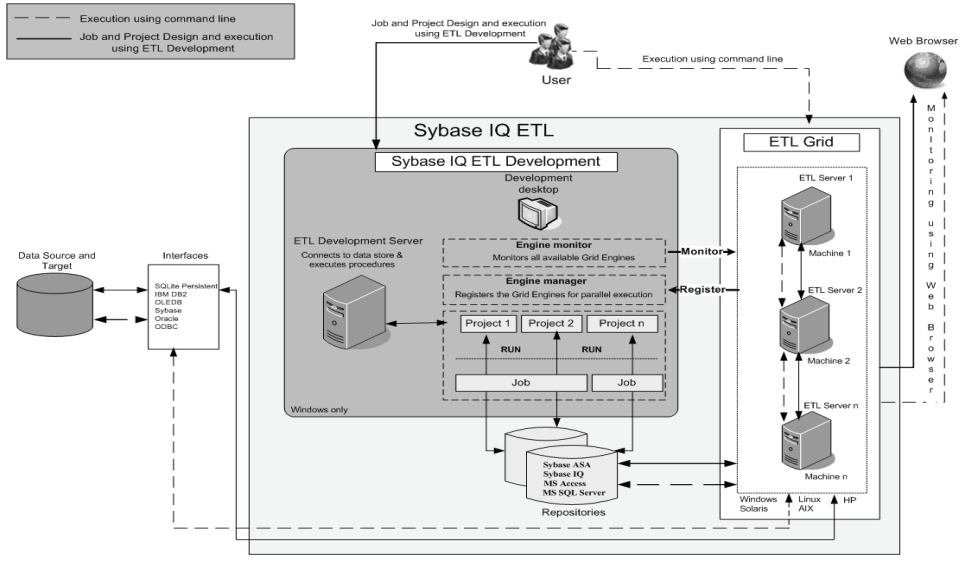
Sybase ETL

使用 Sybase ETL 可以通过一组全面的转换函数从多个异构数据源提取数据，并将其装载到一个或多个数据目标。它提供了可跨操作系统边界和计算机执行并行转换处理的伸缩式网格体系结构。

Sybase ETL 的功能包括：

- 数据提取 – 从各种数据源提取数据。
- 数据转换 – 转换、清除、合并和拆分数据流。
- 数据装载 – 使用 update、 insert、 delete 或 bulk copy 语句将数据装载到目标数据库。

Sybase ETL 体系结构



Sybase ETL 组件

Sybase ETL 包括 Sybase ETL Development 和 Sybase ETL Server。

Sybase ETL Development 只能在 Windows 上使用，它是一种图形用户界面 (GUI) 工具，用于创建和设计数据转换项目和作业。此工具提供了完整的模拟和调试环境，用于加速 ETL 转换流程的开发。

Sybase ETL Server 是一种可伸缩的分布式网格引擎，它通过使用 Sybase ETL Development 设计的转换流程连接到数据源，提取数据并将其装载到数据目标。请参见第 165 页上的 “Sybase ETL Server”。

Sybase ETL Development 包含控制实际处理 （例如连接到数据库和执行过程）的 ETL Development Server。若要并行执行作业和项目，您可以在网络中的不同操作系统上添加多个 ETL 服务器。每个服务器均向所有其它对等服务器公开某些服务。Sybase ETL 使用网格上的各种服务器并行执行多个项目和作业，从而改善了转换速度的可伸缩性。

Sybase ETL Server 采用称为接口的方法或驱动程序连接目标数据库或源数据库。在支持的接口中，用于连接 Sybase IQ 的 ODBC 驱动程序由 Sybase IQ 安装程序自动安装。若要安装支持的其它接口，请参见相应的供应商文档。

注释 若要并行执行使用 Sybase ETL Development 开发的项目和作业，必须安装以单独可执行文件形式提供的 Sybase ETL Server。

您还可以使用命令行在所有支持平台上执行作业和项目。有关详细信息，请参见第 169 页上的 “使用 ETL Server 执行项目和作业”。

注册 ETL 服务器

所有添加到网格中的 ETL 服务器都必须经过注册。您可以使用 Sybase ETL Development 工具中提供的 Engine Manager 注册服务器。请参见第 66 页上的 “使用多个引擎可缩短作业执行时间”。

监控 ETL 服务器

您可以使用 Sybase ETL Development 工具中提供的 Engine Monitor 来监控网络中的服务器。请参见第 67 页上的 “Engine Monitor”。您还可以使用 Web 浏览器来监控从命令行启动的远程项目和作业。请参见第 171 页上的 “使用 Web 浏览器监控项目和作业”。

注释 在本指南中，术语 “网格引擎” 和 “ETL 服务器” 可交换使用。

Sybase ETL 概念

本节介绍 Sybase ETL 的概念。

项目和作业

项目是组件、链接和转换规则的集合。每个项目均包含一个或多个用于在项目运行时按顺序模拟或执行的步骤。当模拟或执行这些步骤时，组件会连接到各种数据源，然后从数据源读取数据，再基于转换规则进行数据转换。项目由可以自由排列的各种组件组成。通过从 “Component Store” 的一个区域将组件拖动到 “Design” 窗口，可以将组件添加到项目。

可以在作业中按顺序或并行运行多个项目。作业控制项目执行的顺序。可以对作业进行安排和监控。

运行项目和作业

可以使用模拟模式或执行模式运行项目。

这两种模式执行项目中所包含组件的所有功能，包括将数据物理传输到数据目标或数据接收器。

模拟 模式允许您：

- 运行带有未保存更改的项目。
- 分步监控和验证转换过程。对于任何链接和任何包含组件之内数据流均可见。也可以检查任何组件并修改映射和计算。

  作出更改后，可以使用新的设置重新初始化该组件，然后转到下一个组件。更改任一组件后，不必再从项目的开头处启动模拟。

可以以交互方式或非交互方式模拟项目。在以上任一方式中，均会在模拟结束时将数据写入数据接收器。有关详细信息，请参见 第 26 页上的“交互式模拟项目”和 第 27 页上的 “非交互式模拟项目”。

注释 注意如果您要以模拟模式运行项目且目标是测试转换规则，则可能需要使用测试数据目标。

执行 模式允许您：

- 运行已保存到存储库的项目和作业。未保存的更改不会执行。
- 直接执行项目并将更改反映在数据接收器中。您无法分步监控转换过程。

注释 项目和作业可以从 Sybase ETL Development 执行，也可以作为预定任务执行。请参见第 50 页上的 “管理作业和预定任务”。

自定义项目

可以创建数据转换项目，而无需手工输入任何一行编程代码或 SQL 语句。例如，您可以执行以下操作：

- 使用 Query Designer 在查询、查找定义、预处理和后处理 SQL 内部生成 select 语句。
- 使用组件之间链接的数据映射功能，在数据源和数据接收器之间映射属性。
- 通过所使用组件的内置 Create Table From Port 命令，创建临时表或持久性 staging 表。
- 通过所使用组件的内置 create table from port 命令在目标数据库中创建表。
- 使用 Content Explorer 浏览所连接的全部数据源的模式信息和数据内容。
- 使用 XML 通过 SQL 数据提供程序组件读取分层 XML 文档并自动生成关系结构。
- 在 Sybase ETL Development 内安排项目的执行并创建作业。

此外，当应对复杂数据转换需求时，可以使用：

- 经过手工优化的 SQL select 语句，用于调整数据提取过程。
- SQL，用于在预处理和后处理命令内部应用数据操作命令。
- JavaScript，用于编写过程、执行复杂计算或在操作系统环境中操作对象。
- 表达式中的间接机制（中括号表示法），用于以动态方式分配值以使用环境或用户变量控制项目。

组件

逐记录分步调试组件

在模拟模式下，许多转换组件允许逐行操作当前的一组数据，并立即显示任何所应用转换的结果。

自适应端口结构和映射

项目内的所有数据流经称为 “输入端口”和 “输出端口”的组件端口。每个端口均具有数据流的结构。可以更改其结构不直接取决于组件配置的所有组件的端口结构。可以在组件内部立即引用添加到端口结构的属性。

连接组件时， Sybase ETL 将尝试在输出端口和输入端口之间创建标准映射。可以在 “Mapping”窗口中修改连接上的映射。若要打开 “Mapping”窗口，请右键单击连接链接，然后选择 “Mapping”。请参见第 27 页上的 “查看当前映射”。

存储库

存储库包含与 Sybase ETL 对象、项目和作业相关的所有数据和信息。

在会话过程中，可以并行方式访问多个存储库。可以在存储库之间复制和传输项目，从而将产品知识库与 development repository 分隔开来。请参见第 23 页上的 “管理项目”。

存储库通常属于一个客户端，例如部门或公司。多个客户端可以使用同一个存储库。每个客户端可以支持任意数量的客户端用户；每个用户均具有用于控制信息访问权限的用户名和口令。

注释 请勿在存储库表中手工操作数据。在存储库经过手动处理后，Sybase 无法保证其功能性。这也可能使存储库无法使用，并可能丢失您的工作。

数据类型和数据格式

在转换过程中将保留数据源的数据类型。

Sybase ETL 在内部区分字符串数据类型与数值数据类型。数据提供程序或数据接收器的 “标准化数据格式”选项会自动将数据转换为标准格式，然后从标准格式转换为目标数据库可以处理的格式。使用不同数据库时，您不必手工转换各种日期和数字格式。

缺省情况下，已选中 “标准化数据格式”选项。但是，如果在日期或数字字段中出现问题，可以在 “Preference”窗口中禁用此设置并手工转换数据。请参见第 18 页上的 “自定义首选项”。

SQL

由数据提供程序提供的大多数数据通过存储在 Query 属性中的 SQL 语句进行定义。 Sybase ETL 支持经过修改的 SQL92 标准。

可以手动将 SQL 语句从现有项目写入或复制到 Query 属性中。如果不想涉及 SQL92 的细节，可使用 Query Designer 以图形方式设计查询并自动生成 SQL 语句。

工具

可通过 Sybase ETL 工具访问所有已连接数据源中的结构和 catalog 信息。可以浏览模式信息或数据，甚至使用这些工具创建新的数据库对象。有关各种 Sybase ETL 工具的信息，请参见第 45 页上的 “高级概念和工具”。

对 Unicode 的支持

所有组件均可处理和支持 Unicode 和多字节数据。可以在计算、脚本和过程中使用启用 Unicode 的转换函数。 Sybase ETL 的 Unicode 支持级别允许提取、转换和装载：

- Unicode 字符
- 组件属性中的以下 Unicode 字符：
  - 文件名或目录名
  - 元数据，例如表或属性名称
  - 连接设置，例如数据库、模式、用户或口令
  - 转换规则
  - 中括号表示法 (SBN) 表达式

表达式

中括号表示法 (SBN) 是一种在 Sybase ETL 环境中广泛适用的间接机制。可以在表达式、SQL 语句和文件名规范中应用中括号表示法。使用中括号表示法可以在运行时以动态方式计算和分配值。

第 2 章

快速入门

启动 Sybase ETL

1 在 Windows 中，选择“开始”|“程序”|“Sybase”|“Sybase ETL Development 4.8”|“Sybase ETL Development”。

登录窗口显示：

- Connection – `Repository`
- Client – `transformer`
- Client user name – `TRANSFORMER`
- Password – `transformer`

这些值在第一次登录时自动设置。在后续登录时，可能需要选择或输入此信息。

单击“Logon”。

2 在“Navigator”中，单击“Repository”|“*TRANSFORMER.transformer.Repository*”|“Projects”打开可用项目的列表。

注释 项目列表显示与产品打包在一起的演示项目。每个演示项目包含如何使用组件或实现方案的示例。

3 双击现有项目名称打开项目，或右键单击“Projects”，然后选择“New”创建新的项目。

在 Demo Repository 中设置新的用户帐户

将新用户添加到 Demo Repository：

1 从 Sybase ETL Development 界面选择“File”|“Open Repository”。

2 在 “Client User”字段中输入新的客户端用户名。

注释 如果要访问演示项目和作业，请勿更改客户端名称。

3 输入口令。

4 选择 “Register new user”选项。

5 选择 “Show all objects”选项。如果不选择此选项，将无法访问演示项目和作业。

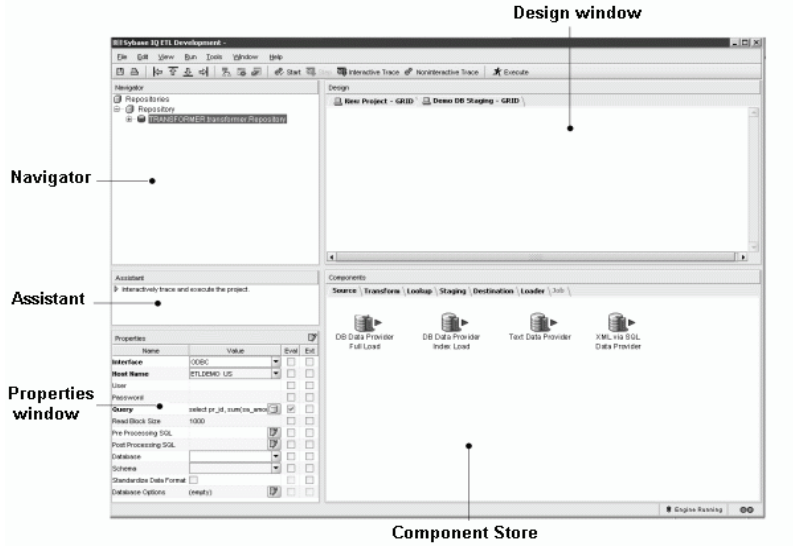
6 单击 “Logon”。

7 重新输入口令，然后单击 “OK”。

使用 Sybase ETL Development 界面

Sybase ETL Development 界面由以下内容组成：

- “Design”窗口 – 用于基于转换规则创建项目和作业。
- Component Store – 用于查找项目的组件。
- Navigator – 用于查找项目、作业和模板。还会显示最近访问的项目、作业和模板。
- “Assistant”窗口 – 用于协助完成当前任务。
- “Properties”窗口 – 用于设置组件的属性。



Navigator

使用 “Navigator”可执行以下操作：

- 管理存储库
- 导航和浏览存储库
- 管理项目和作业
- 执行项目和作业
- 管理用户帐户

管理存储库

Sybase ETL 存储库是用于保存和维护项目、作业和会话参数所有相关数据的表的集合。您可以使用 Sybase IQ、 Microsoft SQL Server、 Sybase Adaptive Server ® Anywhere (ASA) 和 Microsoft Access 数据库作为 Sybase ETL 存储库。

注释 请勿在存储库表中手工操作数据；这样做可能使存储库无法使用，并且还可能会丢失数据。在存储库经过手动处理后， Sybase 无法保证其功能性。

若要访问项目或作业，必须登录到相应的存储库。若要打开存储库，必须至少分配一个客户端和一个客户端用户。一个客户端可以有多个客户端用户。

❖ **打开存储库**

1 选择 “File” | “Open Repository”。或者，在 “Navigator”中右键单击 “Repositories”，然后选择 “Open Repository”。

2 从 “Connection”列表中选择存储库，然后单击 “Logon”。

❖ **关闭存储库连接**

1 在 “Navigator”中右键单击存储库名称，然后选择 “Close Connection”。

2 单击 “Yes”确认要关闭连接及所有打开的项目和作业。

关闭存储库将结束当前连接到该存储库的所有用户会话。

❖ **关闭客户端用户会话**

1 在 “Navigator”中右键单击存储库名称，然后选择 “Close Client”。

2 单击 “Yes”确认要关闭客户端及所有打开的项目和作业。

❖ **添加存储库**

1 选择 “File” | “Open Repository”，打开 “Repository Logon”窗口。

2 单击 “Add”。

3 输入新的存储库连接的参数，然后单击 “Save”。

若要访问新的存储库，必须至少创建一个客户端和一个客户端用户定义。

❖ **创建客户端和客户端用户**

1 在 “Repository Logon”窗口的 “Client”字段中，输入客户端的名称。

2 在 “Client User”字段中输入客户端用户名。名称必须为字母数字字符，最多可包含 255 个字符，并且不能以数字开头。

3 输入口令。

4 选择 “Register New User”选项。

5 如果客户端用户有权查看客户端中的所有现有项目，请选择 “Show All Objects”选项。

6　单击 “Logon”。

7　重新输入口令，然后单击 “OK”。

注释 也可以从 “Use Accounts”窗口创建用户。请参见第 14 页上的 “创建用户”。

❖ **编辑存储库**

1　选择 “File” | “Open Repository”，打开 “Repository Logon” 窗口。

2　选择要修改的存储库，然后单击 “Edit”。

3　作出更改，然后单击 “Save”。

❖ **删除存储库**

1　从 “Connection”列表中选择存储库，然后单击 “Remove”。

2　单击 “Yes”确认删除。

导航和浏览存储库

在 “Navigator”中，分层树列表表示：

- 打开的存储库。
- 与打开的存储库的客户端用户会话。
- 存储在存储库中的对象，如项目、作业和模板。
- 最近打开的项目、作业和模板。

存储库可以由多个客户端用户会话同时打开。客户端用户是客户端的一部分。客户端用户和客户端均在登录到存储库时进行注册。

以下示例显示树结构：

```
Repositories
-- <RepositoryName1>
---- <ClientUser1>.<Client1>.<Repository Name1>
------ Recent
-------- Recently opened projects
-------- Recently opened jobs
-------- Recently opened templates
------ Projects
-------- Project_1
-------- Project_2
-------- Project_N
------ Jobs
-------- Job_1
-------- Job_2
-------- Job_M
------ Templates
-------- Template_1
-------- Template_L
---- <ClientUser1>.<ClientM>.<Repository Name1>
---- <ClientUserN>.<Client1>.<Repository Name1>
-- <RepositoryName2>
```

管理项目和作业

从 “Navigator”可以管理项目和作业。请参见第 3 章 “项目和作业”。

管理用户帐户

从浏览器可以执行以下操作：

- 创建用户。
- 删除用户。
- 更改口令。
- 更改可见性。

只有注册客户端用户才可以访问存储库。可以在 “Repository Logon”窗口或 “User Accounts”窗口中注册客户端用户。

❖ **创建用户**

1 在 “Navigator”中，右键单击存储库名称并选择 “User Accounts”。

2 单击 “Add User”。

3　输入用户名。用户名必须满足以下条件：
- 仅包含字母数字字符。
- 以字母字符开头。
- 最多包含 255 个字符。
- 不能为空。

4　输入口令。

5　重新输入口令。

6　选择 “Show All Objects”选项以显示属于其他存储库用户的对象。

7　单击 “OK”。

❖ **删除用户**

1　在 “Navigator”中，右键单击存储库名称，然后选择 “User Accounts”。

2　选择要删除的用户，然后单击 “Remove User”。

如果所选的用户当前连接到存储库，系统将提示您确认是否要删除用户并关闭所有打开的项目和作业。单击 “Yes”。

3　输入要删除的用户的口令，然后单击 “OK”。

❖ **更改口令**

1　在 “Navigator”中，右键单击存储库名称，然后选择 “User Accounts”。

2　选择要为其更改口令的用户。

3　单击 “Change Password”。

4　输入用户的现有口令和新口令。重新输入新口令。

5　单击 “OK”。

“Properties”窗口

使用 “Properties”窗口可以执行以下操作：
- 查看和编辑组件属性。
- 标识强制组件属性。
- 允许对组件属性进行动态评估。
- 加密组件属性。
- 将自定义属性添加到组件并编辑其值。
- 访问组件配置窗口。

有关特定于组件的属性设置，请参见第 5 章 “组件”。

查看和编辑组件属性

若要查看和编辑组件的属性和值，请在 “Design”窗口中选择组件。所选组件的所有属性将显示在 “Properties”窗口中。

标识必要属性

“Properties”窗口中以**粗体**文本显示的属性名称表示该属性为组件正常工作所必需。所有其它属性均为可选属性，可用于调优和配置组件。

允许动态表达式

选择 “Evaluate”选项可以允许对动态间接表达式 （SBN 表达式）进行评估。请在相应字段中使用中括号表示法 [ ] 输入 SBN 表达式。通过 “Evaluate”选项可以在运行时计算和评估动态属性设置，而不是在设计时分配静态值。

对于某些属性项，缺省情况下已选中 “Evaluate”选项。

❖ **启用或禁用属性的评估**

1 在 “Properties”窗口中，右键单击您要在运行时评估的属性名称。

2 选择 “Evaluate”。

加密属性

项目、作业数据及属性值均存储在 Sybase ETL 存储库中。 Sybase ETL 存储库中的大多数记录未加密，而是以一种可读格式表示。对于口令属性，缺省情况下已选中 “Encrypt”选项。

❖ **加密属性值**

1 在 “Properties”窗口中右键单击属性名称。

2 单击 “Encrypt”。

或者，选择属性值旁的 “Encrypt”复选框。

添加和编辑自定义属性

使用 “Properties”窗口可以添加或编辑自定义组件属性。与其它属性一样，自定义属性也包含可在表达式或用户定义的过程中引用的变量。

请参见第 77 页上的 “自定义属性”。

访问组件的 “Configuration”窗口

在 “Properties”窗口中，单击 “Property”图标打开所选组件的配置窗口。

注释 某些组件没有配置窗口。

“Design”窗口

使用 “Design”窗口可以执行以下操作：

- 创建和修改项目及作业。请参见第 23 页上的 “项目和作业”。
- 模拟和执行项目。请参见第 25 页上的 “模拟项目”和第 32 页上的 “执行项目”。
- 执行作业。请参见第 32 页上的 “管理作业”。

❖ **将组件添加到 “Design”窗口**

若要创建项目或作业，必须添加和连接组件并设置其属性。可以通过以下任一方式将组件添加到 “Design”窗口：

1. 在 “Component Store”中选择组件并将其拖动到 “Design”窗口。请使用此方法将组件添加到项目及作业中。其它方法可用于仅将组件添加到项目中。
2. 在 “Component Store”中双击组件。
3. 在 “Component Store”中，右键单击组件，然后选择 “Add”。
4. 在 “Design”窗口中右键单击现有组件的端口，然后选择 “Add Component”。指向组件类型，然后选择要添加的组件。该组件在现有组件之前还是之后添加取决于所选端口是输入端口还是输出端口。

❖ **从 “Design”窗口删除组件**

1. 在 “Design”窗口中，选择要删除的组件。
2. 右键单击并选择 “Delete”。

❖ **处理常规命令**

1. 在 “Design”窗口中，右键单击任何位置以打开常规项目菜单。此菜单显示常规命令，例如 “Close”和 “Print”。右键单击组件打开 “Component”弹出菜单。 “Component”菜单显示特定于组件的命令。
2. 选择要执行的操作。

Component Store

“Component Store”由多个部分组成，各部分按常规目的对组件进行分组。

将组件从 “Component Store”添加到项目：

- 将组件拖动到 “Design”窗口。
- 右键单击组件，然后选择 “Add”。
- 双击组件。

自定义首选项

在 Sybase ETL Development 中使用 “Preferences”窗口可以自定义以下各组的设置：

- Workbench
  - Appearance
  - Data Viewer
  - Query Designer
- 引擎
- Performance Logging

❖ **自定义首选项**

1 从 Sybase ETL Development 主窗口中选择 “File”|“Preferences”。

2 在 “Preferences”窗口上，选择 “Appearance”并设置下列选项：

- Locale for user interface display – 选择环境的区域设置语言。可以选择 _de （德语）、 _en_US （美国英语），或者 _en_GB （英国英语）。缺省值为 _en_US。
- Show assistant for creating projects – 选择此选项可查看指导您完成项目的助手，并显示有关打开项目的当前状态的信息。
- Default font for displaying text – 选择用于在“Text Data Provider”和 “Text Data Sink”组件窗口中显示文本文件内容的字体。在使用非西欧字符集 （如 UNICODE）时，此设置十分有用。缺省字体为 “Monospaced”。用于显示文本的建议字体为 “Dialog”或 “Monospaced”。

- Default font for displaying data – 选择用于在整个应用程序中显示端口数据的字体。在使用非西欧字符集 （如 UNICODE）时，此设置十分有用。如果是第一次安装 Sybase ETL Development，缺省字体为 “Dialog”。如果先前已安装 Sybase ETL Development，则字体将设置为先前定义的值。Sybase 建议您将字体设置为 “Dialog”。

  注释 为使用 “Default font for displaying text”和 “Default font for displaying data”首选项，Sybase 建议您安装针对东亚语言的文件。在 Windows “控制面板”中，单击 “区域和语言选项”，选择 “语言”选项卡，然后选择 “为东亚语言安装文件”选项。启用此选项将安装显示 Unicode 字符所需的字体。

- Create a new project on startup – 选择此选项可在启动 Sybase ETL 时自动创建新的项目。
- Create links automatically when components are added – 选择此选项可在现有组件和添加到项目的新组件之间自动创建链接。
- Default action on double-clicking a connection – 指定在模拟过程中双击连接时是否打开 “Mapping”窗口或 “Preview”窗口。“Mapping”窗口显示映射信息，“Preview”窗口提供连接数据的预览。缺省值为 “Mapping”。
- Display qualified transformation objects – 选择此选项可在 “Navigator”中的对象名称前加上所有者名称作为前缀。例如，如果选择了此选项，则项目名称将以如下形式出现在 “Navigator”中：

  ```
  TRANSFORMER.Demo Character Mapper
  ```

  其中， TRANSFORMER 是创建项目的客户端用户的名称。
- Use unique object names – 选择此选项可对存储库连接强制唯一的项目和作业名称。
- Show password in component tooltips – 如果希望组件工具提示显示用于登录到基础数据库的口令，可选择此选项。缺省情况下，口令在工具提示中显示为一系列星号。
- Use enhanced color accessibility – 选择此选项可更改组件端口的色彩以加强对色盲用户的支持。选择此选项会将端口的缺省色彩从黄色更改为蓝色，从而使色盲用户可以区分不同的端口状态。
- Use vertical component layout – 选择此选项可从上到下以垂直对齐方式显示项目和作业，而不是缺省情况下的从左到右垂直对齐。
- Show information dialogs – 如果希望执行操作且不被信息提示中断，请取消选择此选项。

- Number of recently opened projects, jobs, and templates to display – 指定要在 “Navigator”和 “File”菜单中查看的最近访问的项目、作业和模板的数目。
- Open the last repository automatically – 选择此选项可在重新启动时自动连接到上次登录的存储库。
- Save repository client password – 如果您希望在登录到存储库之后保存客户端口令，请选择此选项。如果选择此选项，您不必在下次登录时提供口令，该口令将在 “Repository”窗口的 “Password”字段中自动提供。

3 选择 “Data Viewer”并指定导出数据字段的最大长度。缺省值为 255。长度超过在此指定的值的数据字段将在导出文件中截断。

4 选择 “Query Designer”并设置以下选项：

- Enable delete functionality of database objects – 选择此选项可在 “Query Designer”中右键单击表并选择 “Truncate Object”时，删除所选表的所有记录。
- Default number of records to retrieve from the Query Designer – 指定将由 Query Designer 检索的缺省数据记录数。缺省值为 25。
- Create joins automatically – 选择此选项可在表或视图内使用的相同属性名称之间自动创建连接。
- Use brackets when creating joins – 选择此选项可在生成的查询中用括号自动将连接子句括起。例如， select 语句将以以下形式显示：

```
select * FROM SALES ((<join statement 1>)
<join statement 2>)
```

- Default number of recently accessed tables and views to display – 指定希望 Query Designer 在 “Recent”选项卡中显示的最近访问的表和视图的数量。缺省值为 25。

5 选择 “Engine”并设置以下选项：

- Start local engine during application startup – 选择此选项可在启动 Sybase ETL 时启动本地引擎。
- Interval between engine monitor updates (sec) – 指定在 Engine Monitor 的两次更新之间等待的秒数。缺省值为 5 秒。
- Rate of simulation (msec) – 通过设置 “Simulation trace delay”控制模拟速率。 “Simulation trace delay”选项接受 10 到 9999 毫秒之间的值。缺省值为 250 毫秒。
- Grid engine ping timeout (sec) – 指定在启动或重新启动本地网格引擎之前允许访问网格引擎的秒数。缺省值为 60 秒。

- Interval between progress monitor updates (sec) – 指定在 Progress Monitor 更新之间等待作业执行的秒数。缺省值为 5 秒。
- Allow selection of the execution engine – 如果希望能够指定将用于项目执行的网格引擎，请选择此选项。
- Execution engine server – 指定主网格引擎服务器的 IP 地址。
- Execution engine port – 指定主网格引擎服务器的端口地址。

6 选择 “Performance Logging”，并指定记录性能数据的详细信息级别：
- 0 – 性能数据不会写入到存储库。
- 1 – 性能数据写入到存储库。该值为缺省值。

7 单击 “Save”。为使某些更改生效，系统可能会提示您重新启动 Sybase ETL。如果不重新启动，更改将在下一次启动 Sybase ETL 时生效。

故障排除

在安装过程中，Sybase ETL 安装程序将创建一组初始数据源。如果这些存储库数据源由于任何原因而丢失，则必须将其恢复，否则 Sybase ETL 无法打开。恢复 Demo Repository 的初始 ODBC 数据源组：

1 配置 ODBC 用户数据源。
   a 选择 “开始” | “设置” | “控制面板” | “管理工具” | “数据源 (ODBC)”。
   b 单击 “添加”。
   c 从列表中选择 “Microsoft Access Driver”。单击 “完成”。
   d 在 “数据源名”字段中，输入 DEMO_Repository。
   e 从安装目录的 *Demodata* 文件夹中选择 *IQETLDEMO_REP.MDB* 数据库。单击 “确定”。

2 在 “Repository Logon”窗口中设置存储库连接：
   - 选择 “File” | “Open Repository”，打开 “Repository Logon” 窗口。
   - 从 “Connection”列表中选择 “Repository”并选择以下其中一项：
     - Edit
     - Add and enter a name for the connection

- 从 “Interface” 列表选择 “ODBC”。
- 在 “Host” 列表中选择 “DEMO_Repository”。
- 单击 “Save”。

3 配置 Demo Repository 中的项目所需的其它 ODBC 用户数据源：

- 驱动程序 – Microsoft Access
- 名称 – ETLDEMO_DWH ； 数据库 – DEMO_DWH.MDB
- 名称 – ETLDEMO_GER ； 数据库 – DEMO_GER.MDB
- 名称 – ETLDEMO_US ； 数据库 – DEMO_US.MDB

  这些用户数据源的数据库文件也位于安装目录的 *Demodata* 文件夹中。

第 3 章

项目和作业

管理项目

项目由组件和链接组成，通过其端口连接组件。与项目有关的基本操作包括创建、删除、重命名和保存等操作，复杂操作包括模拟等操作。

Sybase ETL 项目以一个或多个源组件开始并以一个或多个目标组件结束。

以下是 Sybase ETL 组件：

- “数据提供程序”组件通常与 “转换”组件、“处理”组件或“数据接收器”组件连接。
- “转换”组件和 “处理”组件具有输入与输出端口，并具有任何其它类型的相邻组件。
- 如果 “转换”组件允许通过多个输入数据流，则需要多个始发源组件。
- 如果 “转换”组件具有多个数据流输出，您可将每个数据流与组件连接。

❖ **创建项目**

1 在 “Navigator”中，右键单击项目并选择 “New” | “Project”。也可从 “File”菜单中选择 “New” | “Project”。

2 将项目组件从 “Component Store”拖至 “Design”窗口。

❖ **修改项目**

1 在 “Navigator”中，双击要修改的项目。

2 进行更改并保存项目。

❖ **为项目解锁**

另一用户客户端锁定的项目将以只读模式打开。

- 若要使项目能够进行读取或写入，请在打开锁定的项目时显示的窗口中单击 “Unlock”和 “Open”。

❖ **复制项目**

1 双击要复制的项目以在 “Design”窗口中打开该项目。

2 在 “Navigator”中，右键单击该项目并选择 “Save As”。或者，在 “File”菜单中选择 “Save As”。

如果使用多个存储库，请选择目标存储库。

3 为新项目输入名称。现有项目的副本已创建，原始项目会保持不变，而且不会存储对原始项目的引用。

❖ **传输项目**

如果您正使用多个存储库，则可将完整的项目从一个存储库复制至另一个存储库且保持对原始项目的引用。例如，您可能希望将项目从 development repository 移至测试或产品知识库。通过存储对原始项目的引用，传输将在原始项目下次启动时识别该项目并选择性地替换与传入对象相关的一切。

1 在 “Navigator”中，右键单击要传输的项目并选择 “Transfer”。或者，从 “File”菜单中选择 “Transfer”。

2 选择希望传输其中项目的存储库。

注释 传输将复制项目定义和执行属性。相关数据 （如参数集）不会进行传输。

❖ **删除项目**

1 在 “Navigator”中，右键单击项目并选择 “Delete”。

2 单击 “Yes”确认删除。

❖ **重命名项目**

1 在 “Navigator”中，右键单击项目并选择 “Rename”。

2 输入项目的新名称并单击 “OK”。

❖ **重置执行属性**

重置增量装载的装载选项：

1　在 “Navigator”中，右键单击项目并选择 “Reset Execution Properties”。

2　单击 “Yes”以确认您希望重置执行属性。Load Index Value（DB Index Load 组件）的当前值已重置。

模拟项目

对项目进行模拟可逐步监控并验证转换过程。与执行项目不同，利用模拟操作可以：

- 运行带有未保存更改的项目。
- 在转换过程的任何阶段查看数据。

在模拟的最后几步中，数据会写入数据接收器中。使用 Data Calculator 等多种转换组件可在模拟期间更改转换规则和示例值，并且可验证所有潜在内容的规则库。

您可以采用以下方式模拟项目：

- 交互式 – 选择 “Run” | “Start”初始化要模拟的项目，然后选择 “Run” | “Step”手动逐步执行组件以模拟项目执行。

  或者，选择 “Run” | “Interactive Trace”以初始化要模拟的项目并采用预定义的速度自动逐步执行组件。
- 非交互式 – 选择 “Run” | “NonInteractive Trace”以非交互模式模拟项目。请参见第 27 页上的 “非交互式模拟项目”。

注释 只有正确初始化所有组件后，才可模拟项目。

模拟的基本功能包括以下高级步骤：

- 以交互式或非交互式启动模拟
- 逐步执行组件
- 查看连接链接上或组件中的数据流
- 修改并重新初始化组件以继续模拟数据流

在模拟操作时，在更详细的级别上，您可以：

- 查看连接链接上的数据内容
- 查看组件内的输入数据与输出数据
- 修改属性或计算，以便更改转换规则和示例值以验证规则库

- 修改计算或属性后再次逐步执行组件
- 执行 “假设”情况
- 采用多步执行项目

交互式模拟项目

若要交互式运行整个项目，选择 “Run” | “Interactive Trace”。如果选择交互式模拟项目，您可在任意时刻停止模拟，并选择 “Run” | “Step”或 “Run” | “Step Through”来手动逐步执行其余项目组件。

❖ **交互式模拟项目**

1 若要启动模拟，单击工具栏上的 “Start”。单击 “Start”时：
   - 将初始化项目的所有组件。
   - 将验证项目中的所有连接。
   - 将执行项目中的所有预编译的 SQL 语句。
   - 检索并高速缓存所有静态查找组件的数据。查询表中数据在模拟期间的任何更改不会在模拟过程中反映出来。

2 选择某个组件并单击工具栏上的 “Step”图标以执行该组件。

   逐步执行组件意味着执行或处理单个组件。在一个步骤中处理的数据记录是当前填充该组件输入端口的记录。

   如果多次逐步执行组件且期间没有执行其它组件，则已收到或已转发的记录数保持不变。可在项目视图中从组件内部以及外部逐步执行多个组件。

3 查看连接链接上或组件中的数据流。
   - 若要在整个转换过程中查看数据，请检查组件或组件端口之间的链接。Data Calculator 和 Data Splitter 等其它组件包含内置预览功能。
   - 若要查看连接链接上的数据，请右键单击并选择 “Preview”。
   - 若要查看当前端口的数据，请右键单击该端口并选择 “Preview”。

     注释 如果没有已处理的记录或没有可用的模拟数据，将禁用 “Preview”选项。

   - 若要从组件内部查看数据，双击该组件或单击 “Property”窗口中的 “Rule”图标。使用此选项，从 Data Calculator 或 Data Splitter 等组件内部查看转换规则的影响。

4　修改并初始化组件。

修改某个组件后，可重新初始化该组件以继续模拟数据流。您无需重新启动对当前项目的完整模拟，

a　双击该组件以便在 “Properties”窗口中修改其属性。

b　保存更改。

c　右键单击组件，然后选择 “Initialize”。

注释　“交互式跟踪”不执行后处理任务，例如在 DB 组件上执行后 SQL 语句。

❖ **设置跟踪延迟**

通过设置模拟交互式跟踪延迟选项可控制模拟速率。

1　选择 “File” | “Preferences” | “Engine”。

2　修改模拟字段的 “Rate”的值。可输入介于 10 至 9999 毫秒之间的值。缺省值是 250 毫秒。

非交互式模拟项目

若要无需交互地模拟项目，选择 “Run” | “Noninteractive Trace”。“非交互式跟踪”逐步执行项目并在执行结束时更新连接记录数量。

如果尚未启动模拟，选择此选项将自动启动模拟，并且模拟会继续进行直至处理了所有数据。

与交互式跟踪不同，如果使用非交互式跟踪来模拟项目，则无法随时停止模拟。

查看当前映射

“Mapping Definition”窗口显示相邻输入结构与输出结构的属性之间的当前映射。

查看当前映射：

1　右键单击连接链接，然后选择 “Mapping”。

2　在 “Mapping Definition”窗口中：

- 选择 “Display structure”以查看已连接端口及其当前映射的所有属性。
- 选择 “Display structure and values”以查看当前记录的字段和值。该视图显示连接至该链接的端口的当前内容。如果该端口未含任何数据，则此窗口中仅显示该端口的结构。通过逐步执行项目直至达到该端口，便可在端口中填充数据。

应用自动映射

若要创建映射，从 “Mapping”菜单上选择以下预定义映射序列之一：

- Create mapping by Order – 按顺序映射
  输入结构和输出结构的端口属性。如果两端的属性数目不同，则说明部分端口属性未映射。
- Create mapping by Name – 按名称映射输入和输出结构的端口属性。
- Create mapping by Name Case Sensitive – 按照其区分大小写的名称映射输入结构和输出结构的端口属性。
- Create mapping by Prefix – 按名称映射输入结构和输出结构的端口属性，忽略指定的前缀。
- Create mapping by Best Match – 映射类似的输入结构和输出结构的端口属性。

应用手动映射

若要手动创建单个映射，请选择连接点并将其拖至端口属性的连接点。

若要更改当前映射，请选择连接点上的映射行并将其拖至未映射的端口属性。

若要删除单个映射，请右键单击该映射并选择 “Delete”。

若要删除链接的所有映射，请从 “Mapping”菜单上选择 “Remove All”。

查看已映射的属性

缺省情况下，“Mapping Definition”窗口显示输入结构和输出结构的所有端口属性。若仅查看已映射的属性，请单击工具栏上 “Display only mapped attributes”图标。

管理端口属性

您可在 “Structure Viewer”窗口中添加并删除端口属性，或者修改设置或现有属性。

❖ **将属性添加至端口结构**

1 在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2 在 “Structure Viewer”窗口中选择 “Actions” | “Add”，或右键单击属性并选择 “Add”。

3 输入属性的名称。端口属性的名称必须以字母字符开头，且只能包含字母数字字符。它不能是保留的 JavaScript 关键字。请参见第 53 页上的 “变量”。

选择 “Populate Attribute”选项将属性添加到多个端口结构。此时会将新属性添加到参与所选连接的所有端口结构并自动进行映射。单击 “Ok”。

4　指定其它详细信息。

❖ **从端口结构中删除属性**

1　在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2　在 “Structure Viewer”窗口中选择 “Actions” | “Remove”，或右键单击属性并选择 “Remove”。

❖ ***修改端口属性***

1　在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2　在 “Structure Viewer”窗口中更改属性设置，然后单击 “Save”。

有关详细信息，请参见第 78 页上的 “管理端口结构”。

修改数据类型

当修改记录结构的数据类型时，您可修改转换期间 Sybase ETL 用于记录结构的内部逻辑表示。此操作不会更改源表或目标表的数据结构定义。确保最终 “数据接收器”的数据结构与正在生成的内容兼容。

查看模拟流

启动模拟后，以下方式可使模拟流可见：

- 绿色虚线框，指示活动组件并且随着每一步都会从一个组件移至下一个组件。
- 链接上显示记录的数目，它会随该框移动。

每一步中正在处理的记录数取决于带 “Read Block Size”属性的上一组件的 “Read Block Size”的当前值。

当执行模拟时，选择一个较小的数字比较合适。执行项目时，大数值的 “Read Block Size”可显著改善性能。

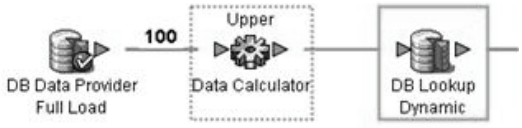
逐步执行当前组件选定组件

启动模拟时，要执行的第一个组件会用绿色虚线框进行指示。

如果逐步执行项目而未进行修改，则该框会在组件间移动，并会显示成功或失败图标直至模拟结束。

您可以选择除当前组件外的其它组件以检查或更改其属性。选定的组件由绿色实线框进行指示。

图 3-1： 当前组件与选定组件



当单击工具栏上的 “Step”或选择 “Run” | “Step”时，下一步将执行当前组件。模拟期间，若要检查或更改与当前组件不同的组件，请单击该组件。绿色实线框会突出显示选定组件。

完成更改后，可从选定组件或当前组件恢复模拟：

- 若要从选定组件恢复模拟，右键单击并选择 “Step”。
- 若要从当前组件恢复模拟，单击工具栏上的 “Step”。

注释 如果右键单击未处理的组件并选择 “Initialize”和 “Step”，将会初始化和逐步执行该组件，并且会突出显示将要执行的下一组件。

前移和后移组件

由框指示的模拟视觉流在大多数项目中是向前移动的；框从一个组件移至下一个组件。然而，项目模拟流并不只朝一个方向移动；流的方向由项目中所使用的组件决定。

转移 Data Calculator 与 Character Mapper 等组件，接收多个记录，将转换应用于这些记录，并转移记录。单个步骤中处理的记录数仅由上一组件中接收到的记录数决定。

其它组件将覆盖上一 “Read Block Size”设置。Staging 组件旨在处理数据流的整个结果集，如 “数据源”组件的查询所定义。将整个结果集传递至输入端口之前，组件将不处理和转移任何数据记录。Staging 组件使用自己的 “Read Block Size”属性重新调整下一步将转移的记录数。有关模拟期间每一组件的行为的详细说明，请参见第 5 章 “组件”。

从多个位置预览数据

右键单击任何连接链接、端口或组件，并选择 “Preview”以打开 “Content Browser”窗口，该窗口会显示当前在选定位置上可用的数据。

注释 如果没有已处理的记录或没有可用的模拟数据，将禁用 “Preview”选项。

“Content Browser”窗口包括允许您同时从多个位置显示多个预览的选项卡。有时，在并列的选项卡中预览组件的输入端口和输出端口的内容将非常有用。

若要将显示的数据保存到定义文件，请单击工具栏上的 “Export”数据图标。为导出数据指定选项。

模拟期间部分执行或初始化

在对单个组件进行修改后重新启动整个模拟可能非常耗时，特别是如果某个含有大量的组件的项目中包含大量输入记录时更是如此。如果您仅对在复杂模拟流中间的某个位置模拟组件感兴趣，单步执行大型项目同样会令人沮丧。对于多步执行项目，在 “Run”菜单中选择组件并选择 “Step through”或 “Start through”以达到您的兴趣点。

模拟到某一组件

若要从项目中间的某一位置的组件开始验证当前项目，请选择该组件，然后选择 “Run” | “Start Through”。模拟会启动当前项目、处理当前组件与选定组件之间的所有组件并处理选定组件。

“Read/Write Block Size”的影响

作为 “Read Block Size”输入的数字可定义单个模拟步骤中组件所提取的记录数。设置 “Write Block Size”以定义要写入的记录数。大部分 “数据提供程序”组件具有 “Read Block Size”属性。大多数 “数据接收器”组件可自定义 “Write Block Size”。Staging 组件等 “转换”组件可自定义读取和写入的值。

注释　“Block Size”属性是在项目模拟以及项目和作业执行期间进行评估的。较小的数字可能非常适合模拟时使用，但当您单击 “Run” | “Execute”时会降低执行速度。在模拟中，“Block Size”被限制为 32K。

控制多个数据流

虽然绝大多数项目包含通过链接来连接的组件的单个流，您仍可设置拥有多个未连接数据流的单个项目。由于 Sybase ETL 是并行系统，因此您无法预测处理流的顺序。

如果您使用多个数据流， Sybase 建议您为每个数据流设计一个项目，以便项目中的所有组件相互连接。利用此设计指南，您可以通过连接项目来控制数据流以形成作业处理流。

执行项目

使用以下这些方法之一，可在缺省网格引擎中执行项目：

- 选择 “Run” | “Execute” 或单击工具栏上的 “Execute” 图标。“Design” 窗口中当前打开的项目将会执行。

  执行会影响模拟的状态。如果尝试执行未保存的项目，系统将提示您保存该项目。如果保存项目，该项目的所有模拟数据将丢失并且该项目将会执行。

  注释 执行前，您必须保存项目中的更改，因为可从存储库中读取该项目定义。如果尚未保存更改，执行将不会开始。

- 在 “Navigator” 中，选择要执行的项目。右键单击并选择 “Execute Project”。选定的项目将会执行。

“Execution Monitor” 将会显示。请参见第 68 页上的 “Execution Monitor”。

安排项目

若要安排项目，请选择 “Tools” | “Runtime Manager”。使用 “Runtime Manager” 创建、编辑、删除、执行和终止任务。

请参见第 50 页上的 “管理作业和预定任务”。

管理作业

使用作业为一个或多个项目设置功能强大的控制流。您可以安排运行 Sybase ETL 作业，而无需任何用户交互。

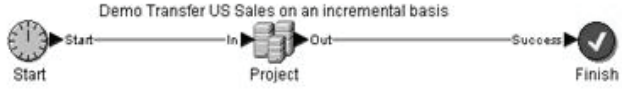
根据作业中项目的成功或失败，您可以控制作业执行。

作业组件

执行单个项目的作业至少包括：

- “Start” 组件
- “Project” 组件
- “Finish” 组件

图 3-2：含最少组件的作业



您可以扩展作业以包含多个：

- 按顺序或并行顺序排列的项目
- 同步程序
- “Finish”和 “Error”组件

“Start”组件后始终跟着一个或多个 “Project”组件。

图 3-3：含多个组件的作业

❖ **创建作业**

1　在 “Navigator”中，右键单击 “Jobs”并选择 “New” | “Job”。可用的作业组件会显示在 “Component Store”中。

2　将 “Start”组件从 “Component Store”添加至 “Design”窗口。

3　添加 “Project”组件并将其与 “Start”组件连接。

4　添加 “Finish”组件并将其与 “Project”组件连接。

5　双击 “Project”组件。

6　选择希望包含在此作业中的项目。单击 “Save”。

现在作业就可以在 Sybase ETL Development 中执行或作为预定任务执行。

从 “Navigator”中，您可以显示和访问作业中包含的项目。

❖ **修改作业**

1　在 “Navigator”中，双击作业名称，或右键单击该作业并选择 “Open”。

2　修改作业并保存更改。

❖ **复制作业**

1 双击要复制的作业以在 “Design”窗口中打开。

2 在 “Navigator”中，右键单击作业并选择 “Save As”。或者，在 “File”菜单中选择 “Save As”。

如果使用多个存储库，请选择目标存储库。

3 提供该作业的名称。现有作业的副本已创建，原始作业保持不变，而且不会存储对原始作业的引用。

注释 将作业复制至其它存储库并不会复制该作业中包含的项目。您必须为作业中的所有项目和多项目组件从新存储库中选择项目。

❖ **传输作业**

如果您正使用多个存储库，则可将所包含的完整作业和项目从一个存储库复制至另一个存储库且保持对原始对象的引用。例如，您可能希望将作业从 development repository 移至测试或产品知识库。通过存储对原始作业的引用，传输将在原始作业下次启动时识别该项目并选择性地替换与传入对象相关的一切。

1 在 “Navigator”中，右键单击要传输的作业并选择 “Transfer”。或者，从 “File”菜单中选择 “Transfer”。

2 选择要将作业传输到的存储库。

将作业和所有包含的项目从一个存储库复制至另一个存储库，但引用原始对象。

注释 传输仅复制作业定义。相关数据 （如参数集或执行属性）不会进行传输。

❖ **删除作业**

1 在 “Navigator”中，右键单击作业并选择 “Delete”。

2 单击 “Delete”。缺省情况下，仅删除选定的作业。若要删除作业和所有包含的项目，请选择 “Delete Included Projects”选项。

注释 在删除作业中使用的项目前，请确保该项目未用于其它作业中。这一检查不会自动进行。当前打开以供设计使用的项目和被任何用户锁定的项目将不受影响。

❖ **重命名作业**

1 在 “Navigator”中，右键单击作业。

2 选择 “Rename”。

控制作业执行

可通过以下操作控制作业执行：

- 使用 synchronizer 组件，您可根据项目的成功或失败转移作业执行。
- 忽略每个项目的错误。

请参见第 158 页上的 “作业组件”。

执行作业

您可直接从 Sybase ETL Development 或以特定时间间隔将作业作为操作系统 “任务管理器” 的预定任务来执行。

- 若要执行 “Design” 窗口中当前打开的作业，请选择 “Run” | “Execute”。
- 若要直接从 “Navigator” 执行作业，请右键单击该作业并选择 “Job Execute”。
- 若要安排作业，选择 “Tools” | “Runtime Manager”。请参见第 50 页上的 “管理作业和预定任务”。

安排作业

若要安排作业，选择 “Tools” | “Runtime Manager”。使用 “Runtime Manager” 创建、编辑、删除、执行并终止任务。由于 “Runtime Manager” 基于 Windows 任务调度管理器，您可找到当前在系统中定义的所有预定 ETL 任务。

请参见第 50 页上的 “管理作业和预定任务”。

使用模板创建项目和作业

使用模板自动创建项目和作业。

使用模板助手构建迁移模板

利用模板助手可创建新模板，或使用现有模板将数据从一个数据库迁移至另一个数据库。

❖ **构建迁移模板**

1 选择 “File” | “New” | “Template”。或者，右键单击 “Navigator”中的 “Templates”并选择此选项。

2 输入迁移详细信息：

- 提供该模板的名称。该名称用于模板对象，并进一步限定用于生成的转换对象。
- 指定迁移类型。

  缺省情况下，可用的 “迁移类型”是 *DB to IQ*。
- 选择 “Allow execution on multiple engines”选项以将多个引擎用于执行。
- 选择 “Use IQ Multiplex”选项，以通过使用多个写入器将数据装载到 IQ 来支持 Multiplex 执行。如果要将多个表迁移到 IQ 数据库，请选择该选项。

  注释 若要支持 Multiplex 执行，必须将 Adaptive Server ® Anywhere (ASA) 11 ODBC 驱动程序安装在 ETL Development 和 ETL Server 所在的同一计算机上。
- 选择 “Use IQ Client Side Load”选项将远程主机上的文件中的数据批量装载到 IQ 数据库。
- 选择 “Use IQ Lock Table”选项可以以排它模式锁定目标表，并防止并发事务更新该表。如果选择该选项，其它任何事务均不能对被锁定的表执行查询或任何更新。 “Use IQ Lock Table”选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。

  如果选择该选项，还必须指定项目在获取锁之前应等待的最大阻塞时间。
- 单击 “Next”。

3 输入源数据库的连接详细信息并选择要传输的表。有关要指定的数据库连接参数的信息，请参见第 80 页上的 “数据库连接设置”。

注释 数据库连接属性与 DB 组件的连接属性相同。

单击 “Logon”可查看指定数据库的可用表的列表。缺省情况下，已选择所有表进行传输。取消选择不希望传输的表。您还可选择一个或多个表的行，右键单击并选择 “Exclude”。要将表纳入传输范围，请右键单击并选择 “Transfer”。

还可以单击：

- Exclude all objects from transfer icon to exclude all tables。
- Include all objects in transfer icon to include all tables。

若要查看有关表的其它信息，选择表的行，右键单击并选择：

- Browse – 查看表数据。
- Count – 查看选定表的记录数。若要查看所有表的记录数，请单击 “Count All”。

单击 “Next”。

4 输入目标数据库的数据库连接属性。有关必须指定的数据库连接参数的详细信息，请参见第 80 页上的 “数据库连接设置”。

单击 “Logon”可查看可用表的列表。若要查看表数据或选定表的记录数，请右键单击 “Browse”或 “Count”。若要查看所有表的记录数，请单击 “Count All”。

单击 “Next”。

5 指定要传输的表的传输设置。

a 选择 “Preserve schema/owner”选项以保留源表的模式或所有者信息。

注释 目标数据库中必须存在同一模式或所有者。

b 输入阶段属性。

在 “Stage”和 “Stage Server”字段中，指定 “DB Bulk Load IQ”组件的装载阶段属性的路径。如果选择了 “Use Pipes”选项，则将自动设置路径。如果未选择 “Use Pipes”，则应手动提供以路径分隔符结尾的值。例如， *C:\ETLStage*。

有关这些属性的详细说明，请参见第 145 页上的 “DB Bulk Load Sybase IQ 属性列表”。

注释 如果在第 2 步中在迁移详细信息窗口中选择了 “Use IQ Client Side Load”选项，“Use Pipes”选项和 “Stage server”字段则处于禁用状态。

c 选择源属性。

缺省情况下，会选择表的所有属性以供传输。若要更改属性选择，单击 “Columns”字段中的图标。

在 “Select Attribute”窗口中，取消选择要从传输中排除的属性。您还可选择一个或多个属性行，右键单击并选择 “Exclude”以取消选择该属性。

d 选择目标表。

假定源表的名称与目标表的名称相同。若要使用不同的名称，请将新名称输入到 “Destination”字段中或选择一个现有表。

e 选择其它选项以对每个表执行适当的操作。

- 数据模型选项 – 传输启动前，验证目标表是否存在。数据模型选项可帮助您设置目标数据模型。它们不影响执行，但会影响通过模板创建的数据模型。

  若要根据选定的源属性创建不存在的目标表，请选择“Create Table”选项或右键单击该选项并选择“Activate”。若要重新创建现有的表，请选择 “Drop Table”选项。

- 执行选项 – 这些选项将影响项目级的执行。

  选择 “Truncate”选项以在装载前从目标表中删除所有记录。此选项对应于目标组件的 Truncate Table 属性。

  关键项目的故障将导致作业停止执行并发出故障信号。“Critical”选项和 “Ignore Errors”选项对应于多个项目作业组件的属性。

  “Ignore Errors”设置并不影响通过此模板生成的项目。

6 选择要对已收集的数据执行的任务。

注释 除保存模板外，您可以执行这里介绍的所有任务，还可通过在“Navigator”中右键单击存储的模板来执行所有任务。

- Save template – 如果选择此选项，模板将存储在存储库中。通过存储可将已收集的数据重新用于类似作业。
- Build projects and jobs – 选择此选项可为每个源表创建一个项目以及一个可控制所有项目执行的迁移作业。
- Create the destination data model – 根据输入的数据模型选项，选择此选项以设置目标数据模型。单击“Advanced”以输入 SQL 命令，该命令将在创建目标表之前执行。
- Execute job – 仅当已选择 “Build projects and jobs”选项时才可激活 “Execute job”选项。如果选择此选项，处理迁移模板数据后将执行生成的作业。

单击 “Finish”。

注释 如果不希望已收集的数据丢失，请确保至少选择 “Save”模板或 “Build Projects and jobs”选项。

注释 在可以执行生成的作业之前，注册引擎或打开该作业和停用“Multi Engine Execution”选项。请参见第 66 页上的 “使用多个引擎可缩短作业执行时间”。

处理数据时，可查看当前状态和进度。

管理迁移模板

❖ **创建模板**

1　在 “Navigator”中，右键单击 “Templates”。

2　选择 “New” | “Template”。

模板助手将指导您配置迁移模板。

❖ **修改模板**

1　在 “Navigator”中，双击模板，或右键单击要修改的模板并选择 “Open”。

2　修改并保存模板。

❖ **复制模板**

1　在 “Navigator”中，右键单击模板。

2　选择 “Copy”。为新模板输入名称。还可将模板复制到不同的存储库中。

❖ **删除模板**

1　在 “Navigator”中，右键单击模板。

2　选择 “Delete”。

注释 删除模板不影响基于该模板的作业和项目。

❖ **重命名模板**

1　在 “Navigator”中，右键单击模板。

2　选择 “Rename”并输入模板的新名称。

❖ **通过模板构建作业**

基于存储模板创建迁移作业以及所有相关的项目：

1　在 “Navigator”中，右键单击模板。

2　选择 “Build”。若要强制使用唯一名称，创建时间戳将会添加到所有对象名称中。

❖ **通过模板创建数据模型**

根据与模板一起存储的数据模型选项来设置目标数据模型：

1　在 “Navigator”中，右键单击模板。

2　选择 “Create Data Model”。

创建和模拟示例项目

本节介绍如何使用组件创建和模拟示例项目。有关组件、属性和功能的详细信息，请参见第 5 章 “组件”。

项目通常包含一个或多个：

- 用于提供项目数据流所需数据的数据提供程序
- 用于转换或重新映射字段值的数据转换程序
- 用于将转换后的值写入目标的数据接收器

注释 您可缺省 repository 中 “Demo Getting Started”项目内查看本部分的结果。

添加数据提供程序

使用以下其中一种方法可将“DB Data Provider Full Load”添加到项目：

- 将组件从 “Component Store”的 “Source”选项卡拖动到 “Design”窗口。
- 在 Component Store 中，右键单击要添加的组件，然后选择 “Add”。
- 在 Component Store 中，双击要添加的组件。

当将组件添加到 “Design”窗口时，组件的缺省配置将会显示。

注释 所有配置窗口中以粗体显示的属性均为必要属性。

❖ **配置数据提供程序**

1 在 “Interface”菜单中选择 “ODBC”。有关不同接口类型的信息，请参见第 80 页上的 “数据库连接设置”。

2 从 “Host Name”菜单中选择 “ETLDEMO_US”。

确认初始组件设置后，该设置将显示在 “Properties”窗口中。

3 若要定义要从数据源中检索的信息，请单击 “Query”字段中的 “Query”图标。

4 输入 SQL 查询或单击 “Query Designer”图标以生成必要的 SQL。

通过 “Query Designer”窗口的左窗格可导航已连接数据库的表目录。

5 若要添加一个或多个表，请将表名拖动到 “Design”窗口，或右键单击表名并选择 “Add Object to Query”。

6　单击并将 PRODUCTS 表拖动到 “Design”窗口。

7　单击 “Save”以关闭 “Query Designer”并返回至 “Query”窗口。自动生成 select 查询。

8　单击 “Execute the Query”图标可运行或测试该查询。还可以修改该查询。

9　单击 “Save”以关闭 “Query”窗口。

注释 成功配置某个组件后，与其关联的端口颜色将由红色或黄色更改为绿色。

添加数据接收器

使用以下其中一种方法将 “DB Data Sink Insert”添加到项目：

- 将组件从 “Component Store”的 “Destination”选项卡拖动到 “Design”窗口。
- 在 Component Store 中，右键单击要添加的组件，然后选择 “Add”。
- 在 Component Store 中，双击要添加的组件。

将组件添加到 “Design”窗口后，该组件将立即显示其缺省配置。

注释 所有配置窗口中以粗体显示的属性均为必要属性。

❖ **配置数据接收器**

1　在 “Interface”菜单中选择 “ODBC”。有关不同接口类型的信息，请参见第 80 页上的 “数据库连接设置”。

2　从 “Host Name”菜单中选择 “ETLDEMO_DWH”。

3　在 “Destination Table”字段中输入 “PRODUCTS”。或者，单击 “Destination”表图标并从显示的列表中选择 “PRODUCTS”。

4　单击 “Finish”以确认设置。

现在项目包含两个组件。如果在 “File” | “Preference”窗口中添加组件选项时已选择 “Create”自动链接，则组件之间的链接将会自动创建。如果没有自动创建行，请单击输出端口并将其拖动到数据接收器的输入端口以创建行。

“DB Data Provider Full Load”组件的输出端口和 “DB Data Sink Insert”组件的输入端口均显示为绿色。这指明两个组件均已配置。

在 “DB Data Sink Insert”组件的 “Property”窗口中，您可检查并设置选定组件的所有属性。

❖ **检查并定义属性映射**

1 右键单击组件之间的链接。链接的颜色更改为绿色。

2 选择 “Mapping”。

将自动创建数据源和目标源之间的映射。若要更改映射，请选择连接行并将其附加到另一连接点。

注释 仅可映射到未分配的目标连接点。如果所有的目标连接点均已分配，请通过选择并删除当前链接至它的映射行来取消分配的连接点。若要删除，请选择映射行并按 “Delete”键，或右键单击并选择 “Delete”。

添加数据计算器

1 单击 “Component Store”中的 “Transform”选项卡。

2 选择并将 Data Calculator JavaScript 组件拖放到连接现有组件的链接上。链接的颜色更改为蓝色。

释放 “Data Calculator”组件后：

- “Data Calculator”组件与左右侧的组件链接。
- “Data Calculator”窗口将会显示。

“Data Calculator”窗口具有 “Tabular”和 “Graph”视图：

- 使用 “Tabular”视图输入转换规则。
- 使用 “Graph”视图可直观定义输入和输出端口之间的映射序列。

3 单击 “Graph”选项卡。两个部分 – “IN”和 “OUT”代表端口属性的当前结构。

4 单击 “Yes”可按顺序分配缺省映射。

5 单击 “Tabular”选项卡可返回值表格视图。

6 将 PR_NAME 属性的所有传入数据更改为大写字母：

```
uUpper(IN.PR_NAME) ' OUT.PR_NAME
```

7 在 “IN.PR_NAME”属性的 “Transformation Rule”列中输入 uUpper(IN.PR_NAME)。无任何新增函数， IN.PR_NAME 值会转移到 OUT.PR_NAME 属性。

8 单击 “Save”以确认设置。项目中所有端口的绿色指示所有组件均已成功配置。

9 选择 “File” | “Save”。

启动模拟

1　单击工具栏上的 “Start”图标以初始化所有组件。

2　单击 “Step”以从组件到组件逐步执行项目。

在模拟期间的任何时刻，都可预览当前数据集。例如，执行第一步时，数据记录将从源组件转移至数据计算器。链接上的数字指示传输的记录数。

3　右键单击链接并选择 “Preview”以预览链接上的数据。

注释 如果没有已处理的记录或没有可用的模拟数据，将禁用 “Preview”选项。

第 4 章

高级概念和工具

Query Designer

使用 Query Designer 可以执行以下操作：

- 浏览当前项目所连接的任何数据库的表目录。
- 使用图形用户界面创建 SQL 查询。
- 查看生成的 SQL 语句。
- 对数据库执行 SQL 查询。
- 浏览所选表或视图中的数据。
- 在模式中创建表。
- 删除表中的所有记录。

  注释 若要删除表中的所有记录，请在 “File” | “Preference” 窗口中选择 “Enable delete functionality of database objects” 选项。请参见第 18 页上的 “自定义首选项”。

- 计算表或视图中记录的数量。

打开 Query Designer

此部分使用 Demo Repository 中的“Demo Getting Started”项目在 Query Designer 中创建查询。

要打开 Query Designer，请执行以下操作：

1 在“Navigator”中，双击“Demo Getting Started”项目以在“Design”窗口中将其打开。

2 在“Design”窗口中双击“DB Data Provider Full Load”组件。或者，选择该组件以在“Properties”窗口中显示其属性。

3 单击“Query”图标。

4 单击“Query Designer”图标。

“Query Designer”界面

“Query Designer”界面包括以下项：

- “Query Definition”窗格 – 包括“Design”窗口，用于自动生成您所处理的记录所特有的 select 语句。
- Navigator – 在“Model”选项卡下显示所有表和视图，并在“Recent”选项卡下显示最近使用的表或视图。您可以设置要在“File”|“Preferences”窗口中的“Recent”选项卡下查看的表或视图的缺省数量。请参见第 18 页上的“自定义首选项”。
- 属性选项卡（Select、Join、Where、Sort、Group）和“Generated Query”选项卡 – 利用这些选项卡，可以查看属性详细信息，以及在查询创建过程中的任意时刻查看生成的查询。

创建查询

本节使用 Demo Repository 中的“Demo Getting Started”项目创建查询。

❖ **创建简单查询**

要生成用于检索表中所有属性的简单查询，请使用 PRODUCTS 表。

1 在“Navigator”中选择“Model”选项卡，然后单击表或视图的名称。若要搜索特定表名或视图名，请按 Ctrl + F。

2 将所选对象拖动到“Design”窗口中。

3 要查看生成的查询的结果，请选择“View”|“Generated Query”或选择“Generated Query”选项卡。

❖ **使用多个表创建查询**

要生成用于连接两个表并检索两个表中信息的查询，请使用 PRODUCTS 和 SALES 表。

1　将 PRODUCTS 表从 “Navigator” 拖动到 “Design” 窗口中。

2　将 SALES 表从 “Navigator” 拖动到 “Design” 窗口中。

3　通过链接两个表的 PR_ID 字段来为这两个表建立连接。如果希望 Query Designer 在表或视图中的同名属性间自动创建连接，请执行以下操作：

a　从 Sybase ETL Development 主窗口中选择 “File” | “Preferences”。

b　选择 “Workbench” | “Query Designer”，然后选择 “Create joins automatically” 选项。请参见第 18 页上的 “自定义首选项”。

4　要查看连接属性详细信息，请在 “Query Designer” 窗口中单击 “Join” 选项卡。

❖ **修改连接的缺省设置**

两个表之间的连接用一条直线 （用于将连接字段连接起来）表示。直线上标有连接运算符 （缺省值为 Equi Join）。

1　右键单击连接两个连接字段的直线。

2　选择 “Modify”。

3　选择连接类型。

❖ **修改连接的排序顺序**

1　在 “Join” 选项卡中，右键单击一行，然后选择：

- Move to start
- Move up
- Move down
- Move to end

2　要在任意时刻还原到连接的缺省状态，请右键单击一行，然后选择 “Sort joins to default order”。

❖ **向 select 子句中添加一个或多个属性**

1　将 PRODUCTS 和 SALES 表拖动到 “Design” 窗口中 （如果它们尚且不在该窗口中）。

2　要添加一个属性，请右键单击要添加的属性，然后选择 “Add Items to Selection”。要添加多个属性，请按住 Ctrl 键，同时选择要添加的属性。

或者，单击 “Query Definition”窗格中的 “PRODUCTS”和 “SALES”选项卡，然后选择要添加到 select 子句中的属性。要搜索属性，请单击 “Search”图标并提供搜索条件。

注释 您可以使用星号 (*) 作为通配符来搜索任意数量的未知字符。

例如：

- 如果属性的数据类型为 integer，则可以使用下列搜索条件之一：

  ```
  int、 int*、 i*ger
  ```

- 如果属性名中包含 “PROD”并且以 “CD”结尾，则可以使用以下搜索条件：

  ```
  *PROD*CD
  ```

❖ **将选定表中的所有属性添加到 select 子句中**

1 在 “Query”选项卡中，单击表的标头。

2 单击右键，然后选择 “Add Items to Select”。

❖ **查看属性详细信息和生成的查询**

1 要查看 Query Designer 生成的查询，请选择 “View” | “Generated Query”，或者选择 “Generated Query”选项卡。

2 单击相应的选项卡以查看属性详细信息。

❖ **向 select 属性添加函数**

1 在 “Select”选项卡中，右键单击要向其添加函数的属性。

2 选择要添加的函数。

Content Explorer

使用 Content Explorer 浏览所连接的全部数据源的模式信息和数据内容。使用 Content Explorer 只能生成即席查询，这些查询无法保存到文件或存储库中。要保存生成的 SQL，请从 “Generated Query”窗口中选择并复制生成的查询。要打开 “Generated Query”窗口，请从 “View”菜单中选择 “Generated Query”。或者，使用 “Generated Query”选项卡选择并复制查询。

使用下列方法之一打开 Content Explorer：

- 选择 “Tools” | “Content Explorer”。 “Choose Data Source”窗口中显示当前连接到数据源的所有组件。当前连接的数据库的列表中的名称是用户定义的名称与组件类型的通用名称的组合。选择一个组件，然后单击 “Start”以打开 “Content Explorer”。
- 右键单击数据库组件，然后选择 “Content Explorer”。

File Log Inspector

使用 File Log Inspector 可以检查有关项目和作业执行的信息以及致命错误，并且可以查看系统日志。

1　选择 “Tools” | “File Log Inspector”。生成的日志文件即会显示。

2　单击要查看的日志文件信息。您可能会看到下列日志文件中的一个或全部日志文件：

注释 日志文件位于安装目录的 *\log* 子目录中。

- *execution.log* – 捕获有关作业和项目执行的信息。
- *fatal.log* – 捕获当系统遇到严重意外行为时写入的低级信息。它包括当系统无法再向 *system* 日志文件中写入信息时出现的致命的系统异常信息。
- *system.log* – 捕获有关系统活动、操作事件和异常事件的信息。您可以查看 *system.log* 文件中的错误代码，以便确定使用 Sybase ETL 时遇到的错误的原因和可能的解决方案。下表列出了 *system.log* 文件中可能显示的错误代码。

错误代码	类型	说明
0	信息	作业或项目执行成功。
100 或 110	错误	ETL 引擎初始化错误。
101	错误	无效许可证错误。
1100	错误	ETL 异常失败，包括错误的命令行用法。
1103 或 1104	错误	因未指定异常而失败。
10001	错误	无法检索存储库中的信息，包括作业、项目和参数集。
10005	错误	作业执行失败。
10006	错误	项目执行失败。
10101	错误	连接存储数据库失败。

写入 *system.log* 文件的数据的详细信息级别取决于设置的跟踪级别。

要设置跟踪级别，请执行以下操作：

- 选择 “Tools” | “Enable System Trace”。
- 在 JavaScript 过程中使用 uTracelevel(*n*) 函数。

  使用 uTracelevel(*n*) 函数（其中 *n* 的值为 0 到 5），可以设置组件中的跟踪级别。
- 通过在安装文件夹的 *etc* 子目录中的 *default.ini* 文件中修改下面一行来指定跟踪级别：

  ```
  Tracelevel=value
  ```

  其中 *value* 是要设置的跟踪级别。您必须重新启动 Sybase ETL，更改才能生效。

注释 Sybase 建议您将跟踪级别设置为 0 或 5。

3 （可选）– 单击工具栏上的 “Truncate log”图标以截断日志。选择 “Yes”确认。

4 （可选）– 单击工具栏上的“Select rows containing a string or regular expression”图标以找到特定日志文件。

注释 Log File Inspector 仅显示最新 1 MB 的日志文件。若要查看大型日志文件的早期记录，请使用文本编辑器打开相应文件。

管理作业和预定任务

使用 Runtime Manager 可以管理项目和作业以及查看当前预定任务的概述。使用 Runtime Manager 调度向导可创建、编辑、删除、执行和终止任务。

由于 Runtime Manager 基于 Windows 任务调度管理器，因此它包括当前在系统中定义的所有预定 ETL 任务。

❖ **创建新日程表**

1 选择 “Tools” | “Runtime Manager”。

2 选择 “Actions” | “Create”。或者，单击工具栏上的 “Create a new schedule”图标。

3 选择要执行的项目或作业。单击 “Next”。

4　输入日程表名称。提供开始时间和结束日期，并指定所希望的任务执行频率：

- Daily – 在每天的指定时间执行。
- Weekly – 在每周的特定几天的指定时间执行。指定每周的哪几天。
- Monthly – 在每个选定月的特定几天的指定时间执行。指定每月的哪几天，并选择适当的日历月。
- Once – 只在指定的日期和时间执行一次。
- At Login – 在您登录时运行。
- At System Startup – 在系统启动时运行。

单击 “Next”。

5　输入用于执行日程表的 Windows 用户帐户的用户名和口令。确认口令。单击 “Next”。

注释 所提供的帐户必须是有效的 Windows 用户帐户。用户必须对诸如安装目录或 Windows 用户目录之类的 ETL 用户文件夹具有读或写访问权，具体取决于安装的类型。

6　如果成功创建了日程表，则会显示一则消息。单击 “Finish”。

新日程表将与现有日程表 （如果有）一起显示在 Runtime Manager 中。

❖ **编辑日程表**

1　选择要编辑的预定项目或作业。

2　单击工具栏上的 “Edit a Schedule”图标，或者从 “Actions”菜单中选择 “Edit”。

❖ **执行日程表**

1　选择要执行的预定项目或作业。

2　单击工具栏上的 “Execute a Schedule”图标，或者从 “Actions”菜单中选择 “Execute”。

❖ **删除日程表**

1　选择要删除的预定项目或作业。

2　单击工具栏上的 “Delete a Schedule”图标，或者从 “Actions”菜单中选择 “Delete”。

❖ **终止日程表**

1　选择要终止的预定项目或作业。

2　单击工具栏上的 “Terminate a running Schedule”图标，或者从 “Actions”菜单中选择 “Terminate”。

调度项目或作业后，便可在 Windows 任务计划程序的管理下执行项目或作业。

下表中显示了 Task Scheduler 中的作业执行状态码。

表 4-1： 作业执行状态码

结果	类型	说明
0	信息	作业或项目执行成功
1	警告	作业或项目执行已被取消
101	警告	没有有效许可证
10001	错误	无法从存储库检索作业数据
10002	错误	找不到作业的初始化组件
10003	错误	无法初始化外部作业设置
10004	错误	初始化失败
10005	错误	作业或项目执行失败
10101	错误	连接存储数据库失败
10102	错误	在所连接的数据库中找不到有效存储库
10103	错误	无法在所连接的存储库上打开会话

自定义 SQL 和转换规则

设置项目或作业时，您可以自定义以下项：

- 用于设置源组件的 SQL 查询
- 用于预处理和后处理任务的 SQL 命令
- 用于处理转换过程的表达式、条件和过程

SQL 命令的格式取决于连接到组件的数据库系统。但是， Sybase ETL (JavaScript) 支持的转换语言的格式不会改变，无论在项目中使用的源系统或目标系统是什么都是如此。

您可以在中括号表示法 (SBN) 表达式中添加 JavaScript 表达式，这会大大减少您的自定义工作量。SBN 表达式可以是组件属性 （如果为特定属性激活 “Evaluate”选项）、 SQL 语句或任何预处理或后处理命令的一部分。SBN 表达式位于中括号的左中括号和右中括号之间，在运行时计算，这一点与在设计时定义的常量表达式不同。

表达式和过程

表达式是可以计算单个值的标识符和运算符的组合。单个表达式可以是变量、常量、属性或标量函数。您可以使用可用于将两个或多个简单表达式连接成一个复杂表达式的运算符。

下面列出了一些表达式示例：

```
'Miller'
uConcat("Time ", "goes by")
(uMid(SA_ORDERDATE, 1, 10) >= '1998-01-01')
[uTracelevel(3)]
```

过程是指包括表达式、语句和控制结构的编程单位。您可以使用 JavaScript 编写过程，例如：

```
if (IN.PR_PRICE < 250)
       OUT.PR_GROUP2 = 'low end' ;
else {
    if (IN.PR_PRICE < 1000)
       OUT.PR_GROUP2 =  'mid range';
    else
       OUT.PR_GROUP2 = 'high end';
}
```

变量

变量是值的符号名。变量有两个基本属性：

- 作用域
- 数据类型

变量的作用域决定可在环境的哪个范围内引用变量。

变量分为两种类型：

- 端口变量
- 组件变量

端口变量

端口结构的值作为端口变量在组件中引用。输入端口和输出端口都具有自动端口变量。端口变量在组件中有效，并且会继承端口结构的名称和数据类型。变量的名称带有前缀 IN.（表示输入端口）和 OUT.（表示输出端口）。输入端口变量是只读的，而输出端口变量是可写的。

以下示例在表达式中使用端口变量：

```
uUpper(IN.CU_NAME)
```

以下示例在过程中使用端口变量：

```
OUT.CU_NAME = uUpper(IN.CU_NAME);
```

组件变量

组件变量与组件属性相关联，表示当前计算的属性内容。您可以在组件内引用这些变量。变量名称的前缀是 REF，例如：

```
uIsNull(REF.Host)
```

要高度灵活地进行转换，所有端口和组件变量都必须在内部使用数据类型 string。如果您使用数值，则可能会引起意外行为。

如果将字符串变量的数值乘以 1，则可以在计算中使用该数值：

```
IN.Margin="2", IN.Price="10"
IN.Margin>IN.Price  - returns TRUE
```

强制使用数值比较：

```
IN.Margin*1>IN.Price*1  - returns
FALSE
```

端口变量和组件变量的名称不能是下列 JavaScript 关键字中的任何一个：

保留的 JavaScript 关键字

break	case	catch	continue	default
delete	do	else	finally	for
function	if	in	instanceof	new
return	switch	this	throw	try
typeof	var	void	while	with
abstract	boolean	byte	char	class
const	debugger	double	enum	export
extends	final	float	goto	implements
import	int	interface	long	native
package	private	protected	public	short
static	super	synchronized	throws	transient
volatile	const	export		

函数

Sybase ETL 包括一组完整的函数和运算符 （基于支持 Unicode 字符集的设计）。

您可以通过前缀 u 识别 Sybase ETL 函数，例如， uConcat()。

中括号表示法

表达式和 SQL 语句可以包含在 Sybase ETL Server 执行表达式或 SQL 语句之前计算的 SBN 表达式。SBN 表达式用中括号 [..] 括起来。符号 `SBN expression` 用作间接表达式的同义词。

您可以在下列项中使用 SBN 表达式：

- 表达式
- SQL 查询
- Pre-SQL 语句和 post-SQL 语句
- 转换规则
- 文件名
- 路径定义
- 组件属性
- URL

示例

文字是指用引号括起的字符串。如果在文字中使用 SBN，则会首先计算 SBN。

```
'Arrival Date: [uDate( 'now' , 'localtime' )]'
```

以下表达式指定 Text Data Provider 中文件的路径：

```
[uSystemFolder( 'APP DEMODATA' )]\PRODUCTS.TXT
```

注释 在组件的 “Property” 窗口中，“Eval” 列指出是否计算输入的值以解析 SBN 表达式。对于许多 items 属性来说，“Eval” 列都是可选的。要切换 “Eval” 复选框，请右键单击属性项，然后选择 “Evaluate”。

SQL 语句

SQL 查询用于所有提取数据的组件 （主要是数据提供程序和 staging 组件）。因为这些组件定义输出端口结构，所以必须对它们执行查询。

要为组件输入查询，请单击 “Properties” 窗口中的 “Query” 图标。您可以：

- 输入查询。
- 运行查询。
- 保存查询。
- 打开 Query Designer。请参见第 45 页上的 “Query Designer”。
- 查找数据库模式。

输入查询

1 在 “Query” 字段中输入 select 语句。或者，使用 Query Designer 创建即席查询。请参见第 45 页上的 “Query Designer”。您可以从可用数据库模式中选择用于查询的表和属性。请参见查找数据库模式。

2 单击 “Execute Query” 图标。

注释 更改现有的 select 语句后，请初始化组件，然后再执行下一步骤。

验证查询

将立即根据连接到组件的数据库系统验证查询。因此，查询语法必须符合所连接的数据库系统所使用的本地 SQL 方言。使用 SQL92 ANSI 标准查询，可以在不更改 select 语句的情况下切换到其它数据库系统。

查找数据库模式

您可以从可用数据库模式中选择用于查询的表和属性。

1 在 “Properties” 窗口中单击 “Query” 图标。

2 单击 “Database Lookup” 图标。

3 选择要在查询中使用的表和属性，然后单击 “OK”。

在查询中使用 SBN 表达式

以下示例说明如何在查询中使用 SBN 表达式。

示例

用于检索特定客户记录的 select 语句可能包括该记录的常量客户记录 CU_NO：

```
select * FROM CUSTOMERS WHERE CU_NO = '12345678'
```

通过 SBN，可以将常量值 CU_NO 分配给自定义组件属性，从而提高灵活性。请参见第 77 页上的 “自定义属性”。假定为属性 CustNo 分配了值 “12345678”，则包含动态表达式的 select 语句如下所示：

```
select * FROM CUSTOMERS WHERE CU_NO = '[REF.CustNo]'
```

您可以在 SBN 表达式中使用任何 Sybase ETL 函数。以下语句对 CustNo1 使用值 “*1234*”，对 CustNo2 使用值 “*4567*”，从而返回相同的记录：

```
select * FROM CUSTOMERS WHERE CU_NO = '[uConcat
(REF.CustNo1, REF.CustNo2)]'
```

预处理和后处理 SQL 语句

对于任何与数据库相连接的组件，您都可以在组件的 “Property”窗口中输入预处理和后处理 SQL 语句。使用这些属性，可以输入在组件的初始化 （预处理）过程中或在项目完成 （后处理）后执行的 SQL 语句。

一些注意事项：

- SQL 语句在执行后不返回输出。
- 允许使用连接的数据库系统所接受的 SQL 语句。
- 您可以使用分号作为分隔符，在预处理或后处理 SQL 属性中输入多个 SQL 语句。
- 允许在预处理和后处理 SQL 中使用 SBN 表达式。

以下示例显示了预处理和后处理 SQL：

```
delete from products;
update customers
set cu_desc = 'valid';
```

使用 JavaScript 编辑器和调试程序

JavaScript 是面向对象的脚本语言，它嵌入在其它产品和应用程序中。使用 JavaScript 可以处理对象，以便对其进行编程控制。

JavaScript 功能通过网格函数得到增强，这些函数可提高语言的灵活性。使用 JavaScript 编辑器和调试程序，可以交互方式编辑、调试和执行 JavaScript 代码。

功能

JavaScript 编辑器和调试程序主要用于 （但并不仅限于）对传入数据设置转换规则。在 JavaScript 编辑器和调试程序内，您可以使用一个输入记录执行和测试脚本。

JavaScript 编辑器和调试程序具有以下功能：

- 彩色编码语法，用于提高可读性
- 监视列表，用于在运行或单步执行代码时控制变量和属性的赋值
- 多个用户定义的断点，用于在行的任意位置停止代码执行
- 用户定义的执行点，用于任意选择代码的执行位置
- 逐语句模式，用于逐行执行代码
- 调试期间单步跳过
- JavaScript 表达式的计算
- 验证代码执行结果

启动 JavaScript 编辑器和调试程序

1 双击 “Data Calculator JavaScript”组件，或者在 “Property”窗口中单击 “Rule”图标。

2 在 “Transformation Rule”列中选择一行，然后单击 “Edit”图标。

“JavaScript Editor and Debugger”窗口包括以下项：

- Navigator – 由 “Variables”选项卡和 “JavaScript”选项卡组成。“Variable”选项卡包括输入端口和输出端口变量、临时变量和预定义的变量。 “JavaScript”选项卡中包括可在过程中应用的所有函数、命令和系统变量。
- “Edit/Debug”窗格 – 可用于编辑实际代码。
- 窗口底部显示下列选项卡：
  - Tasks – 在编译完过程之后显示验证结果。
  - Watch List – 在调试过程中单步执行代码时显示选定的变量及其值。
  - Input Records – 显示当前输入记录的内容。要对输入记录和输出记录进行同步，请单击工具栏上的 “Start debugging”图标。
  - Output Record – 显示当前输出记录的内容。
  - Expression – 在输入 JavaScript 表达式并单击工具栏上的 “Evaluate”后显示表达式的结果。

编辑模式和调试模式

当启动 JavaScript 编辑器和调试程序时，其将在编辑模式下打开。要切换到调试模式，请选择 “Debug” | “Start”。

编辑区域的背景颜色为深灰色表示调试模式。要切换到编辑模式，请单击工具栏上的 “Stop debugging and start editing”图标。

编辑和调试 JavaScript

要验证 JavaScript 代码，请选择 “Debug” | “Start”。验证结果显示在 “Tasks”选项卡上

JavaScript 编辑器提供了可有效跟踪脚本执行的功能。您可以逐行单步执行代码，也可以从一个断点到另一个断点单步执行代码。您可以随时检查变量的当前值。

注释 注释行以两条正斜杠 // 开头。

❖ **单步执行代码**

注释 JavaScript 编辑器和调试程序可在未输入数据的情况下在组件的输入端口运行。但是，为了实现最佳效果，请在使用调试功能之前用数据填充输入端口。

1　验证脚本或切换到调试模式。

绿色箭头 （最初指向行 1）表示单步执行时的执行进度。

2　确保 “Task”选项卡中的结果消息为 “Successful compilation”。

3　要移动到下一行，请单击工具栏上的 “Step”图标。

您可以在单步执行的任意时刻检查变量的名称和当前值。为此，请在 “Navigator”中选择变量，然后右键单击该变量。

❖ **添加和删除断点**

最好不要逐行单步执行过程，应在选定行中添加断点。

1　要添加或删除断点，请单击要在其中设置断点的行。

2　右键单击，然后选择 “Add/Remove breakpoint”。

❖ **步进到断点**

1　针对每一步单击工具栏上的 “Go”图标。

2　单击最后一个断点上的 “Go”图标以执行脚本的其余部分。

变量的行内检查

当在调试模式下单步执行代码时或在代码执行后，可以对变量的当前值执行行内检查。右键单击变量以查看变量的名称和值。

监控监视列表中的值

您可以在执行代码期间使用监视列表监控变量值的变化情况。单步执行代码时，您可以看到监视列表中的一个或多个变量所发生的任何变化。

❖ **向监视列表中添加变量**

1　右键单击变量。

2　选择 “Add to Watchlist”。

❖ **从监视列表中删除变量**

1　在 “Watch List”选项卡中，选择变量行，然后右键单击该变量行。

2　选择 “Remove Watch Variable”。

JavaScript 的特有功能

中断执行

在编辑器内，单击工具栏上的 “Cancel a running script”图标以中断 JavaScript 执行。

创建用户定义的错误

使用 throw("*xx*") 函数可强制执行错误并中断项目的执行。例如，在产品名称 (PR_NAME) 的长度超过 20 个字符时停止执行：

```
if (uLength(IN.PR_NAME) > 20) (
      throw( “Product name exceeds maximum length” );
)
```

创建用户定义的函数

您可以定义脚本内的函数并创建嵌套函数。例如，以下脚本会使得变量 *b* 的值为 6：

```
var a = 2;
var b = 20;
b = IncA(a);
// end
function IncA (a)
{
     var b = 3;
     a = IncB(b) + a++;
     return a;
     function IncB(b)
     {
          b = b + 1;
          return b;
     }
}
```

转换数据类型

Sybase ETL 中的所有变量均以字符串形式表示。如果您使用数值，则可能会引起意外行为。您可以使用函数 parseInt() 和 parseFloat() 将字符串转换为整数或浮点数，例如：

```
var a = "123";
var b = "22";

a > b

will return FALSE while

parseInt(a) > parseInt(b)
returns TRUE.
```

包括文件

使用 uScriptLoad(“*filename*”) 函数可将外部文件包括在脚本中。外部文件可以包含任何有效的 JavaScript 构造 （包括函数），因而允许使用可重用类型的代码，例如：

```
uScriptLoad("C:\scripts\myfunc.js");
var a = 11;
var b = 2;
var c = 0;
b = gcd(a, b);
// gcd function defined in C:\scripts\myfunc.js
```

执行 SQL 查询和命令

您可以为与源组件、查找组件、Staging 组件和目标组件关联的数据库输入并执行 SQL 查询或一系列 SQL 命令。

❖ **执行 SQL 查询**

1　右键单击 “Design”窗口中的组件，然后选择 “Execute SQL Query”。

2　在 “Query”字段中输入 select 语句。您可以使用所连接的数据库的任何有效 SQL 符号来构建查询。要使用可用数据库模式中的表和视图创建查询，请单击 “Database Lookup”图标。

3　单击 “Execute the query”图标。

成功执行后，结果会显示在 “Data Viewer”窗口中。

❖ **执行 SQL 命令**

1　右键单击 “Design”窗口中的组件，然后选择 “Execute SQL Commands”。

2　在 “Command”字段中输入 SQL 命令。要使用可用数据库模式中的表和视图创建命令，请单击 “Database Lookup”图标。

3　单击 “Execute the command”图标。

成功执行 SQL 命令后，会显示一则消息。

注释 执行一系列命令不会返回一个结果集。

参数集

执行项目时，将使用存储在存储库中的值初始化所有组件属性。通过参数集，可以覆盖这些值中的一部分。例如，当将项目或作业从开发转至生产时，您可以使用参数集更改数据库连接设置。若要使用参数集，请执行以下操作：

- 选择要作为参数使用的组件属性。
- 存储参数值集。
- 执行时分配存储参数集。

❖ **选择组件属性作为执行参数**

1 在 “Properties”窗口中，右键单击要作为执行参数使用的所有组件属性，然后选择 “External”。

2 右键单击要通过参数集分配表达式的所有组件属性，然后选择 “Evaluate”。包括所有非打印值，如 Tab、 CRLF 等。

3 保存该项目。

注释 至少为所有提供项目参数的组件赋予唯一名称。您可能还想更改属性的提示符和说明。有关如何修改属性的提示符和说明的信息，请参见第 77 页上的 “自定义属性”。

管理参数集

您可以将参数集分配给项目和作业。

要打开 “Parameter Set”窗口，请在 “Design”窗口或 “Navigator”中右键单击项目或作业，然后选择 “Parameter Sets”。

“Parameter Set”窗口显示选定项目或作业的已定义参数集列表。

注释 有一些属性的值 （在项目设计中显示）可能与您必须在参数集中提供的值不同。有关这些属性及其值的列表，请参见特殊属性值部分。

❖ **创建参数集**

1 在 “Parameter Set”窗口中，单击 “Set” | “New”。此窗口在底部显示已定义的参数及其当前值的列表。

2 使用要添加的值覆盖当前值。

3 单击 “Save”。

4　输入参数集的名称。

5　单击 “OK”。

❖ **修改参数集**

1　在 “Parameter Set”窗口中选择参数集。

2　单击 “Set” | “Open”。

3　使用新值覆盖当前参数。

4　单击 “Save”。

❖ **删除参数集**

1　在 “Parameter Set”窗口中选择参数集。

2　单击 “Set” | “Delete”。

❖ **复制参数集**

1　在 “Parameter Set”窗口中选择参数集。

2　单击 “Set” | “Copy”。

3　命名新参数集。

4　单击 “OK”。

❖ **使用参数集执行项目或作业**

1　在 “Navigator”中，右键单击项目或作业。

2　选择 “Execute Project with Parameter Set”或 “Job Execute with Parameter Set”。 “Parameter Set”窗口即会打开，其中含有另外一个 “Execute”按钮。

3　从列表中选择一个参数集或添加用于单次执行的参数值。

4　单击 “Execute”。请参见第 68 页上的 “Execution Monitor”。

注释　如果没有可用的存储参数集，则窗口会自动在编辑模式下打开。有关如何将参数集分配给预定项目和作业的其它信息，请参见第 50 页上的 “管理作业和预定任务”。

分配参数值

选择参数

- 要选择一个参数，请单击相应的列表行。
- 要选择多个参数，请在相应的行上拖动鼠标，或按住 Ctrl 并单击以选择其它行。

输入或编辑参数值

选择参数后，请执行以下操作：

- 输入新值。旧值将会被覆盖。
- 直接在 “Value”单元格中编辑现有值。
- 从 “Value”单元格中的弹出菜单上选择 “Edit”，然后编辑该值。单击 “Save”。

❖ **将同一值分配给多个属性：**

因为参数集基于组件属性，所以您可能需要将同一值分配给多个属性。

1 选择要分配其值的参数。

2 输入新值，并确认将该值用于所有选定行。

或者，单击 “Edit Selected values”按钮或者从 “Edit”菜单或弹出菜单中选择 “Edit selected”，进行编辑，然后确认值。

对参数列表进行排序

您可以使用一个或多个列对参数列表进行排序。

❖ **按一个列对参数进行排序**

1 单击列标题。

2 单击多次以在升序、降序和原始排序顺序之间切换。

❖ **按多个列对参数进行排序**

1 单击主列标题多次以按适当的顺序排序。

2 按 Ctrl 并单击辅助列标题以对列进行排序。

特殊属性值

有一些属性的值 （在项目设计中显示）可能与您必须在参数集中提供的值不同。

复选框

有关项目设计中的复选框表示的属性，您必须提供 0 （已停用）或 1 （已激活）作为值。

表达式

要在参数集中使用变量值，请像在 “Design”窗口中那样输入表达式。“Eval”列指出是否为表达式启用某属性。请参见第 55 页上的 “中括号表示法”。

设置包含非打印字符 （如 Tab、 CRLF 等）的值时，表达式显得尤为重要。设计项目时，必须为这些属性设置 “Evaluate”选项。

注释 无法在 “Parameter Set”窗口中验证表达式。

下拉菜单

有些菜单不显示基础参数值。这些值中的一部分值需要您设置 “Evaluate”选项以便通过参数集对其进行分配。

使用下表可确定哪个值 （值）对应于显示的值 （提示符）以及是否必须启用 “Evaluate”。

组件	属性	提示符	值	Evaluate
数据库组件	Interface	ODBC	dbodbc	
		Sybase	dbsybase	
		Oracle	dboracle	
		IBM DB/2	dbdb2	
		SQLite Persistent	dbpersistent	
		OLE DB	dbole	
文本组件	Row Delimiter	位置		
		LF	[uChr(10)]	x
		CR	[uChr(13)]	x
		CRLF	[uConcat(uChr(13),uChr(10))]	x
	Column Delimiter	位置		
		Tab	[uChr(9)]	x
		Comma	,	
		Semicolon	;	
	Column Quote	无		
		单引号	'	
		双引号	"	

使用多个引擎可缩短作业执行时间

网格体系结构可在多个分布式引擎上并行执行项目，因而可缩短作业执行时间。

要充分利用此可伸缩性，您必须：

- 安装多个网格引擎。
- 注册您的网格引擎。
- 准备要采用多引擎执行的作业。

注释 在本指南中，术语 “网格引擎”和 “ETL 服务器”可交换使用。

注册网格引擎

安装网格引擎后，可以针对特殊存储库注册网格引擎。从该存储库执行多引擎作业时，将在这些引擎上执行所有项目。

要注册网格引擎，请选择 “Tools”|“Engine Manager”。如果与多个存储库的连接处于活动状态，请选择其中一个连接。 “Engine Manager”窗口显示已经为选定存储库注册的引擎的列表。

已注册的引擎的属性包括：

- Name：用户为引擎定义的名称。
- Host：引擎主机的名称或 IP 地址。
- Port：引擎所监听的端口的编号。
- Base Rank：用户为引擎定义的排名。作业首先尝试在排在首位的引擎上执行项目。
- Description：服务器的说明。

您可以手动注册单个或多个 ETL 服务器。

❖ **手动注册网格引擎**

1 选择 “Engine”|“New”，或者单击 “New Insert”图标。
2 输入所需的值。
3 单击 “OK”。

❖ **注册多个引擎**

1 选择 “Engine”|“New network search”。 “New Engines”窗口即会显示，窗口中显示注册和其它信息以及引擎的在线状态。
2 选择要注册的引擎。
3 单击 “Add”。

注释 单击工具栏上的 “Refresh”图标或者按 F5 键，可在任意时刻更新和重新装载网格引擎的列表。

❖ **修改引擎注册**

1　选择列表中的引擎，然后选择 “Engine”｜“Edit”，或者单击 “Edit selected engine”图标。或者，双击列表中的引擎。

2　用新值重写当前值。

3　单击 “OK”。

❖ **删除引擎注册**

1　选择要删除的引擎。

2　选择 “Engine”｜“Delete”，或者单击工具栏上的 “Delete selected engine”图标。

定义多引擎作业

您可以使用并行网格体系结构在多个引擎上运行作业。典型的多引擎作业包含多个项目，这些项目相互之间没有依赖性或依赖性很小。因此，可以同时在多个引擎上执行项目。

1　双击作业。

2　在 “Design”窗口中，右键单击选择 “MultiEngine Execution”。

在执行期间，作业使用注册的引擎分配项目。

执行多引擎作业

要执行多引擎作业，请在 “Navigator”中右键单击作业，然后选择 “Job Execute”。或者，将它预定为在 Runtime Manager 中运行。

Engine Monitor

Engine Monitor 显示有关环境中所有可用网格引擎的信息，无论这些引擎是否在 Engine Manager 中运行或进行了注册都是如此。

注释 您只能使用在 Engine Manager 中注册的引擎来执行多引擎作业。

要查看有关可用网格引擎的信息，请选择 “Tools” | “Engine Monitor”。Engine Monitor 显示有关所有引擎的详细信息。您可以在 “Update Interval”字段中指定要在两次更新之间等待的秒数。缺省值为 5 秒。

Execution Monitor

Execution Monitor 显示当前作业或项目的属性。要显示 Execution Monitor，请执行以下操作：

1 选择要执行的项目。

2 单击工具栏上的 “Execute”。

“Execution Monitor”窗口分为两个面板：“Job”和 “Projects”。

“Execution Monitor”窗口的上半部分显示当前执行的作业的属性。

属性	说明
名称	作业或项目的名称
State	当前作业执行状态
Start	开始日期和时间
Stop	停止日期和时间
Message	错误消息

“Projects”列表包含作业中的每个项目所对应的行。

属性	说明
名称	项目名
State	当前项目执行状态
Start	开始日期和时间
Stop	停止日期和时间
Engine Name	执行引擎的名称
Engine Host	执行引擎的主机
Engine Port	执行引擎的端口
Message	错误消息

保存或复制执行结果

若要将执行的项目的结果保存到 HTML 文件中，请单击 “Save Results”。

若要将作业或项目的执行结果复制和粘贴到不同应用程序 （例如，记事本或 Microsoft Word），请右键单击在 Execution Monitor 中显示的作业或项目，然后选择 “Copy”。

取消作业执行

要取消作业执行，请单击 “Execution Monitor”窗口中的 “Cancel Execution”。

GRID 引擎尝试取消正在运行的项目。未启动等待执行的项目。

分析性能数据

当执行作业和项目时， Sybase ETL 收集性能相关数据，并将其存储到存储库表中。

❖ **收集性能数据**

1 若要收集性能相关数据并将其存储到存储库中，请选择 “File” | “Preferences” | “Performance Logging”。

2 从 “Logging Level”列表中，选择 “1”。

❖ **查看性能数据**

1 **查看选定项目或作业的性能数据概述**

在 “Navigator”中右键单击项目或作业，然后选择 “Performance Data”。或者，打开项目，右键单击 “Design”窗口中的任意位置，然后选择 “Performance Data”。

“Performance Data”窗口随即显示，并在 “Overview”选项卡下显示选定项目或作业的执行详细信息。缺省情况下，显示在最近一个月中执行的执行操作的性能数据。若要更改显示的执行日期范围，请在工具栏中更改 “Execution Date Range”的值。

注释 一次只能查看一个项目或作业的性能数据。

若要删除任何执行的性能数据，请选择特定行，然后单击工具栏上的 “Delete selected performance log entries”图标。或者，按 Ctrl+D。

警告！ 这将从性能日志中永久删除选定的执行性能数据。 删除后，无法恢复该数据。

2 **查看项目性能数据**

若要查看项目性能详细信息，请在 “Overview”选项卡中选择某一行，然后单击工具栏上的 “Drill down in performance data”图标。或者，双击选定行或显示图表中的对应栏。此时将显示一个新的选项卡，显示组件名称、持续时间、已读取记录数和已写入记录数等详细信息。

注释 当使用 IQ Loader DB via Insert Location 组件或 IQ Loader File via Load Table 组件查看项目的性能数据时，性能表可能不会显示已读取和写入的记录数的正确值。由于 IQ 直接从文件或远程数据库装载数据，因此，该信息对 ETL 不可用。

若要查看选定组件的性能详细信息，请在显示组件详细信息的选项卡中选择某一行，然后单击工具栏上的 “Drill down in performance data”图标。或者，双击选定行或显示图表中的对应栏。该组件的性能详细信息 （例如，事件名称、持续时间、已读取记录数和已写入记录数）随即显示。

注释 您无法查看 “Load Project”组件的详细信息。该组件表示在执行之前，引擎装载项目所花费的时间。

若要返回到更高级别的性能详细信息，请单击工具栏上的“Drill up in performance data”图标。若要返回到 “Overview”选项卡，请单击工具栏上的 “Return to performance data overview”图标，或者选择 “Navigation”｜“Return to overview”。

3 **查看作业性能数据**

若要查看作业性能详细信息，请在 “Overview”选项卡中选择某一行，然后单击工具栏上的 “Drill down in performance data”图标。或者，双击选定行或显示图表中的对应栏。此时将显示一个新的选项卡，以显示项目名称、持续时间、已读取记录数和已写入记录数等详细信息。

若要查看特定项目的性能详细信息，请选择某一行，然后单击工具栏上的 “Drill down in performance data”图标。或者，双击选定行或显示图表中的对应栏。该项目的性能详细信息 （例如，组件名称、持续时间、已读取记录数和已写入记录数）随即显示。

注释 您无法查看 “Load Project”组件的详细信息。该组件表示在执行之前，引擎装载项目所花费的时间。

若要返回到更高级别的性能详细信息，请单击工具栏上的 “Drill up in performance data”图标。若要返回到 “Overview”选项卡，请单击工具栏上的 “Return to performance data overview”图标，或者选择 “Navigation” | “Return to overview”。

注释 若要在项目中查找选定组件，请选择 “Tools” | “Show component”。或者，单击工具栏上的 “Show selected component”图标。

重置性能数据查询 – 性能数据查询存储在 *HKEY_CURRENT_USER\Software\JavaSoft\Prefs\sybase\sybetl* 下的系统注册表中。如果在使用 “Performance Data”时遇到错误，您需要将这些查询重置为其缺省值。若要将已修改查询随时重置为缺省值，请在注册表中删除 “Performance Data Query”键，并重新启动该应用程序。

警告！ 除非有绝对的必要，否则不要更改查询。

❖ **搜索性能数据**

1 在性能数据表中选择任意行。

2 按 Ctrl+F 打开搜索窗口。在 “Find”字段中输入搜索条件。

被搜索数据随即在该表中突出显示。

❖ **输出性能数据**

1 选择 “Tools” | “Generate Report”。

2 选择要输出的详细信息级别，并选择目标文件。可用选项包括：

- Overview
- Job Performance （仅在输出作业的性能数据报告时显示）
- Project Performance
- Component Performance

3 为生成的报告输入目标文件路径，或者接受缺省值。

4 若要将报告数据限制为 “Overview”选项卡中选择的执行信息，请选中 “Only selected executions”复选框。

注释 如果未在 “Overview”选项卡中选择执行信息，“Only selected executions”复选框显示为禁用状态。

5 单击 “Generate”。成功生成报告之后，将显示一则消息。单击 “Yes”以查看性能数据报告。

该报告显示表格数据和图形数据。它包含一个目录，该目录提供了指向该报告中的对应部分的链接。每份报告都可以包含下列各个部分，具体取决于您选择输出的详细信息级别：

- Overview
- Job Performance （仅在作业性能数据报告中显示）
- Project Performance
- Component Performance

“Performance Overview”部分显示各个项目或作业执行的持续时间。“Project Performance”、“Component Performance”和 “Job Performance”部分为选定执行显示了一个性能数据表和性能数据图，三个部分分别显示不同级别的数据粒度。

性能数据模型和内容

性能数据存储在一个名为 TRON_PERFORMANCE 的非规范化存储库表中，您可以查询该表，以便收集性能数据。本节说明了所含信息。

❖ **事件**

性能日志基于事件。对于每个事件，将存储开始时间 （采用三种变化形式：完整时间戳、日期和时间）和持续时间 （以毫秒为单位）。事件描述包括类、名称和文本。某些事件具有结果 （例如，成功或失败）。此外，还会显示有关事件报告引擎的信息。

下表给出了报告事件的简要说明：

类	名称	说明
control	execute job	作业的总执行时间 （每次作业执行一条记录，PRF_JOB_DURATION 属性中的持续时间 （以毫秒为单位的作业持续时间））。
init	load job	从存储库获取作业定义。
control	execute project	项目的总执行时间 （每次项目执行一条记录，PRF_PRJ_DURATION 属性中的持续时间 [以毫秒为单位的项目持续时间]）。
init	load project	从存储库获取项目定义。
init	create	创建项目和组件实例。
init	configure	配置项目和组件实例。
perform	prepare	执行组件预处理操作。
perform	process	执行组件步骤。
perform	finish	执行组件后处理操作。
perform	read	将数据传输到组件输入端口。
perform	write	从组件输出端口推入数据。

注释 由于分布式多线程的原因，项目执行总时间可能大大少于所有参与组件的执行时间的总和。

一般信息

项目或作业的每次执行都由一个全局唯一 ID 标识。您会发现执行开始时间采用三种变化形式：完整时间戳、日期和时间。此外，还会提供有关启动执行的帐户以及项目或作业所在的存储库的其它信息。

注释 执行和事件的开始时间采用协调通用时间 （UTC，也称为格林威治标准时间 (GMT)）记录。

作业执行信息

对于作业，您将获得 ID、版本 （修改日期）和名称。一个作业执行事件将存储作业的持续时间 （以毫秒为单位）。作业组件 （项目）通过其 ID、类和版本表示。

项目执行信息 （包括作业项目）

对于每个执行项目 ID，将提供版本 （修改日期）、名称和全局唯一执行 ID。一个项目执行事件将存储项目的持续时间 （以毫秒为单位）。

项目组件通过 ID、名称、类、类型和版本表示。过程事件将提供步骤数和已处理记录数。

端口事件提供 ID、名称、类、类型和输入 / 输出块和记录的数量。

第 5 章

组件

本章详细说明了各种 Sybase ETL 组件。

概述

Sybase ETL 组件用于创建项目和作业。Component Store 中提供了这些组件。项目组件包括：

- 源组件 – 用于传递转换流中的数据。项目必须从一个或多个源组件开始执行。源组件没有输入端口，但有一个或多个输出端口。
- 转换组件、查找组件和 Staging 组件 – 用于向转换流中的数据应用特定转换。这些类型的组件至少拥有一个 输入端口和一个输出端口。
- 目标组件 （亦称数据接收器） – 用于将数据写入特定目标。目标组件拥有一个输入端口，但是没有输出端口。
- Loader 组件 – 用于从源数据库或文件中抽取数据并将其装载到 IQ 数据库中，而不执行任何转换。

注释 虽然所有组件从功能角度讲有所不同，但是它们的概念是通用的。

设置组件属性

每个组件都是专用于特定任务的，并且包含特定于任务的功能。但是，所有组件的属性设置过程相同。

只有设置了必要属性之后，才能使用项目中的组件。这些属性可在以下位置设置：

- 向 “Design”窗口添加组件时显示的 “Configuration”窗口。
- 添加到项目中的组件的 “Property”窗口。

对 SBN 表达式进行求值

您可以使用中括号表示法 (SBN) 表达式。这些表达式是在使用 Evaluate 属性的属性值之前求值的。请参见第 55 页上的 “中括号表示法”。

注释 对于某些属性， Evaluate 属性在缺省情况下处于选中状态。

❖ **更改求值属性**

可以通过以下方式之一更改求值属性：

1 在 “Properties”窗口中，为属性选择 “Eval”这一选项。

2 右键单击并选择 “Evaluate”。

注释 如果为 Password 属性选择 “Eval”这一选项，值将以纯文本的形式显示。

加密属性

属性值存储在存储库中。但是， Sybase ETL 存储库中的大多数条目未加密，而以可读字符集的形式表示。

若要切换加密属性，请选中 “Encrypt”复选框，或者在 “Properties”窗口中右键单击属性，然后选择 “Encrypt”。

属性引用变量

使用简单值的属性拥有一个关联变量，而您可以在组件表达式和过程中引用该变量。该变量始终包含求值属性的当前值 （如果设置了 “Evaluate”）。

- 若要显示变量名，请在 “Properties”窗口中右键单击属性，然后选择 “Reference Variable”。
- 在 JavaScript Debugger 中设置转换规则时，请单击 “Variable” | “Parameter”以访问 “Reference Variables”。

自定义属性

可向组件添加自定义属性。与其它属性一样，这些属性包含您可以在组件表达式或过程中引用的变量。它们可以包含在运行时求值或通过参数集赋值的表达式。

❖ **添加自定义属性**

1 在 “Design”窗口中选择组件。
2 在 “Property”窗口中右键单击，然后选择 “Add”。
3 输入属性的名称。它不能是保留的 JavaScript 关键字。请参见第 53 页上的 “变量”。在组件中，此属性是用变量 REF.< *变量名称* > 引用的。
4 在提示字段中输入值。它是显示在 “Properties”窗口中的标签。
5 在说明字段中输入值。说明显示在属性工具提示中。
6 单击 “OK”。

❖ **删除自定义属性**

1 在 “Property”窗口中，选择要删除的自定义属性。
2 右键单击并选择 “Remove”。单击 “OK”进行确认。

注释 确保删除对关联变量的所有引用。

提供对组件的说明

您可以为组件分配名称和说明。该说明和名称显示在工具提示中。当您将鼠标移动到组件上方时，将会显示该工具提示。该名称还显示在组件的顶部。

❖ **向组件添加说明**

1 在 “Design”窗口中右键单击组件，然后选择 “Description”。

2 为组件输入名称和说明。您可以使用 HTML 格式的标记设置说明的格式。

配置端口结构

组件既有输入端口又有输出端口，前者用于接收数据，而后者用于传递处理后的数据。分步调试组件时，通过输出端口转发的数据将被传递到下一组件的输入端口。

管理端口结构

在非结构化端口与结构化端口之间添加连接时，前者将继承后者的结构。

您可以向端口结构添加其它属性。

向项目添加组件之后，组件端口的颜色将指示该组件的状态：

- 绿色 – 组件已被正确配置。
- 黄色 – 定义了端口结构，但未定义一个或多个必要属性。
- 红色 – 既未定义端口结构，也未定义一个或多个必要属性。

注释 通过在 “File” | “Preference”窗口中选择 “Use enhanced color accessibility”这一选项，可以更改组件端口的颜色，以加强对色盲用户的支持。请参见第 18 页上的 “自定义首选项”。

修改端口结构

❖ **修改端口结构**

1 在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2 在 “Structure Viewer”窗口中进行必要的更改，然后单击 “Save”。

注释 对于不能修改的端口结构，“Edit Structure”这一选项不可用。只能使用 “View Structure”选项查看端口结构。

❖ **添加端口属性**

1 在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2 在 “Structure Viewer”窗口中选择 “Actions” | “Add”，或右键单击属性并选择 “Add”。

3　为新属性输入名称。端口属性的名称必须以字母字符开头，且只能包含字母数字字符。它不能是保留的 JavaScript 关键字。请参见第 53 页上的 “变量”。

选择 “Populate Attribute”选项将属性添加到多个端口结构。此时会将新属性添加到参与所选连接的所有端口结构并自动进行映射。单击 “OK”。

❖ **删除端口属性**

1　在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2　在 “Structure Viewer”窗口中选择 “Actions” | “Remove”，或右键单击属性并选择 “Remove”。

❖ **修改端口属性**

1　在 “Design”窗口中右键单击端口，然后选择 “Edit Structure”。

2　在 “Structure Viewer”窗口中更改属性设置，然后单击 “Save”。

❖ **从其它端口复制端口结构**

可以根据当前项目中的其它任何可用端口，为端口分配端口结构：

1　在 “Design”窗口中，选择要为其分配新结构的端口。

2　右键单击并选择 “Assign Structure” | “Copy Structure”。

您可以从同一组件的其它端口复制端口结构。您还可以通过选择 “Copy Structure”复制当前项目中的其它任何端口结构。

如果选择了 “Copy Structure”，将会显示一个窗口，其中显示了当前项目的概要图形。您可以选择该项目中的任何可用端口。

❖ **选择用作新端口结构的源的端口**

1　单击要用作源的端口。此时，选定端口的属性结构将显示在窗口的下方区域中。

2　单击 “Apply”。

模拟组件

初始化组件

1　在 “Design”窗口中选择组件。

2　右键单击并选择 “Initialize”或 “Initialize and Step”。

注释　如果在模拟过程中修改组件的现有属性设置之一，请先重新初始化组件然后再向前移动。

多次分步调试组件

您可以在模拟过程中多次分步调试组件，以便预览具有不同属性设置的组件的行为。重复步骤不会增加传递到下游组件的记录数。

❖ **多次逐步调试组件**

1 在 “Design”窗口中单击组件。

2 在 “Properties”窗口中，修改转换规则或属性设置。

3 右键单击组件，然后选择 “Initialize”。

4 右键单击组件，然后选择 “Step”。

5 返回到第 2 步。

对组件重复进行分步调试时，将会在每个步骤中重新处理输入端口的相同记录集并将其转发到输出端口，而不增加输出端口的记录集数。对于很多组件，可以在组件窗口中或通过使用在 Sybase ETL Development 窗口中右键单击时显示的菜单完成分步调试。

预览转换结果

在模拟模式下工作时，可以 （按照源组件或 staging 组件中的查询所定义的那样）将数据源的整个结果集分成若干数据块。数据块包含记录的子集。每个子集中的记录数与显示在组件的 “Property”窗口中的 “Read Block Size”参数相关。若要在逐步调试项目时提高性能，请为 “Read Block Size”参数选择一个较小的数值。

❖ **预览转换结果**

1 逐步调试项目，包括要预览的组件。

2 右键单击组件、端口或连接链接，然后选择 “Preview”。

如果组件的端口不止一个，请先选择要预览转换结果的端口。

数据库连接设置

您可以在以下位置指定数据库连接参数：

- 向 “Design”窗口添加组件或双击现有的项目组件时显示的 “Database Configuration”窗口。
- 在 “Design”窗口中选择项目组件时显示的 “Properties”窗口。

数据库连接参数如下：

- Interface – 选择要用于连接到目标数据库或源数据库的方法或驱动程序。若要设置特定数据库选项，请单击 “Database options”图标。

  组件的可用接口可以是下面一项或多项：

- Sybase
- ODBC

  注释 ODBC 驱动程序必须安装在 Sybase ETL Development 所在的计算机上，并且必须为目标定义系统数据源名 (DSN)。如果要在 ETL Server 上执行项目， ETL Server 还必须可以访问正确的 ODBC 驱动程序和 DSN。

- OLE DB
- Oracle
- DB2
- SQLite Persistent

  注释 SQLite Persistent 只能在测试环境中使用。请勿将其用于生产环境。

注释 Sybase Open Client™ 必须安装在 Sybase ETL Development 所在的计算机上，并且必须在 Windows 上的 *%SYBASE%\ini\sql.ini* 文件 （UNIX 和 Linux 上的 *$SYBASE/interfaces* 文件）中定义数据库服务器。如果要在 ETL Server 上执行项目， ETL Server 还必须可以访问 Open Client 库。

- Host Name – 选择源数据库或目标数据库。 “Host Name”列表中显示的选项取决于所选的接口。

  注释 如果选择了 SQLite Persistent 接口，可以输入现有的或新的数据库文件名。请参见第 247 页上的 “连接到 SQLite 数据库”。

- Database user name and password – 指定授权数据库用户名和口令。
- Database name – 指定要用作源数据库或目标数据库的数据库。
- Database schema – 指定模式 / 所有者以限制显示的对象，并在该模式下创建新表。
- Standardize Data Formats – 选择此选项可将传入的日期和数值信息转换为标准格式， Sybase ETL 可在支持不同格式的系统之间移动该格式。
- Database options – 选择此选项可覆盖性能缺省值和控制某些事务的行为。有关数据库选项的列表，请参见第 241 页上的 “特定于接口的数据库选项”。

源组件

源组件用于传递转换流中的数据。项目必须从一个或多个源组件开始执行。源组件没有输入端口，但有一个或多个输出端口。

组件	说明
DB Data Provider Full Load	从可通过 ODBC 连接或本地驱动程序 （DB2、 Oracle、 Sybase）访问的任何数据源中抽取数据
DB Data Provider Index Load	执行增量数据装载。增量装载是由包含升序值的 Ascending Index 属性控制的。
Text Data Provider	读取结构化数据，并将其从文本文件转换为表。
XML via SQL Data Provider	将层次结构 XML 数据装载到关系模式中。

DB Data Provider Full Load

DB Data Provider Full Load 是一种源组件，用于从可通过 ODBC 连接或本地驱动程序 （DB2、 Oracle、 Sybase）访问的任何数据源中抽取数据。单个输出端口的结构可镜像查询结果集的结构。

配置 DB DataProvider Full Load 组件

1 将 DB DataProvider Full Load 组件拖动到“Design”窗口。此时将显示 “Database Configuration”窗口。若要打开配置窗口，也可以选择组件，然后单击 “Properties”窗口中的 “Properties”图标。

2 输入连接参数。请参见第 83 页上的 “DataProvider Full Load 属性列表”。

注释 您必须添加有效的接口和主机名。

3 单击 “Query”图标。创建并保存查询，以便从数据源检索数据集。您可以直接在 “Query”窗口中创建简单查询，也可以单击 “Query Designer”图标以打开 Query Designer 来生成查询。请参见第 45 页上的 “Query Designer”。

4 单击 “Finish”。

5 在 “Properties”窗口中指定所有其它可选属性和数据库选项。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1　右键单击 “DB DataProvider Full Load”组件。

2　选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

DataProvider Full Load 属性列表

DataProvider Full Load 属性列表可标识您需要在 “Property”窗口中定义的连接参数和其它项。必要属性以**粗体**文本显示。

必要属性

属性	说明
Interface	标识要用于连接到数据源的方法或驱动程序。
Host Name	标识数据源。 “Host Name”列表中的可用选项取决于所选的接口。
Query	单击 “Query”图标将打开一个窗口，您可在其中创建从数据源检索信息的查询。 您可以使用 “Query”窗口创建简单查询，也可以单击 “Query Designer”图标以打开 Query Designer 来生成查询。请参见第 45 页上的 “Query Designer”。

可选属性

属性	说明
User and Password	二者组合用于标识已授权的数据库用户，以及保护数据库免受未经授权的访问的侵害。
Read Block Size	确定组件在单个步骤中检索的记录数。
Pre-processing SQL	单击 “Pre-Processing SQL”图标将打开一个窗口，您可在其中创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击 “Post-processing SQL”图标将打开一个窗口，您可在其中创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。

属性	说明
Database	标识要用作数据源的数据库。 如果选择此选项，则必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	将传入的日期和数字信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、E 秒钟和秒钟分数的格式：`YYYY-MM-DD hh:mm:ss.s`。 例如， `2005-12-01 16:40:59.123` 数字转换后以 “.” 作为小数分隔符。
Database Options	单击 “Database Options” 图标将打开一个窗口，您可在其中设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

Data Provider Full Load 演示

Sybase ETL 包括 Data Provider Full Load 组件的几个演示。这些演示在 “Help” 菜单上作为 “Flash Demos” 提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Source*”，然后选择 “*DB Data Provider - Full Load*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。然后选择：

- Demo Transfer German Customers
- Demo Transfer German Products
- Demo Transfer German Sales
- Demo Transfer U.S. Customers
- Demo Transfer U.S. Products

DB Data Provider Index Load

DB Data Provider Index Load 是一种源组件，可根据升序索引值执行增量数据装载。在执行过程中，DB Data Provider Index Load 将忽略以前抽取的所有数据记录。

使用 DB Data Provider Index Load，可定期执行跟踪源更改的增量装载。

对模拟序列的影响如下：

- “Read Block Size”的值影响在单个模拟步骤中装载的记录数。
- 如果项目是在模拟过程中执行的，则不会更新存储库中的装载索引的值。

配置 DB DataProvider Index Load 组件

1. 将 DB DataProvider Index Load 组件拖动到 “Design”窗口。此时将显示 “Database Configuration”窗口。若要打开配置窗口，也可以选择组件，然后单击 “Properties”窗口中的 “Properties”图标。
2. 添加连接参数。有关特定字段要求，请参见第 86 页上的 “DataProvider Index Load 属性列表”。

   注释 您必须添加有效的接口和主机名。

3. 单击 “Finish”。
4. 在 “Properties”窗口中单击 “Ascending Index”图标，然后从数据库对象列表中选择升序索引属性。
5. 单击 “Query”图标。创建并保存查询，以便从数据源检索数据集。

   您可以直接在 “Query”窗口中创建简单查询，也可以单击 “Query Designer”图标进行创建。请参见第 45 页上的 “Query Designer”。
6. 在 “Properties”窗口中指定所有其它可选属性和数据库选项。

❖ **重置升序索引值**

不能直接处理存储库中装载索引的永久值。但是，可将该值重置为存储在 Load Index Value 属性中的值。若要重置执行属性，请执行下列操作：

1. 在 “Navigator”中右键单击项目。
2. 选择 “Reset Execution Properties”。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1 右键单击 “DB DataProvider Index Load”组件。

2 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

DataProvider Index Load 属性列表

DB Data Provider Index Load 属性列表可标识您在 “Database Configuration”窗口中定义的连接参数和其它项。必要属性以**粗体**文本显示。

必要属性

属性	说明
Interface	标识要用于连接到数据源的方法或驱动程序。
Host Name	标识数据源。“Host Name”列表中的可用选项取决于所选的接口。 如果使用 SQLite 作为主机，则键入 SQLite 数据库文件的路径和文件名（例如 *c:\mySQLite.db*）。如果具有指定名称的数据库不存在， SQLite 将创建该数据库。

Query	单击 “Query”图标将打开一个窗口，您可在其中创建从数据源检索信息的查询。WHERE 子句中的选择标准需要使用预定义变量 *LoadIndex* 进行限定。LoadIndex 需使用中括号括起来，因为该变量是在查询发送到数据库之前求值的，例如：
	`select * FROM SALES`
	`WHERE SA_DELIVERYDATE`
	`>'[LoadIndex]'`
	`ORDER BY SA_DELIVERYDATE`
	注释 引号字符因数据库系统而异。在 Microsoft Access 数据库上，将 # 用于 datetime 值。
	可以使用 “Query”窗口创建简单查询，也可以单击“Query Designer”图标进行创建。请参见第 45 页上的 “Query Designer”。

Ascending Index	打开一个窗口，您可以在其中选择包含了用于增量装载的升序索引的属性。

可选属性

属性	说明
User and Password	二者组合用于标识已授权的数据库用户，以及保护数据库免受未经授权的访问的侵害。

Load Index Value	在执行 Sybase ETL 作业或日程表时将自动使用并存储升序索引属性的最大值。使用 Load Index Value，可通过用户提供的特定值模拟项目。例如：
	`2005-01-19`
	`100`

Read Block Size	确定组件在单个步骤中检索的记录数。
Pre-processing SQL	单击 “Pre-processing SQL” 图标将打开一个窗口，您可在其中创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击 “Post-processing SQL” 图标将打开一个窗口，您可在其中创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Database	标识要用作数据源的数据库。 如果选择此选项，则必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	将传入的日期和数字信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：`YYYY-MM-DD hh:mm:ss.s`。例如， `2005-12-01 16:40:59.123` 数字转换后以 “.” 作为小数分隔符。
Database Options	单击 “Database Options” 图标将打开一个窗口，您可在其中设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

模拟 Index Load

对包含 Index Load 组件的项目进行全面配置之后，可通过执行以下步骤模拟增量装载：

1 选择 “Run” | “Non Interactive Trace”，以非交互方式运行项目。此时将处理与 Load Index Value 属性指定的条件相匹配的所有记录。

2 以交互方式模拟项目。Index Load 组件将不返回任何记录。请参见第 25 页上的 “模拟项目”。

注释 可以在执行第 1 步和第 2 步期间手动更新源表，以验证是否检索了修改后的正确记录。

3 若要重新模拟，请右键单击组件并选择 “Reset Load Index Value”，或在 “Properties” 窗口中为 Load Index Value 属性输入一个新值。

DataProvider Index Load 演示

Sybase ETL 包括 Data Provider Index Load 组件的几个演示。这些演示在 “Help” 菜单上作为 “Flash Demos” 提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Source*”，然后选择 “*DB Data Provider – Index Load*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer. Repository*” | “*Projects*”。然后选择 “Demo Transfer U.S. Sales on an incremental basis”。

Text Data Provider

Text Data Provider 可读取结构化数据，并将其从文本文件转换为表。文本源必须包含长度固定的字段或分隔字段。

配置 Text Data Provider 组件

1 将 Text Data Provider 拖动到 “Design” 窗口。

2 在 Text Data 组件窗口中，选择要用作数据源的文本文件。

3　在 “Properties”窗口中，修改文件说明属性 （如有必要）。

有关特定字段要求，请参见第 89 页上的 “Text Data Provider 属性列表”。

Text Data Provider 组件窗口

使用 Text Data Provider 组件窗口，可以定义输出端口的数据的结构属性。该窗口包括以下内容：

- “File Content”窗格 – 显示源文档的内容。
- “Properties”窗格 – 显示文件说明属性。
- “Output Port Content”窗格 – 显示输出端口的数据的表格视图。

Text Data Provider 属性列表

Text Data Provider 属性列表可标识与源文件的结构有关的项。属性最初是在您向项目添加组件时设置的。必要属性以**粗体**文本显示。

必要属性

属性	说明
Text Source	标识要用作数据源的文本文件。可以在您向项目添加 Text Data Provider 时选择数据源，也可以从 “Properties”窗口中进行选择。若要从 “Properties”窗口中选择数据源，请单击 “Text Source”图标，然后选择相应文件。
Columns	单击 “Columns”图标将打开一个属性表。使用该属性表，可以为源文件中的数据定义列。

可选属性

属性	说明
Row Delimiter	选择每行的分隔方式： • Position （固定行位置） • LF （换行符） • CR （回车符） • CRLF （回车符加换行符） 也可以输入其它分隔符。
Row Length	指定每个固定行中的字符数，前提是已选择 “Position”作为 “Row Delimiter”。

属性	说明
Column Delimiter	选择各列的分隔方式： • Position （固定列位置） • Tab • Comma • Semicolon 也可以输入其它分隔符。
Column Quote	指定源文件中的值的引号使用方式： • 无 • 单引号 • 双引号
Fixed by Bytes	指定如何解释为行长度、列开头和列结尾提供的值： • Not selected – 将值解释为字符数。这是缺省值。 • Selected – 将值解释为字节数。 示例 – 假定源文件包括 **abc abcdef**，并且具有以下特性： • 文件类型 – 固定 （可变行） • 编码 – GB2312 • 行分隔符 – “\n” • 列定义 – 列 1：1-7 ；列 2：9-10 如果选择了 “Fixed by Bytes”选项： • 列 1 显示前 7 个字节，即， **abc**。 • 列 2 显示第 9 和第 10 个字节，即， **ab**。 如果未选择 “Fixed by Bytes”选项： • 列 1 显示前 7 个字符，即， **abc**。 • 列 2 显示后两个字符，即， **cd**。
Null Byte Substitute	设置要替换空字节的字符。
Skip Rows	跳过行序列中的指定行数。
Encoding	设置当前字符编码。若要设置字符编码，请从下拉菜单中选择合适值。
Support Unicode	设置 Unicode 支持。
Read Block Size	确定组件在单个步骤中检索的记录数。

Text Data Provider 演示

Sybase ETL 为 Text Data Provider 组件包含了若干个演示。这些演示在 “Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Source*”，然后选择 “*Text Data Provider*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。然后选择：

- Demo Transfer German Customers
- Demo Transfer German Sales
- Demo Transfer U.S. Customers

XML via SQL Data Provider

XML via SQL Data Provider 可将分层 XML 数据装载到关系模式中。您可以像查询关系数据库一样查询该关系模式。

XML via SQL Data Provider 是专为销售订单、股票报价或科研数据等以数据为中心的 XML 文档设计的，这些文档一般采用常规的层次结构。

配置 XML via SQL Data Provider 组件

1. 将 XML via SQL Data Provider 组件拖动到 “Design” 窗口。
2. 在 “Properties” 窗口中，单击 “XML Source” 图标，然后选择要用作数据源的 XML。您可以指定 *HTTP*、 *FTP*、 *URL* 或文件名。
3. 单击 “Data Output” 图标。
4. 单击 “Properties” 图标将打开 XML Port Manager。针对每个输出端口指定查询。

   缺省情况下， XML via SQL Data 包括一个输出端口，但您可以添加端口。您在 XML Port Manager 中添加的所有输出端口都将显示在 “Design” 窗口中。有关其它信息，请参见第 92 页上的 “使用 XML Port Manager”。

❖ **更新输出端口结构**

更新输出端口结构以反应对 XML 源文件的更改：

1. 右键单击 “XML via SQL Data Provider” 组件。
2. 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure” 选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取 XML 源文件，并将更新应用到输出端口结构。

使用 XML Port Manager

使用 XML Port Manager 窗口，可以创建针对 XML 源文件的查询，以便定义一个或多个输出数据流。

- XML 源视图 – 显示源文档的内容。
- “Data Model”选项卡 – 显示源文档的关系视图。
- “Reference”选项卡 – 显示可用的组件变量。
- OUT-Port 区域 – 用于编写针对数据模型的查询，并将结果发送到特定输出端口。虽然 XML Port Manager 在缺省情况下配置有一个输出端口，但是，您可以添加其它端口，并且编写其它查询。

编写查询

打开 XML Port Manager 时，OUT-Port 区域将包含一个针对 XML 视图的标准查询，该查询将所有列和所有行返回到 OUT1。假定 XML 源文档包含以每个数据节点的属性值表示的客户数据：

```
<root>
   <data id="101" fname="Michaels" lname="Devlin"
   address="114 Pioneer Avenue" city="Kingston"
   state="NJ" zip="07070"/>
<data id="102" fname="Beth" lname="Reiser"
   address="33 Whippany Road" city="Rockwood"
   state="NY" zip="10154"/>
 <data id="103" fname="Erin" lname="Niedringhaus"
   address="190 Windsor Street" city="Tara"
   state="PA" zip="19301"/>
</root>
```

若要针对 XML 检索客户属性，可以使用与以下内容类似的查询：

```
select * from V_XML_CONTENT WHERE TAB_data_ATT_city =
'Kingston'
```

此查询将打开 Content Browser，并且仅返回那些其 city 值与 Kingston 相匹配的行。在表格视图中，可以编写如下所示的查询：

```
select * from TAB_data where ATT_city='Kingston'
```

注释 XML 数据关系通常表示为父 / 子或节点 / 属性关系。Content Browser 将以列和行的形式返回 XML Port Manager 的查询结果。对于列引用和行引用，引用的是查询结果，而不是 XML 数据。

❖ **从 XML 数据源检索数据**

- 使用标准 SQL 语法在输出端口区域的端口域中直接编写查询：

```
select column
FROM table_name
```

  - Query Designer 可帮助您设计针对表格视图的查询。根据 XML 源的结构，您可能需要在各表之间创建连接，才能返回数据行。
  - 缺省的 XML view_name 为 `V_XML_CONTENT`。若要返回一行，请使用 WHERE 子句对 select 语句进行限定 (`select * from V_XML_CONTENT WHERE TAB_state_ATT_state = 'NY'`)。
  - 在表格视图中，格式为属性表达式的 XML 节点有时会创建一个包装元素，您可以使用 WHERE 子句对该元素进行限定 (`select * from TAB_data where ATT_city='Kingston'`) 以返回一行。

❖ **编写针对表式视图的查询**

- 可以在输出端口区域的端口域中直接编写针对表式视图所做的查询，也可以使用 Query Designer 进行编写。请使用标准 SQL 语法对表式视图中的表进行查询。

❖ **添加和删除端口**

- 右键单击端口部分，然后选择 “Add port” 或 “Remove port”。

  可以添加 Info Port 以将 XML 文档转发到下一组件。退出 XML Port Manager 之后，将显示此端口。

设置样本项目

本节将通过一个简单示例指导您完成对该组件的设置。若要遵照本示例操作，请将 *PRODUCTS.xml* 用作 XML 源；它位于 Sybase ETL 安装目录的 Demodata 子目录中。

XML Port Manager

打开 XML Port Manager，然后在输出端口区域中定义端口。根据 XML Data Model 表，每个端口都是通过 select 语句来描述的。

XML 源

下面的 XML 文档涉及的是一种简单的产品结构。每种产品都是用 ID (PR_ID)、名称 (PR_NAME)、产品组 (PR_GROUP1) 和价格 (PR_PRICE) 来描述的，例如：

```
<?xml version="1.0" encoding="UTF-8"?>
   <dataroot xmlns:od="urn:schemas-solonde-com:demodata"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation="PRODUCTS.xsd"
           generated="2005-01-24T16:13:26"><PRODUCTS>
      <PR_ID>435672</PR_ID>
      <PR_NAME>24 CD Rom Drive</PR_NAME>
      <PR_GROUP1>CD Rom</PR_GROUP1>
      <PR_PRICE>134</PR_PRICE>
   </PRODUCTS>
   <PRODUCTS>
      <PR_ID>435673</PR_ID>
      <PR_NAME>Notebook 235</PR_NAME>
      <PR_GROUP1>Notebook</PR_GROUP1>
      <PR_PRICE>1455</PR_PRICE>
   </PRODUCTS>
</dataroot>
```

数据模型

根元素 (TAB_dataroot) 对应有一个表，随后是与级别 1 的元素对应的一个或多个表。在上例中，只有 (TAB_PRODUCTS) 这一元素位于此级别。在下一级别，可以找到一个与级别 2 的各个元素 （TAB_PR_ID、TAB_PR_NAME、 TAB_PR_GROUP1、 TAB_PR_PRICE）对应的表。XML 文档中可以包含多个嵌套的级别，并且每个级别可以创建另一组表。

注释 可以在 DB Schema Options 属性中更改所生成的表名称的前缀。

根级	级别 1 的元素	级别 2 的元素
TAB_dataroot ATT_ROW_ID ATT_FK_generated ATT_xmlns_od ATT_xsi_no	TAB_PRODUCTS ATT_ROW_ID ATT_FK_dataroot	TAB_PR_ID ATT_ROW_ID ATT_FK_PRODUCTS ATT_PR_ID TAB_PR_NAME ATT_ROW_ID ATT_FK_PRODUCTS ATT_PR_NAME TAB_PR_GROUP1 ATT_ROW_ID ATT_FK_PRODUCTS ATT_PR_GROUP1 TAB_PR_PRICE ATT_ROW_ID ATT_FK_PRODUCTS ATT_PR_PRICE

这些表是通过外键链接在一起的。表 TAB_PRODUCTS 通过 ATT_FK_dataroot 属性链接到 TAB_dataroot。

级别 2 的各表通过 ATT_FK_PRODUCTS 属性链接到表 PRODUCTS。若要创建包含 PRODUCTS 记录的视图，必须将级别 2 的各表与表 TAB_PRODUCTS 连接。

该连接是通过级别 2 的每个表的 ATT_FK_PRODUCTS 属性与 TAB_PRODUCTS 的 ATT_ROW_ID 来限定的。只有级别 2 的各表的值属性才是选定属性：ATT_PR_ID、 ATT_PR_NAME、 ATT_PR_GROUP1 和 ATT_PR_PRICE。

```
select  TAB_PR_ID.ATT_PR_ID,
TAB_PR_NAME.ATT_PR_NAME,
TAB_PR_GROUP1.ATT_PR_GROUP1,
TAB_PR_PRICE.ATT_PR_PRICE
FROM  TAB_PRODUCTS, TAB_PR_ID, TAB_PR_NAME,
TAB_PR_GROUP1, TAB_PR_PRICE
WHERE  TAB_PR_ID.ATT_FK_PRODUCTS =
TAB_PRODUCTS.ATT_ROW_ID AND
TAB_PR_NAME.ATT_FK_PRODUCTS =
TAB_PRODUCTS.ATT_ROW_ID
AND  TAB_PR_GROUP1.ATT_FK_PRODUCTS =
TAB_PRODUCTS.ATT_ROW_ID AND
TAB_PR_PRICE.ATT_FK_PRODUCTS =
TAB_PRODUCTS.ATT_ROW_ID
```

XML via SQL Data Provider 属性列表

XML via SQL Data Provider 属性列表可设置 XML 源文件的处理选项。必要属性以**粗体**文本显示。

必要属性

属性	说明
XML Source	标识数据源。 可以在您向项目添加组件时选择 XML 数据源，也可以从 “Properties”窗口中选择文件。若要从 “Properties”窗口中选择数据源，请单击 “*XML Source*”，然后选择该文件。
Data Output	单击 “Data Output”图标将打开 XML Port Manager。XML Port Manager 是一种管理主控台，您可以在其上对 XML 源进行查询。 有关详细信息，请参见第 92 页上的 “使用 XML Port Manager”。

可选属性

属性	说明
Document Schema	单击 “Document Schema”图标，将标识可用于验证 XML 源的外部模式 (*.xsd*) 或 DTD。
Namespace Schema	指向外部命名空间模式的位置。XML 模式包含类型定义和元素声明等组件，这些类型定义和元素声明可用于评估格式正确的元素和属性信息项的有效性。
Validate Schema	如果要启用模式和 DTD 验证，请选择此选项。
XML Options	单击 “XML Options”将打开一个窗口，您可以在其中设置下列 XML 处理选项： • Full schema check – 若要查看是否存在可能比较耗费时间或内存的项，请将此项设置为 1。粒子唯一归属约束检查和粒子派生限制检查都是由此选项控制的。缺省值为 0。 • Ignore external DTD – 若要忽略在文档中引用的外部 DTD，请将该值设置为 1。缺省值为 0。 • Process namespace – 如果不想在分析过程中考虑命名空间规范，请将该值设置为 0。缺省值为 1。 • Preserve Element whitespace – 请将该值设置为 True (1)，以便保留 XML 元素值中的空格。 如果设置为 False (0)，则会剪裁掉 XML 元素值旁的任何空格，或者如果 XML 元素值仅包含空格，该值则被解释为空元素值。

属性	说明
DB Schema	单击 “DB Schema”图标，可选择数据库模式设置（创建表）脚本。使用此选项可强制执行固定数据模型。
DB Schema Options	单击 “DB Schema Options”图标将打开一个窗口，您可以在其中为通过 XML 结构生成的表和属性自定义设置，包括表名称和属性名的前缀。 “DB Schema”选项如下所示： • Attribute name case – 为通过 xml 生成的属性名设置格式。 “upper”和 “lower”将相应地转换属性名。 “mixed”将名称保留为在 xml 文档中的显示格式。缺省值为 mixed。 • Attribute name prefix – 用于生成的每个属性名的前缀。 • Create indexes – 如果设置为 1，则会对表的主键自动生成索引。缺省值为 1。 • Create flat views – 如果设置为 1，则会自动生成名为 V_XML_CONTENT 的视图。该视图连接所有表，并在宽表中返回所有 xml 数据。缺省值为 1。 如果数据库模式生成的表多于 32 个，该视图则不起作用，并且必须关闭该选项。 • Foreign key prefix – 用于作为外键的属性的前缀。 • Ignore Empty Leaf Element Values – 如果您不希望数据库包括不含有任何数据的特定 XML 叶元素的列条目，请将该值设置为 True (1)。如果设置为 False (0)，从 XML 文档创建的数据库将包括所有 XML 叶元素的列条目，而不管这些元素是否为空。 • Primary key name – 用于主键的属性名。 • Table name case – 为通过 xml 生成的表名设置格式。 “upper”和 “lower”将相应地转换表名。 “mixed”将名称保留为在 xml 文档中的显示格式 • Table name prefix – 用于生成的表名的前缀
Database Options	单击 “Database Options”图标将打开一个窗口，您可在其中设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。
Read Block Size	确定组件在单个步骤中检索的记录数。如果组件的输出端口不止一个，则读取块大小将被忽略，并且该组件将在单个步骤中提供所有端口的数据。

XML via SQL Data Provider 演示

Sybase ETL 为 XML via SQL Data Provider 组件包含了若干个演示。这些演示在 “Help” 菜单上作为 “Flash Demos” 提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Source*”，然后选择 “*XML viaSQL - Data Provider*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。然后选择 “Demo XML via SQL Data Provider”。

转换组件

转换组件至少拥有一个输入端口和一个输出端口，可对转换流中的数据应用特定转换。

组件	说明
Data Calculator JavaScript	对在输入端口与输出端口之间传递到该组件的每条记录执行转换。
Data Splitter JavaScript	根据输入值拆分传入数据流。
Character Mapper	替换输入记录中的字符和字符串。Character Mapper 可向所有选定属性应用替换映射。

Data Calculator JavaScript

Data Calculator JavaScript 是一种转换组件，用于定义对输入端口与输出端口之间传递的记录应用转换的规则。例如，可以使用 Data Calculator JavaScript 定义转换端口属性的规则或添加创建新属性的规则。

Data Calculator 还可以执行属性级查找。您可以提供特定查找端口的查找数据。

对模拟序列的影响如下：

如果不执行任何查找，Data Calculator JavaScript 不影响模拟序列。如果执行查找，系统先将所有数据读入查找端口，然后处理主端口的数据。

配置 Data Calculator JavaScript 组件

1　将 Data Calculator JavaScript 组件拖动到 “Design” 窗口。

2　将 Data Calculator 的输入端口链接到提供入站数据的组件的输出端口 （如有必要）。

3　如果 Data Calculator 组件窗口处于打开状态，请单击 “Save” 以关闭该窗口。

4　在 “Design” 窗口中，右键单击 Data Calculator 的输出端口，然后依次选择 “Assign Structure” 和下列选项之一：

选项	操作
IN	创建与 Data Calculator 的输入端口结构相匹配的输出端口结构。
Copy Structure	打开一个窗口，您可以在其中向输出端口应用现有的端口结构。

5　在 “Design” 窗口中双击 “Data Calculator JavaScript”，或者在 “Properties” 窗口中单击 “Rule” 图标。此时将打开组件窗口，并提示您按顺序分配缺省映射。

6　选择下列选项之一：

选项	操作
Yes	按顺序创建缺省映射。 选择此选项可创建一组转换规则，这些规则用于将输入端口中的每个属性映射到输出端口中的相应属性。
No	自行定义输入端口到输出端口的映射。 选择此选项可将输入端口中的各属性分别映射到输出端口中的相应属性。请参见第 101 页上的 “映射端口属性”。

有关映射选项和转换规则的详细信息，请参见第 99 页上的 “使用组件窗口”。

使用组件窗口

Data Calculator 组件窗口提供了数据流的表格视图和图形视图。您还可以使用此窗口映射端口属性和定义转换规则。

表格视图

表格视图提供了当前记录结构、端口属性和转换规则的结构视图。该视图包括以下内容：

- “Current Input Record”窗格可标识输入端口的当前记录中的每个属性及其属性值。所有的输入端口属性都包括 IN. 这一前缀。
- “Transformation rules and Current Output Record”窗格以表格形式显示了每个转换规则和每个输出端口属性。缺省的转换规则可将每个输入端口属性映射到相应的输出端口属性。缺省情况下，输出端口值反映输入端口值。更改属性值或转换规则时，将会更改输出端口值。

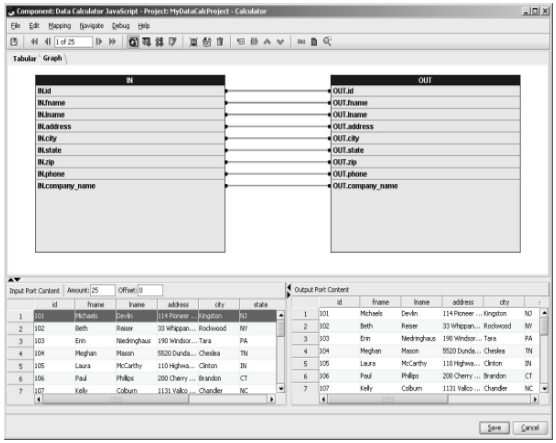
  如果转换规则是函数表达式，则必须仅在一行中输入该规则。缺省情况下，将返回值赋给相应的输出端口属性。但是，如果输入了多行函数表达式，则文本将被解释为脚本，此时您需要设置输出端口属性值。
- “Input Port Content”区域显示输入端口当前可用的记录集。
- “Output Port Content”区域显示输出端口当前可用的记录集。

图形视图

图形视图显示了输入端口属性与输出端口属性之间的当前映射。

- “Input Port Content”区域显示输入端口当前可用的记录集。
- “Output Port Content”区域显示输出端口当前可用的记录集。

注释 向 IN.attribute 应用转换规则之后，将不再显示 IN.attribute 与 OUT.attribute 之间的映射行。



❖ **映射端口属性**

虽然作为缺省选项， Data Calculator 可以在输入端口与输出端口之间创建列到列的映射，但是，可能有时需要您将端口属性以单个形式映射。若要创建自己的映射，请使用图形视图。

1　在 Data Calculator 组件窗口中，单击 “Graph”选项卡。

2　采用以下方式之一映射输入端口结构和输出端口结构：

- 选择菜单栏中的 “Mapping”，然后选择以下预定义映射序列之一：
  - Create mapping by Order – 按顺序映射输入结构和输出结构的端口属性。

    注释 如果两端的属性数目不同，则说明部分端口属性未映射。

  - Create mapping by Name – 按照其名称映射输入结构和输出结构的端口属性。
  - Create mapping by Name Case Sensitive – 按照其区分大小写的名称映射输入结构和输出结构的端口属性。
  - Create mapping by prefix – 按照名称 （忽略指定的前缀）映射输入结构和输出结构的端口属性。
  - Create mapping by Best Match – 映射可能相似的输入结构和输出结构的端口属性。
- 分别连接输入端口和输出端口的各属性。

现在便可使用该组件了，它可将记录从输入端口转发到输出端口。

显示转换结果

Data Calculator 的一项基本功能是，可以立即显示转换规则的结果。使用这项模拟功能，可以验证传入数据、测试转换规则，并查看这些规则对数据输出产生的影响。

当前输入记录属性值

缺省情况下，输出端口值反映输入端口值。通过更改输入端口属性值，可以测试转换规则，还可以查看当前输出记录中的结果。在此区域中所做的手动修改只影响输入或输出端口缓冲区中的数据。这一机制为创建针对转换规则的测试案例提供了方便。

编辑转换规则

在 “Transformation Rule”列中，可以添加、修改或删除转换规则。您还可以通过更改当前属性输入字段来编辑单行函数。将函数 uUpper 添加到属性 IN.PR_NAME，如下例所示：

3	IN.PR_PRICE	OUT.PR_PRICE	low end
4	IN.PR_GROUP1	OUT.PR_GROUP1	CD Rom
5	uUpper(IN.PR_NAME)	OUT.PR_NAME	24 CD ROM DRIVE
6	IN.PR_ID	OUT.PR_ID	435672
7	if (IN.PR_PRICE < 250) OUT.P		
8	'valid'	OUT.PR_DESC	valid

您可以使用 Procedure Editor。有关创建复杂过程转换的信息，请参见第 57 页上的 “使用 JavaScript 编辑器和调试程序”。

您还可以添加转换规则。如果 OUT 属性数与 IN 属性数不匹配，或者如果要在开发的后期阶段向项目添加其它属性，这是极为有用的。

注释 图形视图可镜像实际端口结构。您不能在其中添加或删除属性。

❖ **管理转换规则**

1 若要添加转换规则，在 “Transformation Rule”或 “Current Output Port”列中的任意位置单击右键。

2 选择 “Insert”。

现在，便可使用添加的规则执行进一步的赋值或计算

- 若要删除转换规则，请右键单击 “Transformation Rule”列中的某行，然后选择 “Remove”。
- 若要更改转换规则的顺序，请右键单击 “Transformation Rule”列中的该行，然后选择 “Up”或 “Down”。
- 若要添加缺少的输出属性，请单击 “Mapping”，然后选择 “Add missing output attributes”。

处理转换规则的顺序

转换规则是按顺序处理的。首先处理的是列表中的第一个转换规则。请看以下示例：

Transformation Rules and Current Output Record

Order	Transformation Rule (Expression)	Lookup	Output Port	Value	Data Type	Description	Modified
1	IN.CU_NO		OUT.CU_NO		string		2006-08-17 ...
2	IN.CU_NAME		OUT.CU_NAME		string		2006-08-17 ...
3	IN.CU_BUYER		OUT.CU_BUYER		string		2006-08-17 ...
4	IN.CU_CITY		OUT.CU_CITY		string		2006-08-17 ...
5	IN.CU_STREET		OUT.CU_STREET		string		2006-08-17 ...
6	IN.CU_CREATEDATE		OUT.CU_CREATEI		string		2006-08-17 ...
7	IN.CU_ZIPCODE		OUT.CU_ZIPCODE		string		2006-08-17 ...
8	uConcat(IN.CU_NAME,',',IN.CU_CITY)		OUT.CU_FULLNAI		string		2006-08-17 ...
9	IN.CU_CITY	LOOKUP1.Zip	OUT.CU_CITYSIZE		string		2006-09-16 ...
10	if (OUT.CU_CITYSIZE == 0) OUT.CU_C						2006-09-16 ...

第 9 行用于查找城市的邮政编码的数值，并将结果存储在 CitySize 中。第 10 行中的过程可根据 CitySize 中的数值计算 CitySize 字符串。

模拟 Data Calculator

Data Calculator 的设计目的为，使用户可在数据于整个转换规则中移动时看到对数据所应用的更改。您会发现这是很有用的，例如可以了解更改转换规则对传出数据的影响。根据 “Auto-Synchronization”按钮的状态，转换规则将在输入之后立即应用于整个输入端口记录集。

切换自动同步：

自动同步立即将对转换规则所做的全部更改应用于输入端口的当前记录集。

注释 如果禁用 “Auto-Synchronization”，可以通过选择 “Step”选项手动触发对输入端口数据的处理。

手动应用所有转换规则：

若要向输入端口的当前所有记录手动应用所有转换规则，请单击工具栏上的 “Step”图标。

提取其它记录集：

若要提取其它记录集，请单击工具栏上的“Step through the next incoming data buffer”图标。

逐条处理输入端口记录：

若要逐条处理输入端口的记录，请执行下列操作：

- 单击工具栏上的相应记录控件。
- 单击 “Navigate”，然后选择相应选项。
- 从 “Input Port Content”列表中选择相应记录。

注释 “Current Input Record”区域中显示的值将随着当前记录的更改而更新。

搜索转换规则中的关键字：

单击工具栏上的 “Search the Content of Transformation Rules”图标。

突出显示 null 值和空值

单击工具栏上的 “Highlight NULL-Values and Empty Values”图标。

在 Data Calculator 中使用查找

Data Calculator 可执行属性级查找。您必须输入特定查找端口的查找数据。

添加查找端口

若要向 Data Calculator 添加查找端口，请将数据提供组件的输出端口直接连接到 Data Calculator 组件（不是端口）。查找端口是以自动方式创建和连接的。或者，右键单击 Data Calculator，然后选择“Add Input Port”。将数据提供组件的输出端口与新端口相连。查找端口数不受限制。

准备查找数据

每个查找端口必须至少拥有两个属性。第一个属性代表键。其它所有属性代表返回值。

如果要查找复合键，则必须并置前一组件中的键值，并在用于查找的键表达式中使用同类并置。

设置一般查找选项

使用 Lookup Options 属性可对查找进行配置。“Properties”窗口显示了所有查找端口及其当前选项值的列表。

- Lookup Name – 继承自关联端口，并且不能在此更改。若要更改端口的名称，请从端口菜单中选择 “Description”。
- Lookup Size – 输入查找记录的估计数，以优化内存分配和查找性能。请参见第 112 页上的 “DB Lookup”。
- Lookup Empty / Null – 空值和 null 值通常作为 “unknown”键处理，因而返回缺省查找值。如果空值或 null 值是对查找有效的键，则可通过激活此选项强制查找与这些键对应的值。

构建查找规则

若要设置查找规则，请在 Data Calculator 窗口中打开“Tabular”选项卡。如果查找端口可用，则还会显示一个 “Lookup”列。

提供下列与每个查找规则有关的信息：

- Key Expression – 这是要在查找列表的第一列中搜索的值。输入键表达式 （例如 IN.PR_ID）作为转换规则。
- Return Value – 由于查找列表可以包含多个返回值列，您必须指定要返回的值 （如果找到了键）。若要指定要返回的值，请从“Lookup”列的弹出菜单中选择关联的端口属性。例如：LOOKUP1>>LOOKUP1.PR_NAME。

注释 虽然返回值是按照名称选择并显示的，但是，查找内部使用了列号。这意味着，只要修改了查找端口结构，就必须检查查找规则，在添加或删除属性之后更要如此。

- Output Variable – 将查找返回值赋给此变量。您可以从 “Output Port”列的弹出菜单中选择一个变量 （例如 OUT>>OUT.PR_NAME）。
- Default Expression （可选） – 如果找不到给定键值，则查找返回 Null。若要返回其它缺省值，请在 “Lookup Options”窗口中输入一个表达式。若要打开 “Lookup Option”窗口，请在 “Transformation Rules and Current Output Record”面板下单击 “Lookup”列中的图标。 “Lookup Options”窗口用于指定缺省表达式。该窗口包括一些弹出菜单，用于选择 Lookup 和 Output 属性。

Data Calculator JavaScript 演示

Sybase ETL 为 Data Calculator 组件包含了若干个演示。这些演示在 “Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Transform*”，然后选择 “*Data Calculator - JavaScript*”。

Demo Repository

在 “Navigator”中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。

对于不包含查找的演示，请选择：

- Demo Transfer U.S. Products
- Demo Transfer German Products
- Demo Data Calculator

对于包含查找的演示，请选择：

- Demo Transfer German Customers
- Demo Transfer German Sales
- Demo Transfer U.S. Customers

Data Splitter JavaScript

使用 Data Splitter JavaScript，可以轻松地过滤和分配输入数据。

配置 Data Splitter JavaScript 组件

1 将 Data Splitter JavaScript 组件拖动到 “Design”窗口。

2 将 Data Splitter 的输入端口链接到提供入站数据的组件的输出端口。

3 将 Data Splitter 的输出端口链接到要将出站数据引导至的组件的输入端口。

- OUT1 是顶部输出端口。
- 而 OUT2 是底部输出端口。

您必须配置要将出站数据引导至的组件的输入端口结构。

4 双击 Data Splitter JavaScript 组件。

5 添加要用于引导数据流的条件：

a 右键单击要向其添加条件的端口 （OUT1、 OUT2），然后选择 “Edit”。

b 在 “Condition”窗口中，针对每列定义要应用的条件。

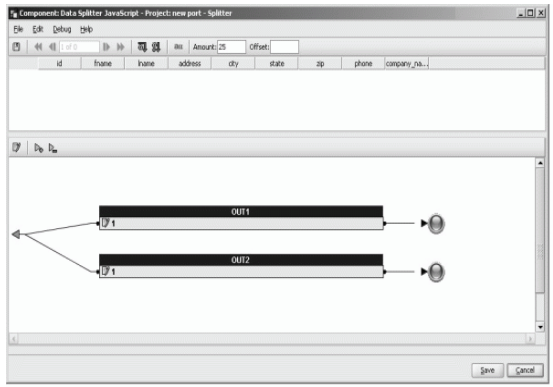
c 单击 “Save”。

拆分入站数据

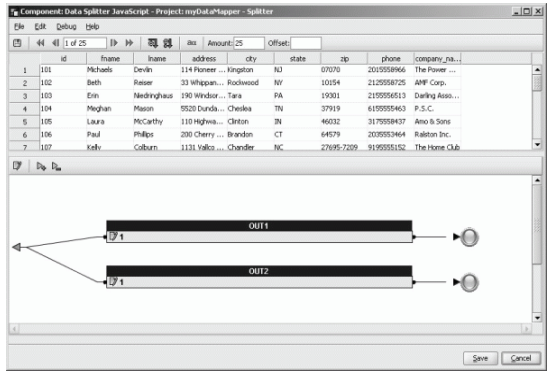
向项目添加 Data Splitter 时，将打开一个组件窗口，其中会显示入站数据属性和输出端口条件。

Data Splitter 配置有两个输出端口，即 OUT1 和 OUT2。OUT1 端口是顶部输出端口，而 OUT2 端口是底部输出端口。
对于 OUT1 和 OUT2 端口，其输入端口条件都预先设置为 1。因为该条件总是为 true，所以，无论当前值为何 （如果是输入端口）， Data Splitter 都会将所有的传入记录复制到这两个输出端口。

由于输入端口的数据缓冲区最初为空，仅显示入站数据属性。OUT1 和 OUT2 的输出端口结构与输入端口结构相匹配。



若要填充输入属性，请单击工具栏上的“Step to the next input buffer”图标。此时，输入数据将显示在该组件窗口的上部。



“Input”区域中的记录与输出端口的记录之间具有对应关系。由于 OUT1 和 OUT2 与输入端口共享同一种端口结构，在窗口的上部选择任何记录时，将会使输出端口指示该记录是否满足端口条件。如果记录满足端口条件，则输出端口的颜色为绿色。如果记录不满足端口条件，则输出端口的颜色改为红色。

注释 Data Splitter 转发到输出端口的记录数可能与传入的记录数不同。如果有一条记录与多个端口条件相匹配，则该记录可以位于所有这些端口上。如果记录与任何条件都不匹配，则会将其从数据流中删除。

自定义端口条件

可以为每个端口分配条件。条件由一个或多个表达式组成。多个表达式可通过运算符并置。当 Data Splitter 对条件进行求值时，其结果是 TRUE (1) 或 FALSE (0)。

例如，您可能需要为 OUT1 编写一组条件，而为 OUT2 编写另一组条件。

❖ 修改端口条件

1 双击 Data Splitter JavaScript 组件，以打开 Data Splitter 组件窗口。右键单击端口并选择 “Edit”。

2 在 “Condition”窗口中，创建要向端口应用的条件。您可以：

- 在 “Condition”窗口的文本区域中手动输入条件。
- 将要添加到条件中的变量和函数从左窗格拖放到文本区域。“Variables”选项卡列出了可以使用的所有变量，而 “Functions”选项卡列出了可添加到条件中的所有可用函数和运算符。
- 右键单击文本区域，然后从 “IN”弹出菜单中选择要添加到条件中的变量。

❖ 添加新的输出端口

虽然 Data Splitter 配置有两个输出端口，但是，您可以创建其它输出端口。新端口是通过名称来标识的。

1 单击端口工具栏上的 “Add new port”图标。

2 为新的输出端口输入名称。单击 “OK”。

❖ **删除输出端口**

1　选择要删除的端口。

2　单击端口工具栏上的 “Remove selected port”图标。或者，右键单击端口并选择 “Delete”。

注释 不能删除所有端口。该组件必须至少拥有一个输出端口。

特定端口条件

Data Splitter 分配记录的方式由端口条件决定。您可以定义互不排斥的交叠端口条件。

- 1 – TRUE。所有记录都被转发到此端口，包括同时与其它任何端口条件相匹配的记录。
- （空） – 条件为空的端口可收集与 Data Splitter 中定义的其它任何条件不匹配的所有记录。

Data Splitter JavaScript 演示

Sybase ETL 为 Data Splitter 组件包含了若干个演示。这些演示在 “Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Transform*”，然后选择 “*Data Splitter - JavaScript*”。

Demo Repository

在 “Navigator”中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。然后选择：

- Demo Data Splitter
- Demo Text Data Sink Delimited/Fixed

Character Mapper

Character Mapper 是一种转换组件，用于替换输入记录中的字符和字符串。Character Mapper 可向几乎所有的选定属性应用替换映射。

使用 Character Mapper 可替换字符或字符串。例如，可将德语的变音符号替换为 ae 或 Unicode 字符。

配置 Character Mapper 组件

1 将 Character Mapper 组件拖动到 “Design” 窗口。

2 将 Character Mapper 的输入端口链接到提供入站数据的组件的输出端口。将 Character Mapper 的输出端口链接到要将出站数据引导至的组件的输入端口。

您需要配置要将出站数据引导至的组件的输入端口结构。

3 打开 Character Mapper 组件窗口。如有必要，请单击工具栏上的 “Step to next incoming data buffer” 按钮，以填充输入和输出内容。

4 添加映射定义。请参见第 110 页上的 “创建新的映射定义”。

使用 Character Mapper 组件窗口

使用 Character Mapper 组件窗口，可为在输入端口与输出端口之间传递的数据定义映射规则。

Character Mapper 组件窗口包括以下内容：

- “Current Input Record” 窗格，显示当前位于输入端口的记录的列、行和内容。
- “Current Output Record” 窗格，显示出现在输出端口的当前记录的列、行和内容。
- “Mapping definition” 窗格，包括 “*From*” 列和 “*To*” 列。

❖ 创建新的映射定义

1 单击工具栏上的 “Insert Mapping” 图标，或者在 “Mapping Definition” 窗格中的任意位置单击右键，然后选择 “Insert Mapping”。此时将为新的映射定义插入一个新行。

2 在 “*From*” 列中输入要替换的字符组合，并在 “*To*” 列中输入要将其替换为的值。Character Mapper 会应用规则，并将结果显示在 “Current Output Record” 窗格中。

- 若要编辑当前显示在 “Current Input Record” 窗格中的记录，请在 “Current Input Port Content” 窗格中单击相应的行，然后进行所需的更改。这样还将更新 “Current Output Record” 窗格中的值以及 “Current Outport Port” 窗格中的选定行。

- 若要删除映射定义，请右键单击相应的映射定义，然后选择“Remove Mapping”。这样将删除选定的映射定义。
- 若要更改映射定义的顺序，请选择工具栏上的 “Move row up”和 “Move row down”图标。
- Character Mapper 可应用映射，并在您编写规则后立即更新输出结果。单击工具栏上的“Enable auto refresh of output values”图标，可切换这种自动同步功能。

注释 Character Mapper 将对所有行和所有列应用映射。如果要将某些列排除在字符映射之外，请在 “Properties”窗口中单击 “Exclude”，然后选择要排除的列。

❖ **重用映射定义**

可将字符映射定义保存到文件，以便可在其它项目中进行重用。若要将字符映射保存到定义文件，并在其它项目中进行重用，请执行下列操作：

- 单击工具栏上的 “Export character mapping”和 “Import character mapping”图标。有关详细信息，请参见导出映射定义和导入映射定义。

❖ **导出映射定义**

1. 在 Character Mapper 组件窗口中，单击工具栏上的 “Export character mapping”图标。
2. 提供文件名，然后单击 “Save”。

注释 Character Mapper 保存的文件不带扩展名。

❖ **导入映射定义**

1. 在 Character Mapper 组件窗口中，单击工具栏上的 “Import character mapping”图标。
2. 选择要导入的文件，然后单击 “Open”。

Character Mapper 演示

Sybase ETL 为 Character Mapper 组件包含了若干个演示。这些演示在“Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*”|“*Demonstrations*”|“*Transform*”，然后选择 “*Character Mapper*”。

Demo Repository

在 “Navigator”中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。然后，选择 “Demo Character Mapper”。

查找组件

查找操作用于在包含一组键和值对的查找表中查找与键对应的值。可在项目执行过程中对静态查找表进行高速缓存，也可在不进行高速缓存的情况下执行动态查找。

组件	说明
DB Lookup	查找数据库中的值。查找数据由正好返回两列的查询的结果集来指定：查找键和查找值。 您可以将查找返回的值 （查找值）赋给当前记录中的任何属性。在项目执行过程中，将对查找表进行高速缓存。在项目执行过程中对基础数据库应用的更改不会影响查找结果。
DB Lookup Dynamic	使用在其 WHERE 子句中引用键值的查询执行动态查找。与 “DB Lookup”组件不同，“DB Lookup Dynamic”不对查找信息进行高速缓存，它针对通过组件的每个记录执行一次 SQL 查找。 在项目执行过程中，查找表的数据可能会被并发数据库用户 （甚至是同一项目内的用户）修改。在这种情况下，执行非动态数据库查找时可能会搜索无效数据。使用 “DB Lookup Dynamic”可确保找到当前值。

DB Lookup

DB Lookup 可查找数据库中的值。查找数据由正好返回两列（查找键和查找值）的查询的结果集来指定。

您可以将查找返回的值 （查找值）赋给当前记录中的任何属性。在项目执行过程中， DB Lookup 将对查找表进行高速缓存。在项目执行过程中对基础数据库应用的更改不会影响查找结果。

配置 DB Lookup 组件

1　将 DB Lookup 组件拖动到 “Design”窗口。

2　在 “Design”窗口中，将 DB Lookup 的输入端口与提供入站数据的组件的输出端口相连。

3　在 “Properties”窗口中，指定有效的接口和主机名。

有关特定字段要求，请参见第 113 页上的 “DB Lookup 属性列表”。

4　在 “Properties”窗口中，指定 “Key Attribute”和 “Value Attribute”。这些就是查找值。

5　在 “Properties”窗口中，单击 “Query”图标以打开 “Query”窗口。

6　在 “Query”窗口中，创建并保存用来检索查找数据的查询。将查询设计为从源表返回查找键和查找值。

示例

假定您想将德国产品所用的产品编号替换为在美国使用的产品编号。德国产品位于表 PRODUKTE(PR_NUMMER, PR_NAME, PR_PREIS) 中。DB Lookup 组件的输入端口包含以上三个属性。

用于查找美国产品编号的表是 LOOKUP_PRODUCTS(SOURCE, DESTINATION)。SOURCE 列包含德国产品编号，而 DESTINATION 列包含美国产品编号。

如果在 LOOKUP_PRODUCTS 中找不到与德国的 PR_NUMMER 对应的值，则当前 PR_NUMMER 将被字符串 “INVALID”所替换。如果查找成功，将会使用相应的美国产品编号替换德国产品编号。

若要针对本示例设置 DB Lookup 组件，请选择：

- Key Attribute：PR_NUMMER
- Value Attribute：PR_NUMMER
- Default Value：INVALID
- Query：select SOURCE, DESTINATION FROM LOOKUP_PRODUCTS

DB Lookup 属性列表

DB Lookup 属性列表可标识您在 “Database Configuration”窗口中定义的连接参数和其它属性。必要属性以**粗体**标记。

必要属性

属性	说明
Key Attribute	从输入端口属性列表中选择一个 “Key Attribute”。此属性与查找表的第一列对应。
Value Attribute	选择用于接收查找从 “Value”属性列表中找到的值的属性。返回的查找值将覆盖现有的任何值。 “Key Attribute”和 “Value Attribute”可能引用记录结构的同一个属性，因此，键可以被其相应的值所覆盖。
Interface	指定要用于连接到数据源的方法或驱动程序。
Host Name	指定数据源。“Host Name”列表中的可用选项取决于所选的接口。
Query	单击 “Query”图标可创建从数据源检索信息的查询。可以使用 “Query”窗口创建简单查询，也可以单击 “Query Designer”图标打开 Query Designer 进行创建。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。
Default Value	指定要赋给值属性的缺省值。如果 DB Lookup 在查找表中找不到键值，将会使用该值。
Use Key Value	选择该选项可将键值而不是缺省值赋给值属性 （如果查找失败）。
Lookup Empty/Null Keys	选择该选项可查找空键值或 NULL 键值。否则，将应用选定的缺省方法。
Lookup Size	指定查找记录的估计数，以优化内存分配和查找性能。
Database	指定要用作数据源的数据库。此外，必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。

属性	说明
Standardize Data Format	选择该选项会将传入的 DATE 和 NUMBER 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：YYYY-MM-DD hh:mm:ss.s。 例如， 2005-12-01 16:40:59.123 数字转换后以 “.” 作为小数分隔符。
Database Options	单击 “Database Options” 图标可设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

DB Lookup 演示

Sybase ETL 为 DB Lookup 组件包含了若干个演示。这些演示在 “Help” 菜单上作为 “Flash Demos” 提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Lookup*”，然后选择 “*DB Lookup*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”。然后选择：

- Demo DB Lookup
- Demo Transfer German Products

DB Lookup Dynamic

DB Lookup Dynamic 使用在其 WHERE 子句中引用键值的查询执行动态查找。与 “DB Lookup” 组件不同，“DB Lookup Dynamic” 不对查找信息进行高速缓存，它针对通过组件的每个记录执行一次 SQL 查找。

在项目执行过程中，查找表的数据可能会被并发数据库用户 （甚至是同一项目内的用户）修改。在此类情况下，执行非动态数据库查找时可能会搜索无效数据。使用 “DB Lookup Dynamic” 可确保找到当前值。

对于本组件，另外一种典型用例就是超出本地计算机上可用内存量的查找表。由于使用 DB Lookup Dynamic 组件时，查找速度较为缓慢，但不需要高速缓存内存，因为查找是逐记录执行的。

配置 DB Lookup Dynamic 组件

1 将 DB Lookup Dynamic 组件拖动到 “Design”窗口。

2 在 “Design”窗口中，将 DB Lookup 的输入端口与提供入站数据的组件的输出端口相连。

3 在 “Properties”窗口中，指定接口和主机名。

有关特定字段要求，请参见第 117 页上的 “DB Lookup Dynamic 属性列表”。

4 指定 “Key Attribute”和 “Value Attribute”。这些就是查找值。

5 单击 “Query”图标以打开 “Query”窗口。

6 在 “Query”窗口中，创建并保存用来检索查找数据的查询。将查询设计为从源表返回查找值。

❖ **重置查找键值**

将组件的 “Key Attribute”和 “Value attribute”属性值重置为缺省值：

1 右键单击 “DB Lookup Dynamic”组件。

2 选择 “Reset Lookup Key value”。

示例

假定您想将德国产品所用的产品编号替换为在美国使用的产品编号。德国产品位于表 PRODUKTE(PR_NUMMER, PR_NAME, PR_PREIS) 中。因此， DB Lookup Dynamic 组件的输入端口包含这三个属性。

用于查找美国产品编号的表为 LOOKUP_PRODUCTS(SOURCE, DESTINATION) 表。SOURCE 列包含德国产品编号，而 DESTINATION 列包含美国产品编号。

如果在 LOOKUP_PRODUCTS 中找不到与德国的 PR_NUMMER 对应的值，则当前的 PR_NUMMER 将被字符串 “INVALID”所替换。如果查找成功，将会使用相应的美国产品编号替换德国产品编号。

若要针对本示例设置 DB Lookup Dynamic 组件，请选择：

- Key Attribute：PR_NUMMER
- Value Attribute：PR_NUMMER
- Default Value：INVALID
- Query：select DESTINATION FROM LOOKUP_PRODUCTS, where SOURCE = ‘[Lookup]’

DB Lookup Dynamic 属性列表

下表列出了 DB Lookup Dynamic 组件的必要属性和可选属性。必要属性以**粗体**文本显示。

必要属性

属性	说明
Key Attribute	从输入端口属性列表中选择一个 “Key Attribute”。此属性与查找表的第一列对应。
Value Attribute	选择用于接收查找从 “Value” 属性列表中找到的值的属性。返回的查找值将覆盖现有的任何值。 “Key Attribute” 和 “Value Attribute” 可能引用记录结构的同一个属性，因此，键可以被其相应的值所覆盖。
Interface	指定要用于连接到数据源的方法或驱动程序。
Host Name	指定数据源。 “Host Name” 列表中的可用选项取决于所选的接口。 如果使用 SQLite 作为主机，则键入 SQLite 数据库文件的路径和文件名 （例如 *c:\mySQLite.db*）。如果具有指定名称的数据库不存在， SQLite 将创建该数据库。
Query	单击 “Query” 图标可创建从数据源检索信息的查询。可以使用 “Query” 窗口创建简单查询，也可以单击 “Query Designer” 图标以打开 Query Designer 进行创建。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。
Default Value	指定要赋给值属性的缺省值 （如果在查找表中找不到键值）。

属性	说明
Use Key Value	选择该选项可将键值而不是缺省值赋给值属性 （如果查找失败）。
Lookup Empty/Null Keys	选择该选项甚至可查找空键值或 NULL 键值。如果未选择该选项，将应用缺省方法。
Lookup Key Value	为查找键指定值。
Database	指定要用作数据源的数据库。此外，必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	选择该选项会将传入的 DATE 和 NUMBER 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：YYYY-MM-DD hh:mm:ss.s。 例如， 2005-12-01 16:40:59.123 数字转换后以 “.” 作为小数分隔符。
Database Options	单击 “Database Options” 图标可设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

DB Lookup Dynamic 演示

Sybase ETL 为 DB Lookup Dynamic 组件包含了若干个演示。这些演示在 “Help” 菜单上作为 “Flash Demos” 提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Lookup*”，然后选择 “*DB Lookup – Dynamic*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”，然后选择 “Demo DB Lookup Dynamic”。

Staging 组件

Staging 组件至少拥有一个输入端口和一个输出端口，可对转换流中的数据应用特定转换。

组件	说明
DB Staging	将传入数据流装载到单个 staging 区域中。DB Staging 可为所有传入数据提供缓冲，然后创建表示给定 select 语句结果集的传出数据流。

DB Staging

DB Staging 是一种 Staging 组件，用于将传入数据流装载到单个 staging 区域中。DB Staging 可为所有传入数据提供缓冲，然后创建表示给定 select 语句结果集的传出数据流。

您可以根据前一组件的输出端口结构创建 staging 表。虽然很多转换组件都是基于逐记录处理的原理设计的，但是 staging 组件的工作可以分为下面两个阶段：

- 阶段 1：从前面的组件收集 “所有”记录。
- 阶段 2：运行查询并以给定大小块的形式提供结果集的记录。

通过在 Query 属性中使用 ORDER BY 或 GROUP BY 子句，可以使用 staging 组件对以前未排序的记录执行排序或集合操作。异构源中的数据可以通过装载到 staging 数据库的多个表中的方式进行连接。您还可以使用本组件创建转换的中间图像，以供进一步检查或处理。

对模拟序列的影响
DB Staging 组件首先从原始数据源检索所有数据，然后用作后续组件的新数据源，因此会影响模拟流。该组件允许覆盖原始源组件的 “Read Block Size”值。

配置 DB Staging 组件

1　将 DB Staging 组件拖动到 “Design”窗口。

2　将 DB Staging 的输入端口连接到提供入站数据的组件。

若要向 DB Staging 组件添加输入流，可以在该 staging 组件上断开与数据提供组件的连接。此时，该组件将自动创建端口。

3　在 “Properties”窗口中，将 “Connection Parameters”添加到要在其中添加 staging 表的数据库。

指定用于创建连接的有效接口和主机名。有关特定字段要求，请参见第 120 页上的 “DB Staging 属性列表”。

注释 如果要使用的 staging 表已经存在，请跳过下一步。

4 在 “Design”窗口中，右键单击 DB Staging 组件，然后选择以下选项之一：

- Create Staging Table from Input。
- Create Staging Table from Port。

5 如果已选择 “Create Staging Table from Input”这一选项，请选择相应的端口结构，然后单击 “OK”。为新表输入名称。

如果已选择 “Create Staging Table from Port”这一选项，请为新表输入名称，然后选择相应的端口结构。单击 “Apply”。

6 在 “Add table”窗口中，验证新表中包含的信息是否正确，然后单击 “Create”。

7 在 “Properties”窗口中，单击 “Stage Options”图标以打开 “Stage Options”窗口。

使用 “Stage Options”窗口，还可以为每个 staging 表定义 “Truncate Table”和 “Write Block Size”选项。

8 在 “Properties”窗口中，单击 “Query”图标，然后创建用于从 staging 区域中选择数据的查询。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1 右键单击 “DB Staging”组件。

2 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

DB Staging 属性列表

下表列出了 Data Staging 组件的必要属性和可选属性。必要属性以**粗体**文本显示。

必要属性

属性	说明
Interface	指定要用于连接到数据源的方法或驱动程序。
Host Name	指定数据源。“Host Name”列表中的可用选项取决于所选的接口。 如果使用 SQLite 作为主机，则键入 SQLite 数据库文件的路径和文件名（例如 *c:\mySQLite.db*）。如果具有指定名称的数据库不存在， SQLite 将创建该数据库。
Stage Options	单击“Stage Options”图标将设置暂存选项。
Query	单击“Query”图标可创建从数据源检索信息的查询。您可以使用“Query”窗口创建简单查询，也可以单击“Query Designer”图标生成查询。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。
Read Block Size	确定组件在单个步骤中检索的记录数。
Pre-processing SQL	单击“Pre-processing SQL”图标可创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击“Post-processing SQL”图标可创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Database	指定要用作数据源的数据库。此外，必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	选择该选项会将传入的 `DATE` 和 `NUMBER` 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：`YYYY-MM-DD hh:mm:ss.s`。 例如， `2005-12-01 16:40:59.123` 数字转换后以“.”作为小数分隔符。

属性	说明
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode”选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。如果指定 “00:00:00.000”作为时间参数，则会在项目启动时立即获取排它锁。
Database Options	单击 “Database Options”图标可设置覆盖性能缺省值和控制某些事务行为的选项。有关这些属性的列表，请参见相关内容。 请参见第 80 页上的 “数据库连接设置”。

DB Staging 演示

Sybase ETL 包括 DB Staging 组件的若干演示。这些演示在“Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations* | *Staging*”，选择 “*DB Staging*”。

Demo Repository

在 “Navigator”中选择 “*Repository*” | “*TRANSFORMER.transformer. Repository*” | “*Projects*”，然后选择 “Demo DB Staging”。

目标组件

目标组件 （也称作数据接收器）用于将数据写入特定目标。此类组件有一个输入端口，但没有输出端口。

组件	说明
DB Data Sink Insert	向数据库表添加记录。可以排除属性或赋予缺省值以确定插入表中的记录。 使用此组件可将来自组件输入端口的所有记录添加到数据库表。
DB Data Sink Update	更新或覆盖与所选键匹配的所有记录。此组件不会插入新记录。 此组件用于更新现有记录，而不是插入新记录。
DB Data Sink Delete	DB Data Sink Delete 从目标表删除与所选键的传入值匹配的记录。 使用此组件可从目标表删除记录。
Text Data Sink	Text Data Sink 以分隔格式或固定长度格式将转换结果写入文本文件。
DB Bulk Load Sybase IQ	DB Bulk Load Sybase IQ 对 Sybase IQ 表执行批量操作。 使用该组件可以根据组件输入端口中的记录来操作 Sybase IQ 数据库中的表记录。

DB Data Sink Insert

DB Data Sink Insert 是将来自输入端口的记录添加到数据库表的目标组件。可以排除属性或赋予缺省值以确定插入表中的记录。

配置 DB Data Sink Insert 组件

1　将 DB Data Sink Insert 组件拖到 “Design” 窗口上。

2　在 “Database Configuration” 窗口中，为目标数据库添加连接参数。

3　有关特定字段要求，请参见第 125 页上的 “DB Data Sink Insert 属性列表”。

4　选择或输入目标表。

可以写入现有表，也可以基于项目中的现有端口添加表。如果要基于现有端口结构添加表，请跳过此步骤。有关其它信息，请参见第 124 页上的 “添加目标表”。

5　单击 “完成”。

6　在 “Properties” 窗口中指定所有其它可选属性。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1 右键单击 “DB Data Sink Insert”组件。

2 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

❖ **装载输入端口中的数据**

将输入端口当前可用的数据装载到数据库或文本文件：

1 右键单击 “DB Data Sink Insert”组件。

2 选择 “Flush Buffer”。

“Flush Buffer”选项将缓冲行写出到目标。已为其指定写入块大小的所有组件均可缓冲数据，直到达到写入块大小为止。在模拟期间的任意时间，如果缓冲区内含有数据，“Flush Buffer”选项还会显示尚未写入到目标的行数。如果缓冲区为空，该选项则为灰显。

添加目标表

可以将转换结果写入现有表，也可以基于项目中的现有端口添加目标表。在 “Database Configuration”窗口或 “Properties”窗口中，无法基于端口结构添加表。必须在 “Design”窗口中选择端口结构。

❖ **写入现有表**

1 在 “Database Configuration”窗口中，指定要在其中写入转换结果的表。可以单击 “Destination Table”图标选择表，也可以在 “Destination Table”字段中手动输入表的名称。

2 单击 “Finish”。

❖ **基于输入端口添加目标表**

1 在 “Design”窗口中，右键单击该组件，然后选择 “Add Destination Table from Input”。

2 为新表命名。单击 “OK”。

3 在 “Add table”窗口中验证表信息是否正确，然后单击 “Create”。

❖ **从现有端口添加目标表**

1 在 “Design”窗口中，右键单击该组件，然后选择 “Add Destination Table from port”。

2 为新表命名。单击 “OK”。

3　选择要将其结构指定给新表的端口。单击 “Apply”。

4　在 “Add table”窗口中验证表信息是否正确，然后单击 “Create”。

注释 您也可以使用自己的工具箱创建目标表，或从 “Properties”窗口选择现有表。

DB Data Sink Insert 属性列表

下表列出了 DB Data Sink Insert 组件的必要属性和可选属性。必要属性以**粗体**文本显示。

必要属性

属性	说明
Interface	指定要用于连接到数据源的方法或驱动程序。
Host Name	指定数据源。“Host Name”列表中的可用选项取决于所选的接口。 如果使用 SQLite 作为主机，则键入 SQLite 数据库文件的路径和文件名 （例如 *c:\mySQLite.db*）。如果具有指定名称的数据库不存在， SQLite 将创建该数据库。
Destination Table	单击 “Destination Table”图标可以从一组现有表中选择目标表，也可以手动输入目标表。 此外，还可以基于组件的端口结构创建新的目标表。请参见第 124 页上的 “添加目标表”。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。
Insert Options	单击 “Insert Options”图标可确定将如何插入属性。“Include”列指定将插入哪些属性。SQL 表达式将用执行时的表达式值覆盖相应属性值。 为将覆盖所选属性的传入值的每个属性指定 INSERT 语句。SQL INSERT 值子句的示例如下： `'valid' '[uDate("now")]'` 将使用在 “Insert Options”窗口中选择的所有属性插入记录。选择或取消选择任何属性。可以指定将插入的属性的值，这将覆盖相应的传入属性值。

属性	说明
Truncate Table	选择该选项可在开始进行转换时从目标表删除所有记录。
Write Block Size	指定在单项写入操作中要写入文件或管道的记录数。
Pre-processing SQL	单击 “Pre-processing SQL” 图标可创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击 “Post-processing SQL” 图标可创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Opening Attribute Quote	指定 SQL 语句中属性名的前缀。
Closing Attribute Quote	指定 SQL 语句中属性名的后缀。
Database	选择要用作数据源的数据库。此外，必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	选择该选项会将传入的 DATE 和 NUMBER 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：YYYY-MM-DD hh:mm:ss.s。 例如， 2005-12-01 16:40:59.123 数字转换后以 “.” 作为小数分隔符。
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode” 选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。如果指定 “00:00:00.000” 作为时间参数，则会在项目启动时立即获取排它锁。
Database Options	单击 “Database Options” 图标可设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

DB Data Sink Insert 演示

Sybase ETL 包括 DB Data Sink Insert 组件的多个演示。这些演示在“Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*”|“*Demonstrations*”|“*Destination*”，选择 “*DB Data Sink - Insert*”。

Demo Repository

在 “Navigator”中选择 “*Repository*”|“*TRANSFORMER.transformer.Repository*”|“*Projects*”。然后选择：

- Demo Transfer German Customers
- Demo Transfer German Products
- Demo Transfer German Sales
- Demo Transfer U.S. Customers
- Demo Transfer U.S. Products

DB Data Sink Update

DB Data Sink Update 是更新或覆盖与所选键匹配的所有记录的目标组件。此组件不会插入新记录。如果没有任何匹配记录，则 DB Data Sink Update 不显示错误消息。

此组件用于更新现有记录，而不是插入新记录。

注释　如果更新值违背基础表或对象的限制 （如约束、参照完整性或唯一索引定义），将显示错误消息。所选 Key Value 属性与任何现有索引定义无关。

配置 DB Data Sink Update 组件

1　将 DB Data Sink Update 拖到 “Design”窗口上。

2　在 “Database Configuration”窗口中，为目标数据库添加连接参数。

有关特定字段要求，请参见第 128 页上的 “DB Data Sink Update 属性列表”。

3 指定要在其中写入转换结果的目标表。可以单击 “Destination Table”图标选择表，也可以在 “Destination Table”字段中手动输入表的名称。

可以写入现有表，也可以基于项目中的现有端口添加表。如果要基于现有端口结构添加表，请跳过此步骤。有关其它信息，请参见第 128 页上的 “添加目标表”。

4 单击 “完成”。

5 在 “Properties”窗口中，单击 “Key”图标以选择用来标识要更新的记录的目标表列。

必须先指定目标表才能选择键，可以选择多个键列。这是逻辑选择操作，与数据库模式中的任何基础索引无关。

6 在 “Properties”窗口中指定所有其它可选属性。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1 右键单击 “DB Data Sink Update”组件。

2 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

❖ **装载输入端口中的数据**

将输入端口当前可用的数据装载到数据库或文本文件：

1 右键单击 “DB Data Sink Update”组件。

2 选择 “Flush Buffer”。

“Flush Buffer”选项将缓冲行写出到目标。已为其指定写入块大小的所有组件均可缓冲数据，直到达到写入块大小为止。在模拟期间的任意时间，如果缓冲区内含有数据，“Flush Buffer”选项还会显示尚未写入到目标的行数。如果缓冲区为空，该选项则为灰显。

添加目标表

可以将转换结果写入现有表，也可以基于项目中的现有端口添加目标表。有关如何添加目标表的信息，请参见第 124 页上的 “添加目标表”。

DB Data Sink Update 属性列表

下表列出了 DB Data Sink Update 组件的必要属性和可选属性。必要属性以**粗体**文本显示。

必要属性

属性	说明
Interface	指定要用于连接到数据源的方法或驱动程序。
Host Name	指定数据源。“Host Name”列表中的可用选项取决于所选的接口。 如果使用 SQLite 作为主机，则键入 SQLite 数据库文件的路径和文件名 （例如 *c:\mySQLite.db*）。如果具有指定名称的数据库不存在， SQLite 将创建该数据库。
Destination Table	单击 “Destination Table”图标可以从一组现有表中选择目标表。 此外，还可以基于组件的端口结构创建新的目标表。请参见第 128 页上的 “添加目标表”。
关键字	选择用来标识要更新的记录的目标表列。 必须先选择目标表，才能选择键，可以选择多个键列。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。

Update Options	单击 “Update Options”图标可选择要包括在更新中的属性 （未列出键属性）。缺省情况下，会选择所有属性。取消选择不想更新的那些属性。
	在 SQL UPDATE SET 子句列中，可以用新值覆盖传入属性的值。新值可以是常量，也可以是表达式。
	在 SQL 语言表示法中，列的内容将处理为：
	`UPDATE customers`
	`SET cu_createdate = '2005-02-26'`
	`WHERE …`.
	可以使用基础数据库的 SQL 语言中允许的任何表达式。中括号中的动态表达式将在组件初始化期间进行计算。

Write Block Size	指定在单项写入操作中要写入文件或管道的记录数。
Pre-processing SQL	单击 “Pre-processing SQL”图标将打开一个窗口，您可在其中创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。

属性	说明
Post-processing SQL	单击 “Post-processing SQL” 图标将打开一个窗口，您可在其中创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Opening Attribute Quote	指定 SQL 语句中属性名的前缀。
Closing Attribute Quote	指定 SQL 语句中属性名的后缀。
Database	指定要用作数据源的数据库。 此外，必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	选择该选项会将传入的 DATE 和 NUMBER 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：YYYY-MM-DD hh:mm:ss.s。 例如， 2005-12-01 16:40:59.123 数字转换后以 “.” 作为小数分隔符。
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode” 选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。如果指定 “00:00:00.000” 作为时间参数，则会在项目启动时立即获取排它锁。
Database Options	单击 “Database Options” 图标可设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

DB Data Sink Update 演示

Sybase ETL 包括 DB Data Sink Update 组件的多个演示。这些演示在 “Help” 菜单上作为 “Flash Demos” 提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Destination*”，选择 “*DB Data Sink - Update*”。

Demo Repository

在 “Navigator” 中选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”，然后选择 “Demo DB Datasink Update”。

DB Data Sink Delete

DB Data Sink Delete 是从数据库目标表中删除与所选键传入值匹配的记录的目标组件。如果没有任何匹配记录，则 DB Data Sink Delete 不显示错误消息。

配置 DB Data Sink Delete 组件

1. 将 DB Data Sink Delete 组件拖到 “Design” 窗口上。
2. 在 “Database Configuration” 窗口中，为目标数据库添加连接参数。指定有效接口和主机名。

   有关特定字段要求，请参见第 132 页上的 “DB Data Sink Delete 属性列表”。
3. 指定要在其中写入转换结果的表。可以单击 “Destination Table” 图标选择表，也可以在 “Destination Table” 字段中手动输入表的名称。

   可以写入现有表，也可以基于项目中的现有端口添加表。如果要基于现有端口结构添加表，请跳过此步骤。请参见第 132 页上的 “添加目标表”。
4. 单击 “完成”。
5. 在 “Properties” 窗口中，单击 “Key” 图标以选择用来标识要从目标表删除的记录的列。

   必须先选择目标表，才能选择键，可以选择多个键列。这是逻辑选择操作，与数据库模式中的任何基础索引无关。
6. 在 “Properties” 窗口中指定所有其它可选属性。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1. 右键单击 “DB Data Sink Delete” 组件。
2. 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

❖ **装载输入端口中的数据**

将输入端口当前可用的数据装载到数据库或文本文件：

1 右键单击 “DB Data Sink Delete”组件。

2 选择 “Flush Buffer”。

“Flush Buffer”选项将缓冲行写出到目标。已为其指定写入块大小的所有组件均可缓冲数据，直到达到写入块大小为止。在模拟期间的任意时间，如果缓冲区内含有数据，“Flush Buffer”选项还会显示尚未写入到目标的行数。如果缓冲区为空，该选项则为灰显。

添加目标表

可以将转换结果写入现有表，也可以基于项目中的现有端口添加目标表。有关如何添加目标表的信息，请参见第 124 页上的 “添加目标表”。

DB Data Sink Delete 属性列表

下表列出了 DB Data Sink Delete 组件的必要属性和可选属性。必要属性以**粗体**文本显示。

必要属性

属性	说明
Interface	标识要用于连接到数据源的方法或驱动程序。
Host Name	标识数据源。“Host Name”列表中的可用选项取决于所选的接口。 如果使用 SQLite 作为主机，则键入 SQLite 数据库文件的路径和文件名（例如 *c:\mySQLite.db*）。如果具有指定名称的数据库不存在， SQLite 将创建该数据库。
Destination Table	单击 “Destination Table”图标可打开一个窗口，在此窗口中可以从一组现有表中选择目标表。 此外，还可以基于组件的端口结构创建新的目标表。有关详细信息，请参见第 132 页上的 “添加目标表”。
Key	选择用来标识要删除的记录的目标表列。 必须先选择目标表，才能选择键，可以选择多个键列。

可选属性

属性	说明
User and Password	二者组合用于标识已授权的数据库用户，以及保护数据库免受未经授权的访问的侵害。
Write Block Size	指定在单项写入操作中要写入文件的记录数。
Pre-processing SQL	单击 “Pre-processing SQL” 图标将打开一个窗口，您可在其中创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击 “Post-processing SQL” 图标将打开一个窗口，您可在其中创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Opening Attribute Quote	SQL 语句中属性名的前缀。
Closing Attribute Quote	SQL 语句中属性名的后缀。
Database	标识要用作数据源的数据库。 如果选择此选项，则必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	将传入的 DATE 和 NUMBER 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：YYYY-MM-DD hh:mm:ss.s。 例如， 2005-12-01 16:40:59.123 数字转换后以 “.” 作为小数分隔符。
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode” 选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。

属性	说明
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。 如果指定 “00:00:00.000”作为时间参数，则会在项目启动时立即获取排它锁。
Database Options	单击 “Database Options”图标将打开一个窗口，您可在其中设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

DB Data Sink Delete 演示

Sybase ETL 包括 DB Data Sink Delete 组件的多个演示。这些演示在 “Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击 “*Help*” | “*Demonstrations*” | “*Destination*”，选择 “*DB Data Sink - Delete*”。

Demo Repository

在 “Navigator”中，选择 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Projects*”，然后选择 “Demo DB Datasink Delete”。

Text Data Sink

Text Data Sink 是以分隔格式或固定长度格式将转换结果写入文本文件的目标组件。

配置 Text Data Sink 组件

1　将 Text Data Sink 组件拖动到 “Design” 窗口中。

2　当将 Text Data Sink 组件添加到项目中时，Text Data Sink 组件的输出端口应链接到该组件的用来提供入站数据的输入端口。

有关特定字段要求，请参见第 137 页上的 “Text Data Sink 属性列表”。另请参见：

- 导出和导入文件定义 – “Component” 窗口中的 “Export” 和 “Import” 选项可用于将文件属性保存到定义文件，并可将这些属性重用于其它组件。
- 修改端口结构 （分隔文件） – 分隔文件的列值反映了当前输入端口结构。您可以基于另一个端口分配端口结构，也可以重新创建当前端口结构。
- 使用固定长度文件 – 如果使用固定长度文件类型，则必须创建列，并且为每个列提供位置参数。

3　单击 “Save”。

4　在 “Properties” 窗口中指定所有其它可选属性。

注释　如果 “Design” 窗口中存在一个相邻组件，则 Sybase ETL 会自动在 Text Data Sink 组件的输入端口和输出端口之间创建一个链接。对于分隔文件，此链接提供了在打开该窗口时所看到的初始端口结构。

如果不存在，则您可能需要关闭 “Design” 窗口并将 Text Data Sink 的输入端口与一个相邻组件的输出端口相连接。

注释　Text Data Sink 不会对模拟序列产生任何影响。

❖ **装载输入端口中的数据**

将输入端口当前可用的数据装载到数据库或文本文件：

1　右键单击 “Text Data Sink” 组件。

2　选择 “Flush Buffer”。

“Flush Buffer” 选项将缓冲行写出到目标。已为其指定写入块大小的所有组件均可缓冲数据，直到达到写入块大小为止。在模拟期间的任意时间，如果缓冲区内含有数据，“Flush Buffer” 选项还会显示行数。如果缓冲区为空，该选项则为灰显。

导出和导入文件定义

“Component”窗口中的 “Export”和 “Import”选项可用于将文件属性保存到定义文件，并可将这些属性重新用于其它组件。 “Export”选项可将组件属性保存到定义文件中。 “Import”选项可装载您使用 Export 命令创建的定义文件。

❖ **导出文件定义**

1 双击 “Text Data Sink”打开 “Component”窗口。
2 单击 “Properties”|“Export”。
3 选择要使用的定义文件。

❖ **导入文件定义**

1 双击 “Text Data Sink”打开 “Component”窗口。
2 单击 “Properties”|“Import”。
3 选择要使用的定义文件。

修改端口结构 （分隔文件）

“Text Data Sink Components”窗口的列值反映了当前输入端口结构。您可以分配新的端口结构，也可以重新创建当前端口结构。

❖ **分配新的端口结构**

1 双击 “Text Data Sink”打开 “Component”窗口。
2 单击 “Column Names”面板中的 “Assign Port Structure”图标。
3 选择要分配其结构的端口。

❖ **重新生成列定义**

1 双击 “Text Data Sink”打开 “Component”窗口。
2 右键单击 “Column Names”面板中的 “Regenerate the column definition”图标。

使用固定长度文件

对于固定长度文件类型，您必须创建列并提供每个列的位置参数。

❖ **将列添加到输出**

1 双击 “Text Data Sink”打开 “Component”窗口。
2 单击 “Column Names”面板中的 “Insert a New Attribute”图标。您可以为已生成的列编辑名称。

❖ **从输出中删除列**

1　双击 “Text Data Sink”打开 “Component”窗口。

2　选择相应列并单击 “Remove an attribute”图标。

Text Data Sink 属性列表

下表列出了 Text Data Sink 组件的必要属性和可选属性。必要属性标签以**粗体**文本显示。

必要属性

属性	说明
Text Destination	指定输出文件。 当向项目添加 Text Data Sink 组件时，该组件会提示您指定目标文件。若要指定目标文件，请单击 “from the Properties”窗口中的 “Destination File”图标，然后选择现有文件，或在项目执行期间键入完整路径和文件名以创建一个文件。
Columns	单击 “Columns”图标以为源文件中数据定义列。如果属性值不为空，则列值会反映您在 “Component”窗口中定义的端口结构或属性值。

可选属性

属性	说明
Row Delimiter	选择每行的分隔方式： • Position （固定行位置） • LF （换行符） • CR （回车符） • CRLF （回车符加换行符） 也可以输入其它分隔符。
Row Length	指定每个固定行中的字符数，前提是已选择 “Position”作为 “Row Delimiter”。
Column Delimiter	选择列的分隔方式： • Position （固定列位置） • Tab • Comma • Semicolon 也可以输入其它分隔符。

属性	说明
Column Quote	指定将输出文件中的值引起来的方式 （仅限分隔文件）： • 无 • 单引号 • 双引号
Fixed by Bytes	指定如何解释为行长度、列开头和列结尾提供的值： • Not selected – 将值解释为字符数。这是缺省值。 • Selected – 将值解释为字节数。 示例 – 假定源文件包含二进制数据：**0x61 62 63 d6 d0 ce c4 61 62 63 64 65**，并且具有以下特性： • 文件类型 – 固定 （可变行） • 编码 – GB2312 • 行分隔符 – “\n” • 列定义 – 列 1：1-7 ；列 2：9-10 如果选择了 “Fixed by Bytes”选项： • 列 1 显示前 7 个字节，其二进制数据为 **0x61 62 63 d6 d0 ce c4**。 • 列 2 显示第 9 和第 10 个字节，其二进制数据为 **0x62 63**。 如果未选择 “Fixed by Bytes”选项： • 列 1 显示前 7 个字符，其二进制数据为 **0x61 62 63 d6d0 cec4 61 62**。 • 列 2 显示后两个字符，其二进制数据为 **0x64 65**。 **注释** 在 GB2312 中， 0xd6d0、 c4c4 表示两个中文字符。
Encoding	从下拉菜单中选择适合的值来设置当前字符编码。
Column Header	选择此选项可将列名写入文件。
Header	单击 “Header”图标可创建要写入文件的报告标头。Text Data Sink 会先写入标头，然后写入传入数据。这是一个可选输出属性。 键入要写入标头的标头文本。表达式允许用中括号表示法。
Append Data	选择此选项可将传入数据附加到目标文件。如果未设置此值，则 Text Data Sink 会覆盖目标文件中的任何现有数据。

属性	说明
Write Block Size	指定在单项写入操作中 Sybase ETL 写入文件的记录数。

Text Data Sink 演示

Sybase ETL 包括针对 Text Data Sink 组件的几种演示。这些演示在“Help”菜单上作为 “Flash Demos”提供，在 Demo Repository 中作为样本项目提供。

Flash 演示

若要运行 Flash 演示，请单击“*Help*”|“*Component Demonstrations*”|“*Destination*”，然后选择 “*Text Data Sink*”。

Demo Repository

在“Navigator”中选择 “*Repository*”|“*TRANSFORMER.transformer.Repository*”|“*Projects*”。然后选择：

- Demo XML via SQL Data Provider
- Demo Text Data Sink Delimited/Fixed

DB Bulk Load Sybase IQ

DB Bulk Load Sybase IQ 是对 Sybase IQ 表执行 bulk 操作的目标组件。您可以使用该组件根据组件输入端口中的记录来操作 Sybase IQ 数据库中的表记录。

配置 DB Bulk Load Sybase IQ 组件

1　将 DB Bulk Load Sybase IQ 组件拖动到 “Design”窗口中。

2　将 DB Bulk Load Sybase IQ 组件的输入端口连接到该组件的提供入站数据的输出端口。

3　在 “Database Configuration”窗口，添加 Sybase IQ 连接参数。

4　单击 “Destination”图标并选择要向其装载入站数据的表。如果要写入新的目标表，请参见第 140 页上的 “添加新的 Sybase IQ 目标表”。

5 单击 “Load Stage”图标。您可以：

- 选择或输入要用作临时数据文件的文件名，然后单击 “Save”。

– 或 –

- 在 “Load Stage”字段中添加管道名 （语法：`pipe://<name>`）。

有关特定字段要求，请参见第 145 页上的 “DB Bulk Load Sybase IQ 属性列表”。

6 单击 “完成”。

要将 DB Bulk Load IQ 添加到项目中，请确保：

- 必须先启动 Sybase IQ 并且确保其处于运行状态，然后才能将 DB Bulk Load IQ 添加到项目中。如果在同一计算机上同时运行 ETL Server 和 IQ 数据库，则可提高性能。这不是必需的。
- 若要连接到 IQ 中的目标数据库，需要选择一个有效的主机名和接口。有关特定字段要求，请参见第 145 页上的 “DB Bulk Load Sybase IQ 属性列表”。
- 如果要将数据装载到新的 IQ 表中，您可以基于 DB Bulk Load 的输入端口或项目中任何可用端口的结构来创建目标表。请参见第 140 页上的 “添加新的 Sybase IQ 目标表”。
- 如果要自定义脚本，请右键单击 “DB Bulk Load IQ”组件，然后选择 “Generate Load Script”。单击 “Properties”窗口中的 “Load Script”图标，编辑并保存您的脚本。

❖ **更新端口结构**

根据对数据库所做的更改更新端口结构：

1 右键单击 “DB Bulk Load IQ”组件。

2 选择 “Reconfigure”。

当数据库模式发生更改时，“Reconfigure”选项会更新组件配置。该选项关闭当前连接，打开数据库的新连接，读取查询元数据，并将更新应用到端口结构。

添加新的 Sybase IQ 目标表

可以将入站数据写入现有表，也可以基于项目中的现有端口添加新的目标表。

❖ **根据输入添加目标表**

基于 DB Bulk Load Sybase IQ 的输入端口创建 IQ 目标表：

1 右键单击 “DB Bulk Load Sybase IQ”组件，然后选择 “Add Destination Table from Input”。

2　为新表输入名称。

3　单击 “OK”。

❖ **从现有端口添加目标表**

基于可用组件端口创建 IQ 目标表：

1　右键单击 “DB Bulk Load Sybase IQ”组件，然后选择 “Add destination table from port”。

2　输入新表的名称，然后单击 “OK”。

3　选择要用于创建表的端口。单击 “Apply”。

启用客户端装载支持

您可以使用该组件将数据从位于其它主机上的文件 （而不是 Sybase IQ）添加到 Sybase IQ 表。您无需在同一计算机上安装 Sybase ETL 和 Sybase IQ； ETL Server 和 Sybase IQ 可以在网络环境中通信，从而使您通过一个步骤即可从远程计算机批量装载。若要支持客户端，请执行下列操作：

- 将 Sybase IQ 15 客户端安装在 ETL Server 所在的相同计算机上。
- 将 Sybase Adaptive Server Anywhere (ASA) 11 ODBC 驱动程序安装在 ETL Development 和 ETL Server 所在的同一计算机上。
- 目标 IQ 数据库版本必须为 Sybase IQ 15。
- 启用 Sybase IQ 15.0 服务器的 allow_read_client_file 和 allow_write_client_file 选项。对各 IQ 服务器一次性设置这些选项。若要设置这些选项，请执行下列操作：

  a　转至 Sybase Central，连接到 Sybase IQ 15.0 服务器。

  b　右键单击 Sybase IQ 服务器的数据库名称，然后选择 “Options”。此时将显示所有数据库选项的列表。

  c　选择 allow_read_client_file 和 allow_write_client_file 选项，并将其值更改为 On。缺省情况下，该值为 Off。

  d　使用 isql 或 dbisql 实用程序启用 allow_read_client_file 服务器选项属性。

```
set option allow_read_client_file=on
GRANT READCLIENTFILE TO <group | user>
```

满足上述前提条件之后，在该组件的 “Properties”窗口中选择 “Use IQ Client Side Load”选项。此外，选择 ODBC 作为接口，否则，在装载远程主机上的数据时可能会遇到错误。

注释 当使用的 ODBC 驱动程序为 IQ 15 ODBC 驱动程序时，客户端装载仅适用于 ODBC。

配置 IQ 多写入器以装载数据

Sybase ETL 支持 Sybase IQ 15 多写入器功能，您可以使用该功能添加多个写入器，以便将数据装载到一个 IQ 数据库中。通过多写入器功能，您可以并行装载多个 Sybase IQ 表，其速度快于顺序装载。您可以在下列情况下使用多写入器功能：

- 从源数据库中选择了多个表，并且打算迁移到目标 IQ 数据库中的多个表。
- 创建的作业具有多个涉及多个表的多项目组件，或者已选择多个链接到并行执行拓扑的项目以执行作业。

若要使用多写入器功能，您必须在目标 IQ 数据库中具有下列权限：

对象名	类型	所需权限
ETL_MULTIPLEX_STATE	表	create
ETL_MULTIPLEX_VERSION	表	create
sp_iqstatistic	Stored Procedure（存储过程）	execute

注释 ETL 在 IQ 数据库中创建两个表：ETL_MULTIPLEX_STATE 和 ETL_MULTIPLEX_VERSION。ETL_MULTIPLEX_STATE 表中的每一行都表示一个由 ETL 网格节点选择的 IQ 写入器，该行将在每次执行之后自动删除。为了避免网格节点因意外错误而崩溃，*必须* 手动清除该表中的数据。

您可以使用 Sybase Central 来设置所需权限：

1 转至 Sybase Central，连接到 Sybase IQ 15.0 服务器。

2 展开 “Users & Groups”，然后选择要为其设置创建表权限的用户。

3 右键单击该用户，然后选择 “Properties”。

4 选择 “Authorities”选项卡，查看 “Resource”选项，为该用户授予在 IQ 数据库中创建数据库对象的权限。

5 选择 “Permissions”选项卡，然后选择 “Procedures & Functions”选项，以查看所有可用权限的列表。

6　选择 sp_iqstatistics，并单击对应的 “Execute”列，以便为该用户授予在 IQ 数据库中执行存储过程的权限。

7　单击 “OK”保存设置。

注释 若要支持 Multiplex 执行，必须将 Sybase Adaptive Server Anywhere (ASA) 11 ODBC 驱动程序安装在 ETL Development 和 ETL Server 所在的同一计算机上。

您必须在 *IQMultiplex.ini* 文件中定义写入器，以便对这些写入器进行配置，从而用于 Multiplex 执行。

❖ **在 *IQMultiplex.ini* 文件中定义首选写入器**

1　导航至安装文件夹中的 *etc* 目录，使用文本编辑器打开 *IQMultiplex.ini* 文件。

2　为要使用的每个 Multiplex 组添加一个段。缺省情况下，*IQMultiplex.ini* 文件为空。如果未指定任何内容，网格引擎则使用内部缺省值。

下面显示了一个样本段：

```
[dbsybase15+iq15m+sample]
/* 这是段名 */
Enabled=true
Workload=OperationsWaiting
MinimalUpdateInterval=60
SelectWriterTimeout=6
MostIdleNode=select NAME from TEST_CUSTOMER_NODE
order by WORKLOAD asc
MostBusyNode=select NAME from TEST_CUSTOMER_NODE
order by WORKLOAD desc
```

必须提供段名，它包括协调器的接口名称、主机名和数据库名称，这些名称用加号 (+) 分隔开来。不要使用冒号 (:)、哈希 (#) 或等号 (=) 字符。各个组的其它属性为可选属性。请参见第 144 页的表 5-1。

3　保存并关闭文件。

表 5-1：Multiplex 组可选属性

名称	类型	Value	说明
Enabled	boolean	True （缺省值）或 False。	在 IQ 服务器数据库中启用或禁用此功能。
Workload	text	Operations Waiting（缺省值）。	指定应当使用 EXEC sp_iqstatistics 结果中的哪一行来充当工作负荷。 调度程序执行存储过程 EXEC sp_iqstatistics 以查询写入器工作负荷。查询结果集在第二列中返回操作状态名称，并在第四列中返回状态值。 调度程序查找第二列与您指定的 “Workload”选项值相符合的行，并将该行的第四列用作最终工作负荷值。
MinimalUpdate Interval	integer	大于零 (0)。缺省值为 6。	调度程序通过查询协调器来刷新写入器信息的最小时间间隔 （以秒为单位）。
SelectWriter Timeout	integer	大于等于零 (0)。缺省值为 0。	当选择但未释放所有写入器时，调度程序应当等待的秒数。如果指定 0，调度程序将无限期地等待。在出现超时时，系统将生成错误。
MostIdleNode	SQL	缺省为空。	SQL 执行返回的第一行的第一列应当为写入器节点名称。调度程序假定返回的写入器是 multiplex 中最空闲的节点，并将其用作下一表装载请求的目标。 例如，以下 SQL 查询创建自定义调度表： DROP TABLE TEST_CUSTOMER_NODE; CREATE TABLE TEST_CUSTOMER_NODE(NAME varchar(100), WORKLOAD int /* 必须为整数 */); INSERT INTO TEST_CUSTOMER_NODE (NAME,WORKLOAD)VALUES('iq15w1',78); INSERT INTO TEST_CUSTOMER_NODE (NAME,WORKLOAD)VALUES('iq15w2',34); INSERT INTO TEST_CUSTOMER_NODE (NAME,WORKLOAD)VALUES('iq15w3',12); /* iq15w1-w3 为写入器 */ 若要从表中获取最空闲的节点，请在 *IQMultiplex.ini* 文件中添加以下 SQL 查询： Select NAME from TEST_CUSTOMER_NODE order by WORKLOAD asc iq15w3 作为最空闲的节点返回。

名称	类型	Value	说明
MostBusyNode	SQL	缺省为空。	SQL 执行返回的第一行的第一列应当为写入器节点名称。调度程序假定返回的写入器是 multiplex 中最繁忙的节点，并延迟将其用作装载表目标。 例如，以下 SQL 查询创建自定义调度表： DROP TABLE TEST_CUSTOMER_NODE; CREATE TABLE TEST_CUSTOMER_NODE(NAME varchar(100), WORKLOAD int /* 必须为整数 */); INSERT INTO TEST_CUSTOMER_NODE (NAME,WORKLOAD)VALUES('iq15w1',78); INSERT INTO TEST_CUSTOMER_NODE (NAME,WORKLOAD)VALUES('iq15w2',34); INSERT INTO TEST_CUSTOMER_NODE (NAME,WORKLOAD)VALUES('iq15w3',12); /* iq15w1-w3 为写入器 */ 若要从自定义调度表中获取最繁忙的节点，请在 *IQMultiplex.ini* 文件中添加以下 SQL 查询： Select NAME from TEST_CUSTOMER_NODE order by WORKLOAD desc iq15w1 作为最繁忙的节点返回。

DB Bulk Load Sybase IQ 属性列表

DB Bulk Load Sybase IQ 属性列表用于标识在 “Database Configuration” 窗口中定义的连接参数和其它项。

必要属性

属性	说明
Interface	指定要用于连接到数据源的方法或驱动程序。
Host Name	指定 Sybase IQ 目标运行所在的主机。
Destination	单击 “Destination Table” 图标可以从一组现有表中选择目标表。

属性	说明
Load Stage	指定数据文件路径或管道名。Load Stage 文件与 IQ Server 必须位于同一台计算机上。 如果使用 UNIX 或 Linux 中的命名管道，则 ETL Server 和 IQ Server 必须位于同一台计算机上。对于 Windows，这并不是必需的。 **注释** 如果已选择“Use IQ Client Side Load”选项，则不支持命名管道。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。
Key	选择用于标识选定函数的记录的目标键属性。 如果未选择任何键，接口则使用主键信息，该信息从 DB 主机传送。如果未提供有任何主键信息，则会显示错误。

函数	选择下列装载函数之一：
	• Insert （缺省值） – 使用指定文件路径或管道名将记录直接装载到选定目标表中。
	• Upsert – 更新现有记录并插入新记录。当选择“Upsert”时，将替换现有记录，并且不会执行属性级更新。您可以使用 “Key”属性指定目标属性，以标识要更新的记录。
	在下列情况下，将忽略 “Upsert”函数：
	  • 当使用“Generate Load Script”时。生成的 LOAD TABLE 脚本用于 “Insert”函数。
	  • 当选择了 “Truncate”选项时。在装载数据之前，将删除目标表中的所有记录。该组件将忽略 “Upsert”函数，并改为执行 “Insert”函数，这会将所有属性装载到目标表中。
	• Delete – 根据传入数据中的键删除目标表中的记录。您可以使用 “Key”属性指定目标属性，以标识要删除的记录。
	如果选择了 “Delete”函数和 “Truncate”选项，则仅执行预处理和后处理 SQL。

Truncate	选择该选项可将目标表中装载之前的所有记录删除。

属性	说明
Use IQ Client Side Load	选择该选项可以使用 LOAD TABLE 语句将记录从远程主机上的文件添加到 Sybase IQ 表。
Load Script	如果此属性为空，则会在运行时根据组件设置生成 LOAD TABLE 语句。 要使用自定义脚本，请右键单击组件，然后选择“Generate Load Script”。生成脚本后，可以单击“Load Script”并编辑脚本。
Load Stage (Server)	指定数据文件的服务器路径，或在使用管道时将其保留为空。 如果 Sybase IQ 服务器需要使用与 Load Stage 属性中指定的路径不同的临时数据文件的其它路径，则必须在此处输入该路径。
Write Block Size	指定在单项写入操作中要写入文件或管道的记录数。
Pre-processing SQL	单击 “Pre-processing SQL”图标可创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击 “Post-processing SQL”图标可创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Database	指定要用作数据源的数据库。此外，必须选择相应的接口，在某些情况下还必须指定相应的用户 ID 和口令。
Schema	标识要用作数据源的模式 / 所有者。显示的对象将具有相应限制，新表将在该模式下创建。
Standardize Data Format	选择该选项会将传入的 DATE 和 NUMBER 信息转换为 Sybase ETL 可以在支持不同格式的系统之间移动的标准格式。 日期将转换为包括年、月、日、小时、分钟、秒钟和秒钟分数的格式：YYYY-MM-DD hh:mm:ss.s。 例如， 2005-12-01 16:40:59.123 数字转换后以 “.”作为小数分隔符。
Database Options	单击 “Database Options”图标可设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的 “数据库连接设置”。

属性	说明
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode”选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。如果指定“00:00:00.000”作为时间参数，则会在项目启动时立即获取排它锁。
Use IQ Multiplex	选择该选项可以使用多个写入器将数据装载到 IQ 数据库中，从而支持 Multiplex 执行。

DB Bulk Load Sybase IQ 和数据库空间

如果使用 Bulk Load Sybase IQ 组件并且项目或作业的执行需要很长时间，请检查 Sybase IQ 控制台或日志。如果看到 “out of space”消息，则必须添加另一个数据库空间。IQ 消息文件中的消息指示哪个数据库空间的空间不足以及要添加的最小兆字节数。如果插入数据时出现该问题，则可能需要更多的 IQ 存储空间。如果在执行大量排序和合并操作的查询过程中出现该问题，则可能需要更多的临时存储空间。

此 SQL 语句可为 Windows 上现有的 asiqdemo 数据库增加 100MB 的数据库空间：

```
CREATE DBSPACE asiqdemo2 AS
'd:\\sybase\\ASIQ-12_7\\demo\\asiqdemo2.iq'
IQ STORE
SIZE 100;
```

此 SQL 语句可为现有的 asiqdemo 数据库增加 200MB 的临时空间：

```
CREATE DBSPACE asiqdemotmp AS
'd:\\sybase\\ASIQ-12_7\\demo\\asiqdemo2.iqtmp'
IQ TEMPORARY STORE
SIZE 200 ;
```

注释 有关诊断潜在内存问题的其它信息，请参见 *Sybase IQ 故障排除和恢复指南* 中的 “资源问题”。

自定义 IQ Loader 数据格式

在写入数据文件或管道时以及在生成 LOAD TABLE 脚本时，IQ Loader 接口会使用分隔符、空值处理和字符集的缺省值。缺省值可以在 ETL Server *INI* 文件中进行指定，如下所示：

组	Key	值	缺省值	说明
iq_loader	rowdelim	任意字符串。'\n' 表示换行符。	' \n '	行分隔符
iq_loader	coldelim	任意字符串。'\t' 表示制表符。	' |@#& '	列分隔符
iq_loader	nullreplace	任意字符串。如果为空，则 NULL 子句不会添加到“Load Script”中。	' [NULL] '	用于 NULL 值的字符串
iq_loader	characterset	IQ 支持的任何字符集。	' ' (=auto)	数据文件中使用的编码

Loader 组件

Loader 组件有助于将源数据库或文件中的数据装载到 IQ 数据库，而无需执行任何转换。

组件	说明
IQ Loader File via Load Table	使用此组件可通过 LOAD TABLE 语句将文件中的数据装载到 IQ 目标数据库中。
IQ Loader DB via Insert Location	使用此组件可通过 INSERT LOCATION 语句将源数据库中的数据装载到 IQ 目标数据库中。

IQ Loader File via Load Table

IQ Loader File via Load Table 组件通过自动生成的 LOAD TABLE 语句将文件中的数据装载到 IQ 目标数据库中。

这是一个独立组件，它在读取源文件时充当数据源，在写入 Sybase IQ 数据库时充当数据接收器。此组件没有输入和输出端口。因此，不需要创建一个组件用于从源文件读取，再创建另一个组件用于在 Sybase IQ 中调用“Load Table”。ETL 会自动生成 Load Table 语句，用于从分隔的文本文件中提取数据以及将数据装载到 Sybase IQ 中。

配置 IQ Loader File via Load Table 组件

1 将 IQ Loader File via Load Table 拖动到 “Design”窗口中。
2 在 “Database Configuration”窗口中，为 IQ 目标数据库添加连接参数。有关特定字段要求，请参见第 156 页上的 “IQ Loader DB via Insert Location 属性列表”。
3 选择或输入目标表。
4 单击 “Finish”。
5 在 “Properties”窗口中指定所有其它可选属性。

使用 “Text Source”属性窗口

“Text Source”属性窗口可为源文件中的数据定义结构属性。该窗口包括以下内容：

- File Content 面板 – 显示源文档的内容。
- Properties 面板 – 显示文件说明属性。
- Preview 面板 – 基于当前选定的属性显示源文件中数据的表格视图。

启用客户端装载支持

您可以使用该组件将远程主机上的文件中的数据装载到 Sybase IQ 表中。有关如何启用客户端装载支持的信息，请参见第 141 页上的 “启用客户端装载支持”。

配置 IQ 多写入器以装载数据

为了启用对 Multiplex 执行的支持 （通过使用多个写入器将数据装载到 IQ 中），您需要执行某些其它配置。有关详细的配置步骤，请参见第 142 页上的 “配置 IQ 多写入器以装载数据”。

IQ Loader File via Load Table 属性列表

下表列出了 IQ Loader File via Load Table 组件的必要属性和可选属性。

必要属性

属性	说明
Interface	指定要用于连接到 IQ 目标数据库的方法或驱动程序。支持的接口为 Sybase 和 ODBC。
Host Name	指定 Sybase IQ 目标运行所在的主机。

属性	说明
Destination	单击该属性可以从一组现有表中选择目标表。
Text Source	标识要用作数据源的文本文件。在 “Properties” 窗口中，单击 “Text Source” 图标，选择文件，然后指定格式。请参见第 150 页上的 “使用 “Text Source” 属性窗口”。
Row Delimiter	指定每一行的分隔方式： • LF （换行符） • CR （回车符） • CRLF （回车符加换行符） 也可以输入其它分隔符。
Column Delimiter	指定列的分隔方式： • Tab • Comma • Semicolon 也可以输入其它分隔符。

可选属性

属性	说明
User and Password	指定获得授权的数据库用户和口令，以保护数据库免受未经授权的访问的侵害。
Key	选择用于标识选定函数的记录的目标键属性。 如果未选择任何键，接口则使用主键信息，该信息从 DB 主机传送。如果未提供有任何主键信息，则会显示错误。
函数	选择下列装载函数之一： • Insert – 将记录从文件直接装载到选定目标表中。 • Upsert – 更新现有记录并插入新记录。当选择 “Upsert” 时，将替换现有记录，并且不会执行属性级更新。您可以使用 “Key” 属性指定目标属性，以标识要更新的记录。 在下列情况下，将忽略 “Upsert” 函数： • 当使用 “Generate Load Script” 时。生成的 LOAD TABLE 脚本用于 “Insert” 函数。 • 当选择了 “Truncate” 选项时。在装载数据之前，将删除目标表中的所有记录。该组件将忽略 “Upsert” 函数，并改为执行 “Insert” 函数，这会将所有属性装载到目标表中。
Use IQ Client Side Load	选择该选项可以使用 LOAD TABLE 语句将远程主机上的文件数据批量装载到目标 IQ 数据库。

属性	说明
Load Script	如果此属性为空，则会在运行时根据组件设置生成 Load Table 语句。 要使用自定义脚本，请右键单击组件，然后选择“Generate Load Script”。生成脚本后，可以单击“Load Script”并编辑脚本。
Truncate	选择该选项可将目标表中装载之前的所有记录删除。
Pre-processing SQL	单击“Pre-processing SQL”图标可创建于组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Post-processing SQL	单击“Post-processing SQL”图标可创建于所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Database	指定要用作数据目标的数据库。该数据库与指定的用户名、口令和主机名一起使用。
Schema	指定所有者以过滤表目录。
Database Options	单击“Database Options”图标可设置覆盖性能缺省值和控制某些事务行为的选项。 请参见第 80 页上的“数据库连接设置”。
Null Indicator	指定用于表示源文件中空值的字符串。
Skip Rows	指定一次装载要从输入文件的开头跳过的行数。缺省值为 0。
Parallel format	当所有列（包括最后一列）用单个 ASCII 字符分隔时，选择此选项可允许 LOAD TABLE 命令以并行方式运行。
Strip	如果希望在插入值之前去除值后的尾随空白，请选择此选项。该选项仅适用于长度可变的非二进制数据。
Byte Order	将在读取期间指定字节顺序。此选项适用于所有二进制输入字段。如果未定义，则此选项将会被忽略。Sybase ETL 始终以运行二进制数据的计算机的本机格式读取二进制数据（缺省值为 NATIVE）。您还可以指定以下值： • HIGH，当多字节数量以高位字节优先时。 • LOW，当多字节数量以低位字节优先时。
Block Size	指定读取输入时应使用的缺省大小（以字节为单位）。
Limit	指定要插入表中的最大行数。缺省值为 0，表示无限制。

属性	说明
ON File Error	指定 Sybase IQ 在因为文件不存在或没有读取文件的正确权限而不能打开输入文件时应执行的操作。对于所有其它原因或错误，则会终止整个插入操作。可以指定下列选项之一： • ROLLBACK 中止整个事务 （缺省设置）。 • FINISH 结束已完成的插入操作并结束装载操作。 • CONTINUE 返回一个错误，但仅跳过该文件继续执行装载操作。不能将此选项用于部分宽度的插入。
Word Skip	遇到长度超过创建单词索引时指定的限制时，允许装载继续进行。
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode”选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。如果指定 “00:00:00.000”作为时间参数，则会在项目启动时立即获取排它锁。
Use IQ Multiplex	选择该选项可以使用多个写入器将数据装载到 IQ 数据库中，从而支持 Multiplex 执行。

IQ Loader DB via Insert Location

IQ Loader DB via Insert Location 组件通过 Insert Location 语句将数据从源数据库装载到目标 IQ 数据库。

这是一个自我包含组件，它在读取源数据库时充当数据源，在写入 Sybase IQ 数据库时充当数据接收器。此组件没有输入和输出端口。因此，不需要创建一个组件用于从源文件读取，再创建另一个组件用于在 Sybase IQ 中调用 Insert Location。ETL 会自动生成 Insert Location 语句，用于将数据从源数据库传输到 Sybase IQ。Insert Location：

- 可用于将 Sybase IQ 12.0 版之前的列移动到 12.0 版或更高版本。
- 能够优化从 Adaptive Server Enterprise 或 Sybase IQ 到 Sybase IQ 的装载操作。

- 还可与 Sybase Enterprise Connect Data Access (ECDA) 配合使用，从 Sybase Adaptive Server Enterprise、Oracle、IBM DB2 和 Microsoft SQL Server 装载 Sybase IQ。ETL 4.8 支持 Sybase ECDA 15.0 与 IBM DB2 9.1、 Oracle 10g 和 Microsoft SQL Server 2005。

  有关 Sybase ECDA 文档，请参见位于 http://www.sybase.com/support/manuals 的 Sybase Product Manuals Web 站点。

配置 IQ Loader DB via Insert Location 组件

1 将 IQ Loader DB via Insert Location 组件拖动到 “Design”窗口。

2 输入目标数据库的 IQ 数据库连接属性。
   - Host – 选择 IQ 主机。
   - User – 输入授权数据库用户名。
   - Password – 输入数据库用户的口令。
   - Database – 选择要用作目标数据库的数据库。
   - Schema – 选择模式 / 所有者以限制显示的对象，并在该模式下创建新表。
   - 单击 “Processing”以指定 IQ 目标数据库上的 Pre-Processing SQL 和 Post-Processing SQL。
   - 单击 “Logon”可查看可用表的列表。
   - 单击 “Next”。

3 输入源数据库的连接信息并选择要传送的表。
   - 选择 “Use remote server definition for accessing source database”选项可从源数据库获取数据和元数据。只有已使用 Create Server 命令定义源服务器作为目标 IQ 数据库上的远程服务器时，才选择此选项。如果不选择此选项，将使用 *.INI* 或 *interfaces* 文件中提供的配置信息直接连接到源数据。
   - Host – 选择数据源。
   - Database – 选择要使用的数据库。
   - Schema – 选择模式 / 所有者以限制显示的对象，并在该模式下创建新表。
   - 单击 “Processing”以指定源数据库上的 Pre-Processing SQL 和 Post-Processing SQL。
   - 选择 “Create Target Tables”选项可创建目标表 （如果目标表不存在）。
   - 如果要一直继续处理 （即使当将数据装载到数据库时出现错误），请选择 “Continue on Error”选项。

- 选择 “Encrypted Password”选项可以加密格式传输口令。

  注释 Sybase IQ 在用作远程服务器时不支持口令加密。

- 选择 “Use IQ Multiplex”选项，以通过使用多个写入器将数据装载到 IQ 来支持 Multiplex 执行。如果要将多个表迁移到 IQ 数据库，请选择该选项。
- 选择 “Lock Table”选项可以以排它模式锁定目标表，并防止并发事务更新该表。如果选择该选项，其它任何事务均不能对被锁定的表执行查询或任何更新。 “Lock Table”选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。

  如果选择该选项，还必须指定项目在获取锁之前应等待的最大阻塞时间。
- 在 “Packet Size”字段中输入网络包大小。
- 为 “Limit Rows”输入值。
- 在 “Skip Rows”字段中指定装载时要从输入表的开头跳过的行数。
- 单击 “Logon”可查看指定数据库的可用表的列表。缺省情况下，已选择所有表进行传输。对于不需要传送的表，请取消选择 “Transfer”选项。也可以选择一个或多个表的行，然后右键单击并选择 “Exclude”。要将表纳入传输范围，请右键单击并选择 “Transfer”。

  或者：

  - 单击 “Exclude all objects from transfer”图标可排除所有表。
  - 单击 “Include all objects in transfer”图标可包括所有表。
- 单击 “Next”。

4 检验源表。源表应为以下格式：

```
source_schema.source_table
```

5 从下拉菜单中选择目标表。源与目标之间应存在一对一映射 （每个目标对应一个源）。

6 单击 “Truncate Destination”选项可从目标表中删除所有现有数据行。

7 单击 “Next”。

8 检查装载配置摘要。单击 “Finish”。

设置 *Insert Location* 语句的设置区域

只要执行 Insert Location 语句，Sybase IQ 就会装载确定语言、归类序列、字符集和日期 / 时间格式所需的本地化信息。如果数据库使用平台的非缺省区域设置，则必须在本地客户端设置环境变量以确保 Sybase IQ 装载正确的信息。

如果设置了 LC_ALL 环境变量，则 Sybase IQ 将该变量的值用作区域设置名称。如果未设置 LC_ALL，Sybase IQ 将使用 LANG 环境变量的值。如果这两个变量均未设置，Sybase IQ 将使用区域设置文件中的缺省条目。有关示例，请参见 *Sybase IQ 12.7 系统管理指南* 第 11 章 “国际语言和字符集”中的 “设置区域设置”。

配置 IQ 多写入器以装载数据

为了启用对 Multiplex 执行的支持 （通过使用多个写入器将数据装载到 IQ 中），您需要执行某些其它配置。有关详细的配置步骤，请参见 第 142 页上的 “配置 IQ 多写入器以装载数据”。

IQ Loader DB via Insert Location 属性列表

IQ Loader DB via Insert Location 属性列表标识需要在“IQ Loader DB via Insert Location”组件窗口中定义的连接参数及其它项。

必要属性

属性	说明
IQ Host Name	指定 Sybase IQ 目标运行所在的主机。
IQ User	指定获得授权的 IQ 用户，以保护数据库免受未经授权的访问的侵害。
IQ Password	指定口令，以保护数据库免受未经授权的访问的侵害。
Source Host Name	指定数据源。
Source Database	指定源数据库。
Source Transfer List	指定模式限定的源表名称和目标表名称。在目标截断列中，如果要截断目标表，则指定 1，否则为 0。

可选属性

属性	说明
IQ Database	指定 IQ 目标数据库。
IQ Schema	指定 IQ 模式 / 所有者以限制显示的对象，并在该模式下创建新表。

属性	说明
IQ Pre-processing SQL	单击 “Pre-processing SQL”图标可创建在组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
IQ Post-processing SQL	单击 “Post-processing SQL”图标可创建在所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Use Remote Definition	只有已使用 Create Server 命令定义源服务器作为目标 IQ 数据库上的远程服务器时，才选择此选项。如果不选择此选项，将使用 *.INI* 或 *interfaces* 文件中提供的配置信息直接连接到源数据。
Source Schema	指定模式 / 所有者以限制显示的对象。
Source Pre-processing SQL	单击 “Source Pre-processing SQL”图标可创建在组件初始化过程中运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
Source Post-processing SQL	单击 “Source Post-processing SQL”图标可创建在所有组件执行后运行的查询。 查询可包含一个或多个 SQL 语句。如果使用多个语句，应以分号 (;) 进行分隔。
函数	选择下列装载函数之一： • Insert – 将记录从源直接装载到选定目标表中。 • Upsert – 更新现有记录并插入新记录。当选择 “Upsert”时，将替换现有记录，并且不会执行属性级更新。使用的表必须具有预定义的主键。 在下列情况下，将忽略 “Upsert”函数： • 当使用 “Generate Load Script”时。生成的 INSERT LOCATION 脚本用于 “Insert”函数。 • 当选择了 “Truncate”选项时。在装载数据之前，将删除目标表中的所有记录。该组件将忽略 “Upsert”函数，并改为执行 “Insert”函数，这会将所有属性装载到目标表中。
Create Target Tables	如果目标表不存在，选择此选项可创建目标表。
Continue on Error	选择此选项可一直继续执行 （即使将数据装载到数据库时出现错误）。
Limit Rows	指定要插入表中的最大行数。缺省值为 0，表示无限制。

属性	说明
Skip Rows	指定装载时要从输入表的开头跳过的行数。缺省值为 0。
Encrypted Password	如果要以加密格式传输口令，可选择此选项。
Packet Size	指定网络包大小
Load Script	如果此属性为空，则会在运行时根据组件设置生成 Insert Location 语句。 要使用自定义脚本，请右键单击组件，然后选择“Generate Load Script”。生成脚本后，可以单击“Load Script”并编辑脚本。
Use IQ Multiplex	选择该选项可以使用多个写入器将数据装载到 IQ 数据库中，从而支持 Multiplex 执行。
IQ Lock Table in Exclusive Mode	选择该选项可以锁定目标表，并防止并发事务更新该表。当应用排它表锁时，其它任何事务均不能对被锁定的表执行查询或任何更新。 “IQ Lock Table in Exclusive Mode”选项还会对 Sybase IQ 中装载有相同表的多个项目进行排队。
Wait Time for IQ Lock Table	指定项目在获取排它锁之前应等待的最大阻塞时间。 请以 hh:nn:ss.sss 格式指定时间参数。如果未输入时间参数，服务器将无限期地等待，直到排它锁可用或发生中断为止。如果指定“00:00:00.000”作为时间参数，则会在项目启动时立即获取排它锁。

作业组件

作业组件控制作业的执行。

组件	说明
Start	表示作业的开头。“Start”是添加到任何作业的第一个组件。
Project	标识要在作业中运行的项目。使用此组件可在作业中运行单个项目。
Synchronizer	控制作业流程的执行。使用此组件可根据以前执行的项目的状态控制作业流程。 可以将各个项目定义为关键项目或非关键项目。关键项目的故障将导致“Synchronizer”出现故障。

组件	说明
Multi-Project	提供作业内项目组的可视化表示形式。Multi-Project 结合了 “Project” 组件和 “Synchronizer” 组件的属性。 当作业由大量独立项目 （也就是说，可以按任何顺序执行的项目）组成时，可使用此组件。
Finish	表示一次成功作业执行的结束。 使用此组件可标记作业的成功结束。
Error	以可视化形式表示一次失败作业执行的结束。 使用此组件可标记一次失败作业的结束。

Start

“Start” 是添加到任何作业的第一个组件。若要将此组件添加到作业，请将其从 “Component Store” 拖动到 “Design” 窗口中。

注释 您可以将多个 “Project” 组件和 / 或 “Multi-Project” 组件与 “Start” 组件相连。

Project

“Project” 组件标识要在作业中运行的项目。使用此组件可在作业中运行单个项目。

将 “Project” 组件添加到作业中

❖ **添加和配置 “Project” 组件**

1. 将 “Project” 组件从 “Component Store” 拖动到 “Design” 窗口中，以将其添加到作业。
2. 将该组件与其相邻组件连接。
3. 双击组件，然后选择要执行的项目。

必要属性

属性	说明
Project Name	单击 “Project Name”图标可选择要添加的项目。

可选属性

属性	说明
Continue on DB Write Errors	如果希望项目一直继续执行 （即使在通过 DB Data Sink 组件将数据装载到数据库时出现错误），可选择此选项。如果由于此选项而导致忽略了错误，则该项目将被声明为 “failed”。Combined with the Reject Log（请参见第 80 页上的 “数据库连接设置”）– 使用此选项可以对已拒绝记录进行 “后处理”。

“Project”组件演示

请参见 DemoRepository 中的样本作业。要运行样本作业，请执行下列操作：

1 在 “Navigator”中单击 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Jobs*”。

2 选择：

- Demo Transfer all German Data
- Demo Transfer U.S. Sales on an incremental basis

Synchronizer

Synchronizer 控制作业执行的流程。

使用此组件可根据以前执行的项目的状态控制作业流程。可以将各个项目定义为关键项目或非关键项目。关键项目的故障将导致作业流程沿着 Synchronizer 的 Error 端口上的分支执行。

可以将 Synchronizer 组件的 Success 和 Error 端口连接到：

- 多个 Project 组件。
- 多个 Multi-Project 组件。
- 多个 Project 和 Multi-Project 组件。
- 一个 Finish 组件或 Error 组件。

❖ **添加和配置 “Synchronizer”组件**

1　将该组件添加到作业，并连接它与所有将其执行状态通知此组件的项目。

2　（可选）在 “Property”窗口中，单击 “Synchronize Options”图标可选择关键项目。

Synchronizer 组件演示

请参见 DemoRepository 中的样本作业。要运行样本作业，请执行下列操作：

1　在 “Navigator”中单击 “*Repository*”｜“*TRANSFORMER.transformer.Repository*”｜“*Jobs*”。

2　选择 “Demo Transfer all German Data”。

Multi-Project

“Multi-Project”以可视化形式表示作业内的项目组。 Multi-Project 结合了 “Project”组件和 “Synchronizer”组件的属性。可以将 Multi-Project 组件的 Success 和 Error 端口连接到：

- 多个 Project 组件。
- 多个 Multi-Project 组件。
- 多个 Project 和 Multi-Project 组件。
- 一个 Finish 组件或 Error 组件。

当作业由大量独立项目 （也就是说，可以按任何顺序执行的项目，这些项目在多引擎作业中甚至可以并行执行）组成时，可使用此组件。

❖ **配置 “Multi-Project”组件**

1　将该组件添加到作业，并将它与其相邻组件连接。

2　在 “Properties”窗口中，单击 “Projects Execution”图标可选择要执行的项目。

3　若要将项目添加到组，请在 “Navigator”中右键单击项目名称，然后选择 “Add Projects”。

4　若要从组中删除项目，请在 “Navigator”中选择该项目，然后选择 “Remove Projects”。

注释 一个项目组可以仅包含项目的一个实例。尝试多次添加一个项目将不起作用。

可用的项目执行选项包括：

- **Continue on Error** – 此选项对应于 “Project”组件的 “Continue on DB Write Errors”属性。如果选择此选项，项目将一直继续执行（即使将数据装载到数据库时出现错误）。
- **Critical** – 可以将各个项目定义为关键项目或非关键项目。关键项目的故障将导致 “Multi-Project”组件出现故障。

“Multi-Project”组件演示

请参见 DemoRepository 中的样本作业。要运行样本作业，请执行下列操作：

1 在 “Navigator”中单击 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Jobs*”。

2 选择 “Demo Transfer all U.S. Data”。

Finish

Finish 以直观方式表示一次成功作业执行的结束。使用此组件可标记作业的成功结束。Finish 可以连接到以下作业组件：Synchronize、 Project 或 Multi-Project。

“Finish”组件演示

请参见 DemoRepository 中的样本作业。要运行样本作业，请执行下列操作：

1 在 “Navigator”中单击 “*Repository*” | “*TRANSFORMER.transformer.Repository*” | “*Jobs*”。

2 选择：

- Demo Transfer all German Data
- Demo Transfer all U.S. Data
- Demo Transfer U.S. Sales on an incremental basis

Error

“Error”组件以直观方式表示一次失败作业执行的结束。使用此组件可标记一次失败作业的结束。Error 可以连接到以下作业组件：Synchronize、 Project 或 Multi-Project。

“Error”组件演示

请参见 DemoRepository 中的样本作业。要运行样本作业，请执行下列操作：

1　在“Navigator”中单击“*Repository*”|“*TRANSFORMER.transformer.Repository*”|“*Jobs*”。

2　选择：

- Demo Transfer all German Data
- Demo Transfer all U.S. Data

第 6 章

Sybase ETL Server

本章提供如何使用 Sybase ETL Server 的相关信息。有关详细信息，请参见第 2 页上的 “Sybase ETL 体系结构”。

Sybase ETL Server 是一种可伸缩的分布式网格引擎，它通过使用 Sybase ETL Development 设计的转换流程连接到数据源，提取数据并将其装载到数据目标。Sybase ETL Server 使用 UDP 广播通知其它服务器有关紧急事件的信息，例如启动、停止、系统故障或崩溃。

用于通信的缺省端口为 5124。可以在 *INI* 文件中或使用命令行更改此缺省端口。

服务器之间的所有通信均在同一端口上通过 TCP / IP 完成。

注释 请确保无防火墙阻止此端口且该端口当前未被使用。根据需要，也可以将安装的所有服务器上的端口更改为另一端口号。

启动 Sybase ETL Server

在命令提示符下，输入：

在 Windows 上：

```
GridNode
GridNode --port 5500
```

在 Linux 和 UNIX 中：

```
GridNode.sh
GridNode.sh --port 5500
```

停止 ETL Server

如果服务器是本地或远程进程，可以从控制台停止服务器。在服务器停止之前，所有当前运行项目会完成执行。若要停止 ETL Server，请在命令提示符下，输入：

在 Windows 上：

```
GridNode --shutdown
GridNode --shutdown --server [remotehost] --port [port]
```

在 Linux 和 UNIX 中：

```
GridNode.sh --shutdown
GridNode.sh --shutdown --server[remotehost] --port [port]
```

将 Sybase ETL Server 作为 Windows 系统服务启动

可以将 Sybase ETL Server 作为 Windows 系统服务安装和运行。若要作为独立于 Windows GUI 的系统服务运行 Sybase ETL Server，请在系统启动后使用 SYSTEM 用户帐户启动 ETL Server。

注释 必须具有管理员特权才能安装、删除、启动和停止系统服务。

若要作为 Windows 系统服务安装服务器，请在命令提示符下，输入：

```
GridNode.exe --install [additional parameters]
```

若要删除服务，请在命令提示符下，输入：

```
GridNode.exe --remove
```

作为 Windows 服务运行 ETL Server 时，基本事件 （失败、成功消息等）将写入 Windows 事件日志。

命令行参数

本节介绍所有 Sybase ETL Server 命令行参数。

若要显示有关可用参数的概述，请在命令提示符下输入 `GridNode - -help` 或 `GridNode - h`。控制台输出会显示每个参数的长格式和简写形式，例如：

```
--version,  -V    Displays version information
```

注释 完整的参数名称通常以两个减号作为前缀，而简写形式则只有一个减号。

表 6-1：命令行参数

命令	缩写	UNIX	Windows	说明
install	inst	是	是	作为 Unix 守护程序或 Windows 服务安装应用程序。
remove	rm	是	是	删除守护程序或系统服务启动。
setoptions	so	否	是	设置作为 Windows 服务运行时使用的命令行选项。
getoptions	go	否	是	输出当作为 Windows 服务运行时使用的命令行选项。
background	bg	是	否	设置不会过度使用系统资源的后台进程。
no_pidfile	nopid	是	否	设置服务器用来记录守护程序进程 ID 的文件。
console	con	是	是	在控制台上写入详细错误信息和跟踪消息。
diagnosis	diag	是	是	列出应用程序环境。

命令	缩写	UNIX	Windows	说明
tracelevel	tl	是	是	设置调试的跟踪级别，从 0（无跟踪）到 5 （很详细）。
server	s	是	是	标识要使用的远程服务器。
port	p	是	是	标识要在其上操作的端口号。
version	V	是	是	标识应用程序版本信息。
help	h	是	是	显示帮助信息。
licenses	ll	是	是	标识有关可用许可证及其状态的简短信息。
nodelist	nl	是	是	列出所有已知对等节点。
shutdown	sh	是	是	关闭节点。
用于执行项目和作业的命令行参数				
dbhost *host*		是	是	存储数据库主机名或数据源名称 (DSN)。
dbinterface *interface*		是	是	存储数据库接口。
dbdatabase *database*		是	是	存储数据库名称。
dbschema *schema*		是	是	存储数据库模式。
dbuser *user*		是	是	存储数据库用户。
dbpassword *encrypted password*		是	是	存储数据库口令。
client *client*		是	是	存储库客户端名。
user *user*		是	是	存储库客户端用户。
password *encrypted password*		是	是	存储库客户端口令。
项目 [*name* | *ID*]		是	是	通过指定项目名称或 ID 执行项目。
job [*name* | *ID*]		是	是	通过指定项目名称或 ID 执行作业。
paramset [*name* | *ID*]		是	是	按名称或 ID 指定项目或作业的参数集。
encrypt *password*		是	是	为口令加密，并显示加密值。使用 encrypt 生成必须与 dbpassword 和 password 一同使用的加密口令。
ping *host:port*		是	是	检查 ETL Server 是否运行在指定的主机和端口上。
env “*variable1=value*; *variable2=value*; ...;*variableN=value*”		是	是	指定要通过 ETL Server 运行的其它环境变量。使用分号分隔多个环境变量，并将整个变量字符串置于双引号中。

使用 ETL Server 执行项目和作业

Sybase ETL Server 可以使用在第 167 页的表 6-1中列出的命令行参数，在所有支持的平台上执行项目和作业。若要执行项目和作业，请使用以下语法：

```
GridNode --project PROJ-1234-5678 --dbinterface dbodbc --dbhost etl_comp
--client transformer--user TRANSFORMER --password 1234ABCD
```

其中，项目 ID 为 “PROJ-1234-5678”，数据库接口为 “dbodbc”，主机名为 “etl_comp”，客户端为 “transformer”，用户为 “TRANSFORMER”，且口令的加密版本为 “1234ABCD”。

此外，也可以使用以下命令行参数执行项目和作业：

- project – 按名称指定项目和参数集。也可以按名称指定作业。与输入复杂的项目、作业或参数集 ID 相比，指定名称更为简单。以下示例使用项目名称 “LoadCustomers” 和参数集名称 “myparams”：

```
GridNode --project LoadCustomers --dbinterface dbodbc --dbhost etl_com
--client transformer --user TRANSFORMER --paramset myparams --password 1234ABCD
```

- encrypt – 生成加密口令。加密口令必须与 dbpassword 和 password 一起使用。若要加密 “mypassword”，请输入：

  ```
  Gridnode --encrypt mypassword
  ```

  ETL Server 将生成并显示加密口令。

- ping – 验证 ETL Server 是否运行在特定主机和端口上。要验证 ETL Server 是否正在 “localhost” 的缺省端口运行，请输入：

  ```
  Gridnode --ping localhost
  ```

  如果 ETL Server 正在运行，将显示以下消息：

  ```
  localhost is alive!
  ```

  如果 ETL Server 没有在您指定的主机和端口运行，则会显示错误消息。

- env – 指定项目和作业的环境变量。使用 uGetEnv 函数，可以在运行时访问这些变量的值。以下示例使用 “LoadCustomers” 项目，环境变量为 INPUT_FILE 和 OUTPUT_FILE：

```
GridNode --project LoadCustomers --dbinterface dbodbc --dbhost etl_com
--client transformer --user TRANSFORMER --paramset myparams --password
1234ABCD --env  “INPUT_FILE=input.txt;
OUTPUT_FILE=output.txt”
```

注释 当您输入命令行时，请在同一行中键入命令、参数和值。为清晰起见，本节中的示例拆分为多行。

在早于 4.5 的 Sybase ETL 版本中，项目和作业的执行是通过 ProcessQ 应用程序完成的， ProcessQ 应用程序随 ETL Server 一起分发并且仅在 Windows 平台内部使用。ProcessQ 目前已不受支持，仅供实现向后兼容性。许多 ProcessQ 参数已被禁用，无法继续使用。

注释 Sybase 建议您不要使用 ProcessQ 执行项目和作业。

INI 文件设置

安装目录的 *etc* 文件夹中提供了包含 Sybase ETL Server 设置的 *INI* 文件。

Default.ini

组	关键字	值	Default	说明
网络	proxy	host:port explorer	explorer	设置用于 Internet 访问的代理。 可以通过使用关键字 “http_proxy”、“https_proxy”、“ftp_proxy”或“ftps_proxy”，针对某一协议 （HTTP、HTTPS、 FTP、 FTPS）调优代理。 代理值 “explorer”采用 Windows 环境中的系统代理。
网络	timeout	1-2147483 秒	600 秒	设置 FTP 连接的超时值。
语言	Default	English_USA	English_USA	基于所选的语言和国家 / 地区调优应用程序的行为。
记录	Console	1/0	0	将日志信息发送到控制台。
记录	LogFile	1/0	1	将日志信息发送到 *system.log* 文件。
记录	Tracelevel	0-5	0	为调试提供不同程度的信息。级别 0 提供最少量的信息，级别 5 提供最详细的信息。在正常执行过程中，将此值设置为 0 可将对性能的影响降至最低程度。
记录	Flushtime	1-n	1	表示内部日志刷新之间间隔的秒数。

使用 Web 浏览器监控项目和作业

您可以使用 Web 浏览器监控所有网格节点的状态，包括作为网格体系结构一部分的 ETL Development 网格节点。除监控通过 ETL Development 启动的项目和作业之外，您还可以：

- 监控远程作业的状态。
- 挂起和恢复远程作业和项目。
- 查看 ETL Server 的远程日志文件。

开始监控之前：

- 如果 ETL Server 尚未运行，则启动 ETL Server。
- 验证您计算机上是否安装有 Internet Explorer (IE) 6.0 或更高版本。

❖ **监控项目和作业**

1　打开 Web 浏览器。

2　请输入：

```
http://<hostname>:<port_number>
```

其中， <*hostname*> 为运行 ETL Server 的计算机的网络名，<*port_number*> 为在其上启动节点的端口。缺省端口号是 5124。

此时将出现显示有以下内容的监控页面：

- Node Summary – 显示当前正在运行的服务器的详细信息，例如，运行该服务器的计算机的主机名和操作系统、该服务器运行的作业数、该服务器利用的 CPU、内存和磁盘空间大小、PID 和帐户信息，以及产品名称和版本号。该选项卡还包括活动作业和最新作业列表。
- Active Jobs – 显示正在运行的作业的详细列表。
- Job History – 提供自前一天执行的所有作业列表。
- Log History – 提供系统日志历史记录。
- Node Overview – 提供所有正在运行的服务器的列表。

查看活动作业

单击 “Active Jobs”选项卡。您将看到所有正在运行的作业及其详细信息，例如，名称、状态、作业中的项目数、启动和结束时间，以及已处理的记录数。

挂起活动作业

在 “Active Jobs”选项卡或 “Node Summary”选项卡中，单击要挂起的作业旁的 “Suspend”图标。该作业的状态随即更改为 “Suspended”。同时还挂起该作业中的所有项目。

恢复作业

1 在 “Active Jobs”选项卡或 “Node Summary”选项卡中，选择要查看其历史记录的作业。或者，您也可以单击 “Job History”选项卡。

2 单击已被挂起的作业旁的 “Resume”图标。

该作业的状态随即更改为 “Running”，同时还会恢复该作业中的所有项目。

取消活动作业

在 “Summary”选项卡中，单击要取消的作业旁的 “Cancel”图标。该作业随即从列表中删除。

挂起活动项目

在 “Active Jobs”选项卡中，单击要挂起的项目旁的 “Suspend”图标。该项目的状态随即更改为 “Suspended”。

恢复项目

1 在 “Active Jobs”选项卡中，选择具有已被挂起的项目的作业。或者，您也可以单击 “Job History”选项卡。

2 单击已被挂起的项目旁的 “Resume”图标。项目状态随即更改为 “Running”，该项目即已恢复。

取消项目

1 在 “Active Jobs”选项卡中，选择活动作业列表中的作业。

2 单击要取消的项目旁的 “Cancel”图标。

项目状态随即更改为 “Cancelled”，该项目即被停止。

查看所有正在运行的服务器

选择 “Node Overview”选项卡。此时将显示所有正在运行的服务器的列表及其详细信息，例如该服务器运行的作业状态、作业数，以及 CPU、内存和磁盘空间大小。

关闭服务器

在 “Node Overview”选项卡中，单击服务器旁的 “Shutdown”图标。该服务器随即从列表中删除。

查看日志历史记录

选择 “Log History”选项卡。此时将显示一个表，显示时间戳、错误类型和错误消息等系统日志详细信息。

如有必要，请单击 “Download”以将日志文件下载到您的计算机。

Sybase ETL Server 故障排除

在与技术支持联系之前，请完成以下步骤：

1　检查错误文本。

2　检查日志文件。

3　在启用系统跟踪的条件下再次运行 Sybase ETL Server。

4　验证版本、修订编号以及计算机 ID。

语法：

GridNode --version

输出：

```
GridNode 4.8.0.27424
```

5　验证可用的许可证。

语法：

GridNode --licenses

输出：

```
GridNode (4.8.0.27424)
Grid Node
-------------------------------------------------
Product ID: SybaseETLServer
Machine ID: 9TuA+igB6298Hys=
SYSAM ID  : 001111eb57f9 DISK_SERIAL_NUM=189e22a0
-------------------------------------------------
Install Date: Tuesday Feb 03 13:53:47 2009
-------------------------------------------------

File: sybase_etl_server.license
Product: Sybase ETL Server (SybaseETLServer)
Version: 4.8
License: ETL Components 4.8(ETL_SERVER)
Status: Valid
```

附 录 A

函数参考

本附录提供了有关 Sybase ETL 函数的参考信息。

集合

函数	说明
uAvg	返回所有输入值的平均值
uMax	返回一列值中的最大值
uMin	返回一列值中的最小值

uAvg

说明 返回所有输入值的平均值

语法 real uAvg(value, ...)

参数 *数字值*

一系列数值参数

示例

```
uAvg(1,2,3,4,5) // 返回 3
```

uMax

说明 返回一列值中的最大值

语法 uMax(value,...)

参数 *数字值*

一系列数值参数

示例

```
uMax(1, 6, 4, -6) // 返回 6

uMax("b", "A", "a") // 返回 "b"

uMax("2004-05_02", "2006-12-12", "1999-05-30") //
返回 "2006-12-12"
```

uMin

说明 返回一列值中的最小值

语法 uMin(value, ...|)

参数 *数字值*

一系列数值参数

示例

```
uMin(1, 6, 4, -6) // 返回 -6

uMin("b", "A", "a") // 返回 "A"

uMin("2004-05-02", "2006-12-12", "1999-05-30")
// 返回 "1999-05-30"
```

位操作

函数	说明
uBitAnd	逐位与操作
uBitOr	逐位或操作
uBitXOr	逐位异或操作
uBitNot	逐位非操作

uBitAnd

说明 逐位与操作

语法 number uBitAnd(value, ...)

参数 *数字值*

一系列数值参数

示例

```
uBitAnd(10, 3) // 返回 "2"
```

uBitOr

说明 逐位或操作

语法 number uBitOr(value, ...)

参数 *数字值*

一系列数值参数

示例

```
uBitOr(10, 3) // 返回 "11"
```

uBitXOr

说明 逐位异或操作

语法 number uBitXOr(value1 , value2)

参数 *number value1, value2*

要计算的值

示例

```
uBitXOr(10, 3) // 返回 "9"
```

uBitNot

说明 逐位非操作

语法 number uBitNot(value)

参数 *数字值*

数字参数

示例

```
uBitNot(10) // 返回 "-11"
```

Boolean

函数	说明
uIsAscending	如果每个参数均等于或大于其前项，则返回 1
uIsBoolean	如果参数是 1、 true 或 yes 中的某一个，则返回 1。
uIsDate	如果参数可以解释为日期，则返回 1
uIsDescending	如果每个参数均等于或小于其前项，则返回 1
uIsEmpty	如果参数为空或 null，则返回 1
uIsInteger	如果参数可以解释为整数值，则返回 1
uIsFloat	如果参数可以解释为浮点值，则返回 1
uIsNull	如果参数为 null，则返回 1
uIsNumber	如果参数可以解释为数字，则返回 1
uNot	如果输入为 1，则返回 0，如果输入为 0，则返回 1

uIsAscending

说明　如果每个参数均等于或大于其前项，则返回 1

语法　number uIsAscending(params, ...)

参数　*params*

表达式列表或任何数据类型的值的列表。

示例　检查多个值是否是升序

```
uIsAscending("A", "B", "C") // 返回 1
uIsAscending("A", "A", "C") // 返回 1
uIsAscending("A", "C", "B") // 返回 0

uIsAscending("1", "2", "3") // 返回 1
uIsAscending("3", "2", "2") // 返回 0
uIsAscending("2004-03-03", "2004-03-05", "2004-03-07")
// 返回 1
uIsAscending("2004-03-03", "2004-03-07", "2004-03-05")
//返回 0
```

uIsBoolean

说明　如果参数是以下值则返回 1：

- 1、 true 或 yes 中的某一个
- 0、 no 或 false

如果参数是以下值则返回 1

语法　number uIsBoolean(param)

参数　*param*

表达式或任何数据类型的值。

示例　检查是否为布尔值：

```
uIsBoolean("1")     // 返回 1
uIsBoolean("yes")   // 返回 1
uIsBoolean("true")  // 返回 1
uIsBoolean("-1")    // 返回 0
uIsBoolean("0")     // 返回 1
```

uIsDate

说明

如果参数可以解释为日期，则返回 1。如果省略第二个参数，则函数尝试应用以下某一个格式：

- `y-M-D H:N:S.s`
- `y-M-D H:N:S`
- `y-M-D`
- `H:N:S`

注释 有关格式字符串的详细信息，请参见 uConvertDate 函数。

语法

number uIsDate(datestring [, format])

参数

string datestring

要检查的字符串。

string format （可选）

输入日期的格式

示例

```
uIsDate("2004-02-29") // 返回 1
uIsDate("2003-02-29") // 返回 0，因为 2003 不是闰年
```

uIsDescending

说明

如果每个参数均等于或小于其前项，则返回 1

语法

number uIsDescending(params, ...)

参数

params

任意数据类型的一系列表达式或值

示例

检查多个值是否是降序

```
uIsDescending("C", "B", "A") // 返回 1
uIsDescending("C", "C", "A") // 返回 1
uIsDescending("A", "C", "B") // 返回 0
uIsDescending("3", "2", "1") // 返回 1
uIsDescending("3", "2", "3") // 返回 0
uIsDescending("2004-03-20", "2004-03-15", "2004-03-
07") // 返回 1
uIsDescending("2004-03-20", "2004-03-07", "2004-03-
15") // 返回 0
```

uIsEmpty

说明　如果参数为空或 null，则返回 1

语法　number uIsEmpty(param)

参数　*param*

要研究的表达式或值

示例

```
uIsEmpty("1")          // 返回 0
uIsEmpty(null)         // 返回 1
uIsEmpty("")           // 返回 1
```

uIsInteger

说明　如果参数可以解释为整数值，则返回 1

语法　number uIsInteger(param)

参数　*param*

要研究的表达式或值

示例

```
uIsInteger ("1")    // 返回 1
uIsInteger ("2.34")    // 返回 0
uIsInteger ("ABC")  // 返回 0
```

uIsFloat

说明　如果参数可以解释为浮点值，则返回 1

语法　number uIsFloat(param)

参数　*param*

要研究的表达式或值

示例

```
uIsFloat("1")       // 返回 1
uIsFloat("2.34")    // 返回 1
uIsFloat("ABC")     // 返回 0
```

uIsNull

说明　如果参数为 null，则返回 1

语法　number uIsNull(param)

参数　*param*

要研究的表达式或值

示例

```
uIsNull("1")  // 返回 0
uIsNull(null) // 返回 1
```

uIsNumber

说明　如果参数可以解释为数字，则返回 1

语法　number uIsNumber(param)

参数　*param*

要研究的表达式或值

示例　检查是否为数值

```
uIsNumber("1")      // 返回 1
uIsNumber("2.34")   // 返回 1
uIsNumber("ABC")    // 返回 0
```

uNot

说明　如果输入为 1，则返回 0，如果输入为 0，则返回 1。此函数只与 uIs- 函数结合使用，因为返回的布尔值不是 true 和 false，而是 0 和 1。

语法　number uNot(expression)

参数　*expression*

应转换为负值的数值

示例

```
uNot(1) // 返回 0
```

转换

函数	说明
uBase64Decode	将字符串从 Base64 表示形式解码
uBase64Encode	将字符串编码为 Base64 表示形式
uConvertDate	将日期字符串转换为缺省或自定义日期格式
uFromHex	将十六进制值转换为整数值
uToHex	将整数值转换为十六进制数字
uHexDecode	以十六进制值编写字符串
uHexEncode	将字符串的字符编码为十六进制表示形式
uToUnicode	将字符串转换为其 Unicode 表示形式
uURIDecode	通过用其原始值替换转义序列将字符串解码
uURIEncode	用转义序列替换 URI 中的某些字符

uBase64Decode

说明 将字符串从 Base64 表示形式解码

语法 string uBase64Decode(input)

参数 *string input*

要解码的字符串

示例

```
uBase64Decode("QSBzZWNyZXQ=") // 返回 "A secret"
```

uBase64Encode

说明 将字符串编码为 Base64 表示形式

语法 string uBase64Encode(input)

参数 *string input*

要编码的字符串。

示例

```
uBase64Encode("A secret") // 返回 "QSBzZWNyZXQ="
```

uConvertDate

说明

将日期字符串转换为缺省或自定义日期格式。第一个参数是要转换的日期字符串；第二个参数是格式字符串，用于指定输入日期字符串的日期格式 （参见下表）。outputformat 参数为可选项，如果省略它，则以格式 `y-M-D H:N:S` 转换日期。

该函数用于处理自 *1582* 年至今的日期。如果不能转换日期，则结果字符串将为空。

语法

string uConvertDate(datestring, inputformat [, outputformat])

参数

string datestring

要转换的日期字符串

string inputformat

输入字符串的日期 / 时间格式

string outputformat （可选）

所需的输出格式。如果省略，则缺省格式为 `y-M-D H:N:S`。

示例

将日期字符串转换为不同格式

```
uConvertDate("2005-06-27 00:00:00","y-M-D H:N:S","D mY") // 返回 "27 JUN 05"

uConvertDate("27 JUN 05", "D m Y") // 返回 "2005-06-27 00:00:00"
```

用法

说明

函数 uConvertDate 使用源格式字符串和目标格式字符串将日期字符串转换为不同格式。第一个参数是要转换的日期字符串。第二个参数是格式字符串，用于指定输入日期的日期格式 （参见下表）。outputformat 参数是可选项。如果省略，则将使用格式 `y-M-D H:N:S` 转换日期。该函数用于处理自 *1582* 年至今的日期。如果不能转换日期，则结果字符串将为空。

标识符说明

标识符	说明
Y	2 位数年份 (06)
y	4 位数年份 (2006)
C	世纪 (20)
M	月份 (03)
m	月份 (JUN)
D	天 (12)
H	小时 (00..23)
h	小时 (01..12)
N	分钟

标识符	说明
S	秒
s	百分之一秒
t	千分之一秒
A	AM/PM
d	周中的某天 (5)
D	周中的某天 (Friday)
E	年中某天 (001..366)
G	年中某周 (01..52)
F	月中周 (1..6)

uFromHex

说明　将十六进制值转换为整数值

语法　integer uFromHex(input)

参数　*string input*

要转换的字符串

示例

```
uFromHex("A3F") // 返回 2623
uFromHex("B")   // 返回 11
```

uToHex

说明　将整数值转换为十六进制数字

语法　string uToHex(input)

参数　*integer input*

要转换的整数值

示例

```
uToHex(45) // 返回 "2D"
```

uHexDecode

说明 以十六进制值编写字符串

语法 string uHexDecode(input)

参数 *string input*

包含十六进制值的十六进制字符串

示例 将十六进制值转换为字符串

```
uHexDecode("313730") // 返回 "170"
uHexDecode(313730)   // 返回 "170"
```

uHexEncode

说明 将字符串的字符编码为十六进制表示形式

语法 string uHexEncode(input)

参数 *string input*

要编码的字符串

示例 将字符串转换为十六进制值

```
uHexEncode("170") // 返回 "313730"
uHexEncode(170)   // 返回 "313730"
```

uToUnicode

说明 将字符串转换为其 Unicode 表示形式

语法 string uToUnicode(input)

参数 *string input*

输入字符串

uURIDecode

说明 通过用其原始值替换转义序列将字符串解码

语法 string uURIDecode(uri)

参数 *string uri*

要解码的 URI

示例

```
uURIDecode("www.myServer.com/filename%20with
%20spaces.txt") // 返回
"www.myServer.com/filename with spaces.txt"
```

uURIEncode

说明 用转义序列替换 URI 中的某些字符

语法 string uURIEncode(uri)

参数 *string uri*

要编码的 URI

示例

```
uURIEncode("www.myServer.com/filename with
spaces.txt")// 返回 "www.myServer.com/filename%
20with%20spaces.txt"
```

日期和时间

大多数 Date 和 Time 函数都是从 uFormatDate 函数派生出来的。唯一的差别是其它 Date 和 Time 函数只返回特殊格式的日期或部分日期，而且它们没有第一个格式参数。因此， uDate() 相当于 uFormatDate("%Y-%m-%d")。

另请参见

- 第 187 页上的 “时间字符串”
- 第 188 页上的 “修饰符”
- 第 189 页上的 “日期和时间计算”
- 第 190 页上的 “已知限制”
- 第 190 页上的 “日期和时间函数列表”

时间字符串

说明 时间字符串可以采用以下任何一种格式：

1 *YYYY-MM-DD*

2 *YYYY-MM-DD HH:MM*

3 *YYYY-MM-DD HH:MM:SS*

4 *YYYY-MM-DD HH:MM:SS.SSS*

5 *HH:MM*

6 *HH:MM:SS*

7 *HH:MM:SS.SSS*

8 now

9 *DDDD.DDDD*

注释
仅指定时间的格式 5 到 7 假设日期为 2000-01-01。格式 8 将转换为使用通用协调时间 (UTC) 的当前日期和时间。格式 9 是以浮点值表示的儒略天数。

示例

获取当前时间 如果没有给定日期，则假定采用时间字符串 now，并将日期设置为当前日期和时间。

```
uDate() // 返回类似 "2006-03-01" 的内容
uDate() 相当于 uDate("now")
```

获取特殊日期 `uDate("2004-01-04 14:26:33")`

```
// 返回日期部分 "2004-01-04"
```

修饰符

说明

时间字符串后面可以是零，也可以是改变日期或改变日期含义的修饰符。可用修饰符如下：

1 *NNN* days

2 *NNN* hours

3 *NNN* minutes

4 *NNN.NNNN* seconds

5 *NNN* months.

6 *NNN* years

7 start of month

8 start of year

9 start of day

10 weekday *N*

11 unixepoch

12　localtime

13　utc

示例

第一个大小修饰符 （1 到 6）只是将指定的时间量添加到由前面的时间字符串指定的日期中。

“start of”修饰符 （7 到 9）将日期向后移到当前月、年或天的开头。

“weekday”(10) 修饰符将日期向前移到星期编号为 *N* 的下一个日期：星期日是 0，星期一是 1，等等。

unixepoch 修饰符 (11) 只有紧跟在 *DDDD.DDDDD* 格式的时间字符串后面时才有效。此修饰符导致将 *DDDD.DDDDD* 解释为非通常情况下的儒略天数，而解释为自从 1970 年以来的秒数。利用此修饰符可将基于 UNIX 的时间方便地转换为儒略天数。

localtime 修饰符 (12) 调整前一时间字符串，以使它显示正确的本地时间。utc 则取消该操作。

日期和时间计算

说明

这些示例显示了一些典型的日期和时间计算。

示例

计算当前日期

```
uDate('now')
```

计算当前月的最后一天。

```
uDate('now','start of month','+1 month','-1 day')
```

在给定 UNIX 时间戳 1092941466 的条件下计算日期和时间

```
uDatetime(1092941466, 'unixepoch')
```

在给定 UNIX 时间戳 1092941466 的条件下计算日期和时间，并针对当地时区进行调整

```
uDatetime(1092941466, 'unixepoch', 'localtime')
```

计算当前 UNIX 时间戳

```
uFormatDate ('%s','now')
```

计算两个日期之间的秒数

```
uJuliandate('now')*86400 - uJuliandate ('2004-01-01 02:34:56')*86400
```

计算在当前年份的 10 月 （1 月 + 9）的第一个星期二的日期

```
uDate('now','start of year','+9 months','weekday 2')
```

已知限制

说明

对本地时间的计算随区域设置的不同而变化。在此实现中，使用标准 C 库函数 localtime() 来帮助计算本地时间。而且， localtime() C 函数正常情况下只对在 1970 年和 2037 年之间的年份有效。对于此范围以外的日期，我们尝试将年份映射到在此范围内的相等年份，然后执行计算，最后映射回该年。

- 对于在儒略天数 0 (-4713-11-24 12:00:00) 之前的日期，日期计算不会给出正确的结果。
- 所有内部计算均假定是公历系统。

日期和时间函数列表

说明

此表列出所有日期和时间函数。

函数	说明
uDate	从格式为 *YYYY-MM-DD* 的日期中返回年、月和日
uDateTime	从格式为 *YYYY-MM-DD HH.MM.SS* 的日期中返回年、月和日
uDay	返回指定日期的天编号
uDayOfYear	返回自某年开头到现在的天数
uHour	返回指定日期的小时
uQuarter	返回年份的季度
uIsoWeek	返回按 ISO 8601 定义的周编号
uJuliandate	以格式 *DDDD.DDDD* 返回自从格林尼治时间公元前 4714 年 11 月 24 日中午以来的天数
uMinute	返回指定日期的分钟
uMonth	返回指定名称的月份
uMonthName	返回以当前区域设置语言指定的日期所属月份的名称
uMonthNameShort	返回以当前区域设置语言指定的日期所属月份的名称简写形式
uSeconds	返回指定日期的秒
uTime	以格式 *HH.MM.SS* 返回日期的时间部分
uTimeDiffMs	返回两个日期之差，单位为毫秒
uWeek	返回指定日期的星期
uWeekday	返回指定日期的星期日期

函数	说明
uWeekdayName	返回以当前区域设置语言指定的日期的星期名称
uWeekdayNameShort	返回以当前区域设置语言指定的日期的星期名称简写形式
uYear	返回指定日期的年份

注释 有关可能的修饰符参数的详细信息，请参见第 187 页上的 “日期和时间”。

uDate

说明 从格式为 *YYYY-MM-DD* 的日期中返回年、月和日

语法 string uDate([modifiers])

参数 *string modifiers （可选）*

指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例 获取时间戳中的日期部分

```
uDate("now") // 以 "YYYY-MM-DD" 格式返回当前日期。

uDate("now", "start of year", "9 months", "weekday 2")
// 返回该年十月中第一个星期二的日期。
```

uDateTime

说明 从格式为 *YYYY-MM-DD HH.MM.SS* 的日期中返回年、月和日

语法 string uDateTime([modifiers])

参数 *string modifiers （可选）*

指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例 获取时间戳中的日期时间部分

```
uDateTime("now") // 返回当前日期，格式为
"YYYY-MM-DD HH:MM:SS"

uDateTime("now", "start of month", "1 months", "-1 day")
// 返回该月最后一天的日期
```

uDay

说明 返回指定日期的天编号

语法 string uDay([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例 根据时间戳获得天编号

```
uDay("now") // 返回当前天编号
uDay("1969-03-13 10:22:23.231") // 返回 "13"
```

uDayOfYear

说明 返回自某年开头到现在的天数

语法 string uDayOfYear([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例 根据时间戳获得天编号

```
uDayOfYear("now") // 返回该年已经过去了多少天
uDayOfYear("1969-03-13 10:22:23.231") // 返回 "72"
```

uHour

说明 返回指定日期的小时

语法 string uHour([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

```
uHour("now") // 返回当前小时
uHour("1969-03-13 10:22:23.231") // 返回 "10"
```

uQuarter

说明　返回年份的季度

语法　string uQuarter([modifiers])

参数　*string modifiers （可选）*

指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

```
uQuarter ("now") // 返回当前季度

uQuarter ("2005-03-13 10:22:23.231") // 返回 "1"
```

uIsoWeek

说明　返回按 *ISO 8601* 定义的周编号

某年的第一周是数字 *01*，它被定义为包含日历年中第一个星期四的那一周，这意味着它还是：

- 通常在日历年内的第一周
- 包含 1 月 4 日的那周
- 从最接近 1 月 1 日的星期一开始的那周

因此某年的最后一周 （数字 52 或 53）是：

- 包含日历年的最后一个星期四的那周
- 通常在日历年内的最后一周
- 包含 12 月 28 日的周
- 以最接近 12 月 31 日的星期日结束的周

语法　number uIsoWeek([modifiers])

参数　*string modifiers （可选）*

指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

```
uIsoWeek("now") // 返回当前周编号
```

uJuliandate

说明　以 *DDDD.DDDD* 格式返回自从格林尼治时间公元前 4714 年 11 月 24 日中午以来的天数。对于日期和时间计算， juliandate 函数是最佳选择。

语法　string uJuliandate([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 “now”修饰符。

示例 将日期转换成数值以便进行计算

```
uJuliandate("now") // 以 "DDDD.DDDD" 格式返回当前日期
```

计算两个日期之间的秒数

```
uJuliandate('now')*86400 - uJuliandate ('2004-01-01
02:34:56')*86400
```

计算自从 Hastings 战役以来的天数

```
uJuliandate('now') - uJuliandate('1066-10-
14','gregorian')
```

在给定 UNIX 时间戳 1092941466 的条件下计算日期和时间，并针对当地时区进行调整

```
uJuliandate(1092941466, 'unixepoch', 'localtime')
```

uMinute

说明 返回指定日期的分钟

语法 string uMinute([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

```
uMinute("now") // 返回当前分钟
```

```
uMinute("1969-03-13 10:22:23.231") //返回 "22"
```

uMonth

说明 返回指定日期的月份

语法 string uMonth([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

```
uMonth("now") // 返回当前月份
```

```
uMonth("1969-03-13 10:22:23.231") // 返回 "03"
```

uMonthName

说明　返回以当前区域设置语言指定的日期所属月份的名称

语法　string uMonthName([modifiers])

参数　*string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例　获取日期中的月份名称

```
uMonthName("now") // 返回当前月份名称
```

将区域设置设置为 “English”

```
uSetLocale("English")
uMonthName("1969-03-13 10:22:23.231") // 返回 "March"
```

将区域设置设置为 “German”

```
uSetLocale("German")
uMonthName("1969-03-13 10:22:23.231") // 返回 "Mrz"
```

uMonthNameShort

说明　返回以当前区域设置语言指定的日期所属月份的名称简写形式

语法　string uMonthNameShort([modifiers])

参数　*string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例　获取日期中的月份名称

```
uMonthNameShort("now") // 返回当前月份名称。
```

将区域设置设置为 “English”

```
uSetLocale("English")
uMonthNameShort("1969-03-13 10:22:23.231") // 返回 "Mar"
```

将区域设置设置为 “German”

```
uSetLocale("German")
uMonthNameShort("1969-03-13 10:22:23.231") // 返回 "Mr"
```

uSeconds

说明 返回指定日期的秒。

语法 string uSeconds([modifiers])

参数 *string modifiers （可选）*

指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

```
uSeconds("now") // 返回当前秒

uSeconds("1969-03-13 10:22:23.231") // 返回 "23"
```

uTime

说明 以 *HH.MM.SS* 格式返回日期的时间部分。

语法 string uTime([modifiers])

参数 *string modifiers （可选）*

指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例 获取时间戳中的时间部分

```
uTime() // 返回当前 UTC 时间

uTime("now","localtime") // 返回当前本地时间
```

uTimeDiffMs

说明 返回两个日期之差，单位为毫秒

语法 string uTimeDiffMs(date1, date2)

参数 *string date1*

更早的日期

string date2

更新的日期

示例

```
uTimeDiffMs ("18:34:20", "18:34:21")     // 返回 1000

uTimeDiffMs ("18:34:20", "18:34:21.200") // 返回 1200
```

uWeek

说明 返回指定日期的星期

语法 string uWeek([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例
```
uWeek("now") // 返回当前周

uWeek("1969-03-13 10:22:23.231") // 返回 "10"
```

uWeekday

说明 返回指定日期的星期编号

语法 string uWeekday([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例
```
uWeekday("now") // 返回当前星期编号

uWeekday("1969-03-13 10:22:23.231") // 返回 "4" 表示
星期四
```

uWeekdayName

说明 返回以当前区域设置语言指定的日期的星期名称

语法 string uWeekdayName([modifiers]);

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例
```
uWeekdayName("now") // 返回当前星期名称
```
将区域设置设置为 “English”
```
uSetLocale("English")
uWeekdayName("1969-03-13 10:22:23.231") // 返回
"Thursday"
```
将区域设置设置为 “German”
```
uSetLocale("German")
uWeekdayName("1969-03-13 10:22:23.231") // 返回
"Donnerstag"
```

uWeekdayNameShort

说明 返回以当前区域设置语言指定的日期的星期名称简写形式

语法 string uWeekdayNameShort([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例
```
uWeekdayNameShort("now") // 返回当前星期名称
```
将区域设置设置为 “English”
```
uSetLocale("English")
uWeekdayNameShort("1969-03-13 10:22:23.231") // 返回 "Thu"
```
将区域设置设置为 “German”
```
uSetLocale("German")
uWeekdayNameShort("1969-03-13 10:22:23.231") // 返回 "Don"
```

uYear

说明 返回指定日期的年份

语法 string uYear([modifiers])

参数 *string modifiers （可选）*
指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例
```
uYear("now") // 返回当前年份
uYear("1969-03-13 10:22:23.231") // 返回 "1969"
```

错误处理

函数	说明
uError	将错误文本写入日志并通知出现错误
uErrortext	返回最后的错误消息
uWarning	将警告消息写入日志

函数	说明
uInfo	将信息性消息写入日志
uTrace	将跟踪消息写入日志
uTracelevel	在日志中设置跟踪消息的详细信息级别

uError

说明 将错误文本写入日志并通知出现错误

语法 string uError(errortext)

参数 *string errortext*

要写入日志文件的文本

示例 通知出错

```
uError("'PP' is no valid country key.")
```

uErrortext

说明 返回最后的错误消息

语法 string uErrortext()

示例

```
uErrortext() // 返回最后的错误文本
```

uInfo

说明 将信息性消息写入日志

语法 string uInfo(infotext)

参数 *string infotext*

要写入日志文件的文本

示例 记录信息性消息

```
uInfo("21445 records selected.")
```

uWarning

说明 将警告消息写入日志

语法 string uWarning(warningtext)

参数 *string warningtext*
要写入日志文件的文本

示例 记录警告消息

```
uWarning("The attribute for the customer name is null.")
```

uTrace

说明 将跟踪消息写入日志。

调用 uTrace() 函数之前，必须手动设置跟踪级别，使其至少为 1。若要将跟踪级别设置为 1，您可以执行下列操作之一：

- 在调用 uTrace() 函数之前先调用 uTracelevel(1)。
- 如果您使用的是 ETL Development，请在安装文件夹的 *etc* 目录的 *Default.ini* 文件中将跟踪级别设置为 1。重新启动 ETL Development。
- 如果您使用的是 ETL Server，请使用 “--tracelevel 1” 选项启动服务器。

语法 string uTrace(tracetext);

参数 *string tracetext*
要写入日志文件的文本

示例

```
uTrace("CUSTOMER_NAME = " + CUSTOMER_NAME)
```

uTracelevel

说明 在日志中设置跟踪消息的详细信息级别。tracelevel 的范围为从 0 （不跟踪）到 5 （非常详细）。

语法 uTracelevel(tracelevel)

注释 大量的日志操作会大幅降低执行速度。

参数 *integer tracelevel*

指定跟踪消息的详细程度。（0 = 关闭， 5 = 非常详细）

示例

```
uTracelevel(5) // 将 tracelevel 设置为“非常详细”
```

文件

函数	说明
uFileInfo	返回有关文件的信息
uFileRead	从文件读取数据
uFileWrite	将数据写入文件

uFileInfo

说明 返回有关文件的信息。将 infotype 设置为 EXISTS 时，函数返回文件的完整路径 （如果文件存在），或返回空字符串 （如果文件不存在）。如果将 infotype 设置为 SIZE，则返回文件的大小或空字符串 （如果文件不存在）。

注释 请记住您必须在 JavaScript 环境中使用双反斜杠，因为反斜杠用作转义序列。

语法 string uFileInfo(file [, infotype])

参数 *string file*

要检查的文件

string infotype （可选）

要获得的信息的种类。缺省值为 EXISTS。

示例 获取文件信息

```
uFileInfo("C:\\windows\\notepad.exe") // 返回
C:\windows\notepad.exe

uFileInfo("C:\\windows\\notepad.exe","SIZE") // 返回
68608
```

uFileRead

说明 从文件读取数据

语法 string uFileRead(URL [, bytes] [, offset] [, encoding])

参数

string URL
指定要读取的源的 URL

integer bytes （可选）
要读取的字节数。缺省值为 *0*，意味着读取整个文件

integer offset （可选）
从文件开头跳过的字节数。缺省值为 *0*。

string encoding （可选）
数据源的编码。缺省编码为 *ISO8859-1*

示例 访问本地文件

```
uFileRead("c:\\myFile.txt")

uFileRead("/home/testuser/myfile.txt")

uFileRead("file:///c:/ myFile.txt")
```

从 Windows 共享位置读取文件

```
uFileRead("\\\\fileserver\\freeShare\\testfile.txt")
```

通过 HTTP 和 HTTPS 读取文件的内容

```
uFileRead("http://http://www.google.com/search?hl=en&q
pizza&btnG=Google+Search")

uFileRead("https://http://www.google.com/search?hl=en&
q=pizza&btnG=Google+Search")
```

通过 FTP 读取文件的内容

```
uFileRead("ftp://myUser:myPasswd@myServer/data/myFile.
txt")
```

uFileWrite

说明　将数据写入文件。如果没有给定 URL，则将数据写入 Sybase ETL 日志目录中的 *write.log* 文件。

语法　string uFileWrite(data [, URL] [, append] [, encoding])

参数

string data
　要写入的数据

string URL （可选）
　用于文件访问和定位的 URL

number append （可选）
　标记 (0/1) 指示是否应当附加数据

string encoding （可选）
　目标文件的编码

示例　通过 CIFS 将数据写入文件

```
uFileWrite("hello",
"//myServer/myShare/data/test.txt")
```

格式设置

函数	说明
uFormatDate	返回带日期信息的用户定义的字符串。

uFormatDate

说明　返回带日期信息的用户定义的字符串。有关格式字符串中将被引用日期部分替换的特殊转义序列的列表，请参见下面的*用法*。

语法　number uFormatDate(format, modifiers, ...)

参数

string format
　返回字符串的格式规范

string modifiers （可选）
　指定日期或日期计算的字符串列表。缺省值为 now 修饰符。

示例

从日期创建字符串

```
uFormatDate("Today is %A the %d of %B in %Y", "now")
//返回类似 "Today is Thursday the 10 of February in 2005" 的内容
```

用法

函数 uFormatDate 返回带日期信息的用户定义的字符串。格式字符串中的特殊转义序列将被引用日期部分替换。

转义序列

转义序列 %A	返回值 星期名称
%a	短星期名称
%B	月份名称
%b	短月份名称
%d	月中某天
%f	小数秒 SS.SSS
%H	小时 00-24
%j	年中某天 000-366
%J	儒略天数
%m	月份
%M	分钟
%s	自从 1970-01-01 以来的秒数
%S	秒 00-59
%w	周中的某天 0-6， 0= 星期日
%W	年中某周
%Y	年份 0000-9999
%%	%

模糊搜索

函数	说明
uGlob	比较类似对其通配符使用 UNIX 文件通配语法区分大小写的值
uLike	比较不区分大小写的值
uMatches	如果给定的字符串与正则表达式匹配，则返回 true

uGlob

说明 比较类似对其通配符使用 UNIX 文件通配语法区分大小写的值。

语法 bool uGlob(pattern, text)

参数

string pattern

说明匹配模式的字符串

string text

要研究的字符串

示例 使用 UNIX 文件通配语法比较值

```
uGlob("Mr. *", "Mr. Smith")   // 返回 1，指示匹配
uGlob("Mr. *", "Mrs. Clarke") // 返回 0
```

通配规则：

“*” – 匹配任何顺序的零个或更多字符。

“?” – 只完全匹配一个字符。

[^...] – 匹配一个不在括起来的列表中的字符。

[...] – 匹配括起来的字符列表中的一个字符。

使用 [...] 和 [^...] 匹配时，可以通过使 “]” 字符成为在 “[” 或 “^” 后面的第一个字符而将该字符包括在列表中。 可以使用 “-” 指定字符的范围。

示例：

“[a-z]” 匹配任何单个小写字母。 若要匹配 “-”，则应使它成为列表中的最后一个字符。

若要匹配 “*” 或 “?”，请将它们放在 “[]” 中。

示例：abc[*]xyz，仅匹配 “abc*xyz”。

uLike

说明

比较不区分大小写的值。

uLike 函数执行模式匹配比较。第一个参数包含模式，第二个参数包含要根据该模式匹配的字符串。模式中的百分比符号 % 与字符串中任何顺序的零个或更多字符匹配。模式中的下划线 _ 与字符串中任何单个字符匹配。其它任何字符与自身匹配，或者与其小写 / 大写等价字符匹配（例如，不区分大小写匹配）。

注释 目前，uLike 只理解大写 / 小写的 7 位拉丁字符，所以 uLike 对 8 位 ISO8859 字符或 UTF-8 字符区分大小写。例如：

`uLike('a' ,'A')` 为 1

`uLike('' ,'')` 为 0

语法

number uLike(pattern, text)

参数

string pattern

说明匹配模式的字符串

string text

要研究的字符串

示例

使用模式匹配比较值

```
uLike("% happy %", "A happy man.") // 返回 1
uLike("% happy %", "A sad man.")   // 返回 0
```

uMatches

说明

如果给定的字符串与正则表达式匹配，则返回 True。

语法

number uMatches(text, regexpr)

参数

string text

要检查的文本

string regexpr

正则表达式规范

示例

检查字符串是否可以解释为浮点数

```
uMatches("abc","[-+]?[0-9]*\\.?[0-9]*")  // 返回 0
uMatches("1.23","[-+]?[0-9]*\\.?[0-9]*")  // 返回 1
```

查找

函数	说明
uChoice	返回由索引指定的给定参数的值
uFirstDifferent	返回与第一个参数不同的第一个参数值
uFirstNotNull	返回第一个非 Null 参数
uElements	返回分隔字符串中的元素个数
uToken	返回分隔字符串中的第 N 个元素

uChoice

说明　返回由索引指定的给定参数的值。索引值是基于零的，因此索引为零时返回第二个参数

语法　string uChoice(index, values, ...)

参数

integer index

引用返回值的索引号。基于零。

string values

值的列表

示例

IF 结构：

```
uChoice(0, "A", "B") // 返回 "A"
uChoice(1, "A", "B") // 返回 "B"
```

CASE 结构：

```
uChoice(2, "n.a.", "Jan", "Feb", "Mar") // 返回 "Feb"
```

模拟您希望用相应的颜色名称替换颜色 ID 的查找函数

```
uChoice(IN.Color, "n.a.", "Red", "Blue", "Green")
```

uFirstDifferent

说明 返回与第一个参数不同的第一个参数值

语法 string uFirstDifferent(params, ...)

参数 *params*

任意数据类型的一系列表达式或值

示例

```
uFirstDifferent("2004-05-01", "2004-05-01", "2005-01-
04", "2005-11-24",) //返回 "2005-01-04"
```

uFirstNotNull

说明 返回第一个非 Null 参数。

语法 string uFirstNotNull(params, ...)

参数 *params*

任意数据类型的一系列表达式或值

示例

```
uFirstNotNull(null, null , "A", "B") // 返回 "A"
```

uElements

说明 返回分隔字符串中的元素个数。如果省略第二个参数，则用空格 (ASCII 32) 作为分隔符。

语法 integer uElements(text [, delimiter])

参数 *string text*

要研究的字符串

string delimiter （可选）

要使用的分隔符。缺省分隔符为*空格*字符。

示例 对分隔字符串中的令牌进行计数

```
uElements("James T. Kirk") // 返回 3
```

uToken

说明　返回分隔字符串中的第 N 个元素。第二个参数指定令牌编号。索引从 1 开始。如果省略第三个参数，则用空格 (ASCII 32) 作为分隔符。

语法　string uToken(text, index [, delimiter])

参数

string text

要研究的字符串

Integer index

要返回的令牌数

string delimiter （可选）

要使用的分隔符。缺省分隔符为**空格**字符。

示例

```
uToken("James T. Kirk", 1) // 返回 "James"

uToken("James T. Kirk", 2) // 返回 "T."
```

杂项

函数	说明
uCommandLine	返回当前进程的命令行字符串
uGetEnv	返回环境变量的值
uGuid	返回全局唯一标识符
uMD5	生成给定字符串的校验和
uScriptLoad	装载和计算 JavaScript，并返回结果
uSetEnv	设置环境变量的值
uSetLocale	将区域设置日期和时间设置更改为不同语言
uSleep	使进程延迟指定的毫秒数
uSystemFolder	返回预定义的应用程序和系统路径

uCommandLine

说明　返回当前进程的命令行字符串

语法　string uCommandLine()

示例

```
uCommandLine() // 返回
"GridNode.exe --port 5124"
```

注释　在 UNIX 上不支持 uCommandLine。

uGetEnv

说明　返回环境变量的值

语法　string uGetEnv(variable)

参数　*string variable*

要读取的环境变量的名称

示例

```
uGetEnv("LOAD_MAX_VALUE")
```

uGuid

说明　返回全局唯一标识符。以下格式可用：

- *numeric* – 仅数字
- *base64* – base64 编码
- *hex* – 没有连字号的十六进制格式

语法　string uGuid([format])

参数　*string format （可选）*

要返回的 GUID 值的格式

示例

```
uGuid() // 例如返回 A8A10D9F-963F-4914-8D6FC8527A50EF2A
```

uMD5

说明　生成固定长度为 32 个字符的给定字符串的校验和。

语法　string uMD5(text)

参数　*string text*

要生成校验和的文本

示例

```
uMD5("Austin Powers") // 返回
"C679A893E3DA2CC0741AC7F527B1D4EB"
```

uScriptLoad

说明　装载和计算 JavaScript，并返回结果

语法　string uScriptLoad(filelocation)

参数 *string filelocation*
要装载的 JavaScript。

示例 装载外部 JavaScript 文件

```
uScriptLoad("\\server3\myScripts\basicFunctions.js")
```

uSetEnv

说明 设置环境变量的值

语法 string uSetEnv(variable, value)

参数 *string variable*
要设置的环境变量的名称

string value
要设置的值

示例

```
uSetEnv("LOAD_MAX_VALUE", IN.Date)
```

uSetLocale

说明 将区域设置日期和时间设置更改为不同语言

语法 string uSetLocale([language] [, country] [, codepage])

参数 *string language （可选）*
要使用的语言字符串 （参见下表）

string country （可选）
要使用的国家 / 地区名称 （参见下表）

string codepage （可选）
Codepage number as string

示例 检索不同语言的月份名称

```
locale:uSetLocale("English")    // 切换到英语
uMonthName("2005-03-22") // 返回 "March"
uSetLocale("German")     // 切换到德语
uMonthName("2005-03-22") // 返回 "Marz"
uSetLocale("C")          // 切换回操作系统缺省值
```

用法

语言字符串

以下语言字符串可以被识别。uSetLocale 只接受操作系统支持的任何语言。

注释 三字母的语言字符串代码仅在 Windows NT 和 Windows 95 中有效。

主语言	子语言	语言字符串
中文	中文	“chinese”
中文	中文 （简体）	“chinese-simplified”或“chs”
中文	中文 （繁体）	“chinese-traditional”或“cht”
捷克语	捷克语	“csy”或“czech”
丹麦语	丹麦语	“dan”或“danish”
荷兰语	荷兰语 （比利时）	“belgian”、“dutch-belgian”或“nlb”
荷兰语	荷兰语 （缺省）	“dutch”或“nld”
英语	英语 （澳大利亚）	“australian”、“ena”或“english-aus”
英语	英语 （加拿大）	“canadian”、“enc”或“english-can”
英语	英语 （缺省）	“english”
英语	英语 （新西兰）	“english-nz”或“enz”
英语	英语 （英国）	“eng”、“english-uk”或“uk”
英语	英语 （美国）	“english”、“americanenglish”、“english-american”、“english-us”、“english-usa”、“enu”、“us”或“usa”
芬兰语	芬兰语	“fin”或“finnish”
法语	法语 （比利时）	“frb”或“french-belgian”
法语	法语 （加拿大）	“frc”或“frenchcanadian”
法语	法语 （缺省）	“fra”或“french”
法语	法语 （瑞士）	“french-swiss”或“frs”

主语言	子语言	语言字符串
德语	德语 （奥地利）	“dea”或“germanaustrian”
德语	德语 （缺省）	“deu”或“german”
德语	德语 （瑞士）	“des”、“german-swiss”或“swiss”
希腊语	希腊语	“ell”或“greek”
匈牙利语	匈牙利语	“hun”或“hungarian”
冰岛语	冰岛语	“icelandic”或“isl”
意大利语	意大利语 （缺省）	“ita”或“italian”
意大利语	意大利语 （瑞士）	“italian-swiss”或“its”
日语	日语	“japanese”或“jpn”
朝鲜语	朝鲜语	“kor”或“korean”
挪威语	挪威语 （博克马尔语）	“nor”或“norwegianbokmal”
挪威语	挪威语 （缺省）	“norwegian”
挪威语	挪威语 （尼诺斯克语）	“non”或“norwegiannynorsk”
波兰语	波兰语	“plk”或“polish”
葡萄牙语	葡萄牙语 （巴西）	“portuguese-brazilian”或“ptb”
葡萄牙语	葡萄牙语 （缺省）	“portuguese”或“ptg”
俄语	俄语 （缺省）	“rus”或“russian”
斯洛伐克语	斯洛伐克语	“sky”或“slovak”
西班牙语	西班牙语 （缺省）	“esp”或“spanish”
西班牙语	西班牙语 （墨西哥）	“esm”或“spanish-mexican”
西班牙语	西班牙语 （现代）	“esn”或“spanish-modern”
瑞典语	瑞典语	“sve”或“swedish”
土耳其语	土耳其语	“trk”或“turkish”

国家 / 地区设置

以下是 uSetLocale 所识别的国家 / 地区字符串的列表。 uSetLocale 只接受操作系统支持的国家 / 地区字符串。三字母国家 / 地区名称代码来自 ISO/IEC （国际标准化组织，国际电工委员会）规范 3166。

国家 / 地区	国家 / 地区字符串
澳大利亚	“aus”或“australia”
奥地利	“austria”或“aut”
比利时	“bel”或“belgium”
巴西	“bra”或“brazil”
加拿大	“can”或“canada”
捷克共和国	“cze”或“czech”
丹麦	“denmark”或“dnk”
芬兰	“fin”或“finland”
法国	“fra”或“france”
德国	“deu”或“germany”
希腊	“grc”或“greece”
香港特别行政区	“hkg”、“hong kong”或“hong-kong”
匈牙利	“hun”或“hungary”
冰岛	“iceland”或“isl”
爱尔兰	“ireland”或“irl”
意大利	“ita”或“italy”
日本	“japan”或“jpn”
韩国	“kor”、“korea”
墨西哥	“mex”或“mexico”
荷兰	“nld”、“holland”或“netherlands”
新西兰	“new zealand”、“new-zealand”、“nz”或“nzl”
挪威	“nor”或“norway”
中华人民共和国	“china”、“chn”、“pr china”或“pr-china”
波兰	“pol”或“poland”
葡萄牙	“prt”或“portugal”
俄罗斯	“rus”或“russia”
新加坡	“sgp”或“singapore”
斯洛伐克共和国	“svk”或“slovak”
西班牙	“esp”或“spain”
瑞典	“swe”或“sweden”
瑞士	“che”或“switzerland”
中国台湾	“taiwan”或“twn”
土耳其	“tur”或“turkey”

国家 / 地区	国家 / 地区字符串
英国	“britain”、“england”、“gbr”、“great britain”、“uk”、“united kingdom”或 “united-kingdom”
美国	“america”、“united states”、“unitedstates”、“us”或 “usa”

uSleep

说明　使进程延迟指定的毫秒数

语法　string uSleep(msecs)

参数　*integer msecs*

要延迟的毫秒数

示例

```
uSleep(1000) // 使进程延迟一秒
```

uSystemFolder

说明　返回预定义的应用程序和系统路径

语法　string uSystemFolder([foldertype])

示例

```
uSystemFolder("APP_LOG")  // 返回日志目录的路径
```

用法

uSystemFolder 函数可以用于访问文件系统上的特殊目录。可以指定以下文件夹：

组	名称	说明
应用程序	APP_MAIN	基本应用程序路径。典型路径为 *C:\Program Files\ETL*
应用程序	APP_LIB	共享库目录，通常位于应用程序目录的 lib 文件夹中
应用程序	APP_LOG	共享库目录，通常位于应用程序目录的 lib 文件夹中
应用程序	APP_CONFIG	配置文件目录，通常在应用程序目录的 etc 文件夹中
应用程序	APP_LICENSE	许可证目录，通常在应用程序目录的 license 文件夹中
应用程序	APP_SCRIPT	脚本目录，通常在应用程序目录的 scripts 文件夹中
应用程序	APP_GRAMMAR	语法目录，通常在应用程序目录的 grammar 文件夹中
应用程序	APP_LANGUAGE	语言文件目录，通常在应用程序目录的 language 文件夹中
应用程序	APP_DATABASE	数据库目录，通常在应用程序目录的 database 文件夹中
应用程序	APP_TEMP	临时目录，通常在应用程序目录的 temp 文件夹中
应用程序	APP_DEMODATA	演示数据目录，通常在应用程序目录的 demodata 文件夹中
应用程序	APP_USERDATA	存储 userspcecific 文件的目录。典型路径为 *C:\Documents and Settings\username\Application Data\ETL*
Windows	ALTSTARTUP	对应于用户的非本地化 “启动”程序组的文件系统目录。
Windows	APPDATA	为特定于应用程序的数据充当公用存储库的文件系统目录。典型路径为 *C:\Documents and Settings\username\Application Data*。
Windows	CDBURN_AREA	充当等待写入 CD 的文件的 staging 区域的文件系统目录。典型路径为 *C:\Documents and Settings\username\Local Settings\Application Data\Microsoft\CD Burning*。
Windows	COMMON_ADMINTOOLS	包含所有计算机用户的管理工具的文件系统目录
Windows	COMMON_APPDATA	包含所有用户的应用程序数据的文件系统目录。典型路径为 *C:\Documents and Settings\All Users\Application Data*。

组	名称	说明
Windows	COMMON_DESKTOPDIRECTORY	文件系统目录，其中包含在所有用户的桌面上出现的文件和文件夹。典型路径为 *C:\Documents and Settings\All Users\Desktop*。仅对于 Windows NT 系统有效。
Windows	COMMON_DOCUMENTS	包含所有用户公用文档的文件系统目录。典型路径为 *C:\Documents and Settings\All Users\Documents*。
Windows	COMMON_FAVORITES	文件系统目录，充当所有用户公用的收藏项的公用存储库。仅对于 Windows NT 系统有效。
Windows	COMMON_MUSIC	文件系统目录，充当所有用户公用的音乐文件的存储库。典型路径为 *C:\Documents and Settings\All Users\Documents\My Music*。
Windows	COMMON_PICTURES	文件系统目录，充当所有用户公用的图像文件的存储库。典型路径为 *C:\Documents and Settings\All Users\Documents\My Pictures*。
Windows	COMMON_PROGRAMS	文件系统目录，其中包含在所有用户的 “开始”菜单上出现的公用程序组的目录。典型路径为 *C:\Documents and Settings\All Users\Start Menu\Programs*。仅对于 Windows NT 系统有效。
Windows	COMMON_STARTMENU	文件系统目录，其中包含出现在所有用户的“开始”菜单上的程序和文件夹。典型路径为 *C:\Documents and Settings\All Users\Start Menu*。仅对于 Windows NT 系统有效。
Windows	COMMON_STARTUP	文件系统目录，其中包含出现在所有用户的“启动”文件夹中的程序。典型路径为 *C:\Documents and Settings\All Users\Start Menu\Programs\Startup*。仅对于 Windows NT 系统有效。
Windows	COMMON_TEMPLATES	文件系统目录，其中包含对所有用户可用的模板。典型路径为 *C:\Documents and Settings\All Users\Templates*。仅对于 Windows NT 系统有效。
Windows	COMMON_VIDEO	文件系统目录，充当所有用户公用的视频文件的存储库。典型路径为 *C:\Documents and Settings\All Users\Documents\My Videos*。
Windows	COOKIES	文件系统目录，充当 Internet cookies 的公用存储库。典型路径为 *C:\Documents and Settings\username\Cookies*。
Windows	DESKTOP	表示 Windows 桌面 （命名空间的根）的虚拟文件夹。

组	名称	说明
Windows	DESKTOPDIRECTORY	文件系统目录，用于物理存储桌面上的文件对象 （不要与桌面文件夹本身混淆）。典型路径为 *C:\Documents and Settings\username\Desktop*。
Windows	FAVORITES	文件系统目录，充当用户收藏夹项目的公用存储库。典型路径为 *C:\Documents and Settings\username\Favorites*。
Windows	FONTS	包含字体的虚拟文件夹。典型路径为 *C:\Windows\Fonts*。
Windows	HISTORY	文件系统目录，充当 Internet 历史项目的公用存储库。
Windows	INTERNET_CACHE	文件系统目录，充当临时 Internet 文件的公用存储库。典型路径为 *C:\Documents and Settings\username\Local Settings\Temporary Internet Files*。
Windows	MYDOCUMENTS	表示 “我的文档” 桌面项目的虚拟文件夹。
Windows	MYMUSIC	文件系统目录，充当音乐文件的公用存储库。典型路径为 *C:\Documents and Settings\User\My Documents\My Music*。
Windows	MYPICTURES	文件系统目录，充当图像文件的公用存储库。典型路径为 *C:\Documents and Settings\username\My Documents\My Pictures*。
Windows	MYVIDEO	文件系统目录，充当视频文件的公用存储库。典型路径为 *C:\Documents and Settings\username\My Documents\My Videos*。
Windows	NETHOOD	文件系统目录，其中包含可能存在于 “网上邻居” 虚拟文件夹中的链接对象。它与 *CSIDL_NETWORK* 不同，后者表示网络命名空间的根。典型路径为 *C:\Documents and Settings\username\NetHood*。
Windows	PERSONAL	表示 “我的文档” 桌面项目的虚拟文件夹。它等同于 MYDOCUMENTS。
Windows	PRINTHOOD	文件系统目录，其中包含可以存在于 “打印机” 虚拟文件夹中的链接对象。典型路径为 *C:\Documents and Settings\username\PrintHood*。
Windows	PROFILE	用户的配置文件文件夹。典型路径为 *C:\Documents and Settings\username*。应用程序不应当在此级别创建文件或文件夹；它们应当将其数据放在 *APPDATA* 或 *LOCAL_APPDATA* 引用的位置下面。

组	名称	说明
Windows	PROGRAM_FILES	“Program Files”文件夹。典型路径为 *C:\Program Files*。
Windows	PROGRAM_FILES_COMMON	跨应用程序共享的组件的文件夹。典型路径为 *C:\Program Files\Common*。仅对于 Windows NT、Windows 2000 和 Windows XP 系统有效。
Windows	PROGRAMS	包含用户的程序组 （它们本身就是文件系统目录）的文件系统目录。典型路径为 *C:\Documents and Settings\username\Start Menu\Programs*。
Windows	RECENT	文件系统目录，其中包含用户最近使用过的文档的快捷方式。典型路径为 *C:\Documents and Settings\username\My Recent Documents*。若要在此文件夹中创建快捷方式，请使用 *SHAddToRecentDocs*。除了创建快捷方式以外，此函数还会更新 shell 的最近文档列表，并将快捷方式添加到 “开始”菜单的 “我最近的文档”子菜单中。
Windows	SENDTO	包含 “发送到”菜单项的文件系统目录。典型路径为 *C:\Documents and Settings\username\SendTo*。
Windows	STARTMENU	包含 “开始”菜单项的文件系统目录。典型路径为 *C:\Documents and Settings\username\Start Menu*。
Windows	STARTUP	对应于用户的 “启动”程序组的文件系统目录。一旦任何用户登录到 Windows NT 或启动 Windows 95，系统就会启动这些程序。典型路径为 *C:\Documents and Settings\username\Start Menu\Programs\Startup*。
Windows	SYSTEM	Windows 系统文件夹。典型路径为 *C:\Windows\System32*。
Windows	TEMPLATES	文件系统目录，充当文档模板的公用存储库。典型路径为 *C:\Documents and Settings\username\Templates*。
Windows	WINDOWS	Windows 目录或 SYSROOT。它对应于 *%windir%* 或 *%SYSTEMROOT%* 环境变量。典型路径为 *C:\Windows*。

网络

函数	说明
uHostname	返回本地网络名称
uSMTP	向 SMTP 服务器发送邮件

uHostname

说明 返回本地网络名称

语法 string uHostname()

示例

```
uHostname()  // 返回类似 "pollux" 或 "castor" 这样的信息
```

uSMTP

说明 向 SMTP 服务器发送邮件

语法

bool uSMTP(serverURL, sender, recipients, subject, body)
bool uSMTP(sender, recipients, subject, body)
bool uSMTP(recipients, subject, body)bool uSMTP(subject, body)
bool uSMTP(body)

参数

string serverURL
URL，用于指定要使用的 SMTP 服务器、端口、用户名和口令

string sender
发件人的电子邮件地址

string recipients
收件人的逗号分隔列表

string subject
邮件的主题

string body
电子邮件的内容

示例

```
uSMTP("Just a mail")

uSMTP("Testmail!", "Just a mail")
```

用法

uSMTP 函数允许使用 SMTP 服务器将电子邮件发送到多个收件人。

指定 SMTP 服务器、端口、用户和口令

用于指定 SMTP 服务器的服务器 URL 的语法如下：

```
protocol://user:password@server:port
```

协议可为下列协议之一

<*empty*> – 带 SSL 加密的 SMTP （如果适用）

SMTP – 无 SSL 加密的 SMTP

SMTPS – 带 SSL 加密的 SMTP

用户和口令 – 为了验证客户端，将使用用户名和口令。如果不提供，则不会进行身份验证。

注释 如果用户名包含 @ 符号，则必须用 # 替换它，以避免含混。

端口 -- 要使用的 TCP 端口，缺省值为 25。

URL 示例

```
myServer
myServer:123
SMTPS://myServer:123
Me:secret@myServer
```

指定收件人 – 可以添加以逗号分隔的地址列表。缺省情况下，直接发送到所有收件人。若要指定抄送或密件抄送，只需添加邮件地址作为前缀。

```
user@host.domain
My Name <user@host.domain>
To: My Name <user@host.domain>
To: user@host.domain
Cc: My Name <user@host.domain>
To: user@host.domain, Bcc: Test User
<test@myserver.com>
```

安全性 – 如果 SMTP 服务器允许使用加密连接进行通信，则这将自动完成。如果提供用户名和口令，则按以下顺序尝试身份验证方法：PLAIN、LOGIN。

自定义缺省值 – 可以在 *INI* 文件中指定个人缺省值。

```
[SMTP]
ServerURL=<your default server URL>
Sender=<your default sender>
Recipients=<your default recipients>
Subject=<your default subject>
```

INI 文件示例：

```
[SMTP]
ServerURL=maxm:secret@mail.gmail.com
Sender= Maxi <Max.Mustermann@ gmail.com>
Recipients=ETLAdmin@MyCompany.com, Cc: QA
qa@MyCompany.com
Subject=ETL Message
```

数字

函数	说明
uAbs	返回实数的大小，忽略其正、负符号
uCeil	返回大于或等于参数的最小整数
uDiv	返回除法整数
uExp	返回以 e 为基数的幂
uFloor	返回小于或等于参数的最大整数
uLn	返回数字的自然对数 （基数 e）
uLog	返回数字的对数
uMod	返回除法的模
uPow, uPower	返回基数表达式取指定次方的值
uRandom	返回随机数字
uRound	返回舍入到最近整数的参数
uSgn	返回给定值的符号
uSqrt	返回给定值的平方根

uAbs

说明 返回实数的大小，忽略其正、负符号

语法 number uAbs(value)

参数 *数字值*

要进行计算的数字

示例

```
uAbs(1522)       // 返回 1522
uAbs('-123.45') // 返回 123.45
uAbs('123ABC')   // 返回 0
```

uCeil

说明　返回大于或等于参数的最小整数

语法　number uCeil(value)

参数　*数字值*
　要进行计算的数字

示例　向上舍入数字

```
uCeil(1523.1) // 返回 1524
uCeil(1523.9) // 返回 1524
```

uDiv

说明　返回除法整数

语法　number uDiv(value, divisor)

参数　*数字值*
　要进行计算的数字
number divisor
　除法的除数。

示例

```
uDiv(10, 3) // 返回 3
```

uExp

说明　返回以 e 为基数的幂

语法　number uExp(value)

参数　*数字值*
　要进行计算的数字

示例

```
uExp(1) // 返回 "2.718281828459045"
```

uFloor

说明 返回小于或等于参数的最大整数

语法 number uFloor(value)

参数 *数字值*
要进行计算的数字

示例
```
uFloor(1523.1) // 返回 1523
uFloor(1523.9) // 返回 1523
```

uLn

说明 返回数字的自然对数 （基数 e）

语法 number uLn(value)

参数 *数字值*
要进行计算的数字

示例
```
uLn(2.718281828) // 返回 0.999999
```

uLog

说明 返回数字的对数

语法 number uLog(value [, base])

参数 *数字值*
要进行计算的数字

number base （可选）
对数的基数。如果省略，则使用 10 作为基数。

示例
```
uLog(100) // 返回 2
uLog(16, 2) // 返回 4
```

uMod

说明 返回除法的模

语法 number uMod(value, divisor)

参数　*数字值*
要进行计算的数字
number divisor
除法的除数
示例

```
uMod(10, 3) // 返回 1
```

uPow, uPower

说明　返回基数表达式取指定次方的值
语法　number uPow(value, exponent)
参数　*数字值*
要进行计算的数字
number exponent
要用作指数的数字
示例

```
uPow(10, 3) // 返回 1000
```

uRandom

说明　返回随机数字
语法　number uRandom()
示例　随机数

```
uRandom() // 返回类似 "0.696654639123727" 这样的值
```

uRound

说明　返回舍入到最近整数的参数
语法　number uRound(value [, scale])
参数　*数字值*
要进行计算的数字
number scale （可选）
数字的位数

示例

```
uRound(10.1) // 返回“10”
uRound(10.49) // 返回“10”
uRound(10.5) // 返回“11”
uRound(10.9) // 返回“11”
uRound(1.235, 2) // 返回 "1.24"
```

uSgn

说明 返回给定值的符号

语法 number uSgn(value)

参数 *数字值*

要进行计算的数字

示例

```
uSgn(-10.4) // 返回 -1
uSgn(0) // 返回 0
uSgn(10.4) // 返回 1
uSgn(null)  // 返回 null
```

uSqrt

说明 返回给定值的平方根

语法 number uSqrt(value)

参数 *数字值*

要进行计算的数字

示例

```
uSqrt(25)     // 返回 5
uSqrt(0)      // 返回 0
uSqrt(null)   // 返回 null
```

脚本

函数	说明
uEvaluate	计算函数或 JavaScript 表达式，并返回结果

uEvaluate

说明　计算函数或 JavaScript 表达式，并返回结果

语法　string uEvaluate(expression)

参数　*数值表达式*
要计算的 JavaScript 代码

示例　计算函数表达式：

```
uEvaluate("3 + 5")
```

```
uEvaluate("parseFloat(IN.Salary) + 1500")
```

定义自定义函数：

```
uEvaluate("function timesTwo(a){ return a*2; }")
```

使用自定义函数：

```
uEvaluate("timesTwo(4)")
```

```
uEvaluate("timesTwo(IN.Salary)")
```

计算脚本：

```
uEvaluate("if ("parseFloat(IN.Salary) > 2000") {2000;}
else {("parseFloat(IN.Salary) + 500");}")
```

字符串

函数	说明
uAsc, uUnicode	返回指定字符的 Unicode 字符值
uChr, uUniChr	返回对应给定数字的 Unicode 字符串，或设置某个字符串的格式

函数	说明
uCap	返回字符串的大写表示
uCon, uConcat	将所有给定的参数并置成单个字符串
uJoin	用特殊的 Null 和空值处理方式并置分隔字符串
uLeft	返回字符串最左侧的 N 个字符
uLength, uLen	返回字符串的长度
uSubstr, uMid	返回字符串的一部分
uLPos	查找字符串中某个子字符串的第一个位置
uLower, uLow	返回小写字母的输入字符串
uLStuff	将字符串的左侧填充到指定的长度
uLTrim	从字符串的左侧删除字符
uRepeat	返回给定字符串重复 N 次后的结果
uReplace	替换字符串的某些部分
uReverse	颠倒字符串
uRight	返回字符串最右侧的 N 个字符
uRPos	在字符串中查找子字符串的最后一个位置
uRStuff	将字符串的右侧填充到指定的长度
uRTrim	从字符串的右侧删除字符
uTrim	从字符串的两侧删除字符
uUpper, uUpp	返回大写字母的输入字符串

uAsc, uUnicode

说明 返回指定字符的 Unicode 字符值。

语法 number uAsc(value [, index])

参数 *string value*

输入字符串

number index （可选）

用于读取 ASCII 值的字符位置

示例

```
uAsc("Big Ben")     // 返回 66
uAsc("Big Ben", 2) // 返回 105
```

uChr, uUniChr

说明 返回对应给定数字的 Unicode 字符串，或设置某个字符串的格式

语法 string uChr(params, ...)

参数 *params*

表达式或值的列表

示例

```
uChr(64) // 返回 "@"
uChr("\u0064\u006f\u0067") // 返回 "dog"
uChr(65, "pple ") // 返回 "apple"
```

uCap

说明 返回字符串的大写表示。换句话说，字符串中的每个单词的第一个字母大写。

语法 string uCap(text)

参数 *Input text*

要大写的字符串

示例

```
uCap('fArmeR, ASTROnaut') // 返回 'Farmer, Astronaut'
uCap('the first weekend') // 返回 'The First Weekend'
```

uCon, uConcat

说明 将所有给定的参数并置成单个字符串

语法 string uConcat(params)

参数 *params*

任意数据类型的一系列表达式或值

示例

```
uConcat("For ", 3, " years.") //返回 "For 3 years."
```

uJoin

说明 用特殊的 Null 和空值处理方式并置分隔字符串

语法 string uJoin(delimiter, allowEmpty, params, ...)

参数 *string delimiter*
要在其它所有字符串部分之间使用的分隔符
number allowEmpty
标志 (0/1) 指示是否允许空字段
string params
要并置的字符串列表

示例
```
uJoin("-", 1, "James", "", "Tiberius", "Kirk") //
返回 "James--Tiberius-Kirk"

uJoin("-", 0, "James", "", "Tiberius", "Kirk") //
返回 "James-Tiberius-Kirk"
```

uLeft

说明 返回字符串最左侧的 N 个字符

语法 string uLeft(input, chars)

参数 *string input*
输入字符串
number chars
要检索的字符数

示例
```
uLeft("James T. Kirk", 5) // 返回 "James"
uLeft(null, 5)            // 返回 null
```

uLength, uLen

说明 返回字符串的长度

语法 number uLength(input)

参数 *string input*
输入字符串

示例
```
uLength("James T. Kirk") // 返回 13
```

uSubstr, uMid

说明　返回字符串的一部分

语法　string uSubstr(input, position, length)

参数　*string input*
　输入字符串
number position
　从其开始读取的位置
number length
　要读取的字符数

示例
```
uSubstr("James T. Kirk", 7, 2) // 返回 "T."
```

uLPos

说明　查找字符串中某个子字符串的第一个位置。结果为零表示未找到子字符串。

语法　string uLPos(input, substring)

参数　*string input*
　输入字符串
string substring
　要搜索的子字符串

示例
```
uLPos("James T. Kirk", "T") //返回 7
```

uLower, uLow

说明　返回小写字母的输入字符串

语法　string uLower(input)

参数　*string input*
　要转换的字符串

示例
```
uLower("James T. Kirk") // 返回 "james t. kirk"
```

uLStuff

说明 将字符串的左侧填充到指定的长度

语法 string uLStuff(input, length [, stuff])

参数 *string input*
要填充的字符串
number length
字符串的新长度
string stuff（可选）
要追加的字符串，缺省值为空格 (ASCII 32)

示例
```
uLStuff("3.5", 5)       // 返回 "3.5"
uLStuff("3.5", 5, "0") // 返回 "003.5"
```

uLTrim

说明 从字符串的左侧删除字符。如果省略第二个参数，则它缺省为空格 (ASCII 32)。

语法 string uLTrim(input, trimstring)

参数 *string input*
要剪裁的字符串
string trimstring
要调整的字符串

示例
```
uLTrim(" 3.5")        // 返回 "3.5"
uLTrim("003.5", "0") // 返回 "3.5"
```

uRepeat

说明 返回给定字符串重复 N 次后的结果

语法 string uRepeat(input, repeats)

参数 *string input*
要重复的字符串
number repeats
输入字符串的重复次数

示例
```
uRepeat("Hello ", 4) // 返回 "Hello Hello Hello Hello "
```

uReplace

说明　替换字符串的某些部分

语法　string uReplace(input, search, replace)

参数　*string input*
　要处理的字符串
string search
　要搜索的模式
string replace
　将替换任何匹配项的字符串

示例
```
uReplace("At four o' clock he became four", "four", "4")
// 返回 "At 4 o' clock he became 4"
```

uReverse

说明　颠倒字符串

语法　string uReverse(input)

参数　*string input*
　要颠倒的字符串

示例
```
uReverse("Smith")  // 返回 "htimS"
```

uRight

说明　返回字符串最右侧的 N 个字符

语法　string uRight(input, chars)

参数　*string input*
　输入字符串
number chars
　要读取的字符个数

示例
```
uRight("James T. Kirk", 4) // 返回 "Kirk"
uRight(null, 5) / / 返回 null
```

uRPos

说明　在字符串中查找子字符串的最后一个位置

语法　string uRPos(input, substring)

参数　*string input*
　输入字符串
string substring
　要查找的子字符串

示例　查找最后一次出现的子字符串

```
uRPos("James T. Kirk", "T") //返回 7
```

uRStuff

说明　将字符串的右侧填充到指定的长度

语法　string uRStuff(input, length [, stuffstring])

参数　*string input*
　输入字符串
number length
　结果字符串的新长度
string stuffstring (optional)
　要附加的字符串

示例

```
uRStuff("3.5", 5)       // 返回 "3.5  "
uRStuff("3.5", 5, "0") // 返回 "3.500"
```

uRTrim

说明　从字符串的右侧删除字符

语法　string uRTrim(input [, trimstring])

参数　*string input*
　输入字符串
string trimstring （可选）
　要调整的字符串

示例

```
uRTrim("3.5 ")        // 返回 "3.5"
uRTrim("3.500", "0") // 返回 "3.5"
```

uTrim

说明　从字符串的两侧删除字符

语法　string uTrim(input [, trimstring])

参数　*string input*
输入字符串
string trimstring （可选）
要调整的字符串

示例
```
uTrim(" 3.5 ")          // 返回 "3.5"
uTrim("003.500", "0") // 返回 "3.5"
```

uUpper, uUpp

说明　返回大写字母的输入字符串

语法　string uUpper(input)

参数　*string input*
输入字符串

示例
```
uUpper("James T. Kirk") // 返回 "JAMES T. KIRK"
```

运算符

函数	说明
uEQ	如果两个参数相等并且均不为 NULL，则返回 1
uNE	如果两个参数不相等并且均不为 NULL，则返回 1
uGT	如果第一个参数大于第二个参数，并且二者均不为 NULL，则返回 1
uGE	如果第一个参数大于或等于第二个参数，并且二者均不为 NULL，则返回 1
uLT	如果第一个参数小于第二个参数，并且二者均不为 NULL，则返回 1
uLE	如果第一个参数小于或等于第二个参数，并且二者均不为 NULL，则返回 1

注释 为了保持兼容，运算符也作为函数提供。

uEQ

说明 如果两个参数相等并且均不为 NULL，则返回 1

语法 number uEQ(value1, value2)

参数 *value1, value2*

要比较的数值或字符串值

示例

```
uEQ(1,2)    // 返回 0
uEQ(1,1)    // 返回 1
uEQ(null,1) // 返回 0
```

uNE

说明 如果两个参数不相等并且均不为 NULL，则返回 1

语法 number uNE(value1, value2)

参数 *value1, value2*

要比较的数值或字符串值

示例

```
uNE(1,2)    // 返回 1
uNE(1,1)    // 返回 0
uNE(null,1) // 返回 0
```

uGT

说明 如果第一个参数大于第二个参数，并且二者均不为 NULL，则返回 1

语法 number uGT(value1, value2)

参数 *value1, value2*

要比较的数值或字符串值

示例

```
uGT(2,1)    // 返回 1
uGT(1,2)    // 返回 0
uGT(1,1)    // 返回 0
uGT(null,1) // 返回 0
```

uGE

说明　如果第一个参数大于或等于第二个参数，并且二者均不为 NULL，则返回 1

语法　number uGE(value1, value2)

参数　*value1, value2*

要比较的数值或字符串值

示例

```
uGE(2,1)     // 返回 1
uGE(1,2)     // 返回 0
uGE(1,1)     // 返回 1
uGE(null,1) // 返回 0
```

uLT

说明　如果第一个参数小于第二个参数，并且二者均不为 NULL，则返回 1

语法　number uLT(value1, value2)

参数　*value1, value2*

要比较的数值或字符串值

示例

```
uLT(2,1)     // 返回 0
uLT(1,2)     // 返回 1
uLT(1,1)     // 返回 0
uLT(null,1) // 返回 0
```

uLE

说明　如果第一个参数小于或等于第二个参数，并且二者均不为 NULL，则返回 1

语法　number uLE(value1, value2)

参数　*value1, value2*

要比较的数值或字符串值

示例

```
uLE(2,1)     // 返回 0
uLE(1,2)     // 返回 1
uLE(1,1)     // 返回 1
uLE(null,1) // 返回 0
```

三角函数

函数	说明
uAcos	返回数字的反余弦值 （以弧度为单位）
uAsin	返回数字的反正弦值 （以弧度为单位）
uAtan	返回数字的反正切值 （以弧度为单位）
uCos	返回数字的余弦值 （以弧度为单位）
uSin	返回数字的正弦值 （以弧度为单位）
uTan	返回数字的正切值 （以弧度为单位）

uAcos

说明 返回数字的反余弦值 （以弧度为单位）

语法 number uAcos(value)

参数 *数字值*
输入值

uAsin

说明 返回数字的反正弦值 （以弧度为单位）

语法 number uAsin(value)

参数 *数字值*
输入值

uAtan

说明	返回数字的反正切值 （以弧度为单位）
语法	number uAtan(value)
参数	*数字值* 输入值

uCos

说明	返回数字的余弦值 （以弧度为单位）
语法	number uCos(value)
参数	*数字值* 输入值

uSin

说明	返回数字的正弦值 （以弧度为单位）
语法	number uSin(value)
参数	*数字值* 输入值

uTan

说明	返回数字的正切值 （以弧度为单位）
语法	number uTan(value)
参数	*数字值* 输入值

附 录 B

连接参数

本附录描述了数据库配置选项。此外，还提供了有关部分所支持接口的其它信息。

特定于接口的数据库选项

DB 选项	接口及缺省值						说明
	Oracle	ODBC/DB2	Sybase 15	Sybase	OLEDB	SQLite Persistent	
Allow fallback	1 （不使用）	1 （不使用）	1	1	1 （不使用）	1 （不使用）	如果设置为 1，则在 bulk 操作不可用时会使用标准装载操作。
Always use logon credentials	不可用	0	不可用	不可用	不可用	不可用	当生成 ODBC 连接字符串时，每次均会将凭证添加到连接字符串中。
API trace	不可用	不可用	False	False	不可用	不可用	启用 CTLIB 跟踪功能。
API version	不可用	不可用	150	125	不可用	不可用	CTLIB API 版本的兼容性。
Auto commit	不可用	1 如果数据库为 IQ 或 ODBC 驱动程序为 ASA ODBC 9，则 ODBC 将使用 0 而非用户输入的值。	0	0	不可用	不可用	将连接置于自动提交模式。
Auto vacuum	不可用	不可用	不可用	不可用	不可用	0	当从数据库删除对象时将空间释放出来。

DB 选项	接口及缺省值						说明
	Oracle	ODBC/DB2	Sybase 15	Sybase	OLEDB	SQLite Persistent	
Block inserts	不可用	auto 只有在 ODBC 驱动程序为 ASA9，值为 auto，不存在 LOB 类型，且 “Write rejected records to file” 选项为空时才会使用 bulk 操作。	不可用	不可用	不可用	不可用	准备用于执行的 SQL 语句时请使用块智能绑定。
BLOB chunk size	1024 从文件提取 LOB 数据时，ETL 每次从文件中提取 1024 字节并将其写入数据库中。	不可用	不可用	不可用	不可用	不可用	确定 LOB 在多大时将被截断。
BLOB fetch mode	LOB_INLINE	INLINE	不可用	不可用	不可用	不可用	BLOB 数据将写入辅助文件或保存在内存中。如果设置为 “INLINE”，则数据将保存在内存中。如果设置为 “FILE”，则数据将暂时写入磁盘中。
Busy timeout	不可用	不可用	不可用	不可用	不可用	10	创建一个处理程序，在遇到锁定的数据库表时等待指定的秒数。
Cache size	不可用	不可用	不可用	不可用	不可用	3000	高速缓存中要使用的页数。
CLIENT_CHARSET	不可用	不可用	-	-	不可用	不可用	与 CTLIB 共同使用的字符集（用户定义）。
Client Conversion	不可用	不可用	0 (ASE) 1 (ASA/IQ)	0 (ASE) 1 (ASA/IQ)	不可用	不可用	控制 Client Library 是否应将数据转换为相应形式。
Connect timeout	0 （不使用）	0 （不使用）	0	0	10	0	在经过 Connect timeout 设置的秒数后停止连接尝试。如果设置为 0，则连接不会超时。
CONVERTER_CHARSET	不可用	不可用	-	-	不可用	不可用	CLIENT_CONVERSION = 1 时使用的字符集。
Database name	不可用	不可用	-	-	不可用	不可用	数据库名称。
DBMS_VER	不可用	不可用	-	-	不可用	不可用	数据库的版本。
Default cache size	不可用	不可用	不可用	不可用	不可用	3000	高速缓存中使用的缺省页数。

DB 选项	接口及缺省值						说明
	Oracle	ODBC/DB2	Sybase 15	Sybase	OLEDB	SQLite Persistent	
Disconnect timeout	不可用	10 在 32 位 Windows 中，ETL 始终使用缺省值。 在其它平台上不使用此选项。	不可用	不可用	不可用	不可用	试图从数据库断开时，如果在 *n* 秒内未收到数据库的答复，则强制从数据库断开。
Enable bulk load	1 （不使用）	1/ 不使用	1	1	1 （不使用）	1	如果设置为 1，则会使用 bulk 操作（如果适用）。
Enable SQL Server fast load	不可用	不可用	不可用	不可用	1	不可用	如果此选项设置为 1，则会启用 MS SQL Server 快速装载功能。如果设置为 0 则会禁用此功能。
Execution timeout	0 （不使用）	0/ 不使用	-1	-1	不可用	0	经过一定的时间间隔（秒）后组件将停止执行。（0 <= 表示不设超时时间）。
Extended connect options	不可用	-	不可用	不可用	-	不可用	允许将其它特定于驱动程序的参数添加到 ODBC 连接字符串中。
Full column names	不可用	不可用	不可用	不可用	不 可用	1	设置为 1 时，将按照以下模式将列名完全限定： <table-name/alias> <column-name>。
Internal database	不可用	不可用	不可用	不可用	不 可用	-	数据库引用。
Isolation level	DEFAULT （不使用）	DEFAULT	DEFAULT （不使用）	DEFAULT （不使用）	不可用	DEFAULT （不使用）	定义某个事务必须与其它事务作出的资源修改或数据修改相互隔离的级别。
Lock resultset data	0 （不使用）	0	0（不使用）	0（不使用）	不可用	0	查询表将被锁定。此选项用于确保当进程正在使用选定的记录集时不会向该记录集中写入任何数据。从选定的记录集中提取完最后一条记录后，将释放该记录集。
Log SQL Statements to a file	0	0	0	0	0	0	如果设置为 1，则所有 SQL 语句均将记录到日志或 *SQL.log* 文件中。

DB 选项	接口及缺省值						说明
	Oracle	ODBC/DB2	Sybase 15	Sybase	OLEDB	SQLite Persistent	
Numeric support	不可用	0 当 DBMS 为 IQ 或 ODBC 驱动程序为 ASA9 时，此值为 0 而非用户输入的值。	不可用	不可用	不可用	不可用	是否要启用 ODBC 数值支持。
Object name end quote	"	- ODBC 使用从 DBMS 查询得到的值，而不使用用户输入。	]	]	-	不可用	创建 SQL 语句时，终止字符将用作引号。
Object name start quote	"	"" ODBC 使用从 DBMS 查询得到的值，而不使用用户输入。	[	[	-	不可用	生成 SQL 语句时将用作引号的起始字符。
PAD_BLANKS	不可用	不可用	0	0	不可用	不可用	使用空格字符保持固定列宽。
Page size	不可用	不可用	不可用	不可用	不可用	4096	每页的字节数（必须是 2 的幂），大于或等于 512 且不超过 32768。
Quote character	"	不可用	不可用	不可用	不可用	不可用	与 QUOTE_START 和 QUOTE_END 相同。
Quote object names	0 （不使用）	0	0	0	0	0	若设置为 1，则将使用在 QUOTE_START 和 QUOTE_END 中指定的字符来括起所生成 SQL 语句中的标识符。
Reject log column delimiter	tab	tab	tab	tab	-	tab	用作拒绝日志中的列分隔符。
Short column names	不可用	不可用	不可用	不可用	不可用	0	如果标志设置为 false，则对列名进行完全限定，否则只需列名即可引用列。
Show all tables	不可用	不可用	0 （不使用）	0 （不使用）	不可用	不可用	显示系统表及用户表。
Show error location	1	1	1	1	1	1	设置为 1 时将显示错误所在的位置。数据库错误包含结果集内的记录所在的位置。

DB 选项	接口及缺省值						说明
	Oracle	ODBC/DB2	Sybase 15	Sybase	OLEDB	SQLite Persistent	
SHOW_ ERROR_ LOCATION_ ABSOLUTE_ ROWS	1	1	1	1	1	1	显示从结果集的绝对起始处算起的错误位置而非当前结果集。
Synchronous	不可用	不可用	不可用	不可用	不可用	0	如果选择 Full =2，则 SQLite 会确保在继续执行前将数据写入磁盘中。 如果选择 Normal=1，则将暂停以在重要时刻写入，但不如 Full 时频繁。 如果选择 Off = 0，则数据将传递给操作系统且 SQLite 会立即继续执行。
Temp store	不可用	不可用	不可用	不可用	不可用	2	如果设置为 1，则临时数据库所在的位置为文件。如果设置为 2，则临时数据库所在的位置为内存。
Treat numeric value as character	不可用	1 如果数据库为 IQ，或者驱动程序的名称包含“SYSYBNT”或“LIBDB2.A”，则 ODBC 将使用 1 而非用户输入的值。	不可用	不可用	不可用	不可用	将数值数据强制转换为字符串。
Truncate reject log	1	1	1	1	1	1	日志将在数据库连接时截断。如果值为 0，则会将数据追加到现有的日志文件中。
Use DELETE instead of TRUNCATE	0	0	0	0	0	0	如果设置为 1，则会使用 DELETE 语句而非 TRUNCATE。
Use system views	True	不可用	不可用	不可用	不可用	不可用	使用 DBA 系统表来显示 元数据而非每个用户的元数据。
Validate result column binding	不可用	不可用	不可用	不可用	1	不可用	如果设置为 1，则会在从数据库读取数据时对结果列映射的绑定进行验证。
Write empty dates as NULL	0	0	0 （不使用）	0 （不使用）	不可用	不可用	如果设置为 1，则会将空日期的值强制为 NULL。

DB 选项	接口及缺省值						说明
	Oracle	ODBC/DB2	Sybase 15	Sybase	OLEDB	SQLite Persistent	
Write error code to reject log	1	1	1	1	1	1	将错误代码添加到拒绝日志中。
Write error text to reject log	1	1	1	1	1	1	将错误文本添加到拒绝日志中。
Write header to reject log	0	0	0	0	0	0	指定拒绝日志是否应包含标头行。
Write rejected records to file	-	-	-	-	-	-	指定拒绝日志的文件路径。 此选项用于记录装载时数据库所拒绝的记录。

注意：

- 连字符 (-) 表示数据库选项没有缺省值。您可以输入合适的值。
- “不可用”表示不为基础接口提供数据库选项。
- “不使用”虽然显示数据库选项，但基础接口并不使用这些选项。

数据库和接口支持

下表列出了适用于各种数据库组件类型的接口和数据库。

表 B-2：数据库和接口支持矩阵

Interface	DB Data Provider 和 DB Lookup	DB Data Sink	DB Bulk Load Sybase IQ	DB Staging	IQ Loader File via Load Table	IQ Loader DB via Insert Location
Sybase	ASE ASA IQ	IQ	IQ	ASE ASA IQ	IQ	IQ
ODBC	可通过 ODBC 访问的任何数据源。	IQ MS-Access	IQ	MS-Access IQ ASA ASE	IQ	不支持
OLE DB	IQ SQL-Server	IQ	IQ	IQ	不支持	不支持
Oracle	可通过 Oracle Call Interface (OCI) 访问的任何 Oracle 数据库系统。	不支持	不支持	不支持	不支持	不支持
DB2	可通过 IBM DB/2 客户端接口访问的任何 DB2 数据库系统。	不支持	不支持	不支持	不支持	不支持
SQLite Persistent	任何 SQLite 数据库文件。	不支持	不支持	不支持	不支持	不支持

Sybase ETL 环境已经过充分测试、评估和验证，它与支持的数据库系统的许多接口驱动程序相符。

如果遇到可能与驱动程序兼容性相关的意外结果，请尝试安装接口驱动程序的支持版本之一。有关 Sybase ETL 4.8 支持的所有接口驱动程序版本的列表，请参见 *Sybase ETL 4.8 发行公告* 的 “接口支持”。

使用 SQLite Persistent 接口

Sybase ETL 技术包含用于临时数据存储和暂存的内置式通用关系数据库。它基于 SQLite 这一速度极高、使用广泛且符合 SQL-92 绝大部分要求的数据库。SQLite 是一种小型 C 库，它实现了自我包含的可嵌入零配置 SQL 数据库引擎。

注释 SQLite 只能在测试环境中使用。请勿将其用于生产环境。

连接到 SQLite 数据库

SQLite 数据库用一个带有 *.db* 扩展名的单个文件来表示。该数据库文件可以包含任意数目的表。

❖ **创建或连接至 SQLite 数据库文件**

1. 在 “Properties”窗口中，从 “Interface”菜单中选择 “SQLite Persistent”。
2. 提供 SQLite 数据库文件的主机名。
   - 若要创建新的 SQLite 数据库文件，请在 “Host Name”字段中提供名称，但不要将 *.db* 扩展名包含在内。将在缺省位置，即安装文件夹下的 *database* 目录中自动创建带有 *.db* 扩展名的新 SQLite 数据库文件。

     若要在非缺省目录中创建 SQLite 数据库文件，请在 “Host Name”字段中指定包含 *.db* 扩展名的完整路径。
   - 如果要连接至缺省目录中的现有 SQLite 数据库文件，请从 “Host Name”菜单中选择文件名。不要输入 *.db* 扩展名。如果要连接至非缺省目录中的 SQLite 数据库文件，请在 “Host Name”字段中指定包含 *.db* 扩展名的完整路径。

示例 – 创建一个新的 SQLite 数据库文件 *mySQLite.db* 或连接至现有 *mySQLite.db* 数据库文件：

- 接口：SQLite Persistent
- 主机名：mySQLite

所创建的数据库文件名为 *mySQLite.db*。

创建 SQLite 表

可以用以下方法之一创建 SQLite 表：

- 右键单击 Staging 组件，然后选择 “Create Staging Table from Input” 或 “Create Staging Table from port”。
- 右键单击 Data Sink 组件中的一个，然后选择 “Add Destination Table from Input” 或 “Add Destination Table from Port”。

从 SQLite 数据库提取数据

为 DB 组件上的 SQLite 数据库文件提供相应的连接参数。

可以在连接到数据库的组件的预处理或后处理 SQL 属性中使用 SQLite 支持的 SQL 命令。

使用 “Tools” 菜单中的 “Content Explorer” 可以在项目中操作或浏览连接至组件的 SQLite 数据库对象。另外，您可以使用从 *sqlite.org* 获取的客户端应用程序连接到 SQLite 数据库文件。

注释 若要使用外部客户端应用程序连接到 SQLite 数据库文件，您必须熟知 SQLite 的锁定策略。

使用 Oracle 接口

表 B-2 列出了从 Sybase ETL 数据类型到 Oracle 数据类型的类级别转换。

表 B-2：从 Oracle 接口到 Sybase ETL 的数据类型映射

ETL 数据类型	Oracle 数据类型	大小 / 精度	最小标度	最大标度
binary	BLOB	2147483647		
binary	BFILE	2147483647		
binary	RAW	2000	0	0
string	CLOB	2147483647	0	0
string	CHAR	2000	0	0
float	DECIMAL	38	0	0
integer	NUMBER	38	0	0
float	DOUBLE PRECISION	15	0	38
datetime	DATE	19	0	0
datetime	TIMESTAMP	28	0	9
string	VARCHAR2	4000	0	0
unicode	NCHAR	1000	0	0
unicode	NVARCHAR2	2000	0	0
unicode	NCLOB	2147483647	0	0

附 录 C

将 ETL 用于渐变维度

本章对渐变维度 (SCD) 进行了概述。它列出了一些常见的 SCD 应用场景并说明如何使用 ETL 项目和作业实现这些应用场景。

概述

渐变维度是一种常用的数据仓库技术应用场景。SCD 利用三种不同的方法类型来处理对数据仓库维度表中列的更改。

第 1 类 SCD

在第 1 类 SCD 中，新数据将覆盖现有数据。现有数据将丢失，并且不会跟踪历史更改。第 1 类 SCD 易于维护，但只有在您不需要跟踪历史更改时才有用。

*示例：*假设有一个保存产品信息的表。

Key	Name	Price
1	Notebook	1200

笔记本电脑的价格涨到 1500。更新后的表直接覆盖当前记录：

Key	Name	Price
1	Notebook	1500

第 2 类 SCD

第 2 类 SCD 保留值的完整历史记录。如果新数据与旧数据不同，将用新数据值创建附加维度记录，并且该记录将成为当前记录。每条记录都包含有效日期和到期日期，用以标识该记录处于活动状态的时间段。使用第 2 类 SCD 可以保留表中维度数据的完整历史记录。

*示例：*请参见第 252 页上的 “案例研究应用场景”。

第 3 类 SCD

第 3 类 SCD 使用单独的列跟踪更改。有一个版本的维度记录用来存储所选属性的先前值和当前值。当您需要跟踪仅在有限时间内发生的历史更改时，请使用第 3 类 SCD。

*示例：*您有一个保存产品信息的表。

Key	Name	Price
1	Notebook	1200

笔记本电脑的价格在 2008 年 7 月 15 日涨到 1500。为容纳第 3 类 SCD，添加了 Current Price 和 Effective Date 这两个新列：

Key	Name	Original Price	Current Price	Effective Date
1	Notebook	1200	1500	2008-07-15

注释 因为只有在进行非常重要的更改时才应修改维度表的结构，所以很少使用第 3 类。

案例研究应用场景

本节提供了第 2 类 SCD 的案例研究应用场景，并说明如何在 Sybase ETL 中创建转换项目来实现此应用场景。

案例说明

您有两个表：

- PRODUCT，位于日常运作数据库或源数据库中。
- PRODUCT_PRICE，位于数据仓库或目标数据库中。此表跟踪源表 (PRODUCT) 中产品随时间的变化发生的修改，如：
  - 现有产品的价格的更改
  - 新增的产品
  - 删除的产品

PRODUCT 表的数据库表模式如下：

列	说明
Key	产品的唯一 ID。
Name	产品名称。
Price	产品价格。

PRODUCT_PRICE 表的数据库表模式如下：

列	说明
Key	源表 (PRODUCT) 中的源键标识符。
Name	产品名称。
Price	产品价格。
Valid From	新记录的插入日期。
Valid To	记录的失效日期。将新记录插入 PRODUCT_PRICE 表时，具有相同源键的记录将失效。

规则

将维度从 PRODUCT 表传送到 PRODUCT_PRICE 表的规则为：

1 如果相应记录在 PRODUCT_PRICE 表中不存在，则使用与 PRODUCT 表相同的列值创建该记录。将 Valid From 日期设置为记录插入日期，将 Valid To 日期设置为 9999-12-31.

2 如果相应记录在 PRODUCT_PRICE 表中存在，但未进行任何更改，则不要插入新记录或更新现有记录。

3 如果相应记录在 PRODUCT_PRICE 表中存在，且记录的价格已更改：

- 将带有旧价格的记录的 Valid To 日期设置为昨天。
- 创建一条新记录。将 Valid From 日期设置为记录插入日期，将 Valid To 日期设置为 9999-12-31。

4 如果相应记录在 PRODUCT_PRICE 表中存在，但已从 PRODUCT 表中删除，则将 PRODUCT_PRICE 表中产品的 Valid To 日期设置为昨天。

注释 基于以上规则， PRODUCT_PRICE 表中维护维度更改的不间断历史记录。

工作原理

在 2008 年 1 月 1 日最初装载后， PRODUCT *表显示如下：*

Key	Name	Price
1	Notebook	1200
2	Monitor	1000
3	Mouse	500

ETL 进程首次运行后， PRODUCT_PRICE 表显示如下：

Key	Name	Price	Valid From	Valid To
1	Notebook	1200	2008-01-01	9999-12-31
2	Monitor	1000	2008-01-01	9999-12-31
3	Mouse	500	2008-01-01	9999-12-31

在 2008 年 1 月 15 日，显示器的价格修改后， PRODUCT *表更新。*

Key	Name	Price
1	Notebook	1200
2	Monitor	1400
3	Mouse	500

重新运行 ETL 进程后， PRODUCT_PRICE 表显示如下：

Key	Name	Price	Valid From	Valid To
1	Notebook	1200	2008-01-01	9999-12-31
2	Monitor	1000	2008-01-01	2008-01-14
3	Mouse	500	2008-01-01	9999-12-31
2	Monitor	1400	2008-01-15	9999-12-31

*在 2008 年 1 月 22 日，添加硬盘这一新产品后，*PRODUCT *表再次更新。*

Key	Name	Price
1	Notebook	1200
2	Monitor	1400
3	Mouse	500
4	Hard Disk	1000

重新运行 ETL 进程后， PRODUCT_PRICE 表显示如下：

Key	Name	Price	Valid From	Valid To
1	Notebook	1200	2008-01-01	9999-12-31
2	Monitor	1000	2008-01-01	2008-01-14
3	Mouse	500	2008-01-01	9999-12-31
2	Monitor	1400	2008-01-15	9999-12-31
4	Hard Disk	1000	2008-01-22	9999-12-31

2008 年 7 月 28 日，PRODUCT 表再次更新：
从该表中删除了一种产品，即鼠标。

Key	Name	Price
1	Notebook	1200
2	Monitor	1400
4	Hard Disk	1000

重新运行 ETL 进程后，PRODUCT_PRICE 表显示如下：

Key	Name	Price	Valid From	Valid To
1	Notebook	1200	2008-01-01	9999-12-31
2	Monitor	1000	2008-01-01	2008-01-14
3	Mouse	500	2008-01-01	2008-07-27
2	Monitor	1400	2008-01-15	9999-12-31
4	Hard Disk	1000	2008-01-22	9999-12-31

为 SCD 设置 ETL 项目

本节介绍使用项目和作业完成第 2 类 SCD 的 ETL 概念。随产品打包提供的 Demo Repository 包括与 SCD 第 2 类使用案例有关的各种 ETL 转换对象，其中包括：

项目

- Demo Product Price SCD – Initial Load

  此项目初始化或重新初始化 SCD - Update 项目
  SCD – 更新项目和作业。

  注释 该项目不是使用案例实现的一部分。在生产环境中，对空目标表首次执行 “Update New and Modified”项目时将执行初始装载，其中所有源记录将作为新记录处理。由于演示环境使用两个不同的表来模拟对源数据的更改，因此始终需要在目标表中恢复原始数据，方法是在运行其它两个更新项目或作业之前执行该项目。

  Demo Product Price SCD – Update New and Modified

  此项目每天更新目标数据库的维度表，以反映源数据库中产品的修改或添加情况。请参见第 252 页上的 “案例研究应用场景”中的规则 1-3。

  若要完成完全更新，还应执行 SCD - Update Deleted 项目。

- Demo Product Price SCD – Update Deleted

  此项目每天更新目标数据库的维度表，以反映源数据库中产品的删除情况。请参见第 252 页上的 “案例研究应用场景”中的规则 4。

  若要完成完全更新，还应执行 “SCD - Update New and Modified” 项目。

作业

- Demo Product Price SCD – Daily Update

  此作业执行 SCD - Update New and Modified 和 SCD - Update Deleted 项目，并提供用于对目标维度表执行完全更新的单个转换对象。在执行此作业之前，应执行 SCD - Initial Load 项目。

了解目标维度表

标识目标记录

目标维度表包含同一源键的多条记录。为区分记录的当前版本和历史版本，目标维度表使用复合键，这种键包括源键及有效日期或到期日期属性。ETL 演示项目使用 Valid From 日期属性作为该键的一部分。

当前目标记录

源表中的每条记录都恰好由目标表中的一条当前记录表示。检查目标维度表时，只有当前记录与 SCD 相关。在 ETL 演示项目中，当前记录的 Valid To 日期为 '9999-12-31'。

检测源更改

本节单独介绍如何捕获源表中的更改，包括新的源记录、修改的源记录以及已从源中删除的记录。可以将所述方法组合使用，在一个步骤中检测不同类型的数据更改。

在此案例研究应用场景中，源数据库不包含任何更改日志信息，所以要检测任何更改，必须将源内容与目标内容进行比较。由于在大多数情况下，源对象和目标对象不在同一数据库中，因此需要执行异构连接。

检测新的源记录

在目标维度表上次更新后添加到源表中的记录在目标维度表中没有对应的当前记录。

1　所有源记录均使用相应的数据提供程序组件进行读取。有关数据提供程序组件的列表，请参见第 82 页上的 “源组件”。

注释 尽管只需要检测键属性，但会读取传送到目标维度表的所有属性。

2　对于每条源记录，都会根据源键属性检查目标维度中是否存在对应的当前记录。

a　选择相应的查找组件。有关查找组件的列表，请参见第 112 页上的 “查找组件”。

如果需要对要传输的数据进行计算，可以考虑使用 Data Calculator 组件的查找功能。请参见第 98 页上的 “Data Calculator JavaScript”。

b　从目标中选择查找数据。因为这是一项简单的存在性检查，所以只需要从所有当前目标记录中选择原始源键。不过，在 ETL 中进行的查找始终返回键的值，因此还必须选择相应的返回值。

c　向端口结构添加一个属性以填充查找结果。根据查找结果可以确定源记录是新添加的还是以前就存在的。此属性指示数据状态，并可用于在转换过程的下一步中对数据进行过滤。请参见第 78 页上的 “修改端口结构”。选择此新属性作为查找组件的值属性或 Data Calculator 中的输出属性。

d　设置适当的查找缺省值。查找不存在的键时，将返回该缺省值。为确保查找值正确指出记录是否存在，请将其设置为不同于任何现有键的所有查找值的常量。

示例：

源数据 – select Key, Name, Price from PRODUCT

查找数据 – select Key, '1' from PRODUCT_PRICE where Valid_To = '9999-12-31'

值属性 – Exists (integer)

缺省值 – 0

执行此查找将得到具有 Key、 Name、 Price、 Exists 属性的记录。对于目标表中的所有现有记录， Exists 将为 1（查找值）；对于所有非现有记录， Exists 将为 0 （缺省值）。

检测修改的源记录

目标维度表上次更新后在源表中修改的记录在目标维度表中具有对应的当前记录，但相关值已更改。

1 所有源记录均使用相应的数据提供程序组件进行读取。有关数据提供程序组件的列表，请参见第 82 页上的 “源组件”。

注释 尽管只需要检测键属性，但会读取传送到目标维度表的所有属性。

2 对于每条源记录，都会根据源键属性检查目标维度表中是否存在对应的当前记录，并对值进行比较。

a 选择相应的查找组件。

因为这需要查找单个键属性的多个值并进行比较，所以应使用 Data Calculator 组件。请参见第 98 页上的 “Data Calculator JavaScript”。

b 从目标中选择查找数据。对于所有当前目标记录，要比较的键属性和所有值将使用相应的数据提供程序组件进行读取。有关数据提供程序组件的列表，请参见第 82 页上的 “源组件”。

c 向端口结构添加一个属性以填充其它目标键属性。根据查找结果可以确定源记录是已经过修改还是新的或未经改动。此属性指示数据状态，并可用于在转换过程的下一步中对数据进行过滤。对目标执行的更新操作需要唯一标识当前记录，因此目标键的日期部分也需要填充。请参见第 78 页上的 “修改端口结构”。

d 设置适当的查找缺省值。查找不存在的键时，将返回该缺省值。为确保查找值正确指出记录是否存在，请将其设置为不同于现有键的所有查找值的常量。

e 查找所有必需的目标值。第一次查找时，将新的目标键属性用作 Data Calculator 中的输出属性，从而指示是否存在。将要比较的值将读入到临时变量中。

f 比较源属性值和目标属性值。将根据现有记录的值比较结果重新计算目标键属性。

示例：

源数据 – select Key, Name, Price from PRODUCT

查找数据 – select Key, Valid_From, Price from PRODUCT_PRICE where Valid_To='9999-12-31'

首先通过从目标读取有效日期来检查是否存在：

输出属性 – Valid_From

缺省值 – 0

将当前目标价格读入到临时变量中：

输出属性 – Tmp_Price

缺省值 – 0

如果当前目标记录存在 （Valid_From 不为 0），则比较 Price 和 Tmp_Price。如果 Price 尚未更改，则将 Valid_From 重新计算为 0。

执行这些查找和计算将产生具有 Key、 Name、 Price 和 Valid_From 属性的记录。Valid_From 包含要更新的目标记录的有效日期，或者包含 0，表示新记录和未更改的记录。

检测删除的源记录

在目标维度表上次更新后已从源表中删除的记录在目标维度表中仍具有对应的当前记录。

1　目标维度表中所有当前记录的键值都使用相应的数据提供程序组件进行读取。有关数据提供程序组件的列表，请参见第 82 页上的“源组件”。

2　对于每条当前目标记录，都会根据源键属性检查源中是否存在对应的记录。

a　选择相应的查找组件。有关查找组件的列表，请参见第 112 页上的 “查找组件”。

如果源数据不在数据库中，请使用 Data Calculator 组件的查找功能。请参见第 98 页上的 “Data Calculator JavaScript”。

b　从源中选择查找数据。因为这是一项简单的存在性检查，所以只需要从所有源记录中选择源键。不过，在 ETL 中进行的查找始终返回键的值，因此还必须选择相应的返回值。

c　向端口结构添加一个属性以填充查找结果。根据查找结果可以确定源记录是已删除还是仍旧存在。此属性指示数据状态，并可用于在转换过程的下一步中对数据进行过滤。选择此新属性作为查找组件的值属性或 Data Calculator 规则的输出属性。请参见第 78 页上的 “修改端口结构”。

d　设置适当的查找缺省值。查找不存在的键时，将返回该缺省值。为确保查找值正确指出记录是否存在，请将其设置为不同于现有键的所有查找值的常量。

示例：

目标数据 – select Key, Valid_From from PRODUCT_PRICE where Valid_To='9999-12-31'

查找数据 – select Key, '0' from PRODUCT

值属性 – Removed (integer)

缺省值 – 1

执行此查找将产生具有 Key、 Valid_From、 Removed 属性的记录。对于所有现有源记录， Removed 都将为 0 （查找值）；对于所有非现有记录， Removed 都将为 1 （缺省值）。

替代方法

- 如果源是为插入、更新或删除操作提供升序指示符 （如自动增量、修改日期等）的数据库，则 DB Data Provider Index Load 组件只能用于读取自上次装载以来更改的记录。请参见第 85 页上的 “DB Data Provider Index Load”。
- 使用 staging 组件将源和目标中的相关数据装载到同一数据库。然后，将通过使用完全外部连接从暂存位置提取数据，检测新的、修改的和删除的记录。请参见第 119 页上的 “DB Staging”。

过滤记录

使用 Data Splitter 组件，除了可以将数据流拆分为多个输出之外，还可以从数据流中删除记录。有关 Data Splitter 组件的详细信息，请参见第 106 页上的 “Data Splitter JavaScript”。

若要从数据流中删除记录，请为每个输出端口定义条件，使要删除的记录不与其中的任何条件匹配。若要输出单个数据流，请通过删除其中一个缺省输出端口，用单个输出端口配置 Data Splitter。

填充目标维度表

为目标属性赋值

使用 DB Data Sink 组件的插入和更新选项，为不包括在入站数据流中的那些目标属性赋值。这些值可以是常量，也可以是表达式的结果。请参见第 123 页上的 “DB Data Sink Insert”和第 127 页上的 “DB Data Sink Update”。

在 ETL 演示项目和作业中，插入选项使用：

- 表达式 – 对 Valid From 日期属性使用 '[uDate('now','localtime')]' （今天）。
- 常量 – 对 Valid To 日期属性使用 '9999-12-31'。

更新选项对 Valid To 日期属性使用下面的表达式：
'[uDate('now','localtime','-1 day')]'（昨天）。

执行部分更新

DB Data Sink 组件的更新选项允许更新一部分属性，而不是更新完整的记录。要将属性排除在更新范围之外，您应在更新选项窗口中取消选择相应属性。请参见第 127 页上的 “DB Data Sink Update”。

在 ETL 演示项目中，通过仅更新 Valid To 日期属性，即可使旧记录过期。

附 录 D

最佳实践

本附录介绍使用 Sybase ETL 的最佳实践。

使用 ETL Server 的最佳实践

避免启动多个 ETL Server 会话

如果在 ETL Development 运行时从命令行启动 ETL Server， ETL Development 会变得不稳定，若在 ETL Development 中执行任何操作，它将显示错误消息。出现此种情况的原因是，从命令行启动的 ETL Server 会话与由 ETL Development 启动的 ETL Server 会话之间发生冲突。

若要避免启动多个服务器会话，请在 “Preferences”窗口中，取消选择 “Engine” | “Start local engine during application startup”。这会防止在您下次启动 ETL Development 时自动启动 ETL Server 会话。

输入缺省端口号以执行命令行

如果您尝试使用缺省端口号 5124，但不在命令行中包括此端口号，则命令行执行会失败。为了避免出现此问题，必须在命令行中输入缺省端口号 5124。例如：

```
GridNode -con --port 5124 --server localhost
-f "..\\testdata\\tpms\\tpms_TestB.xml"
```

访问 ETL 4.2 存储库时重新输入口令

必须重新输入存储库用户口令，才能登录以访问使用 ETL 4.8 之前的 ETL 版本创建的存储库。

在 Query Designer 中输入查询时使用列别名

将函数应用于 Query Designer 中的表时，对于不同的数据库， Content Browser 中的输出列名也不同，输出列名可以与源列名无关。例如：

DBMS	函数	Query Designer 中的查询	源列名	输出列名
IBM DB2	CHAR	`select CHAR(col_1) from DATE_TBL`	COL_1	1
Adaptive Server	convert	`select convert(char(10), ID) from CUSTOMER_TBL`	ID	col_1
Sybase IQ	convert	`select convert(char(10), id) from alt_sales_orders`	id	id
Microsoft Access	SUM	`select SUM(SA_TOTAL) from SALES`	SA_TOTAL	Expr1000

在 Query Designer 中输入查询时必须使用列别名，以便列别名显示为输出列名。例如，使用列别名 mycolumn：

```
select CHAR(col_1) as mycolumn from TEST_TBL
```

使用 ETL 组件的最佳实践

迁移宽表

迁移具有成百上千个列的表会消耗大量内存。在将具有不同列数和行数的多个宽表从 Sybase IQ 源数据库迁移到 Sybase IQ 目标数据库时，为了避免发生错误，请遵循下列建议：

- 使用 SQL Anywhere 存储库，而不是 Microsoft Access 或 SQLite Persistent 存储库。
- 为 SQL Anywhere 存储库分配 1 GB 的数据库空间，并将所有其它值保留为系统缺省值。
- 使用以下方法、组件和接口来迁移具有大量列的表：

列数	迁移方法	使用接口
多达 3000 列	Insert Location 组件	Sybase
多达 3000 列	Load Table 组件	Sybase 或 ODBC
多达 3500 列	迁移向导 **注释** 由于内存限制，或者如果您未使用 SQL Anywhere 存储库，迁移向导可能无法生成迁移项目。	Sybase 或 ODBC
多达 10000 列	Load Table 组件	ODBC

注释 当迁移的表含有的列多于 3000 时，您可能会在移动大量行时遇到性能问题。

导入具有 32 个以上同级元素的 XML 文件

若要使用 XML via SQL Data Provider 组件导入 32 个以上同级元素，请将该组件 “XML Options”属性中的 “Create Flat View”设置为 0。必须使用 Content Explorer 手动设置子查询。

将源文本文件的最后一行装载到 Sybase IQ

当使用 IQ Loader File via Load Table 组件时，如果源文本文件的最后一行不以尾随行分隔符结尾， Sybase IQ 不会从 ETL 接受源文本文件的最后一行。为了避免出现此问题，在源文本文件最后一行的结尾添加行分隔符。例如，在 Windows 中，由于行分隔符指定为 CRLF，因此将光标放在最后一行结尾，按 Enter 即将行分隔符添加到最后一行。

配置 Adaptive Server Enterprise 以进行批量复制

如果要将 Adaptive Server Enterprise 用于 DB Staging，必须首先配置 Adaptive Server 数据库以进行批量复制。如果未配置 Adaptive Server，尽管项目会成功执行，您仍会遇到错误。

添加 35 个以下的 Data Calculator JavaScript 组件

确保不要为一个项目添加 35 个以上的 Data Calculator JavaScript 组件。

增加 Adaptive Server ODBC 驱动程序的文本大小

当 text 或 image 数据值大于在 Microsoft Windows ODBC 配置中为该驱动程序设置的值时， Adaptive Server ODBC 驱动程序会对其进行截断。必须增加 ODBC 控制面板中的文本大小值，或者在数据库连接的数据库选项参数中设置该值，以避免出现此问题。

❖ **使用控制面板增加文本大小值**

1 选择 “控制面板” | “管理工具” | “数据源 (ODBC)”，然后在 “用户 DSN” 或 “系统 DSN” 选项卡中选择 Adaptive Server Enterprise 的数据源。

2 选择 “配置” 以显示 “ODBC Adaptive Server Enterprise Setup” 窗口，然后选择 “高级”。

3 将 “Text Size” 的值更改为大于 32 KB （缺省值）的值。 Adaptive Server ODBC 驱动程序会截断任何大于此处所设置值的数据值。

❖ **使用数据库选项增加文本大小值**

1 在数据库连接的 “Properties” 窗口中，双击 “Database Options” 字段的编辑图标以显示 “Enter Properties” 窗口。

2 在 “Extended Connect Options” 字段中，输入 “TEXTSIZE=*N*”，其中 *N* 为要设置的文本大小值。

在 Windows 上使用文件路径名作为 “Load Stage” 属性

对于 DB Bulk Load Sybase IQ 组件，如果已选择 “Use IQ Client Side Load” 选项将客户端计算机上的文件中的数据批量装载到 IQ 数据库，请在 “Load Stage” 属性字段中提供文件路径名，而不是管道名。客户端装载不支持使用 “Load Stage” 管道名。

注释 仅当选择 ODBC 作为接口时，客户端装载才会正常运行。

当在不同平台上执行项目时，不得更改源文本文件中的分隔符

如果在 Windows 上使用 Text Data Provider 组件创建项目，然后在 UNIX 或 Linux 上执行该项目，请确保不要将源文本文件转换为 UNIX 或 Linux 格式。源文本文件中使用的分隔符应当始终与 Text Data Provider 组件中设计的分隔符相同。如果分隔符不一致，则会发生执行错误。

在 Windows 上设置命名管道权限

对于 DB Bulk Load Sybase IQ 组件，如果希望在“Load Stage”属性字段中提供管道名，必须首先进行以下设置，以免发生任何错误：

1　转至“控制面板”|“管理工具”|“本地安全策略”|“本地策略”|“安全选项”。

2　在“策略”列表中双击“网络访问 : 可匿名访问的命名管道”。

3　将您的命令管道添加到现有列表中。例如，如果管道名为“pipe://mypipe”，请将“mypipe”添加到列表中。单击“应用”。

4　单击“确定”。

使用国际化的最佳实践

正确分析具有字节顺序标记的源文件

如果要使用“Fixed by Bytes”属性分析文件，请确保源文件不包含任何字节顺序标记。分析之前，使用文本编辑器删除源文件中的字节顺序标记。

设置 ETL 以支持 UTF-8 编码

提供给 uSetEnv 函数的参数不能包含多字节字符或非西方字符。必须设置 ETL 以支持 UTF-8。例如：

```
set LANG=zh.UTF-8
```

选择正确的字符集编码以正确显示 Unicode 字符

当您使用 “Text Data Provider”或 “Text Data Sink”组件装载字符数据时，除非为具有字节顺序标记 (BOM) 的 Unicode 文件的字符集编码选择正确的 “endianness”类型，否则字符会显示不正确。

若要正确显示 Unicode 字符，请在组件配置窗口的 “Character Encoding”字段中，为字符数据选择具有正确 “endianness”类型的字符集编码。例如，选择：

- UTF-16LE – 处理以 UTF-16LE 编码的文本文件，这些文件在文件开头具有 BOM，其中 LE 表示 “little-endian”，因为 BOM 位于文件开头。
- UTF-16BE – 处理以 UTF-16BE 编码的文本文件，这些文件在文件结尾具有 BOM，其中 BE 表示 “big-endian”，因为 BOM 位于文件结尾。

索引

D

E

F

G

H

J

K

L

M

N

P

Q

R

S

T

W

X

Y

Z