

Rational Software Development Platform



# VisualAge Generator 到 EGL 迁移 指南

版本 6 发行版 001



Rational Software Development Platform



# VisualAge Generator 到 EGL 迁移 指南

版本 6 发行版 001

**注意**

在使用本文档之前，请阅读第 321 页的『声明』中的一般信息。

**第三版（2005 年 4 月）**

本版本适用于下列许可程序：

- Rational Web Developer
- Rational Application Developer
- WebSphere Development Studio Client Advanced Edition for iSeries
- WebSphere Studio Enterprise Developer

IBM 欢迎您提出宝贵意见。您可以通过邮件将意见发送至以下地址：

IBM 中国公司上海分公司，汉化部中国上海市淮海中路 333 号瑞安广场 10 楼邮政编码：200021

您也可以通过电子的形式给我们提供意见： IBMLink: [ibmcn\(ctscrcf\)](mailto:ibmcn(ctscrcf)) 互联网: [ctscrcf@cn.ibm.com](mailto:ctscrcf@cn.ibm.com) 中国: 021-63857881 其他国家或地区: (86-21)63857881

当您发送信息给 IBM 时，即授予 IBM 非专有权，IBM 可以按他认为适当的任何方式来使用或分发该信息而无须对您承担任何责任。

**© Copyright International Business Machines Corporation 2004, 2005. All rights reserved.**

---

## 前言

本文档面向想要从 VisualAge Generator 4.5 迁移到企业生成语言（EGL）的读者。

---

## 本书面向的读者

本书面向想要将代码从 VisualAge Generator 4.5 迁移到企业生成语言（EGL）的程序员或系统管理员。

---

## 相关信息

以下列格式中的其中一种或多种格式提供了相关文档:

- 产品 CD-ROM 上的联机书籍文件（.pdf）。使用 Adobe Acrobat Reader 来查看联机手册和打印期望的页面。
- 产品 CD-ROM 上的 HTML 文件（.htm）。

以下 Web 站点以在线书籍文件（.pdf）形式提供了本书的最新版本:

<http://www.ibm.com/developerworks/rational/library/egl doc.html>



# 目录

前言	iii
本书面向的读者	iii
相关信息	iii
<b>第 1 部分 迁移概述</b>	<b>1</b>
<b>第 1 章 迁移概述</b>	<b>3</b>
本书中使用的术语	3
EGL 的哪些新增功能决定了需要进行迁移?	3
规划迁移	4
确定是否可以迁移到 EGL	6
在 EGL 中不可用的 VisualAge Generator 功能部件	8
术语差别	8
<b>第 2 章 迁移工具原理</b>	<b>13</b>
VisualAge Generator 到 EGL 迁移工具概述	14
迁移工具术语	14
第 1 阶段详细信息	15
第 2 阶段详细信息	17
第 3 阶段详细信息	18
单文件迁移概述	19
迁移问题	21
精确的 EGL 语法	22
解析部件名的时间及方式	23
公共代码方案	23
VisualAge Generator 到 EGL 迁移工具使用的技术	26
技术概述	26
编辑器和构建描述符首选项	26
程序属性	27
EGL 构建路径和 import 语句	28
containerContextDependent 属性	29
EGL 保留字列表	30
将部件放到 EGL 文件中	30
带程序迁移	33
带关联部件迁移	33
不带关联部件迁移	34
覆盖与合并文件	34
一般规则	36
自从 EGL 5.1.2 以来 VAGen 迁移工具有哪些新增功能?	39
自从 EGL 6.0 iFix 以来 VAGen 迁移工具有哪些新增功能?	39
迁移工具的已知限制	40
概要	40
对 Java 和 Smalltalk 执行的第 1 阶段	40
第 2 阶段和第 3 阶段	40
语法迁移	40
<b>第 3 章 处理不确定情况</b>	<b>41</b>
处理数据项的不确定情况	41

具有偶数长度的 PACK 数据项	41
共享编辑和消息	42
共享数据项的映射项编辑例程	43
共享数据项的填充字符	44
处理记录的不确定情况	45
重新定义的记录	45
记录中的 Level-77 项	46
备用指定记录	47
具有相同记录名的不同定义	48
处理表的不确定情况	48
保留字和表名	48
处理映射组和映射的不确定情况	49
保留字和 formGroup 名	49
映射组和 formGroup 需求	50
浮动区和起始位置	51
映射组、映射和设备大小	51
映射名和帮助映射名	52
数字变量字段	54
映射变量字段和编辑例程	55
映射字段和数字硬件属性	56
映射数组和属性	56
未命名变量字段	57
不受保护映射常量	58
定位在“行 = 0, 列 = 0”处的字段	58
处理程序的不确定情况	59
程序名和保留字	59
程序中的隐式数据项	60
相关联的程序部件	60
迁移所需的中间变量	62
处理函数(包括 I/O 语句)的不确定情况	63
用于映射的 DISPLAY 语句	63
I/O 错误例程	65
SQL I/O 语句	66
SQL I/O 和缺少的必需 SQL 子句	68
SQL I/O 和 !itemColumnName	70
带有多个更新的 SQL I/O	70
处理其它语句的不确定情况	71
语句中的隐式数据项	71
语句中的 Level-77 项	71
赋值语句	72
FIND 语句	72
RETR 语句	73
SET map PAGE 语句	73
SET mapItem 属性	74
检查 IN 文字或标量	75
检查 SQL 项和映射项是否为 NULL	76
I/O 错误值 UNQ 和 DUP	77
I/O 错误值 LOK	78
XFER	79
处理 EZE 字的不确定情况	80
EZESYS	80

EZEWAIT . . . . .	82
-------------------	----

## 第 2 部分 从 VisualAge Generator 4.5 on Java 迁移到 EGL . . . . . 85

<b>第 4 章 第 1 阶段 - 从 Java 中抽取</b>	<b>87</b>
在 VisualAge for Java 中安装第 1 阶段迁移工具 . . . . .	87
添加迁移功能部件 . . . . .	88
创建迁移数据库 . . . . .	88
设置第 1 阶段首选项 . . . . .	88
“构建规划”页 . . . . .	89
“映射”页 . . . . .	91
“重命名”页 . . . . .	92
“执行”页 . . . . .	92
样本 MigPreferences.xml 文件 . . . . .	94
在运行第 1 阶段工具之前 - 提示与技巧 . . . . .	96
提高性能 . . . . .	96
保存工作空间 . . . . .	97
运行第 1 阶段工具 . . . . .	97
迁移规划和高级 PLP 项目 . . . . .	98
创建高级 PLP 项目 . . . . .	99
手工创建迁移规划文件 . . . . .	100

## 第 3 部分 从 VisualAge Generator 4.5 on Smalltalk 迁移到 EGL . . . . . 103

<b>第 5 章 第 1 阶段 - 从 Smalltalk 中抽取</b>	<b>105</b>
在 VisualAge Smalltalk 上安装第 1 阶段迁移工具 . . . . .	105
装入迁移功能部件 . . . . .	105
创建迁移数据库 . . . . .	106
设置第 1 阶段首选项 . . . . .	106
“构建规划”页 . . . . .	107
“映射”页 . . . . .	109
“重命名”页 . . . . .	110
“执行”页 . . . . .	111
样本 MigPreferences.xml 文件 . . . . .	112
根据首选项派生文件名 . . . . .	114
在运行第 1 阶段工具之前 - 提示与技巧 . . . . .	114
提高性能 . . . . .	114
保存映像 . . . . .	115
运行第 1 阶段迁移工具 . . . . .	115
迁移规划和高级配置映射 . . . . .	117
创建高级配置映射 . . . . .	117
手工创建迁移规划文件 . . . . .	118

## 第 4 部分 第 2 阶段和第 3 阶段 - 常用迁移步骤 . . . . . 121

<b>第 6 章 第 2 阶段 - 转换为 EGL 语法</b>	<b>123</b>
设置工作台首选项 . . . . .	123
启动参数 . . . . .	123
必需的 EGL 首选项 . . . . .	123

建议的首选项 . . . . .	124
VAGen 迁移首选项 . . . . .	124
VAGen 迁移语法首选项 . . . . .	126
其它建议设置 . . . . .	128
设置第 2 阶段 VAGen 迁移文件 . . . . .	128
运行第 2 阶段 . . . . .	132
从用户界面中运行第 2 阶段 . . . . .	132
以批处理方式运行第 2 阶段 . . . . .	133

<b>第 7 章 第 3 阶段 - 导入 . . . . . 137</b>	
运行第 3 阶段工具 . . . . .	137
以批处理方式运行第 3 阶段 . . . . .	139
使用已写至临时目录的迁移集 . . . . .	139

<b>第 8 章 以单文件方式运行迁移 . . . . . 141</b>	
使用用户界面来运行单文件迁移 . . . . .	141
使用批处理方式来运行单文件迁移 . . . . .	142

## 第 5 部分 完成迁移 . . . . . 145

<b>第 9 章 完成迁移 . . . . . 147</b>	
导出首选项 . . . . .	147
为 EGL 项目和包保存一条基线 . . . . .	148
用于完成单文件迁移的预备步骤 . . . . .	148
第 1 — 3 阶段迁移与单文件迁移的公共步骤 . . . . .	148
查看 EGL 源代码 . . . . .	148
查看 EGL 构建描述符部件 . . . . .	149
建立一个绑定控制部件以用作模板 . . . . .	151
建立特定于程序的绑定控制部件 . . . . .	152
查看链接编辑命令 . . . . .	152
查看 EGL 链接选项部件 . . . . .	153
查看 EGL 资源关联部件 . . . . .	153
为调试作准备 . . . . .	154
对 COBOL 生成进行生成和测试 . . . . .	154
对 Java 生成进行生成和测试 . . . . .	155
查看标准 . . . . .	155
除去对 VisualAge Generator 兼容性方式的使用 . . . . .	155

## 第 6 部分 语言和运行时差别 . . . . . 161

<b>第 10 章 语言和运行时差别 . . . . . 163</b>	
语言差别 . . . . .	163
运行时差别 . . . . .	163
一般差别 . . . . .	163
在调试方面的差别 . . . . .	164
在生成的 COBOL 方面的差别 . . . . .	165
在生成的 Java 方面的差别 . . . . .	165
主机与工作站环境之间的差别 . . . . .	166
在分布式 CICS 与本机工作站环境之间的差别 . . . . .	166
在生成的 C++ 与生成的 Java 之间的差别 . . . . .	169

## 第 7 部分 附录 . . . . . 171

<b>附录 A. 保留字 . . . . . 173</b>	
EGL 保留字 . . . . .	173

EGL 枚举字 . . . . .	173
SQL 保留字 . . . . .	176
需要进行特殊处理的 SQL 保留字 . . . . .	177
Java 保留字 . . . . .	177

## 附录 B. VisualAge Generator 与 EGL

<b>语言元素的关系 . . . . .</b>	<b>179</b>
一般语约定 . . . . .	179
数据项 . . . . .	180
记录 . . . . .	186
表 . . . . .	193
映射组 . . . . .	195
映射 . . . . .	198
程序 . . . . .	210
函数 . . . . .	215
语句 . . . . .	227
EZE 字 . . . . .	239
程序流 EZE 字 . . . . .	239
SQL EZE 字 . . . . .	240
日期和时间 EZE 字 . . . . .	241
其它数据 EZE 字 . . . . .	242
一般函数 EZE 字 . . . . .	244
字符串 EZE 字 . . . . .	245
数学 EZE 字 . . . . .	245
用户界面 EZE 字 . . . . .	246
对象脚本编制 EZE 字 . . . . .	247
DL/I EZE 字 . . . . .	247
服务例程 . . . . .	248
PSB . . . . .	249
控制部件 . . . . .	249
生成选项部件 . . . . .	250
链接表部件 . . . . .	263
资源关联部件 . . . . .	269
链接编辑部件 . . . . .	273
绑定控制部件 . . . . .	273
符号参数 . . . . .	273

## 附录 C. 迁移工具发出的消息 . . . . . 277

VisualAge Generator 到 EGL 迁移工具发出的消息	
- 第 1 阶段 . . . . .	277
第 1 阶段公共消息 . . . . .	277

VisualAge for Java 上的第 1 阶段 . . . . .	280
VisualAge Smalltalk 上的第 1 阶段 . . . . .	281
VisualAge Generator 到 EGL 迁移工具发出的消息	
- 第 2 阶段 . . . . .	282
从 VisualAge Generator 传送至 EGL 迁移工具的消息	
- 第 3 阶段 . . . . .	297

## 附录 D. “问题”视图中的消息 . . . . . 299

## 附录 E. “问题”视图中的 IWN.xxx 消息 . . . . . 303

IWN.VAL 消息 . . . . .	303
IWN.XML 消息 . . . . .	305

## 附录 F. 不正确的外部源格式会导致创建 EGL 时发生问题的情况. . . . . 307

## 附录 G. 迁移数据库 . . . . . 309

创建 DB2 迁移数据库 . . . . .	309
为 DB2 7.2 设置 JDBC 级别 . . . . .	309
为 DB2 8.1 或更高版本设置 JDBC 级别 . . . . .	309
在 Windows XP 上使用 DB2 . . . . .	309
创建迁移数据库 . . . . .	309
复位迁移数据库 . . . . .	310
使用 DB2 来对远程数据库进行编目 . . . . .	311
使用 DB2 来对远程数据库进行取消编目 . . . . .	312
非常有用的查询 . . . . .	312

## 附录 H. 迁移工具的性能 . . . . . 315

项目、包、部件和程序的数目 . . . . .	315
处理器速度 . . . . .	316
函数部件中的行数 . . . . .	317
对第 1 阶段使用干净的 Java 工作空间 . . . . .	317
磁盘空间要求 . . . . .	318

## 声明 . . . . . 321

商标 . . . . .	322
--------------	-----

## 索引 . . . . . 325



---

## 第 1 部分 迁移概述



---

## 第 1 章 迁移概述

具有企业生成语言 (EGL) 组件的 Rational® Developer 或 WebSphere® Studio 产品是 VisualAge Generator 的后继产品。您需要迁移 VisualAge Generator (VAGen) 源代码以转换到 EGL。

本迁移指南提供有关规划迁移、使用迁移工具来转换源代码以及运行迁移工具后要完成迁移所需执行的其它步骤的信息。

除了本《迁移指南》以外，您还应该查阅下列各项以了解其它的或更多的最新信息：

- EGL 的联机帮助系统。
- 《EGL 参考指南》。
- VisualAge Generator 的 Web 站点和新闻组。Web 站点如下所示：  
<http://www.ibm.com/software/awdtools/visgen/>
- EGL 的 Web 站点和新闻组。Web 站点如下所示：  
<http://www.ibm.com/developerworks/rational/products/egl/>
- 您所使用的产品的 Web 站点和新闻组。

---

### 本书中使用的术语

EGL 是作为若干产品的其中一个组件交付的。本书使用以下术语：

#### 开发者产品

将 EGL 作为组件包括的任何产品。

#### EGL 开发环境

在启动将 EGL 作为组件包括的任何产品之后看到的工作台和其它窗口。

---

### EGL 的哪些新增功能决定了需要进行迁移？

EGL 包括 VisualAge® Generator 的主要更改和增强，包括下列各项：

- 对 VAGen 语言进行的更改，这包括许多增强功能，如新的数据类型、多维结构字段数组、动态数组、*case* 语句以及经过改进的 Web 支持。
- 对用来开发程序的用户界面所作的更改，这包括内容辅助、用于创建部件的代码模板以及适用于大部分部件类型的文本编辑器。
- 对生成和准备过程所作的更改，这包括对于分布式平台来说仅进行 Java™ 生成而不是进行 C++ 和 Java 生成，以及使用构建服务器而不是准备 JCL 模板来进行 COBOL 生成。
- 对运行时进行的更改，这包括使用 Enterprise Developer Server 来提供运行时服务。
- 对库管理方式所作的更改，这包括您能够选择自己的源代码资源库来与 EGL 进行交互。

VAGen 语言与 EGL 之间的差别比较大。过去，当您从跨系统产品或 VisualAge Generator 的一个版本升级到新版本时，语言只有小幅的更改。先前的迁移工具能够以独

立于任何其它部件的方式来迁移每个部件。但是，由于在这两种语言之间存在差别，所以，VisualAge Generator 到 EGL 迁移工具必须在其它被引用或相关联部件的上下文中迁移每个部件以确定下列各项：

- 被引用部件的部件类型
- 由于新的 EGL 语法而必须移至引用部件的信息
- 被引用部件在工作空间中的位置

术语交叉部件迁移用来描述这种一个部件的迁移取决于其它部件的情况。对于生成从 VAGen 语言到 EGL 的最佳可能转换来说，必须进行交叉部件迁移。这反过来又意味着您需要仔细地考虑要将哪些部件组一起迁移。

由于 VisualAge Generator 与 EGL 之间存在差别并且需要进行交叉部件迁移，所以，此迁移是您需要完成的主要工作，并且需要仔细地进行规划。

---

## 规划迁移

在规划迁移项目时，您需要考虑下列任务：

- 规划用于迁移的试验项目：
  - 选择试验项目所涉及的开发者和系统支持人员。
  - 选择要在试验项目中使用的一小组源代码。使用这一小组代码来验证环境设置以及库管理过程和工具。
  - 升级到带有修订包 4 的 VisualAge Generator 4.5。如果将 VisualAge Generator 与 VisualAge for Java 配合使用，则还必须安装 APAR PQ88461 以升级“VisualAge Generator 实用程序”功能部件。请与 IBM® 支持机构联系以获取 APAR 修订，或者访问 [VAGen Web 站点](http://www-306.ibm.com/software/awdtools/visgen/support)（网址为 <http://www-306.ibm.com/software/awdtools/visgen/support>）并遵循 Download 部分中的链接。并且，请回顾第 307 页的附录 F，『不正确的外部源格式会导致创建 EGL 时发生问题的情况』以了解在特定情况下可能需要的其它 VisualAge Generator APAR。
  - 如果尚未安装 DB2®，请安装它。DB2 是必需的迁移数据库。
  - 请查看您计划使用的开发者产品的功能。确保它包含所需的功能部件。例如：
    - 如果您计划为 zOS 环境生成 COBOL，则必须使用包括为 zSeries® 生成 COBOL 的开发产品。
    - 如果计划使用 iSeries™，则必须使用包括为 iSeries 生成 COBOL 的开发者产品。
  - 请查看您计划使用的产品的先决条件。此外，请查看运行时环境的先决条件。例如：
    - 如果您计划为 zOS 环境生成 COBOL，则务必查看 Enterprise Developer Server for zOS 产品的先决条件。还应查看构建服务器的先决条件。
    - 如果计划为 UNIX® System Services (USS) 环境生成 Java，则一定要查看 Enterprise Developer Options for zOS 组件的先决条件。
    - 如果计划为 iSeries 生成，则一定要查看开发者产品的运行时组件的先决条件。
    - 如果计划为工作站环境生成 Java，则一定要查看计划使用的开发者产品的先决条件。
- 对将要运行试验项目的小组进行培训：

- 开发者产品环境
- EGL 语言
- VisualAge Generator 到 EGL 迁移工具
- 运行试验项目规划以执行下列操作:
  - 为试验小组安装开发者产品，并确保安装该产品的机器的区域设置与用于开发 VAGen 程序的区域设置相同。例如:
    - 如果您以前是在德语机器上开发 VAGen 程序的，则应该在德语机器上安装开发者产品。这将确保能够正确地迁移德语元音变音字符以及用作小数点的逗号。
    - 如果您以前是在中文机器上开发 VAGen 程序的，则必须在使用同一代码页的中文机器上安装开发者产品。这将确保能够正确地迁移 DBCS 字符。
  - 对试验代码集运行 VAGen 迁移工具。请参阅下列各节以了解有关迁移工具的信息:
    - 第 13 页的第 2 章，『迁移工具原理』
    - 第 85 页的第 2 部分，『从 VisualAge Generator 4.5 on Java 迁移到 EGL』
    - 第 103 页的第 3 部分，『从 VisualAge Generator 4.5 on Smalltalk 迁移到 EGL』
    - 第 121 页的第 4 部分，『第 2 阶段和第 3 阶段 - 常用迁移步骤』
    - 第 145 页的第 5 部分，『完成迁移』
  - 创建库管理过程:
    - 选择并安装源代码资源库，包括从开发者工作站进行访问。
    - 定义与您的企业文化和所选源代码资源库配合使用的更改管理过程。
    - 开发更改管理过程所需的任何工具，包括以下过程:
      - 检入和检出过程。
      - 版本控制过程。
      - 如果您想要使用批处理生成，还有用于从源代码资源库中检索源代码以及装入工作空间或目录结构的工具。
  - iSeries COBOL 目标环境:
    - 遵循《EGL Server for iSeries 指南》中的指导。
  - zOS COBOL 目标环境:
    - 安装 Enterprise Developer Server for zOS V5.0 产品的必备软件，包括对 COBOL 编译器和运行时进行的任何更改。
    - 安装 Enterprise Developer Server for zOS V5.0 产品。
    - 安装 Enterprise Developer Server for zOS 的任何必需 PTF。
    - 安装构建服务器和任何必需的 PTF。并且定制伪 JCL 构建脚本。构建服务器是随 Enterprise Developer Options for z/OS<sup>®</sup> 交付的。
  - Java 目标环境:
    - 如果您正在切换平台（例如，从 Windows<sup>®</sup> CICS<sup>®</sup> 切换到 Windows 本机），则查看运行时平台差别。进行相应的代码更改。请参阅第 166 页的『在分布式 CICS 与本机工作站环境之间的差别』以获取差别列表。
    - 如果您当前正在使用 ODBC 支持，则从供应商那里获取 JDBC 支持。
  - 生成并准备程序:

- 查看并修改构建描述符部件。请参阅第 149 页的『查看 EGL 构建描述符部件』以获取 VAGen 迁移工具无法处理的更改的列表。
- (可选) 构建 EGL 批处理生成服务器。这要求使用源代码资源库并创建工具来装入具有生成操作所需的所有部件的目录。
- 测试:
  - 测试开发环境, 以确保可以成功地调试程序。在开发环境中进行调试可能要求访问 DB2、远程 VSAM 文件以及仅在运行时环境中运行的非 EGL 程序。
  - 至少测试一个具有代表性的程序样本, 以确保您了解任何运行时差别。请参阅第 163 页的『运行时差别』以获取差别列表。
  - 使用您可能要对 EGL 源代码进行的典型更改来测试库管理过程和工具。务必针对每个目标环境来测试用于更改公共代码、表单、dataTable 和程序的过程。并且, 针对每个目标环境来测试用于添加公共代码、表单、dataTable 和程序的过程。
- 根据试验项目的结果来优化库管理过程和工具。
- 记录试验项目的发现内容, 这包括:
  - 需要进行的代码更改, 当您正在切换目标环境时尤其如此。
  - 开发者对任何个人构建描述符部件所需进行的更改。
  - 根据您在试验项目中遇到的问题, 请参考对于开发者很有用的《迁移指南》章节。
  - 最终用户将会注意到的运行时行为更改。
- 根据试验项目的发现, 构建一个规划来完成迁移。
- 为其余开发者提供培训:
  - 开发者产品环境
  - EGL 语言
  - 新的源代码资源库
  - 新的库管理过程
  - 新的生成过程

---

## 确定是否可以迁移到 EGL

EGL 是 Rational Developer 或 WebSphere Studio 产品中的战略组件, VisualAge Generator 客户应该迁移到这个组件。EGL 支持并未打算完全替换 VisualAge Generator Developer 4.5 (VAGen) 所支持的所有功能和平台。根据目标环境以及您使用 VisualAge Generator 开发的程序类型的不同, 您可能需要等待 EGL 的将来发行版。

以下列表显示了在 EGL 的当前发行版中支持的 VisualAge Generator 目标环境。

- MVS CICS
- MVS™ Batch
- Unix System Services
- Windows Native
- AIX® Native
- Linux™ on Intel® 平台
- HP-UX

- Solaris
- iSeries

注: 虽然 VisualAge Generator 能够为某些平台生成 Java 和 C++, 但是, EGL 只能生成 Java。

要了解这些受支持环境中的其它注意事项, 请参阅下列各项:

- 与迁移到 EGL 相关的特殊注意事项 - 文件和数据库访问 (第 7 页的表 1)
- 与迁移到 EGL 相关的特殊注意事项 - 用户界面 (第 7 页的表 2)
- 第 8 页的『在 EGL 中不可用的 VisualAge Generator 功能部件』

下表列示了受支持环境的特殊注意事项。

表 1. 与迁移到 EGL 相关的特殊注意事项 - 文件和数据库访问

VAGen 文件和数据库访问	特殊注意事项
SQL	在 EGL 中受支持。
串行记录、带索引记录和相对记录	在 EGL 中受支持。
消息队列记录	在 EGL 中受支持。
DL/I	在 EGL 中不受支持。您必须等待将来发行版才能进行迁移。
GSAM	在 EGL 中不受支持。您必须等待将来发行版才能进行迁移。
IMS™ 消息队列	在 EGL 中不受支持。您必须等待将来发行版才能进行迁移。
Btrieve	在 EGL 中不受支持。
本地 VSAM	对于下列各项受支持: <ul style="list-style-type: none"> <li>• 面向 AIX 的 Java 生成。</li> <li>• COBOL 生成。</li> </ul> 对于调试或者对于面向其它环境的 Java 生成, 不受支持。
远程 VSAM	对于下列各项受支持: <ul style="list-style-type: none"> <li>• 调试 (如果远程文件位于 ZOSCICS 或 iSeries 上的话)。</li> <li>• 面向 Windows 的 Java 生成 (如果远程文件位于 ZOSCICS 或 iSeries 上的话)。</li> <li>• 面向 ZOSCICS 的 COBOL 生成。</li> </ul>

表 2. 与迁移到 EGL 相关的特殊注意事项 - 用户界面

VAGen 用户界面	特殊注意事项
文本用户界面, 包括打印	在 EGL 中, 对于 COBOL 生成和 Java 生成来说都受支持。
Web 事务和用户界面 (UI) 记录	在 EGL 中不受支持。您必须等待将来发行版才能进行迁移。

表 2. 与迁移到 EGL 相关的特殊注意事项 - 用户界面 (续)

VAGen 用户界面	特殊注意事项
使用 VAGen Java 包装器的 JSP 和 Java servlet	<ul style="list-style-type: none"> <li>您可以使用新开发者产品提供的信息来将 JSP 和 Java servlet 迁移到该产品。</li> <li>您可以使用本《VAGen 迁移指南》来将 VAGen 服务器程序迁移到 EGL。您可以使用 EGL 来生成 Java 包装器。</li> </ul>
在自由格式面板上未使用 VAGen 部件但却使用了 VAGen Java 包装器的 Java 图形用户界面应用程序或 applet。	<ul style="list-style-type: none"> <li>使用用于从 VisualAge for Java 迁移 Java 代码的信息，可将 Java 应用程序或 applet 迁移至新的开发者产品。</li> <li>您可以使用本《VAGen 迁移指南》来将 VAGen 服务器程序迁移到 EGL。您可以使用 EGL 来生成 Java 包装器。</li> </ul>
在自由格式面板上使用 VAGen 部件的 Java 图形用户界面应用程序或 applet。	在当前发行版中不受支持。
Smalltalk 图形用户界面视图或可视部件。	在当前发行版中不受支持。必须将带有 VAGen 部件的视图迁移到基于 Java 的解决方案。EGL 没有任何基于 Smalltalk 的解决方案。

## 在 EGL 中不可用的 VisualAge Generator 功能部件

除了表 1 和表 2 中列示的特殊注意事项以外，如果您需要以下列表中的任何功能部件，则您应该评估现在进行迁移与将来进行迁移的影响：

- 专用的编辑器和列表，如在生成期间生成的程序的列表。
- 专用的功能：
  - 在一组所选部件中搜索引用以及将列表限制为只包含某个程序的关联项。
  - 按部件类型或按子类型过滤部件。EGL 提供了搜索功能，因此，您可能能够对特定部件类型或子类型执行搜索。
- VisualAge Generator 模板。
- 如果您计划使用 Java 生成，则还需确定是否使用不能被转换到本机运行时环境的特定于 CICS 的功能。请参阅第 166 页的『在分布式 CICS 与本机工作站环境之间的差别』以了解差别的详细信息。

## 术语差别

VisualAge Generator Developer on Java、VisualAge Generator Developer on Smalltalk 和 EGL 全都使用不同的术语。为了帮助您将 VAGen 术语与 EGL 术语对应起来，下列 6 个表显示了这三组术语。

表 3. 代码组织术语差别

VisualAge Generator on Java	VisualAge Generator on Smalltalk	企业生成语言 (EGL)
工作空间	映像	工作空间
项目	配置映射	EGL 项目
包	应用程序	包含一个或多个 EGL 文件的 EGL 源文件夹和 EGL 包

表 3. 代码组织术语差别 (续)

VisualAge Generator on Java	VisualAge Generator on Smalltalk	企业生成语言 (EGL)
(没有对应的概念)	(没有对应的概念)	文件 (通常, Java 包或 Smalltalk 应用程序将分割成多个文件)。EGL 文件包含一种或多种部件类型的 EGL 一个或多个部件。
类或类型	类	EGL 部件类型
方法或成员	方法	(没有对应的概念)
VAGen 部件	VAGen 部件	文件中的 EGL 部件

表 4. VAGen 部件和概念术语差别

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
共享的数据项	共享的数据项	DataItem 部件的类型定义
不共享的数据项	不共享的数据项	基本项定义
数据项部件	数据项部件	DataItem 部件
记录部件	记录部件	记录部件 注: 迁移工具将把所有 VAGen 记录定义转换为 EGL 固定记录以保留 VAGen 行为。
结构项 (记录中的字段的结构)	结构项 (记录中的字段的结构)	结构字段
数组 (记录或映射中多次出现的项)	数组 (记录或映射中多次出现的项)	结构字段数组
映射组部件	映射组部件	FormGroup
映射部件: • 显示器映射 • 打印机映射	映射部件: • 显示器映射 • 打印机映射	表单: • textForm • printForm
I/O 选项和 I/O 对象	I/O 选项和 I/O 对象	EGL I/O 语句
Java 应用程序或 applet (图形用户界面)	Smalltalk 视图或可视部件 (图形用户界面)	<ul style="list-style-type: none"> <li>不支持 Smalltalk 视图和可视部件。</li> <li>如果 Java 应用程序和 applet 在自由格式面板上未使用 VAGen 部件, 则该 Java 应用程序和 applet 受支持。如果 Java 应用程序和 applet 在自由格式面板上使用了 VAGen 部件, 则在当前发行版中该 Java 应用程序或 applet 不受支持。</li> </ul>
生成选项部件	生成选项部件	构建描述符部件
生成选项	生成选项	构建描述符选项
链接表部件	链接表部件	链接选项部件

表 5. VAGen 与 IDE Windows 的术语差别

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
日志 <ul style="list-style-type: none"> <li>显示错误消息</li> <li>仅当同时关闭了“日志”和“工作台”时，产品才会关闭</li> <li>关闭产品时，始终保存工作空间</li> </ul>	系统抄本 <ul style="list-style-type: none"> <li>显示错误消息</li> <li>如果关闭“系统抄本”或 VisualAge Organizer，则产品关闭</li> <li>关闭产品时，可以选择保存映像</li> </ul>	控制台 <ul style="list-style-type: none"> <li>显示消息。</li> <li>“问题”视图</li> <li>显示消息，尤其是那些与语法验证相关的消息。</li> <li>关闭产品时，始终保存工作空间。</li> </ul>
工作台 <ul style="list-style-type: none"> <li>显示工作空间中的项目和包。</li> </ul>	VisualAge Organizer <ul style="list-style-type: none"> <li>显示映像中的应用程序。</li> </ul>	EGL 透视图和 Web 透视图: <ul style="list-style-type: none"> <li>“导航器”和“项目资源管理器”视图显示工作空间中的项目、源文件夹、包和文件。</li> </ul>
代码片断编辑测试窗	工作空间	代码片断编辑测试窗页面编辑器
资源库资源管理器	应用程序版本浏览器	如果您决定使用资源库，则该资源库可能有对应的概念。
VAGen 部件浏览器 <ul style="list-style-type: none"> <li>在三个窗格中显示了包、部件类型和 VAGen 部件</li> <li>浏览器包含过滤功能</li> </ul>	VAGen 部件浏览器 <ul style="list-style-type: none"> <li>在三个窗格中显示了应用程序、部件类型和 VAGen 部件</li> <li>浏览器包含过滤功能</li> </ul>	EGL 透视图和 Web 透视图: <ul style="list-style-type: none"> <li>“导航器”和“项目资源管理器”视图显示工作空间中的项目、源文件夹、包和文件。</li> <li>“大纲”视图显示了文件中的部件。</li> </ul>
VAGen 选项	VAGen 首选项	EGL 首选项
VAGen 选项	VASmalltalk 首选项	其它产品首选项
“引用”工具，用于查找使用特定部件名或文本字符串的部件	“引用”工具，用于查找使用特定部件名或文本字符串的部件	EGL 搜索或文件搜索
“关联”工具，用于查找被特定部件引用的所有部件	“关联”工具，用于查找被特定部件引用的所有部件	EGL 部件引用

表 6. VAGen 工作空间管理术语差别

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
资源库	库	无。根据您使用的产品的不同，提供了 CVS 和 Clear Case LT。您可以选择自己的存储库管理系统。
添加 / 删除	装入 / 卸装	如果您决定使用资源库，则该资源库可能有对应的概念。
替换为	装入另一个版本	替换为本地历史记录 注：您决定使用的资源库可能有其它工具。

表 6. VAGen 工作空间管理术语差别 (续)

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
进行比较	浏览更改	与本地历史记录进行比较 <b>注:</b> 您决定使用的资源库可能有其它工具。

表 7. VAGen 资源库管理术语差别

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
管理程序	库管理程序	如果您决定使用资源库, 则该资源库可能有对应的概念。
资源库管理: • 清除 / 复原 • 压缩	库管理: • 清除 / 恢复 • 克隆	如果您决定使用资源库, 则该资源库可能有对应的概念。

表 8. VAGen 源代码管理术语差别

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
所有权: • 项目所有者 • 包所有者 • 类所有者	所有权: • 配置映射管理员 • 应用程序管理员 • 类所有者	如果您决定使用资源库, 则该资源库可能有对应的概念。
版本和发行版	版本和发行版	如果您决定使用资源库, 则该资源库可能有对应的概念。
项目: 1. 项目是必需的。 2. VAGen 项目列表部件指定了项目之间的关系。 3. 包所有者始终可以将包发布到项目中。	配置映射: 1. 可以选择使用。 2. 必需映射指定了配置映射之间的关系。 3. (可选) 可以委托配置映射管理员发布应用程序或者进行限制以便只允许配置映射管理员发布应用程序。	项目: 1. 项目是必需的。 2. 项目的“EGL 构建路径”属性。但是, 这并不会自动地将项目一起装入到工作空间中。 3. 除非资源库提供了对应的概念, 否则没有对应的概念。
包: 1. 没有对应的概念 2. 没有对应的概念 3. 没有对应的概念 4. 组成员 5. 在设置项目的版本时, 会自动地设置所包括的包的版本。	应用程序: 1. 必备应用程序 2. 子应用程序 3. 特权 4. 组成员 5. 在设置配置映射的版本之前, 必须设置应用程序的版本	文件夹或包: • 如果您决定使用资源库, 则该资源库可能有对应的概念。
类或类型: • 在设置包或项目的版本时, 会自动地设置所包括的类的版本。	类: • 在设置应用程序的版本之前, 必须设置类的版本并发布该类。	EGL 部件类型: • 在 EGL 中没有对应的概念。

表 8. VAGen 源代码管理术语差别 (续)

VisualAge Generator on Java	VisualAge Generator on Smalltalk	EGL
<p>VAGen 部件:</p> <ul style="list-style-type: none"> <li>• 每个部件都有日期和时间戳记。</li> <li>• 可以将包含重复部件名的包添加到工作空间中。</li> <li>• 提供了用于找出重复部件的重复部件工具。</li> </ul>	<p>VAGen 部件:</p> <ul style="list-style-type: none"> <li>• 每个部件都有日期和时间戳记。</li> <li>• 不能将包含重复部件名的应用程序装入到映像中。</li> </ul>	<p>EGL 部件:</p> <ul style="list-style-type: none"> <li>• 部件包含在 EGL 文件中; 只有 EGL 文件才能具有日期和时间戳记。</li> <li>• 工作空间中可以包含重复的部件。通过使用项目的 EGL 构建路径、文件的 import 语句以及 containerContextDependent 属性的组合, EGL 可以确定为了解析部件名引用而搜索的名称空间。部件名在名称空间中必须是唯一的。项目的 EGL 构建路径限制了在查找部件名时要考虑的其它项目。文件的 import 语句限制了在查找部件名时在 EGL 构建路径中要考虑的其它包和 / 或部件。记录或函数的 containerContextDependent 属性指定 EGL 应该使用包含程序的文件的 EGL 构建路径和 import 语句而不是包含记录或函数的文件的那些构建路径和 import 语句。</li> </ul>

---

## 第 2 章 迁移工具原理

VisualAge Generator 到 EGL 迁移工具实际上是一系列工具。本章提供工具的高级概述并描述了工具使用的技术。

---

VisualAge Generator 到 EGL 迁移工具的设计具有几项主要目标:

- 在从 VisualAge Generator 迁移到 EGL 时, 保留程序的行为。
- 在从 VisualAge Generator 迁移到 EGL 时, 在适当的时候, 保留 Java 项目和包结构。
- 在从 VisualAge Generator 迁移到 EGL 时, 在适当的时候, 保留 Smalltalk 配置映射和应用程序结构。
- 使您能够执行递增式子系统迁移 (每次迁移一个子系统)。
- 使您能够迁移子系统的多个版本。

VisualAge Generator 到 EGL 迁移工具的设计还具有几项次要目标:

- 尽最大可能使用批处理方式处理, 并且您有机会在关键点选择查看规划的迁移, 然后再继续执行下一个步骤。
- 将关于已规划的迁移的信息存储在数据库中, 以便可以跨多个项目版本以及跨多个子系统保留该信息。这还使您能够将中间结果保存为备份。如果资源库包含大量的部件, 这一点就很重要。
- 为从资源库中抽取 VAGen 源代码并装入迁移数据库的工具提供一组样本程序。(可选) 您可以定制样本程序以便更准确地反映您所处的环境。

VisualAge Generator 到 EGL 迁移工具的设计基于下列假定:

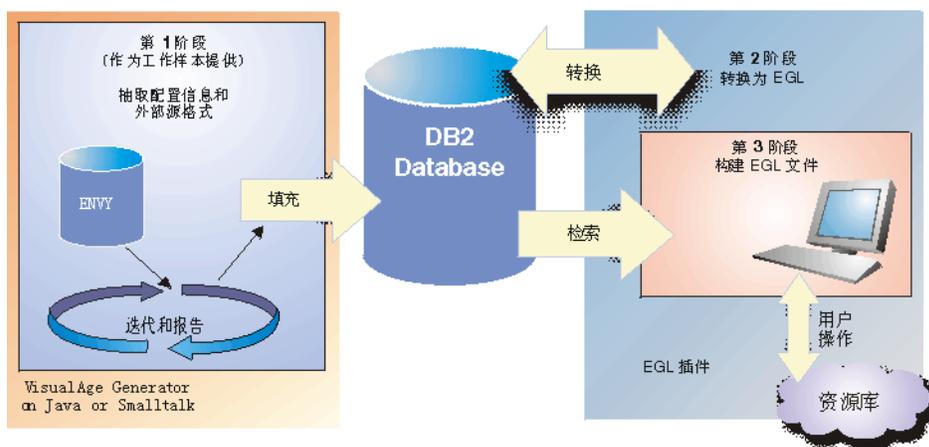
- 迁移是使用由 **VisualAge Generator 4.5** 生成的外部源格式从 VisualAge Generator 4.5 进行的。
- 要迁移的部件是有效的 **VisualAge Generator** 部件。在 VisualAge Generator 4.5 中, 可以验证和 / 或生成程序、表和映射组。

可以通过两种方法来使用 VisualAge Generator 到 EGL 迁移工具:

- 第 1 - 3 阶段迁移, 第 14 页的『VisualAge Generator 到 EGL 迁移工具概述』对此作了描述。这是主要的源代码迁移技术。
- 单文件迁移, 第 19 页的『单文件迁移概述』对此作了描述。对于要通过迁移几个程序来验证环境的运作是否正常的情况, 此技术是非常有用的。

## VisualAge Generator 到 EGL 迁移工具概述

为了实现上面列示的目标，VisualAge Generator 到 EGL 迁移工具实际上包含一系列的工 具，这些工具组织成下图中显示三个阶段。



- 用于第 1 阶段的工具在 VAGen 环境中运行。第 1 阶段工具从 Java 资源库或 Smalltalk 库中抽取关于源代码组织以及源代码本身的信息。第 1 阶段工具将此信息装入到迁移数据库中。VAGen 源代码是以外部源格式存储的。
- 用于第 2 阶段的工具在 EGL 环境中运行。第 2 阶段工具使用迁移数据库中存储的信息来为第 1 阶段期间存储在迁移数据库中的 VAGen 部件创建 EGL 语法。第 2 阶段工具将得到的 EGL 源代码存储在迁移数据库中。
- 用于第 3 阶段的工具也在 EGL 环境中运行。对于您想要创建的每个 EGL 项目，第 3 阶段工具为属于该项目的部件抽取 EGL 源代码并在文件系统中创建 EGL 项目。（可选）如果您仅使用一组项目的一个版本，则第 3 阶段工具可以将项目导入到工作空间中。

将项目导入工作空间之后，您接着可以使用您决定使用的源代码资源库来设置项目的版本。您使用源代码资源库提供的工具来管理源代码。

### 迁移工具术语

为了进行交叉部件的迁移，在迁移部件时，您不仅必须提供部件本身，还必须提供它所引用的所有部件。例如，在迁移程序时，您不仅应该提供该程序，还应该提供该程序所引用的所有部件。对于一个程序，在迁移该程序时所需的部件集就是在 VisualAge Generator 中生成该程序时所需的部件集。这组部件就是该程序的关联项列表。

在 VisualAge Generator 中，通常使用下列技术来提供生成所需的所有部件：

- VisualAge Generator on Java 中的项目列表部件（PLP）
- VisualAge Generator on Smalltalk 中的配置映射。

迁移工具利用这两种技术。此工具使用下列术语：

- 如果正在从 VisualAge Generator on Java 迁移：
  - 高级 PLP 项目是包含项目列表部件（PLP）并且未被任何其它 PLP 引用的 Java 项目。
  - 迁移集由 Java 高级 PLP 项目中引用的所有 VAGen 项目组成，这包括整个 PLP 链中从该高级 PLP 项目开始的所有 VAGen 项目。

- 如果正在从 VisualAge Generator on Smalltalk 迁移：
  - 高级配置映射是未被任何其它配置映射列示为必需映射的 Smalltalk 配置映射。
  - 迁移集由 Smalltalk 高级配置映射中列示为必需映射的所有 Smalltalk 配置映射组成。迁移集包含整个 Smalltalk 必需映射链中从该高级配置映射开始的所有配置映射。
- 迁移规划是一个文件，它指定了一个或多个迁移集的信息。如果在第 1 阶段首选项中指定迁移规划文件名，则所有与存储库过滤器相匹配的迁移集都将放在同一个迁移规划文件中。如果您未指定迁移规划文件名，则每个迁移集都放在独立的迁移规划文件中。

注意：如果正在从 VisualAge Generator on Java 迁移并且当前未使用 PLP 项目，则可以创建仅用于迁移的 PLP 项目。此外，可以执行下列其中一项操作：

- 如果数据库或其它系统中的信息指定了特定的 Java 项目版本进行迁移所需的内容，则您可以编写一个工具来根据数据库自动创建迁移规划文件。
- 手工创建迁移规划文件。

如果正在从 VisualAge Generator on Java 迁移，则请参阅第 98 页的『迁移规划和高级 PLP 项目』一节以了解更多详细信息。

## 第 1 阶段详细信息

第 1 阶段工具是作为样本程序与 Rational Developer 或 WebSphere Studio 产品一起交付的。根据您当前所使用的 VisualAge Generator Developer 4.5 产品的不同，可以安装不同的样本程序以便在 VisualAge Generator Developer on Java 或 VisualAge Generator Developer on Smalltalk 上运行。由于 Java 环境与 Smalltalk 环境之间存在差别，所以，这两个样本程序略有不同。但是，在这两个环境中，使用第 1 阶段样本程序的基本步骤是相同的。第 1 阶段的基本步骤如下所示：

- 步骤 1. 定义规则和首选项以便对第 1 阶段迁移进行指引。
- 步骤 2. 运行工具并生成下列其中一项或多项输出：
  1. 一个或多个迁移规划文件
  2. 一个报告，此报告显示每个迁移规划文件的迁移方式
  3. 一个日志文件，此日志文件包含关于检测到的任何问题的消息
  4. 一个迁移数据库

### 步骤 1

定义规则和首选项，它们向第 1 阶段工具提供有关所要迁移的内容的信息，这些信息包括：

1. 如何过滤 Java 项目名以便只考虑您想要迁移的项目。对于 Smalltalk 来说，您指定如何对 Smalltalk 配置映射名进行过滤。这样，由于第 1 阶段工具只需要处理那些与过滤器相匹配的 Java 项目或 Smalltalk 配置映射，所以，第 1 阶段的性能会有所提高。
  - 从那些与过滤器匹配的 Java 项目中，第 1 阶段工具选择任何包含高级项目列表部件的 Java 项目。对于 Java 项目来说，如果该 Java 项目未被任何其它 PLP 引用，它就包含高级项目列表部件（PLP）。
  - 从与过滤器相匹配的 Smalltalk 配置映射中，第 1 阶段工具选择任何高级配置映射。高级配置映射是未被任何其它配置映射列示为必需映射的配置映射。

2. 您是想要创建一个对可以根据过滤器迁移的所有内容都产生影响的迁移规划，还是想要创建多个迁移规划（为每个 Java 高级 PLP 项目版本或 Smalltalk 高级配置映射版本创建一个迁移规划）。
3. 如何根据 Java 项目名和包名或者根据 Smalltalk 配置映射名和应用程序名来创建 EGL 项目名、包名和文件名。您可以指定的信息包括：
  - 一些规则，这些规则指示哪些 Java 项目和包或者 Smalltalk 配置映射和应用程序包含公共代码。
  - 在创建 EGL 项目名和包名时要使用的重命名规则。
  - 一些名称，对于包含公共部件或未使用的部件的 EGL 文件，将使用这些名称。
4. 迁移数据库的名称以及访问该数据库所需的用户标识和密码。
5. 您想让第 1 阶段工具在步骤 2 中生成哪些输出。您可以选择在单个步骤中创建所有输出，也可以依次创建输出以便在创建下一项更耗时的输出之前有机会查看规则和首选项。

## 步骤 2

根据已定义的规则和首选项，第 1 阶段工具生成下列可能的输出：

1. **迁移规划文件。** 迁移规划文件包含迁移集。每个迁移集都是 Java 资源库中的一个高级 PLP 项目版本或者 Smalltalk 库中的一个高级配置映射版本。从属 Java 项目版本或必需的 Smalltalk 配置映射版本是在迁移集中指定的。
  - 如果迁移首选项文件未指定迁移规划文件名选项值，则将创建多个迁移规划文件。对于 Java 来说，每个高级 PLP 项目版本都将产生一个包含一个迁移集的迁移规划文件。同样，对于 Smalltalk 来说，每个高级配置映射版本都将产生一个包含一个迁移集版本的迁移规划文件。
  - 如果迁移首选项文件指定了迁移规划文件名选项值，则对于 Java 来说，每个高级 PLP 项目版本都将在单个迁移规划文件中产生一个迁移集条目。同样，对于 Smalltalk 来说，每个高级配置映射版本都将在单个迁移规划文件中产生一个迁移集条目。
  - 例如，考虑由 6 个 Java 项目组成的订单输入系统，其中，第 6 个 Java 项目包含一个 PLP，该 PLP 指定了其它 5 个项目的版本。如果您请求多个迁移规划和 3 个版本，则将创建 3 个迁移集 - 为每个包含 PLP 部件的 Java 订单输入项目版本创建一个迁移集。同样，对于 Smalltalk 来说，如果要迁移配置映射（此配置映射反映了构成订单输入系统的代码）的 3 个版本，则将有 3 个迁移集 - 此高级配置映射的每个版本都有 1 个迁移集。

此时您可以指示停止第 1 阶段工具，以便您有机会查看迁移规划文件以确保迁移规划文件正确反映了所要迁移的 Java 项目版本或 Smalltalk 配置映射版本。

2. **一个报告，此报告显示了每个迁移集的迁移方式。** 第 1 阶段工具可以在不装入数据库的情况下生成此报告。这可以帮助您确保过滤器和首选项选择正确的一组 Java 项目或 Smalltalk 配置映射，并确保您对重命名规则所产生的 EGL 项目、包和文件命名约定感到满意。通过查看报告，如果您对建议的 EGL 结构不满意，则可在第 1 阶段工具实际装入数据库之前，您就有机会优化规则和首选项。如果有必要的话，您可以将上述步骤迭代任意次，直到您对迁移的内容以及建议的 EGL 结构感到满意为止。此报告显示了下列内容：
  - 对于 Java，每个迁移集都列示了所包括的项目版本。对于每个项目版本，您可以查看包版本，对于每个包版本，您可以查看 VAGen 部件列表。

- 对于 Smalltalk 来说，每个迁移集都列示了所包括的配置映射版本。对于每个配置映射版本，您可以查看应用程序版本，对于每个应用程序版本，您可以查看 VAGen 部件列表。

对于每个 VAGen 部件，您可以查看相应的 EGL 项目、包和文件名，这将是该部件的放置位置。对于每个 VAGen 部件，您还可以查看 VisualAge Generator 创建的关联项列表以及存放关联项的 EGL 文件。请参阅第 30 页的『将部件放到 EGL 文件中』一节以了解有关在第 1 - 3 阶段迁移期间如何将 VAGen 部件分配到文件中的信息。

3. 一个日志文件，如果任何 VAGen 程序、表、映射组或控制部件名与 EGL 保留字列表有冲突，此日志文件就会提供消息。在迁移期间，这些部件不会被重命名。您可以在 VisualAge Generator 中将这此部件重命名，也可以等待迁移到 EGL 之后再执行此操作。
4. 一个迁移数据库，此迁移数据库将根据迁移规划文件随信息和 VAGen 源代码一起装入。您可以选择一个迁移规划以便在装入该数据库时使用，也可以使用目录中的所有迁移规划文件。第 1 阶段迁移工具将随下列各项一起装入该数据库：
  - 关于所选迁移规划文件中的每个迁移集的信息。
  - 该迁移集的相关 Java 项目集或 Smalltalk 配置映射集。
  - Java 项目集或 Smalltalk 配置映射集中的每个 VAGen 部件的外部源格式 VAGen 部件定义。
  - 每个 Java 项目、包和 VAGen 部件或者每个 Smalltalk 配置映射、应用程序和 VAGen 部件的对应 EGL 项目名、包名和文件名。
  - 在此步骤期间还将创建一个报告，因此，对装入到迁移数据库中的内容作了完整的记录。此报告与上一个报告具有相同的格式。

对于 VisualAge Generator 的 Java 版本和 Smalltalk 版本来说，第 1 阶段工具都是以样本程序形式提供的。您可以“按现状”使用第 1 阶段工具，也可以对样本程序进行修改以使其更适合于您的环境。例如，您当前可能将配置信息存储在 Java 资源库或 Smalltalk 库外部。此配置信息可能指定了生成操作所需的源代码版本。在这种情况下，您在编写自己的工具来根据配置信息和 Java 资源库或 Smalltalk 库的组合来装入迁移数据库时，可以将样本程序用作指南。

如果您修改第 1 阶段样本程序，则您可能想修改迁移数据库以使其包含其它信息，以便帮助进行代码分析。您可以对现有的 SQL 表添加其它列，也可以对迁移数据库添加其它表。但是，在第 2 阶段和第 3 阶段迁移中将不会使用这些新列和新表。此外，如果您修改第 1 阶段样本程序，则必须确保用样本程序中显示的信息来填充 SQL 表。如果未这样做，则第 2 阶段和第 3 阶段将无法迁移代码。

请参阅第 87 页的第 4 章，『第 1 阶段 - 从 Java 中抽取』以了解有关在 VisualAge Generator Developer on Java 上安装和运行第 1 阶段工具的详细信息。请参阅第 105 页的第 5 章，『第 1 阶段 - 从 Smalltalk 中抽取』以了解有关在 VisualAge Generator Developer on Smalltalk 上安装和运行第 1 阶段工具的详细信息。

## 第 2 阶段详细信息

第 2 阶段工具是在 Eclipse 插件 `com.ibm.etools.egl.vagenmigration` 中交付的，并且在 EGL 环境中运行。由于要迁移的信息现在位于迁移数据库中，所以，无论是从

VisualAge Generator on Java 迁移还是从 VisualAge Generator on Smalltalk 迁移，使用的是同一个第 2 阶段工具。运行第 2 阶段工具的基本步骤如下所示：

1. 您定义规则和首选项以告知第 2 阶段工具要迁移的内容，这些内容包括：
  - 有关 EGL 源代码创建方式的特定详细信息。例如，第 2 阶段迁移工具必须将 VAGen 工作存储器记录分割成两个 EGL 基本记录：
    - a. 第一个记录与原始工作存储器记录同名并包含所有非 Level-77 项。
    - b. 第二个记录与原始工作存储器记录同名、具有后缀并包含所有 Level-77 项。

有一个第 2 阶段迁移首选项，它使您能够指定每当第 2 阶段工具创建包含 Level-77 项的新记录时要让它使用的后缀。

  - 要迁移的迁移集。例如，如果您创建了 3 个迁移集来迁移订单输入系统的 3 个不同版本，您最初可能只想迁移 1 个版本。这使您能够灵活地对迁移进行限制，而不必同时迁移迁移数据库中的所有内容。
  - 迁移数据库的名称以及访问该数据库所需的用户标识和密码。如果未显式地指定数据库用户标识和密码，则第 2 阶段和第 3 阶段迁移工具都将尝试使用用来登录至 Windows 机器的用户标识和密码来登录至数据库。
  - 在第 2 阶段完成后是否要自动启动第 3 阶段工具。如果自动运行第 3 阶段，则您可以选择将 EGL 项目的一个版本装入到工作空间中。您还可以选择将 EGL 项目装入到临时目录中，以便以后可以与源代码资源库进行交互。
2. 根据已定义的规则和首选项，第 2 阶段工具执行下列操作：
  - a. 从数据库中检索一个迁移集的部件。
  - b. 将外部源格式源代码转换为 EGL 源代码。
  - c. 将 EGL 源代码存储在迁移数据库中。与部件迁移相关联的消息也存储在迁移数据库中。这将提高第 2 阶段的性能，这是因为，如果在另一个 Java 项目版本或 Smalltalk 配置映射版本中使用同一个部件版本，就已经有了 EGL 源代码，从而不需要再次进行转换。
  - d. 创建一个所遇到的任何潜在问题的日志文件，这些问题包括与 EGL 保留字列表有冲突的可生成部件或者迁移工具无法解决的不确定情况。
  - e. 进行迭代以处理下一个所选迁移集。

第 2 阶段迁移工具能够以批处理方式运行。请参阅第 123 页的第 6 章，『第 2 阶段 - 转换为 EGL 语法』以了解有关在 EGL 开发环境中安装和运行第 2 阶段工具的详细信息。您不能修改第 2 阶段迁移工具。

## 第 3 阶段详细信息

第 3 阶段工具与第 2 阶段工具是在同一个 Eclipse 插件 (com.ibm.ertools.egl.vagenmigration) 中交付的，并且也在 EGL 环境中运行。由于要迁移的信息现在位于迁移数据库中，所以，无论是从 VisualAge Generator on Java 迁移还是从 VisualAge Generator on Smalltalk 迁移，使用的是同一个第 3 阶段工具。运行第 3 阶段工具的基本步骤如下所示：

1. 您定义规则和首选项以告知第 3 阶段工具要迁移的内容，这些内容包括：
  - 要迁移的迁移集。例如，如果您创建了 3 个迁移集来迁移订单输入系统的 3 个不同版本，您可能已通过第 2 阶段工具迁移了全部 3 个版本，但只想通过第 3 阶

段工具迁移 1 个版本。在第 3 阶段只迁移 1 个版本的最常见原因是，您想要在源代码资源库中设置此代码的版本，然后使用第 3 阶段工具迁移下一个版本并在源代码资源库中设置它的版本。

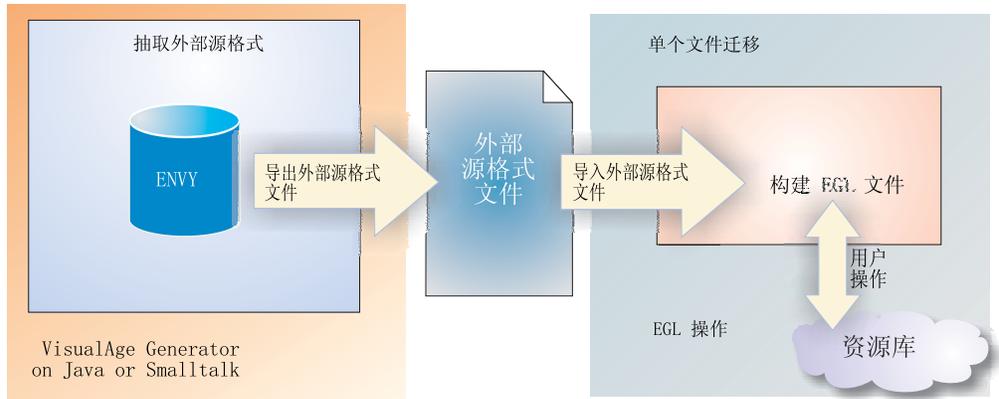
- 迁移数据库的名称以及访问该数据库所需的用户标识和密码。
- 2. 根据已定义的规则和首选项，第 3 阶段工具执行下列操作：
  - a. 为迁移集创建一个任务列表。这个任务列表包含由第 2 阶段生成的消息的汇总表，那些消息可能要求您执行其它任务以完成迁移。
  - b. 根据第 1 阶段期间在迁移数据库中存储的信息来在工作空间中创建 EGL 项目和包结构。
  - c. 根据第 2 阶段期间为 VAGen 部件存储的 EGL 源代码来创建 .egl 源文件。 .egl 源文件包含解析 EGL 部件引用所需的大部分 import 语句。请参阅第 28 页的『EGL 构建路径和 import 语句』以了解有关 import 语句的详细信息。
  - d. 根据第 2 阶段期间为 VAGen 控制部件存储的 EGL XML 源代码来创建 .eglbld 文件。控制部件包括生成选项部件（EGL 构建描述符部件）、链接选项部件、资源关联部件、绑定控制部件和链接编辑部件。
- 3. 此刻，您应该执行下列操作：
  - a. 查看工作空间中的“问题”视图是否包含任何消息，以了解是否发生了任何验证错误。您可以结合第 2 阶段生成的日志或者第 3 阶段生成的任务列表来进行此项验证错误检查。
  - b. 生成（但不准备）所有程序和 dataTable，以确保能够针对目标环境进行正确的迁移。当生成程序时，务必使用 genFormGroup 和 genHelpFormGroup 构建描述符选项，以便生成所有 formGroup。此步骤是可选的，但强烈建议您执行此步骤。
  - c. 设置 EGL 项目的版本或者将那些项目提交到源代码资源库中。
  - d. 生成并测试已迁移的代码。此步骤是可选的，但强烈建议您执行此步骤。

第 3 阶段迁移工具能够以批处理方式运行。请参阅第 137 页的第 7 章，『第 3 阶段 - 导入』以了解有关对 EGL 运行第 3 阶段工具的详细信息。第 3 阶段工具是在您安装第 2 阶段工具时自动安装的。您不能修改第 3 阶段迁移工具。

---

## 单文件迁移概述

当您刚开始熟悉 EGL 和设置环境时，您可能只想迁移几个程序以对环境进行验证、确保生成和准备过程正常进行并确保正确地 EGL 配置了运行时环境。在这种情况下，您可能不想全面地完成第 1 - 3 阶段迁移。第 2 阶段迁移工具提供了一种机制以“单文件迁移”方式迁移程序，如下图所示：



与第 1 - 3 阶段迁移相比，单文件迁移是一个更加手工化的过程。在单文件方式下，您使用 VisualAge Generator 来将外部源格式源代码导出到一个文件中。然后，使用 EGL 开发环境来创建 EGL 项目和 EGL 包。然后，可以使用“导入”向导将“外部源格式”文件导入到 EGL 源中。于是，单文件迁移工具将运行并执行下列操作：

- 如果目标 EGL 源文件还不存在，则创建该文件。如果该文件已存在，则您可以选择覆盖该文件或对该文件进行追加。根据首选项以及外部源格式文件中包含的部件的不同，迁移工具可能会创建其它 EGL 文件。
- 将外部源格式源代码转换为 EGL 源代码。
- 创建一个所遇到的任何潜在问题的日志文件，这些问题包括与 EGL 保留字列表有冲突的可生成部件或者迁移工具无法解决的不确定情况。

在单文件迁移期间执行的外部源格式到 EGL 转换基本上就是在第 1 - 3 阶段迁移的第 2 阶段期间执行的语法转换。但是，单文件迁移具有几项局限性，这使得它不适合于大规模迁移。这些局限性包括：

- 在迁移期间，只考虑一个外部源格式文件中的部件。要实现可能的最佳迁移，请将程序及其所有关联项都包括在该外部源格式文件中。
- 将 VAGen 部件放到文件中的方式与第 1 - 3 阶段迁移所采用的方式不同。在单文件方式下，假定您指定了 *targetFile.egl* 作为目标 EGL 文件名，则会发生下列情况：
  - 所有控制部件都放在名为 *targetFile.egl* 的文件中。
  - 如果您未选择“将可生成的各部件分别保存到不同的 EGL 文件中”首选项，则所有其余部件都将放在名为 *targetFile.egl* 的文件中。
  - 如果您选择了“将可生成的各部件分别保存到不同的 EGL 文件中”首选项，则会发生下列情况：
    - 每个程序部件都放到名为 *programName.egl* 的文件中，其中 *programName* 是程序名。
    - 每个表部件都放到名为 *tableName.egl* 的文件中，其中 *tableName* 是表名。
    - 每个映射组以及该映射组中的所有映射都将放到名为 *mapGroupName.egl* 的文件中，其中 *mapGroupName* 是该映射组的名称。
    - 所有其余部件都将放到名为 *targetFile.egl* 的文件中。如果要将所有其余部件都与程序放在同一个文件中，则 *targetFile* 可以与 *programName.egl* 相同。不会尝试确定哪些部件被多个可生成部件共享。
  - 所有文件都放在同一个 EGL 项目、源文件夹和包中。

- 由于所有输出文件都放在同一个 EGL 包中，所以，迁移工具不会包括任何 import 语句。另外，由于所有部件都放在同一个 EGL 包中，所以，原始 Java 项目和包结构不会保留下来。同样，原始 Smalltalk 配置映射和应用程序结构也不会保留下来。
- 如果同一个部件在外部源格式文件中出现多次，则只迁移最后一个定义。
- 当以单文件方式进行迁移时，可以选择采用三种技术来处理公共部件。在从这两种技术中进行选择之前，您务必了解每种技术的缺点。这三种技术如下所示：
  - 如果将两个外部源格式文件迁移到同一个 EGL 包中，这两个文件包含同一个部件，并且您指定了不同的 **targetFile** 名，就会产生 EGL 无法解析的重复部件。如果迁移一个带有 Program1 及其关联项的文件以及第二个带有 Program2 及其关联项的文件，并且这两个程序共享一些公共部件，就会发生存在重复部件的情况。迁移工具将把公共部件放到这两个 targetFile 中。您可以通过将每个程序及其关联项迁移到不同的 EGL 包中来避免此问题。这仍然会导致工作空间包含重复的部件，但是，由于它们位于不同的包中，所以 EGL 能够解析部件引用。
  - 如果将两个外部源格式文件迁移到同一个 EGL 包中，这两个文件包含同一个部件，并且您指定了同一个 **targetFile** 名，则会询问您是否要覆盖工作空间中的现有文件。
    - 如果您指定不想覆盖现有文件，则第二个导入中所有放到现有文件中的部件都将添加到现有文件中。第二个导入中的所有公共部件都将导致 targetFile 包含重复部件。
    - 如果您指定要覆盖现有文件，则第二个导入中所有放到现有文件中的部件都将完全地替换现有的 targetFile。这将导致包括在第一个导入中但未包括在第二个导入中的那些部件丢失。

与第一种技术相同，您可以通过将每个程序及其关联项迁移到不同的 EGL 包中来避免此问题。这仍然会导致工作空间包含重复的部件，但是，由于它们位于不同的包中，所以 EGL 能够解析部件引用。

- 如果将公共部件分到另一个外部源格式文件中，就可能没有以单文件为基础执行良好的 VisualAge Generator 到 EGL 迁移所必需的所有信息。例如，如果 SQL 记录位于一个外部源格式文件中，并且使用该记录的已修改的 SQL 的函数位于另一个文件中，则迁移工具无法全面地为该函数构建 I/O 语句。另外，如果公共部件位于另一个包中，则必须对每个需要引用公共包的部件添加 EGL import 语句。
- 在单文件方式下，也不执行由第 2 阶段和第 3 阶段执行的附加处理。一些示例如下所示：
  - 表单不嵌套在 formGroup 中。
  - 对属于该部件类型的文件中的部件没有按部件名排序。
  - 在输出文件中，各部件之间仅由一条线分隔。

请参阅第 21 页的『迁移问题』和第 26 页的『VisualAge Generator 到 EGL 迁移工具使用的技术』以便更好地了解单文件迁移与第 1 - 3 阶段迁移之间的差别。

---

## 迁移问题

在 VisualAge Generator 与 EGL 之间，编写和管理源代码的方法有几项差别。下列差别对于迁移来说特别重要：

- 在某些情况下，EGL 语法比 VisualAge Generator 更精确
- 在解析部件引用的时间以及方式方面的差别

- 在处理公共代码方面的差别

下列各节对这些差别作了更详细的说明。

## 精确的 EGL 语法

尽管这两种语言的语法有着很大的不同，但是，对于大多数部件来说，可以将 VAGen 语言迁移到 EGL 语言并保持行为与原始 VAGen 程序相同。但是，在许多情况下，EGL 语法比 VisualAge Generator 更精确或者限制性更强。这些情况在典型程序中很罕见。但是，当发生这些情况时，迁移工具要求进行交叉部件迁移以确定能够保留 VisualAge Generator 中的所需行为的精确 EGL 语法。交叉部件迁移意味着迁移工具要求一个或多个其它被引用部件可用，这样才能正确地迁移当前部件。以下是一些示例：

- 在 VisualAge Generator 中，对显示器（文本）映射和打印机映射都可以使用 DISPLAY I/O 选项。EGL 提供了用于文本表单的 display 语句以及用于打印表单的 print 语句。为了方便地从 VisualAge Generator 迁移，有一个 EGL 首选项用于指示您想要采用“VisualAge Generator 兼容性”方式。“VisualAge Generator 兼容性”首选项允许对打印表单使用 display 语句。在迁移期间，如果程序、它的映射组以及映射全都可用，则迁移工具可以确定是迁移到 display 语句还是迁移到 print 语句。但是，如果在不迁移程序的情况下迁移 DISPLAY 函数，则迁移工具无法决定性地确定是使用 EGL display 语句还是使用 print 语句。在这种情况下，迁移工具将使用 display 语句，这是因为，在“VisualAge Generator 兼容性”方式下，此语句可用于打印表单。
- 在 VisualAge Generator 中，对显示器（文本）映射和打印机映射都可以使用 SET<sup>TM</sup> map PAGE 语句。如果下一个 CONVERSE 或 DISPLAY 用于显示器映射，则此语句将导致屏幕被清除，如果下一个 DISPLAY 用于打印机映射，则此语句将导致换页。EGL 提供了用于文本表单的 clearScreen 系统库函数以及用于打印表单的 pageEject 系统库函数。“VisualAge Generator 兼容性”首选项不会影响 clearScreen 或 pageEject 的使用。在迁移期间，如果程序、它的映射组以及映射全都可用，则迁移工具可以确定是迁移到 clearScreen 系统库函数还是迁移到 pageEject 系统库函数。但是，如果在不迁移程序的情况下迁移的函数中使用 SET map PAGE 语句，则迁移工具无法决定性地确定是使用 clearScreen 系统库函数还是使用 pageEject 系统库函数。在这种情况下，迁移工具使用 EZE\_SETPAGE，这将故意生成的无效 EGL 语法。这将导致“问题”视图中出现一个错误，以使您知道需要更正该函数。
- 在 VisualAge Generator 中，您可以指定编辑表或编辑函数来作为映射变量字段的编辑例程。不能同时指定这两者。在 EGL 中，可以同时指定 validatorDataTable 和 validatorFunction 属性。如果该编辑表或编辑函数在迁移期间可用，则迁移工具可以确定是设置 validatorDataTable 还是 validatorFunction 属性。但是，如果编辑例程指定的部件不可用，则迁移工具无法决定性地确定是设置 EGL validatorDataTable 还是 validatorFunction 属性。在这种情况下，迁移工具通过使用诸如编辑例程名长度以及编辑消息存在与否之类的信息来尝试确定该编辑例程是表还是函数。如果迁移工具仍无法进行确定，它将使用 validatorFunction 属性。仅当 validatorFunction 不是函数或者找不到时，在“问题”视图中才会出现错误。

迁移工具使用迁移集中的所有可用部件来解析不确定情况。为了最大程度地减少这些不确定情况，在进行迁移时始终应该包括关联部件。例如，在迁移程序时，务必包括在 VAGen 中生成该程序所需的所有部件。这将确保对部件进行可能的最佳迁移。要了解有关迁移工具如何解析不确定情况的概述，请参阅下列各节：

- 第 33 页的『带程序迁移』

- 第 33 页的『带关联部件迁移』
- 第 34 页的『不带关联部件迁移』

请参阅第 41 页的第 3 章,『处理不确定情况』以获取一个完整的情况列表,在那些情况下,迁移工具必须执行交叉部件迁移才能完成正确的迁移,该列表还提供了当其它部件不可用时迁移工具为了尝试进行智能选择而需执行的步骤。

## 解析部件名的时间及方式

在定义时, VisualAge Generator 不要求所有部件都存在。在程序结构图中, VisualAge Generator 用问号指示缺少的映射、记录、表和函数。但是,在其它位置(例如在使用共享数据项的位置),当部件当前不存在时,并没有指示。当您在 VisualAge Generator 中保存部件时,会进行一些基本的语法验证,但是,在进行测试、验证或生成之前,不会进行交叉部件验证。在 EGL 中,每当保存文件时,都会进行更广泛的验证 - 包括验证是否能够解析所有部件名。当存在问题时,这能够提供最早期的警告。

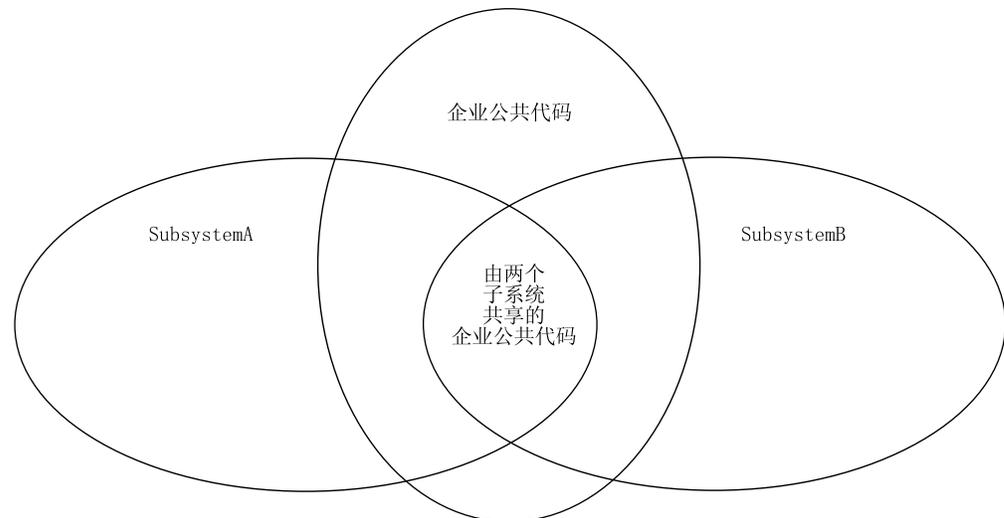
VisualAge Generator 在工作空间中搜索所有部件以查找特定部件名。如果在 VisualAge for Java 中有重复的部件名,则测试和生成将停止,直到您修正了重复部件问题为止。VisualAge for Smalltalk 不允许将重复部件装入到映像中。在 EGL 中,允许工作空间包含重复的部件名。EGL 使用项目的 EGL 构建路径、文件中的 import 语句以及记录和函数的 containerContextDependent 属性的组合来确定所要使用的部件定义。

当您使用第 1 - 3 阶段迁移来进行迁移时,迁移工具将为项目设置 EGL 构建路径并根据迁移集中的可用部件来将 import 语句包括在文件中。要获取正确的 EGL 构建路径和 import 语句,在进行迁移时始终应该包括所有关联部件。例如,在迁移程序时,务必包括在 VisualAge Generator 中生成该程序所需的所有部件。这将确保对部件进行可能的最佳迁移。要了解更多详细信息,请参阅下列各节:

- 第 28 页的『EGL 构建路径和 import 语句』
- 第 29 页的『containerContextDependent 属性』

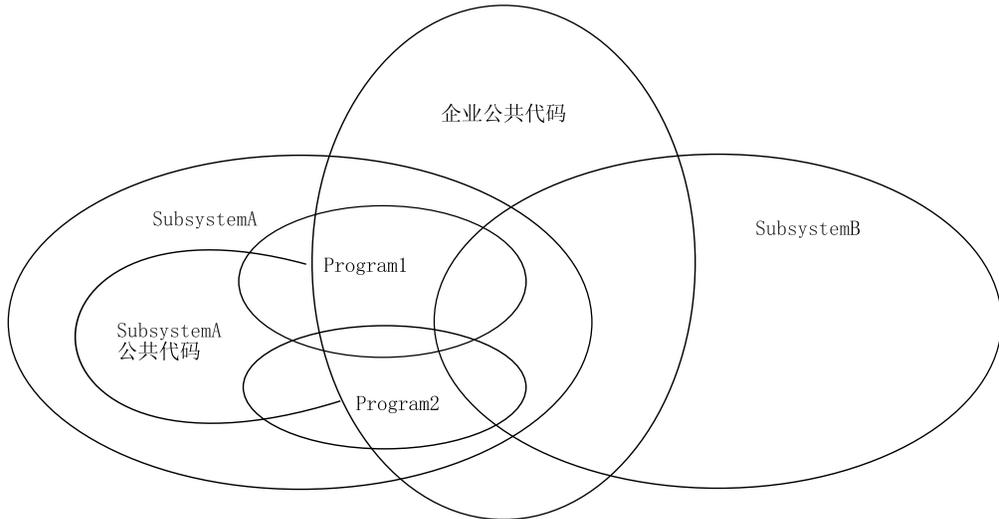
## 公共代码方案

公共代码是在子系统或程序之间共享的代码。下图显示了由两个子系统共享的公共代码。



在此实例中，有一个或多个 Java 项目或 Smalltalk 配置映射包含“企业公共代码”。这些项目或配置映射中的代码可以由多个子系统共享。在这个示例中，SubsystemA 和 SubsystemB 使用公共代码的子集。一部分“企业公共代码”同时被这两个子系统使用。例如，“企业公共代码”可能包括由许多子系统使用的 SQL 记录定义。

下图显示了两个子系统对“企业公共代码”进行相同的基本共享，并且，SubsystemA 的显示更为详细。



SubsystemA 具有由 SubsystemA 中的多个程序使用并且只能由 SubsystemA 中的程序使用的“SubsystemA 公共代码”。在此实例中，Program1 和 Program2 每个都使用部分“SubsystemA 公共代码”以及部分“企业公共代码”。在这两个程序之间，“SubsystemA 公共代码”与“企业公共代码”有一定程度的重叠，这包括与 SubsystemB 使用的“企业公共代码”的重叠。例如，“SubsystemA 公共代码”可能包括只由 SubsystemA 中的程序使用的 SQL 记录定义。“SubsystemA 公共代码”也可能包括由 SubsystemA 中的数个程序使用的映射组定义。

## 公共代码和 VisualAge Generator

为了使公共代码便于使用，VisualAge Generator 在测试和生成时确定应该如何解释特定的源代码部分。这样做的好处是，每个子系统或程序都可以通过改变程序所使用的特定映射组或通过改变工作空间在生成期间包含的数据项或记录定义来使代码略有不同。以下是三个示例：

- 与终端进行交互的联机程序以及打印类似报告的批处理程序可以共享几乎相同的逻辑，如下所示：
  - ProgramA 是使用 MapGrpA 的主事务程序，MapGrpA 包含名为 HEADER、DETAIL 和 TRAILER 的显示器映射。ProgramA 显示部分 HEADER 映射，在浮动区显示 DETAIL 行，然后转换 TRAILER 映射，该映射包含一个输入字段，用户可以在该字段中请求下一个报告。ProgramA 使用 SET HEADER PAGE 语句来清除屏幕。
  - ProgramB 是使用 MapGrpB 的主批处理程序，MapGrpB 包含名为 HEADER、DETAIL 和 TRAILER 的打印机映射。ProgramB 生成 ProgramA 在终端上显示的那个报告的硬拷贝版本。ProgramB 显示部分 HEADER 映射，在浮动区显示 DETAIL 打印行，然后在页面底部显示 TRAILER 映射。ProgramB 使用 SET HEADER PAGE 语句来强制换页。

- 主事务程序与主批处理程序的浮动区中的行数有所不同。但是，对于这两个程序，数据检索逻辑、数据处理逻辑以及 HEADER 和 DETAIL 映射的显示逻辑是相同的。因此，ProgramA 和 ProgramB 被设计成使用公共函数来从数据库中检索数据、处理数据以及显示 HEADER 和 DETAIL 映射。
- 在 VisualAge Generator 中，由于同一个 DISPLAY I/O 选项既可以用于显示器映射又可以用于打印机映射，所以这种公共代码技术能够起作用。另外，同一个 SET HEADER PAGE 语句既可以用于显示器映射又可以用于打印机映射。VisualAge Generator 根据它所测试或生成的特定程序来解释 DISPLAY I/O 选项和 SET map PAGE 语句。
- EGL 对显示表单和打印表单需要不同的语句 - 对文本表单使用 display 和 clearScreen; 对打印表单使用 print 和 pageEject。在“VisualAge Generator 兼容性”方式下，display 语句也可以用于打印表单。但是，clearScreen 仅适用于文本表单输出，即使在“VisualAge Generator 兼容性”方式下亦如此。
- 一个不太典型的示例是使用名为 SET-MESSAGE-TEXT 的公共错误处理程序函数，该函数从名为 MSGTBLE 的 VAGen 表中检索消息文本，并将该消息文本存储在名为 MESSAGE-TEXT 的函数参数中，其中 MESSAGE-TEXT 是共享数据项。
  - 假定 SubsystemA 和 SubsystemB 在不同的 CICS 区域中运行。在这种情况下，两个子系统都可以提供它们自己的 MSGTBLE 定义以及它们自己的 MESSAGE-TEXT 函数参数定义。如果这两个子系统在它们各自的映射定义中提供大小不同的错误消息字段，就可能发生这种情况。
  - 当 VisualAge Generator 生成程序时，它将使用当前装入在工作空间中的定义。只要每个子系统总是在进行测试或生成之前将自己的 MESSAGE-TEXT 数据项定义装入到工作空间中，VisualAge Generator 就将使用对该子系统正确的定义。此技术的缺点是您在进行生成时必须控制工作空间的内容，并且工作空间不能同时包含这两个子系统。
  - EGL 允许工作空间同时包含这两个子系统。在这种情况下，EGL 使用 EGL 构建路径、import 语句与 SET-MESSAGE-TEXT 函数的 containerContextDependent 属性的组合来解析 MESSAGE-TEXT dataItem 部件定义引用。
- 一个略有不同的示例是，使用名为 ERROR-RECORD 的公共错误记录，该记录包含名为 MESSAGE-TEXT2 的共享数据项。
  - 假定 SubsystemA 和 SubsystemB 具有不同的 MESSAGE-TEXT2 定义。如果子系统需要为不同的屏幕大小构建消息文本，则可能发生这种情况。
  - 当 VisualAge Generator 生成程序时，它将使用当前装入在工作空间中的定义。只要每个子系统总是装入它自己的 MESSAGE-TEXT2 定义，VisualAge Generator 就将使用对该子系统正确的定义。这种技术的缺点与 SET-MESSAGE-TEXT 函数示例的缺点类似。您在进行生成时必须控制工作空间的内容，并且工作空间不能同时包含这两个子系统。
  - EGL 允许工作空间同时包含这两个子系统。在这种情况下，EGL 使用 EGL 构建路径、import 语句与 ERROR-RECORD 的 containerContextDependent 属性的组合来解析 MESSAGE-TEXT2 dataItem 部件定义引用。

## 公共代码和迁移工具

公共代码通常是在许多程序中使用的代码。由于公共代码以下列方式影响迁移工具，所以，您需要将其包括在每个迁移集中：

- 如果公共代码可用，则迁移工具能够解决大部分不确定情况。这将最大程度地减少或者除去您必须手工进行的代码更改。

- 如果公共代码可用，则迁移工具可以为项目正确地设置 EGL 构建路径并为 EGL 文件包括正确的 `import` 语句。这将最大程度地减少您必须手工进行的 EGL 构建路径更改数以及需要添加的 `import` 语句数。
- 迁移工具第一次迁移部件版本时，该工具将把为该部件创建的 EGL 存储到迁移数据库中。原始的“外部源格式”也保留在迁移数据库中。如果另一个迁移集使用同一个部件版本，则迁移工具在为其它迁移集中的新部件创建 EGL 时将使用原始“外部源格式”来进行引用，但不再将该部件转换为 EGL。在为其它迁移集构建 EGL 项目、包和文件时，迁移工具也使用该部件版本的 EGL。这种技术为迁移工具提供了在交叉部件迁移期间解决不确定情况所必需的引用信息，同时通过只将每个部件版本迁移一次来提高性能。

为了确保进行可能的最佳迁移，在迁移子系统时，您始终应该在迁移集中包括“企业公共代码”和“子系统公共代码”。

---

## VisualAge Generator 到 EGL 迁移工具使用的技术

### 技术概述

迁移工具使用许多技术来确定对应的 EGL 语法以及保留 VisualAge Generator 行为。这些技术如下所示：

- 编辑器和构建描述符首选项
- 程序属性
- EGL 构建路径和 `import` 语句
- `containerContextDependent` 属性
- EGL 保留字列表
- 将部件放到 EGL 文件中
- 带程序迁移
- 带关联部件迁移
- 不带关联部件迁移
- 覆盖与合并文件

下列各节对这些技术作了说明。此外，还有一些一般规则控制着迁移工具。

### 编辑器和构建描述符首选项

在启动第 2 阶段迁移之前，您应该对工作空间打开 *VisualAge Generator* 兼容性首选项。EGL *VisualAge Generator* 兼容性首选项提供了对下列 VAGen 行为的支持：

- 在部件名中使用连字符 (-) 以及本地语言字符 @ 和 #
- 基本数据类型 `numc` 和 `pacf`
- 将下标缺省为 1 (表示一维结构字段数组)
- `dataTable` 的 `use` 声明中的 `deleteAfterUse` 属性，这个属性替换了 `keep after use`
- SQL 字段属性 `sqlDataCode`
- `call` 语句选项 `externallyDefined` 和 `noRefresh`，这些选项替换了 `NONCSP` 和 `NOMAPS` 选项
- `transfer` 和 `show` 语句的 `externallyDefined` 选项，这个选项替换了 `DXFR` 和 `XFER` 语句的 `NONCSP` 选项。

- `display printForm` 语句的实现方式与 `print printForm` 语句相同
- 仅当在屏幕上显示未被赋予值的字段时，才使用表单字段的初始值。此首选项并不设置存储器中的字段的初始值。
- **handleSysLibraryErrors** 系统变量，这个系统变量替换了 `EZEREPLY`
- **handleHardDliErrors** 系统变量，这个系统变量替换了 `EZEDLERR`
- **getVAGSysType** 系统函数，这个系统函数提供了 `EZESYS` 的旧 `VAGen` 值
- **connectionService** 系统函数，这个系统函数替换了 `EZECONCT`
- **segmentedMode** 系统变量，这个系统变量替换了 `EZESEGM`
- **sqlIsolationLevel** 系统变量，这个系统变量替换了 `EZESQISL`
- 如果从数据库中检索到的值为空，则未包含在 `SQL` 记录中的主变量根据基本类型的不同将被初始化为空白或零。仅当为字段设置了 `isNullable=yes` 属性时，`SQL` 记录中的主变量才会被初始化。
- 除了 `SQL WHERE` 子句和 `EGL prepare` 语句中的主变量引用以外，对于数字字段（`VAGen PACK` 字段），将偶数精度增加 1。

`VAGen` 迁移工具对它迁移到 `EGL` 构建描述符部件的每个 `VAGen` 生成选项部件自动添加 `vagCompatibility="YES"` 选项。`vagCompatibility` 构建描述符选项指示生成操作提供与 `VisualAge Generator` 兼容性首选项相同的支持。

注：如果您认为您可能想除去对 `VisualAge Generator` 兼容性方式的使用，则在进行迁移之前，请参阅第 155 页的『除去对 `VisualAge Generator` 兼容性方式的使用』一节以了解详细信息。例如，如果需要从部件名中除去连字符、`@` 或 `#`，则您可能想在迁移期间使用“重命名用户出口”。

## 程序属性

迁移工具对每个程序包括 5 个程序属性：

- `includeReferencedFunctions = yes`。迁移工具始终包括这个程序属性以使函数不必嵌套在程序中。这使您能够仅仅在一个独立的项目或包中维护公共函数的一个副本并导入它们，而不是将公共函数包括在每个程序中。当您使用第 1 - 3 阶段迁移时，迁移工具还将针对与程序不在同一个包中的函数来包括任何必需的 `import` 语句。
- `allowUnqualifiedItemReferences = yes`。迁移工具始终包括这个程序属性以便不需要对字段（`VAGen` 数据项）引用进行限定。`EGL` 的未限定字段规则合并了 `VAGen` 规则，因此，未限定字段所解析为的记录、`dataTable`（`VAGen` 表）或表单（`VAGen` 映射）与在 `VisualAge Generator` 中相同。迁移工具不添加限定。
- `localSQLScope = yes`。迁移工具始终包括这个程序属性，即使 `yes` 是缺省值亦如此。迁移工具在创建结果集标识和 `prepare` 语句标识时使用的命名约定并不保证在各个程序之间是唯一的。因此，需要包括 `localSQLScope = yes` 以保留 `VAGen` 行为。
- `throwNrfEofExceptions = yes`。迁移工具始终包括这个程序属性以使 `NRF`（`noRecordFound`）和 `EOF`（`endOfFile`）被视为错误条件。在 `EGL` 中，`NRF` 和 `EOF` 通常不会被视为错误条件。因此，需要包括 `throwNrfEofExceptions = yes` 以保留 `VAGen` 行为。
- `handleHardIOErrors = no`。迁移工具始终包括这个程序属性以使 `VGLib.handleHardIOErrors` 的缺省值设置为 0。`VGLib.handleHardIOErrors` 是 `EZEFEC`

的替换项。VGLib.handleHardIOErrors 的正常 EGL 缺省值是 1。但是，EZEFEFC 的 VAGen 缺省值是 0。因此，需要包括 handleHardIOErrors = no 以保留 VAGen 行为。

## EGL 构建路径和 import 语句

EGL 允许工作空间同时包含一个部件名的多个定义。项目的 EGL 构建路径限制了在查找部件名时要考虑的其他项目。文件中的 import 语句确定了在查找部件名时要考虑 EGL 构建路径中除当前包以外的哪些包以及哪些部件。

在大部分情况下，EGL 构建路径和 import 语句对于解析任何部件引用来说已经足够了。例如，如果在程序的记录声明中将一个记录用作类型定义，则程序的 EGL 构建路径和 import 语句对于解析记录名来说已经足够了。如果只有一个 dataItem 部件定义可以与记录定义、函数本地存储器或函数参数列表配合使用，则 EGL 构建路径和 import 语句对于解析 dataItem 部件引用来说也已经足够了。

例如，您可能正在 SubsystemA 和 SubsystemB 上工作，这两个子系统使用 RECORDX 的两个不同定义。SubsystemA 中的所有程序都需要使用 RECORDX 的 SubsystemA 定义。实现方式可以是：

- SubsystemA 中的项目的 EGL 构建路径属性需要包括负责提供 SubsystemA 的 RECORDX 定义的项目。
- 在 SubsystemA 中，程序的将 RECORDX 用作记录类型声明的文件需要包括一个 import 语句来导入 SubsystemA 中包含 RECORDX 定义的包。

SubsystemA 项目的 EGL 构建路径属性把将要搜索的项目限制为仅包括 SubsystemA 中的项目以及公共项目。SubsystemA 中的文件中的 import 语句限制了 EGL 构建路径中将被考虑的包。即使 RECORDX 使用 dataItem 部件 ITEM1 来作为类型定义并且两个子系统具有不同的 ITEM1 定义，EGL 构建路径和 import 语句对于解析 ITEM1 引用来说也已经足够了。每个子系统中包含 RECORDX 的项目必须指定一个 EGL 构建路径属性，该属性包含一个子系统项目，后者包含 ITEM1 的相应子系统定义。每个子系统中包含 RECORDX 的文件必须具有一个 import 语句，该 import 语句指定一个子系统包，该子系统包包含 ITEM1 的相应子系统定义。

当您使用第 1 – 3 阶段迁移时，迁移工具将根据**迁移集中的部件**来执行下列操作：

- 根据每个项目在其他项目中需要引用的部件来为该项目设置 EGL 构建路径。
- 对于包含某个文件的项目，根据该文件需要引用的部件（此部件位于该项目的 EGL 构建路径内的其他包中），为该文件包括大多数 import package 语句。这些 import 语句基于 VisualAge Generator 在第 1 阶段迁移期间确定的部件关联。
- 迁移工具为数据项部件添加 import 语句。在第 2 阶段迁移期间，迁移工具确定数据项部件是否具有编辑例程。如果被指定为编辑例程的表或函数包括在迁移集中，则迁移工具更新迁移数据库以包括作为数据项关联项被指定为编辑例程的部件。
- 对于下列情况，由于它们在 VisualAge Generator 中不是关联项，所以迁移工具不为它们添加 import 语句：
  - 对于使用 CALL、DXFR 或 XFER 语句来转移至程序的函数。如果正在为 Java 生成，则在包含该函数的文件中，必须为包含该程序的包添加 import 语句，或者用包名对程序名进行完全限定。此外，可以在链接选项部件中使用一个条目来指定程序所在的包的名称。

- 对于 `.eglbl` 文件中的构建部件。VAGen 控制部件（如生成选项部件）不列示它们的关联部件，因此，因此，没有信息可供迁移工具使用。此外，EGL 处理构建描述符部件的方式导致您可能需要对 `nextBuildDescriptor` 值进行重新排序（VAGen/OPTIONS）。此项重新排序操作反过来又要求对迁移工具可能已完成的任何导入进行修改。

**注：**第 1 阶段迁移工具将对迁移集中的部件进行分析以确定每个部件的关联项。为了确保只分析迁移集的部件，第 1 阶段迁移工具将首先从工作空间中删除任何 Java 项目，然后再装入由高级 PLP 项目指定的迁移集。同样，第 1 阶段迁移工具将首先删除任何 Smalltalk 配置映射，然后再装入由高级配置映射指定的迁移集。由于关联项分析限制为仅对迁移集进行，因此，迁移工具不需要设置 EGL 构建路径属性以指定未包括在迁移集中的 EGL 项目。此外，迁移工具不会为未包括在迁移集中的 EGL 包包括 `import` 语句。

当使用单文件迁移时，迁移工具的工作如下所示：

- 由于始终输出到单个现有项目中，所以不设置 EGL 构建路径。
- 由于将把创建的所有文件放到同一个包中，所以不包括 `import package` 语句。因此，不需要 `import` 语句。

## containerContextDependent 属性

**注：**本节描述在 EGL V6.0 中只是部分实现的功能。如果对函数指定了 `containerContextDependent`，则该函数内的函数调用解析是在生成时进行的（而不是在开发时进行的），并且包括对使用调用函数的程序或 `pageHandler` 的名称空间的引用。目前，`containerContextDependent` 不影响记录或 `dataItem` 部件的名称解析。

下列描述反映的是最终实现的意图。

如第 28 页的『EGL 构建路径和 `import` 语句』所述，EGL 构建路径和 `import` 语句对于提供所需的部件名解析来说通常已足够了。但是，每当您保存文件时，EGL 期望解析所有部件名引用。如果 EGL 无法解析部件名，它会将一条错误消息添加到“问题”视图中。根据体系结构的不同，还可能需要对记录或函数使用 `containerContextDependent` 属性。

考虑 RECORDX 被用作 FUNCTIONY 中函数参数的类型定义的情况。假定 RECORDX 与 FUNCTIONY 位于不同的项目和包中，EGL 期望下列情况：

- 对于包含 FUNCTIONY 的项目的 EGL 构建路径，包括的项目必须包含 RECORDX 的定义
- 包含 FUNCTIONY 的文件必须包括一个 `import` 语句以导入包含 RECORDX 的包。

如果所有子系统具有相同的 RECORDX 定义，则 EGL 构建路径和 `import` 语句已足够了，并且，每当您保存包含 FUNCTIONY 的文件时，EGL 都能够解析 RECORDX 的部件引用。

但是，考虑 SubsystemA 和 SubsystemB 都使用 FUNCTIONY 但具有不同的 RECORDX 定义的情况。在这种情况下，EGL 构建路径和 `import` 语句无法同时指向这两个子系统。EGL 对函数支持 `containerContextDependent` 属性。在这种情况下，可以对 FUNCTIONY 指定 `containerContextDependent=yes`。这指定在程序内使用 FUNCTIONY 之前，不解析函数参数和本地存储器的部件名引用。当测试或生成使用 FUNCTIONY 的程序时，包含该程序的项目的 EGL 构建路径以及包含该程序的文件的 `import` 语句确定了在什么

位置查找 RECORDX 的定义。通过使用 `containerContextDependent=yes`，您就能够获得 VisualAge Generator 为函数提供的灵活性。子系统中每个项目的 EGL 构建路径以及子系统中包含程序的任何文件的 `import` 语句都指向该子系统的 RECORDX 定义。

记录部件也支持 `containerContextDependent` 属性。例如，SubsystemA 和 SubsystemB 都可以使用相同的 RECORDZ 定义。但是，RECORDZ 使用的类型定义引用名为 ITEM1 的 dataItem 部件。子系统使用不同的 ITEM1 定义。在这种情况下，可以对 RECORDZ 指定 `containerContextDependent=yes`，以便在程序中使用 RECORDZ 之前 EGL 验证不尝试解析 ITEM1。包含该程序的子项目的 EGL 构建路径以及包含该程序的文件的 `import` 语句确定了在什么位置查找 ITEM1 的定义。

迁移工具不会尝试为您设置 `containerContextDependent` 属性。这是因为迁移工具不要求同时迁移所有子系统，并且不会对所有部件的所有定义执行全面的分析以确定何时存在重复的部件定义。如果您确定在测试和生成时（如在 VisualAge Generator 中）而不是在定义时（如在 EGL 中）需要解析重复的部件名，则可以根据需要添加 `containerContextDependent` 属性。

## EGL 保留字列表

EGL 具有一个保留字列表。部件不能与任何 EGL 保留字同名。另外，EGL 不允许使用符号 # 作为 EGL 部件名的第一个字符。如果 VAGen 部件与 EGL 保留字同名，或者如果 VAGen 部件以符号 # 开头，则迁移工具根据部件类型执行下列操作：

- 由于非 VAGen 程序或运行时环境（例如，CICS PROGRAM 定义）会频繁地引用程序、映射组或表，所以迁移工具不会将这些部件重命名。
- 迁移工具通过在部件名前面添加“重命名前缀”来将数据项、记录、映射和函数重命名。“重命名前缀”是其中一个“VAGen 迁移语法首选项”，可以对第 2 阶段或单文件方式迁移指定此首选项。
- 除执行下列操作以外，迁移工具不对控制部件进行重命名：
  - 迁移工具从绑定控制部件名末尾除去 .BND 后缀。
  - 迁移工具从链接编辑部件名末尾除去 .LKG 后缀。
  - 迁移工具将控制部件名中的任何其它点更改为下划线。此工具还将 /OPTIONS、/RESOURCE 和 /LINKEDIT 生成选项中引用的控制部件名中的点更改为下划线。

第 1 阶段迁移工具能够提供与 EGL 保留字列表有冲突的程序名、映射组名、表名和控制部件名的列表。如果在进行迁移前未将这些部件重命名，则第 2 阶段迁移工具（或单文件方式）还将发出错误消息。在“问题”视图中会出现错误。您可以在 EGL 中通过将程序、formGroup 或 dataTable 重命名并使用 EGL *alias* 属性（可选）来更正问题。

## 将部件放到 EGL 文件中

当您使用第 1 – 3 阶段迁移来进行迁移时，每个 Java 包或 Smalltalk 应用程序都将根据第 1 阶段重命名规则迁移到对应的 EGL 包。原始 Java 包或 Smalltalk 应用程序中的 VAGen 部件将根据下列各项而放到相应 EGL 包中的一个或多个 EGL 文件中：

- 部件的类型：
  - 可生成部件 – 程序、表或映射组
  - 控制部件 – 生成选项、资源关联、链接表、链接编辑或绑定控制
  - 其它可迁移的部件 – 数据项、记录、映射或函数

- 不可迁移的部件 - 用户界面记录、DL/I 记录和 PSB 不会被迁移到任何文件。
- 一个第 1 阶段首选项，该首选项使您能够标识包含公共部件的 Java 项目或包的名称。同样，有一个用于 Smalltalk 的第 1 阶段首选项，该首选项使您能够标识包含公共部件的配置映射或应用程序的名称。
- 该部件是否被某个其它部件使用。第 1 阶段迁移工具根据下列各项来确定某个部件是否被使用：
  - 如果一个部件在**迁移集中**任何可生成部件的 VAGen 关联项列表中出现，则该部件“被使用”。
  - 如果一个部件位于第 1 阶段首选项所指定的公共 Java 项目或包中，或者位于第 1 阶段首选项所指定的公共 Smalltalk 配置映射或应用程序中，则该部件“被使用”。

第 1 阶段迁移工具确定了所有部件的放置位置。第 1 阶段迁移工具将单个 Java 包或 Smalltalk 应用程序中的 VAGen 部件放到相应 EGL 包中的 EGL 文件中，如下所示：

- 所有控制部件都放在名为 `eglPackageName.egl` 的单个文件中，其中 `eglPackageName` 是相应 EGL 包的名称。
- 每个程序部件都放到名为 `programName.egl` 的文件中，其中 `programName` 是程序名。
- 每个表部件都放到名为 `tableName.egl` 的文件中，其中 `tableName` 是表名。
- 每个映射组以及该映射组中的所有映射都放到名为 `mapGroupName.egl` 的文件中，其中 `mapGroupName` 是该映射组的名称。如果没有映射组部件，则第 1 阶段迁移工具将创建哑元映射组部件。由于映射组以及该映射组中的所有映射都必须放在同一个文件中，所以，这些部件必须被视为一个组。如果部件最初不在同一个 Java 包或 Smalltalk 应用程序中，这就可能会导致某些部件被移到另一个 EGL 包或项目中。迁移工具按照以下方式来确定 `mapGroupName.egl` 文件的放置位置：
  - 如果映射组及其所有映射**都在同一个 Java 包中**，则 `mapGroupName.egl` 文件将放在相应的 EGL 包中。同样，如果映射组及其所有映射都位于同一个 Smalltalk 应用程序中，则 `mapGroupName.egl` 文件将放到与该 Smalltalk 应用程序相对应的 EGL 包中。在这种情况下，迁移工具按照处理程序和表文件的方式来处理 `mapGroupName.egl` 文件。这也是最常见的情况。
  - 如果映射组及其映射分布在**项目中的数个 Java 包中**，则将使用项目名以及后缀来为要包含 `mapGroupName.egl` 文件的新的 EGL 包创建名称。这个新的 EGL 包将放到原始项目中。同样，如果映射组及其映射分布在配置映射中的数个 Smalltalk 应用程序中，则将使用配置映射名以及后缀来为要包含 `mapGroupName.egl` 文件的新的 EGL 包创建名称。对于 Java 和 Smalltalk 来说，您都可以通过一个第 1 阶段首选项来控制后缀。
  - 如果映射组及其映射分布在**数个 Java 项目中**，则将使用迁移集名以及后缀来为包含 `mapGroupName.egl` 文件的新的 EGL 项目创建名称。同样，如果映射组及其映射分布在若干个 Smalltalk 配置映射中，则将使用迁移集名以及后缀来为包含 `mapGroupName.egl` 文件的新的 EGL 项目创建名称。对于 Java 和 Smalltalk 来说，您都可以通过一个第 1 阶段首选项来控制后缀。
- 其余所有部件都是按照以下方式放置的：
  - 如果部件**仅被迁移集中的一个程序使用**，则该部件是按如下方式放置的：

- 如果该部件与程序位于同一个包中，则该部件将与程序放在同一个文件中。例如，对于程序的主函数（ProgramA-MAIN）来说，如果该函数未在任何其它程序中使用，则它将与程序（ProgramA）放在同一个文件中。该文件将与程序同名 - ProgramA.egl。
- 如果部件与使用它的程序不在同一个包中，则该部件是按如下方式放置的：
  - 如果该部件位于公共项目或包中，则该部件将放到该部件的原始包中名为 commonParts.egl 的文件中。您通过一个第 1 阶段首选项来控制 commonParts 的名称。
  - 即使该部件并非位于公共项目或包中，该部件也将放到该部件的原始包中名为 commonParts.egl 的文件中。
- 如果该部件被迁移集中的数个程序使用，则该部件将放到它的原始包中名为 commonParts.egl 的文件中。例如，如果 ProgramA 调用 ProgramB 并传递 RecordR，则 RecordR 将放到与包含 RecordR 的原始 Java 包或 Smalltalk 应用程序相对应的 EGL 包中名为 commonParts.egl 的文件中。
- 如果部件未被迁移集中的任何程序使用，则该部件按如下方式放到文件中：
  - 如果部件位于公共 Java 项目或包中，则该部件将放到与包含该部件的原始 Java 包相对应的 EGL 包中名为 commonParts.egl 的文件中。同样，如果部件位于公共 Smalltalk 配置映射或应用程序中，则该部件将放到与包含该部件的原始 Smalltalk 应用程序相对应的 EGL 包中名为 commonParts.egl 的文件中。
  - 如果部件并非位于公共 Java 项目或包中，则该部件将放到与包含该部件的原始 Java 包相对应的 EGL 包中名为 unusedParts.egl 的文件中。同样，如果部件并非位于公共 Smalltalk 配置映射或应用程序中，则该部件将放到与包含该部件的原始 Smalltalk 应用程序相对应的 EGL 包中名为 unusedParts.egl 的文件中。您通过一个第 1 阶段首选项来控制 unusedParts 的名称。
- 请注意下列特殊事项：
  - 任何在映射中被用作编辑例程的函数都会对该函数是否被使用产生影响。但是，迁移工具始终将该函数与程序放在一起或者放到 commonParts 文件中。迁移工具决不会将该编辑函数放到为该映射组创建的文件中。
  - 任何在表中被使用的共享项都会对该数据项是否被使用产生影响。但是，迁移工具始终将该数据项与程序放在一起或者放到 commonParts 文件中。迁移工具决不会将该数据项放到为该表创建的文件中。

**注：**如果迁移多个迁移集或迁移集版本，而未清除迁移数据库，则第 1 阶段处理的第一个包含部件版本的迁移集版本控制着 EGL 部件的项目名、包名和文件名。为了确保根据每个迁移集版本的定义来放置部件，您应该在各版本之间清除迁移数据库。

用于 Java 和 Smalltalk 的第 1 阶段迁移工具是作为样本代码提供的。您可以对第 1 阶段迁移工具进行修改以便根据自己的库管理原理来以不同的方式放置部件。例如：

- 如果 ProgramX 调用 ProgramY 并传递记录 ProgramY-Parm 和 Common-Parm，您可能想将 ProgramY-Parm 放到 ProgramY 所在的文件中，并将 Common-Parm 放在 commonParts 文件中。在了解命名约定的情况下，您可以修改第 1 阶段迁移工具以更改文件放置算法。
- 对于大型的包，您可能想按部件类型将部件分割成独立的文件。
- 如果同一个部件版本在多个迁移集版本中出现，但应该根据迁移集版本的不同而放到不同的 EGL 项目、包或文件中，则每当您处理迁移集版本时，您可能想针对每个

部件的新的 EGL 项目、包和文件名来更新迁移数据库。如果进行此项更改，则务必全面地处理每个迁移集版本（完成第 1 - 3 阶段），然后才开始迁移下一个迁移集版本。

## 带程序迁移

通常，在进行迁移时，您指定一个迁移集，该迁移集标识了应该成组迁移的所有 Java 项目或 Smalltalk 配置映射。通过使用迁移集，迁移工具首先迁移程序及其关联项。这使得此工具能够使用特定程序的上下文来帮助处理 EGL 语言比 VisualAge Generator 更精确或限制性更强的情况。第一个要迁移的程序与其关联项一起确定了该程序或其关联项中的任何不确定情况的 EGL 语法。另一个程序可能会对共享数据项、公共记录、映射、表或函数中的同一种不确定情况采用另一种处理方式。由于部件版本只被迁移一次，所以，第一个使用公共部件的程序控制着其关联项的任何不确定情况的处理方式。

考虑这样一个示例：ProgramA 是主事务程序，它使用显示器映射，ProgramB 是主批处理程序，它使用打印机映射。这两个程序共享一些公共函数，那些函数显示 HEADER 和 DETAIL 映射。这些公共函数还使用 SET map PAGE 语句来清除屏幕或者强制换页。在这种情况下，如果 ProgramA 首先迁移，则迁移工具将把那些函数的 EGL 源代码创建为使用 display 语句和 clearScreen 系统库函数。如果 ProgramB 首先迁移，则迁移工具将把 EGL 源代码创建为使用 print 语句和 pageEject 系统库函数。

每当迁移程序及其关联项时，第一个使用共享数据项、公共记录、映射、表或函数的程序控制着产生的 EGL 代码。在大多数情况下，由于程序以相同的方式使用公共代码，所以，这种技术能够提供最合适 VAGen 源代码迁移。但是，正如您从本示例中可以看到的那样，您希望公共代码实现的特定目标在得到的 EGL 源代码中可能并未反映出来。在本示例中，无论哪个程序首先迁移，您都无法测试或生成所迁移的第二个程序。在 VisualAge Generator 兼容性方式下，可以使用 display 语句来解决 I/O 语句问题。但是，通过选择 clearScreen 或 pageEject 来解决此问题可能要求添加一个新变量 TEXT-OR-PRINT，每个程序都初始化此变量，并且，公共函数对此变量进行测试以确定是执行 clearScreen 系统库函数还是执行 pageEject 系统库函数。

## 带关联部件迁移

如果程序及其关联项不可用，则迁移工具将使用迁移集中的所有部件（或者外部源格式文件中的所有部件，如果正在以单文件方式进行迁移的话）。在这种情况下，如果交叉部件迁移所需的其它部件可用，则迁移工具基本上可以确定它的选择是正确的。

考虑映射变量字段指定了编辑例程的示例。如果在同一个迁移集（或者外部源格式文件）中存在与编辑例程同名的 VAGen 表，则迁移工具假定这就是始终要使用的表并且将其迁移为 validatorDataTable 属性。如果存在与编辑例程同名的函数，则迁移工具将该函数迁移为 validatorFunction 属性。在这两种情况下，由于存在与编辑例程同名的部件，所以迁移工具作出正确选择的可能性较高。如果没有与编辑例程同名的表或函数，则迁移工具将按照不带关联部件迁移方式来处理映射变量字段。

在许多情况下，带关联部件迁移所执行的迁移操作与您在迁移程序及其关联项时执行的迁移操作非常类似。不带程序迁移的缺点是，您可以根据所迁移的特定语句以及迁移集中包括的其它部件来快速地从带关联部件迁移切换到不带关联部件迁移，即使在单个函数中亦如此。

## 不带关联部件迁移

有时，即使程序可用，它的关联项也不是都包括在迁移集中。或者，您可能正在迁移某些过去被子系统使用但当前未被使用的公共部件。在这种情况下，所迁移的部件的关联项可能不可用。迁移工具仍使用下列其中一种技术来转换部件：

- **EGL 语法灵活性。**例如，DISPLAY I/O 选项是在不带关联映射的情况下迁移的。在这种情况下，迁移工具选择使用 `display` 语句并将一条警告消息包括在迁移日志中。即使迁移工具的猜测不正确，但是，由于您使用 VisualAge Generator 兼容性方式，所以，即使表单是打印表单，`display` 语句也能够被接受。
- **智能猜测。**例如，映射变量字段指定了一个编辑例程，但在迁移集（或外部源格式文件）中没有与该编辑例程同名的部件。在这种情况下，迁移工具使用其它信息。此工具将检查下列各项以尝试确定是使用 `validatorDataTable` 属性还是使用 `validatorFunction` 属性：
  - 编辑例程名的长度，这是因为，当长度为 8 个或更多个字符时，表示该编辑例程是函数。
  - 编辑例程名是否是 EZEC10 或 EZEC11，这表示该编辑例程是函数。
  - 是否还指定了编辑消息，这是因为该消息只能与编辑表或者 EZEC10 或 EZEC11 配合使用。

以上各项中的任何一项都使迁移工具能够进行智能选择以确定是设置 `validatorDataTable` 属性还是设置 `validatorFunction` 属性。如果无法作出最终的选择，则迁移工具使用 `validatorFunction` 属性并在迁移日志中包括一条错误消息。如果迁移工具所作的猜测不正确，则在“问题”视图中还应该会出现一个错误。

- **故意无效的语法。**例如，SET map PAGE 是在不带关联映射的情况下迁移的。在这种情况下，迁移工具可以在使用 `EGL converseLib.clearScreen` 函数（用于文本表单）与使用 `EGL converseLib.pageEject` 函数（用于打印表单）之间进行选择。但是，这两个选项具有相同的可能性。因此，迁移工具故意创建无效的语法并将 SET map PAGE 转换为 `converseLib.EZE_SETPAGE`。这将导致“问题”视图中出现一个错误，您必须更正问题。
- **在由于缺少关联项而没有信息的情况下进行直接转换。**（缺少的关联项会引起迁移工具无法检测到的问题。）例如，RecordA 指定它正在重新定义 RecordB 的存储器。在 VisualAge Generator 中，重新定义信息存储在 RecordA 的记录定义中。在生成时，RecordA 和 RecordB 都必须可用，并且在程序中为 RecordA 完成重新定义。在 EGL 中，重新定义信息仅存储在程序中。如果迁移程序时 RecordA 不可用，迁移工具就无法检测出 RecordA 需要在程序中包括 `redefines` 属性。在没有 `redefines` 属性的情况下，EGL 的测试和生成操作将 RecordA 和 RecordB 都视为独立的数据区。**程序的运行方式与在 VisualAge Generator 中的运行方式不同** – 可能无法正确地初始化数据，并且可能会发生异常终止。这就是我们强烈鼓励您生成并测试已迁移的程序的原因。

## 覆盖与合并文件

第 2 阶段和第 3 阶段迁移向导提供了数个相关的首选选项来控制对同一个迁移集的多个版本进行的处理。这些首选选项如下所示：

- 迁移余下的 VAGen 部件。
- 导入到工作空间中 – 覆盖或不覆盖现有文件。
- 将迁移的文件保存至临时目录。

迁移余下的 VAGen 部件控制着迁移工具是否将迁移集中的所有部件都转换到 EGL。

- 如果未选择迁移余下的 VAGen 部件，则只把可生成部件及其关联项转换到 EGL 并存储在迁移数据库中。除非数据项、记录和函数是一个或多个可生成部件的关联项，否则不转换它们。不转换控制部件。如果正在单个迁移集中迁移一个子系统项目和一个公共项目，则取消选择迁移余下的 VAGen 部件可能很有用。在这种情况下，将发生下列情况：
  - 对于子系统项目，只转换在该子系统中实际使用的部件。
  - 对于公共项目，将转换任何可生成部件及其关联项。此外，还转换该子系统所使用的任何数据项、记录和函数。不会将其它可能被其它子系统使用但未被当前子系统使用的数据项、记录和函数转换到 EGL。

取消选择迁移余下的 VAGen 部件有两个优点：

- 对于子系统项目，由于迁移工具仅转换实际使用的部件，因此您有机会对代码进行清理。
  - 对于公共部件，可以将部件转换操作推迟到它们被另一子系统实际使用时才进行。当包括另一个子系统的迁移集中的公共部件时，将把此子系统使用的任何其它部件都转换到 EGL 并存储在迁移数据库中。如果公共部件在各个子系统具有关联项或者包含的部件是各个子系统可生成部件的关联项，这样做就尤其有用。通过将公共部件的迁移操作推迟到子系统使用该部件时才进行，就能够以“带关联项”方式迁移那些公共部件。当迁移下一个包含该公共部件的迁移集时，您对覆盖现有文件所作的选择控制着对最初迁移的公共部件文件执行的操作。
- 如果选择了迁移余下的 VAGen 部件，则把可生成部件及其关联项转换到 EGL 并存储在迁移数据库中。然后，将把任何不是可生成部件关联项的数据项、记录和函数转换到 EGL。并且，还将把所有控制部件都转换到 EGL。选择迁移余下的 VAGen 部件有两个优点：
    - 对于子系统项目来说，将把所有部件都转换到 EGL，而无论它们是否被使用。如果必须保留代码以记录历史，这样做就很有帮助。
    - 对于公共项目来说，如果您知道公共项目中的部件在您计划将来迁移的子系统项目中没有关联项，则选择迁移余下的 VAGen 部件就特别有用。您可以一次性地转换所有公共部件并将 EGL 存储在迁移数据库中。于是，如果其它子系统的迁移集包括了该公共项目，EGL 就已经转换完毕了，并且可以与新的子系统一起被导入到工作空间中或者保存到临时目录中。

如果对第一个迁移集版本选择了迁移余下的 VAGen 部件，则对其它迁移集版本应该继续选择迁移余下的 VAGen 部件。您还应该指定覆盖现有文件。通过同时指定这两个选项，您确保迁移集中的所有部件都包括在 EGL 文件中。

导入到工作空间中控制着迁移工具是否在工作空间中构建 EGL 项目、包和文件。如果您选择了导入到工作空间中，则还可以选择一些其它选项。

- 如果您正在迁移迁移集的多个版本，则可以选择在迁移结束时将哪个版本导入到工作空间中。您可以选择最新版本或最旧版本。选择最新版本的优点是：这就是您最有可能要生成以便进行其它测试的版本。选择最旧版本的优点是，这使您首先将与最旧版本相对应的 EGL 项目、包和文件存储到源代码资源库中。
- 您可以指定要如何处理当前迁移所创建的 EGL 文件在工作空间已存在的情况。
  - 如果选择覆盖现有文件，则 EGL 文件将被一个新文件替换，这个新文件只包含当前迁移集中的部件。如果已经为先前迁移集转换了某些 VAGen 部件版本，则迁移

工具不会再次转换那些 VAGen 部件版本。但是，对于当前迁移集中包括的部件，迁移工具将把那些部件版本的 EGL 包括在文件中。如果您决定更改 VAGen 迁移语法首选项或者修改“重命名用户出口”首选项，则请选择覆盖现有文件。在这种情况下，您可以将数据库复原到第 1 阶段末尾并接着使用新的首选项来再次运行第 2 阶段和第 3 阶段。指定覆盖现有文件使您能够在不必首先清除 EGL 工作空间的情况下运行第 2 阶段和第 3 阶段。在指定了迁移余下的 VAGen 部件的情况下选择覆盖现有文件也是很有益的。在这种情况下，如果您迁移另一个版本的迁移集，则 EGL 文件将被新文件替换，这些新文件包含当前迁移集版本所包括的部件版本。

- 如果未选择覆盖现有文件，则现有 EGL 文件将被修改为包含当前迁移集中的任何其它部件。**已包含在 EGL 文件中的部件不会被更改，即使当前迁移集使用另一个部件版本亦如此。**仅当您未指定迁移余下的 VAGen 部件，并且只是正在迁移公共项目的一个版本但却是对若干个不同的子系统进行迁移时，取消选择覆盖现有文件才是有用的。在这种情况下，您可以象是在迁移不同子系统那样分阶段为公共项目逐渐地构建 EGL 文件。最初，只有第一个子系统使用的公共部件才包括在 EGL 文件中。在迁移第二个子系统时，迁移工具将把第二个子系统所需的任何其它部件添加到 EGL 文件中。迁移工具将原始文件与其它部件合并到一起，以使属于该部件类型的部件仍然是按字母顺序组织的。

在迁移迁移集的多个版本时，您可以选择导入到工作空间中。但是，更好的技术是取消选择导入到工作空间中并改为选择将迁移的文件保存至临时目录。通过使用临时目录，迁移工具就可以创建所有迁移集版本。

将迁移的文件保存至临时目录使您能够迁移迁移集的多个版本并将所有版本存储在工作空间外部。此选项要求使用 EGL 开发环境的多个并发实例。因此，由于需要大量的内存资源，所以只建议在批处理方式下使用此选项。当您选择将迁移的文件保存至临时目录时，还必须指定一个高层目录。迁移工具将在这个高层目录中为每个迁移集版本创建一个子目录。当同时指定了迁移余下的 VAGen 部件时，将迁移的文件保存至临时目录的作用尤其明显。在这种情况下，每个与迁移集版本相对应的子目录都包含该迁移集版本中包括的 VAGen 项目版本的所有部件。

## 一般规则

有一些一般规则控制着迁移工具在迁移源代码时的行为。这些规则如下所示：

- 必须将为了支持 EZE 字或其它语句而添加的任何新变量添加到程序中。不能安全地将它们添加为函数中的本地存储器。这是因为，如果堆栈当前包含的任何函数具有本地存储器、参数或返回值，就不支持分段转换。由于函数上下文是未知的，所以，没有办法确定函数是否将位于导致分段转换的函数堆栈中。因此，将把任何新变量添加到程序中，而不是将它们添加为函数本地存储器。请参阅第 62 页的『迁移所需的中间变量』以了解详细信息。
- 您可能有已迁移至 VisualAge Generator 4.5 但在 VisualAge Generator 4.5 中从未进行修改的来自跨系统产品或旧发行版 VisualAge Generator 的部件。在某些情况下，在外部源格式文件中缺少信息，或者信息是以 EGL 所不支持的方式指定的。迁移工具将尝试提供缺少的信息或更正该信息。这包括：
  - 对于主事务来说，如果缺少 VAGen 分段信息，则迁移工具使 EGL segmented 属性的缺省值为 *no*。
  - 对于 SQL 记录来说，如果缺少 SQL 数据代码，则迁移工具将该项转换为固定长度项。

- 对于 SQL 函数来说，迁移工具尝试根据被指定为 I/O 对象的记录来创建任何缺少的必需 SQL 子句。
- 在函数中，迁移工具将所有语句都转换为某些内容。这将保留 IF / ELSE / END 和 WHILE / END 逻辑。但是，得到的语句在语法上可能是不正确的。例如：
  - 如果某个 EZESYS 值在 EGL 中不受支持，迁移工具就会使用 VAGen 值。如果值不受支持，在“问题”视图中就会有一条消息。
  - 尽管 EGL 当前不支持 DL/I，但是 EZE DL/I 状态字 (EZEDL\*) 仍将被迁移到 EGL 系统变量。对于可能的系统变量名，EGL 系统变量是“最佳的猜测”。迁移工具将发出错误消息。在“问题”视图中不会出现错误。如果您尝试生成使用函数的程序，则生成操作将失败。
  - 尽管 CONVERSE I/O 选项可能是与用户界面记录配合使用的，但是，该选项始终被迁移到 EGL *converse* 语句。迁移工具不发出警告消息。如果该 CONVERSE 是用于用户界面记录的，在“问题”视图中就会有一个错误，这是因为用户界面记录不能被迁移。
  - 尽管 XFER 语句可能是与用户界面记录配合使用的，但是，该语句始终被迁移到 EGL 语句。迁移工具将发出错误消息。在“问题”视图中也会出现错误。
  - EZESCRPT 特殊功能字将被注释掉。这些特殊功能字没有对应的 EGL 替换项。由于无法在 IF、WHILE 或 TEST 语句中使用 EZESCRPT，所以函数的逻辑结构将被保留下来。迁移工具将发出错误消息。在“问题”视图中不会出现错误。
- 当在没有程序的情况下尝试解析部件引用时，迁移工具将检查迁移集中的部件。因此，将迁移集定义成包括有理由一起使用的项目组来说非常重要。例如：
  - 不要包括若干子系统中具有冲突部件名的项目。
  - 在迁移子系统时，包括公共 Java 项目或公共 Smalltalk 应用程序。
- 在迁移期间，迁移工具不执行某些操作：
  - 对于下列情况，由于它们在 VisualAge Generator 中不是关联项，所以迁移工具不为它们添加 import 语句：
    - 对于使用 CALL、DXFR 或 XFER 语句来转移至程序的函数。如果正在为 Java 生成，则在包含该函数的文件中，必须为包含该程序的包添加 import 语句，或者用包名对程序名进行完全限定。此外，可以在链接选项部件中使用一个条目来指定程序所在的包的名称。
    - 对于 .eglblid 文件中的构建部件。VAGen 控制部件（如生成选项部件）不列示它们的关联部件，因此，没有信息可供迁移工具使用。此外，EGL 处理构建描述符部件的方式导致您可能需要对 *nextBuildDescriptor* 值进行重新排序（VAGen /OPTIONS）。此项重新排序操作反过来又要求对迁移工具可能已完成的任何导入进行修改。

**注：**第 1 阶段迁移工具将对迁移集中的部件进行分析以确定每个部件的关联项。为了确保只分析迁移集的部件，第 1 阶段迁移工具将首先从工作空间中删除任何 Java 项目，然后再装入由高级 PLP 项目指定的迁移集。同样，第 1 阶段迁移工具将首先删除任何 Smalltalk 配置映射，然后再装入由高级配置映射指定的迁移集。由于关联项分析被限制为仅对迁移集进行，因此，迁移工具不需要设置 EGL 构建路径属性以指定未包括在迁移集中的 EGL 项目。此外，迁移工具不会为未包括在迁移集中的 EGL 包包括 import 语句。

- EGL 不允许使用隐式项。VisualAge Generator 允许使用隐式项，但不建议使用隐式项。隐式项是在程序中被使用但却未在该程序所使用的记录、表或映射中被显

式定义的项。由于评估每个未限定数据项以确定它是否是隐式项时会对性能产生影响，所以，迁移工具不为隐式项创建定义。您应该在迁移前为隐式项提供定义。为了在进行迁移前解决问题，请在 VisualAge Generator 中验证程序以确定该程序是否确实使用隐式项。如果使用了隐式项，则 VAGen 验证操作将提供一条消息，该消息包含该项的正确定义。如果在迁移前没有为隐式项创建定义，则在“问题”视图中将有一个错误，您可以在 EGL 中更正问题。

- 迁移工具不会尝试设置 `containerContextDependent` 属性。以后，对于需要引用其它由某个子系统提供的部件的特定公共记录或函数，可以添加此属性。请参阅第 29 页的『`containerContextDependent` 属性』一节以了解有关如何对记录和函数使用此属性的更多详细信息。
- 迁移工具假定 VAGen 语法有效并且可以在 VisualAge Generator 中成功地对使用迁移集所包括的部件的程序进行验证。迁移工具不会尝试更正无效的语法。例如：
  - 如果映射数组的元素具有不同的编辑特征或属性，则第一个数组元素的特征确定了迁移到 EGL 的内容。迁移工具不发出消息。
  - 如果记录中共享数据项的长度已更改，导致父项的长度与其子结构中的项长度之和不匹配，迁移工具不会更改任何项长度，并且不发出消息。在“问题”视图将有一个错误，该错误指示子代长度之和与父代长度不匹配。
- 即使效率低下的代码在 EGL 中导致语法错误，迁移工具也不会尝试改进那些代码。例如：
  - 即使同一个记录在程序的“表及其它记录”列表中列示了两次，迁移工具也不会将其除去，并且不会发出消息。在“问题”视图中会出现错误。同样，即使同一个表在程序的“表及其它记录”列表中列示了两次，迁移工具也不会将额外的表除去，并且不会发出消息。在“问题”视图中会出现错误。
  - 即使某个记录在程序中未被使用但却列示在该程序的“表及其它记录”列表中，迁移工具也不会将其除去，并且不会发出消息。在“问题”视图中不会出现错误。同样，即使某个表在程序中未被使用但却列示在该程序的“表及其它记录”列表中，迁移工具也不会将该表除去，并且不会发出消息。在“问题”视图中不会出现错误。
  - 即使工作存储器记录列示在程序的“表及其它记录”列表中但该记录仅包含 Level-77 项，迁移工具也不会将该记录除去，并且不会发出消息。在“问题”视图将出现一个错误，该错误指示找不到该记录。这是因为，现在存在的唯一一个记录包括了 Level77 后缀首选项作为记录名的一部分。
  - 如果 VAGen 程序包括映射组或帮助映射组，则实际的映射组部件不必存在。例如，如果该程序从未显示（DISPLAY）或转换（CONVERSE）映射，或者该程序从未将映射用作被调用参数，则实际映射组部件不必存在。在这种情况下，由于该映射组在 VisualAge Generator 中不是该程序的关联项，所以，迁移工具对该 `formGroup` 包括 `use` 语句，但不会在程序中包括 `import` 语句。迁移工具不发出错误消息。EGL 需要实际的 `formGroup` 部件。如果没有 `formGroup` 部件，或者如果 `formGroup` 部件与程序不在同一个包中，则在“问题”视图将会出现一个错误，该错误指示找不到该程序的 `use` 声明中指定的 `formGroup`。
- 对于 VisualAge Generator 中未归档的工具的使用，即使没有等效的 EGL 支持，迁移工具也不必进行检测，也不会提供警告消息。例如：
  - VAGen `EZEBYTES` 函数仅被归档为支持项和记录。实际上，此函数还具有未归档的支持映射的功能。EGL `sysLib.bytes` 函数仅支持项和记录。如果对映射使用 `EZEBYTES`，迁移工具不会提供警告消息。在“问题”视图中会出现错误。

- 如果 CALL 语句指定了未限定的项来作为自变量，并且在程序中有此项名的多个定义，则 VAGen 使程序的主工作存储器记录中的 Level-77 项具有优先权。EGL 要求对该项进行限定。迁移工具并不会为您添加限定，并且不会提供警告消息。在“问题”视图中会出现错误。

---

## 自从 EGL 5.1.2 以来 VAGen 迁移工具有哪些新增功能？

下面是“EGL 5.1.2 General Availability”中的新增功能或改进功能。根据 WebSphere Studio 5.1.2 产品维护级别的不同，您可能已经拥有了某些第 1 阶段增强功能。

- 第 1 阶段已进行了如下更改：
  - 已经为迁移数据库更新了 DDL。您必须重新运行 SetupTables.bat 并重新运行第 1 阶段才能将源代码放到迁移数据库中。
  - 已将 EGL 保留字列表外部化并且已将保留字列表版本包括在日志信息中。
  - 已添加了新的 checkStage1.bat 文件以便检查第 1 阶段的结果是否有错误。
- 第 2 阶段和第 3 阶段迁移以及单文件方式迁移现在都包括下列更改：
  - 支持新的 EGL 语法。
  - （可选）调用一个用户出口以使您能够在第 2 阶段迁移期间或单文件迁移期间将部件重命名。例如，您可以编写一个用户出口以将连字符 (-) 更改为下划线 (\_)。
  - 新的 VAGen 语法迁移首选项，这些首选项使您能够为 SQL 记录中的字段指定下列内容：
    - 如果 SQL 列名与项名相同，则迁移工具应该省略该列名。
    - 迁移工具始终应该省略 *isNullable = yes* 属性。
    - 仅当只对 SQL 记录指定了一个 SQL 表，并且 VAGen 只读属性设置为 *yes* 时，迁移工具才应该包括 *isReadOnly = yes* 属性。
  - 新的 VAGen 语法迁移首选项，此首选项使您能够指定迁移工具应该将逗号小数点更改为句点小数点，即使工作站使用的语言环境在缺省情况下使用句点小数点亦如此。
- 第 2 阶段到第 3 阶段迁移现在还执行下列操作：
  - 为指定了 *validatorFunction* 或 *validatorDataTable* 属性的 *dataItem* 部件添加 *import* 语句。
  - 对属于该部件类型的文件中的部件按 EGL 部件名进行排序。

下列 EGL 更改提供了对已迁移的 VAGen 程序的更好支持：

- 现在，可以在字符串并置中使用数字变量。这提供了对已迁移到 EGL *prepare* 语句的 VAGen SQL I/O 执行时语句构建的支持。
- 现在，Java 运行时环境允许您更改 EGL 产品消息，这与 VisualAge Generator 中提供的功能类似。

---

## 自从 EGL 6.0 iFix 以来 VAGen 迁移工具有哪些新增功能？

下面是自 EGL 6.0 iFix 以来的新增功能或改进功能：

- 第 2 阶段和第 3 阶段迁移以及单文件方式迁移现在都包括下列更改：

- 能够更好地支持用户出口，这使得您能够在第 2 阶段迁移期间或单文件迁移期间重命名部件。
- 新增 VAGen 迁移首选项，这使得您能够尽量减少使用 VisualAge Generator 兼容性方式。

---

## 迁移工具的已知限制

### 概要

务必查看 EGL 限制，尤其是有关验证、调试或生成的任何限制。

### 对 Java 和 Smalltalk 执行的第 1 阶段

- 如果迁移多个迁移集或迁移集版本，而未清除迁移数据库，则第一个包含部件版本的迁移集版本控制着 EGL 部件的项目名、包名和文件名。如果下列情况成立，这通常不会导致问题：
  - 如果部件未在 Java 包之间移动，并且不存在包含相同 Java 包名的不同名迁移集。
  - 如果部件未在 Smalltalk 应用程序之间移动，并且没有包含相同 Smalltalk 应用程序的不同名配置映射。

如果您的情况与上面描述的情况不同，则变通方法是迁移一个迁移集版本并彻底完成第 1 阶段至第 3 阶段，接着清除迁移数据库，然后迁移下一个迁移集版本并彻底完成第 1 阶段至第 3 阶段。

### 第 2 阶段和第 3 阶段

- VAGen 迁移向导的限制：
  - 进度窗口上的“取消”按钮不起作用。在启动第 2 阶段或第 3 阶段迁移工具之后，除非使用“任务管理器”，否则无法取消迁移。
- 如果要在迁移数据库与应用程序数据库之间来回切换，则每次进行切换时都必须关闭 EGL 开发环境。
- Linux 环境不支持第 2 阶段和第 3 阶段。

### 语法迁移

- 迁移工具能够正确地转换用作 SQL 记录结构列名的 SQL 关键字。但是，迁移工具无法处理用作记录的 SQL defaultSelectCondition 列名或者函数的 SQL 子句列名的 SQL 关键字。变通方法是修改 SQL defaultSelect 条件或 SQL 子句，如第 177 页的『需要进行特殊处理的 SQL 保留字』所述。本节提供了 SQL defaultSelectCondition 和 SQL I/O 语句所需的语法的 SQL 关键字列表和详细信息。
- 请参阅第 29 页的『containerContextDependent 属性』以了解有关此属性的使用限制的详细信息。

---

## 第 3 章 处理不确定情况

在许多情况下，迁移工具可能无法从 VisualAge Generator 中获得生成相应 EGL 语句所需的所有信息。由于相应的 EGL 语句可能会随着 VisualAge Generator 提供的输入的不同而发生变化或者产生不同的结果，所以，这些情况称为“不确定情况”。在这些不确定情况下，迁移工具可能无法迁移到与 VisualAge Generator 中的预期结果相匹配的 EGL 语句。在许多不确定情况下，生成的 EGL 语句会随着迁移过程的不同而有所变化：

- 是否以带程序及其关联项的方式进行迁移以及以此方式进行迁移时的程序迁移顺序。
- 是否以不带程序但带必需关联部件的方式进行迁移（在这种情况下，仍然能够完成交叉部件迁移）。
- 是否以不带关联项的方式进行迁移（在这种情况下，迁移工具拥有的可用于进行智能选择的信息有限）。

由于带程序及其关联项迁移能够保证信息量最大，所以这是首选的带关联项迁移方法。下列各表说明了下列部件类型的带关联部件迁移与不带关联部件迁移之间的差别：

- 共享的数据项
- 记录
- 表
- 映射组和映射
- 程序
- 函数，包括 I/O 语句
- 其它语句
- EZE 字

这些表还显示了在这些不确定情况下可能会发生的一些潜在问题，并且，对于这些问题，提供了可能的解决方案建议。并没有一种解决方案对于每种情况来说都是最佳的。例如，当有两个具有相同部件名的部件时，将使用频率较低的部件重命名对代码其它部分的影响程度将会较低。

---

### 处理数据项的不确定情况

#### 具有偶数长度的 PACK 数据项

**VisualAge Generator:** PACK 数据项的长度是作为位数指定的，最大长度为 18。偶数长度记录在共享数据项定义中，对于非共享数据项，记录在记录定义中。但是，在大多数编辑器中，以及在测试和生成时，使用的长度是下一个更长的奇数长度，最大值为 18。只有“SQL 记录编辑器”才显示偶数长度。对于被用作 SQL WHERE 子句中的主变量或者指定了“执行时语句构建”的 SQL 语句中的主变量的偶数长度项，测试和生成将创建一个具有偶数长度的临时变量。

**EGL:** 十进制基本类型替换了 VAGen PACK 类型。在 VisualAge Generator 兼容性方式下，EGL 测试和生成提供的支持与 VisualAge Generator 中提供的支持相同。对于具有偶数精度的十进制项，测试和生成将所有记录中的精度增加 1，并在 SQL WHERE 子句或 PREPARE 语句中使用具有偶数精度的临时变量。如果您未指定 VisualAge Generator 兼容性方式，则 EGL 使用对数据项指定的精度。

迁移所需的关联部件：无。

表 9. 具有偶数长度的压缩数据项

带关联部件迁移	不带关联部件迁移
<p>迁移工具根据 VAGen 迁移首选项 <i>不对项或变量使用 evensql=y</i> 来迁移压缩数据项。</p> <p>如果未选择该首选项，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 对于共享数据项定义，使用 VisualAge Generator 中指定的偶数或奇数长度，而不考虑在 SQL 行记录中是否曾经使用该项。</li> <li>• 对于所有记录定义中的非共享项，使用 VisualAge Generator 中指定的偶数或奇数长度，这是因为在 SQL WHERE 子句或 PREPARE 语句中可能已将该项用作主变量。</li> <li>• 对于表、函数参数、函数返回值和函数本地存储器中的非共享项，使用奇数长度（或者，如果该项具有最大长度，则使用 18），这是因为在这些情况下未在 VisualAge Generator 中记录用于确定偶数位数的信息。</li> </ul> <p>如果选择了该首选项，迁移工具将总是对所有项或变量使用奇数长度（或者，如果该项具有最大长度，则使用 18）。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题:</b> 仅当解除 VisualAge Generator 兼容性方式或仅当使用首选项 <i>不对项或变量使用 evensql=y</i> 进行迁移时，才会发生问题。在这种情况下，由于有效位数少于 VisualAge Generator 兼容性方式下的有效位数，所以可能会发生溢出。</p> <p><b>可能的解决方案:</b> 在解除 VisualAge Generator 兼容性方式或选择该首选项之前，执行以下操作：</p> <ul style="list-style-type: none"> <li>• 查看 SQL 表和视图定义以确定是否有任何需要偶数精度的 SQL 列。如果是的话，评估使用与 SQL 列定义不匹配的主变量长度对 SQL 性能的影响。</li> <li>• 查看 EGL 记录中的所有十进制 dataItem 部件定义和基本字段以了解是否有偶数长度项。评估这些项中的任何一项是否可能会发生溢出。</li> </ul>	<p><b>潜在问题:</b> 带关联部件迁移列中列出的潜在问题和可能的解决方案适用。</p>

## 共享编辑和消息

**VisualAge Generator:** 共享 dataItem 部件定义可以为映射和用户界面（UI）记录指定缺省编辑和消息。

**EGL:** dataItem 部件定义只有一组编辑和消息属性。尽管在此发行版的 EGL 中不支持用户界面记录，但是迁移工具还是对数据项合并了映射属性和用户界面属性。这将最大程度地保留数据项信息。

**迁移所需的关联部件:** 无。

表 10. 共享编辑和消息

带关联部件迁移	不带关联部件迁移
<p>迁移工具按如下方式合并映射编辑和用户界面编辑:</p> <ul style="list-style-type: none"> <li>• 对于每个编辑或消息，迁移工具执行下列第一项适用的操作: <ul style="list-style-type: none"> <li>– 如果指定了用户界面编辑，则迁移工具将该用户界面编辑及其相关消息信息转换为对应的 EGL 属性。</li> <li>– 如果仅指定了映射编辑，则迁移工具将该映射编辑及其相关消息转换为对应的 EGL 属性。</li> <li>– 如果指定了用户界面消息，但未指定它的相关用户界面编辑，则迁移工具将该用户界面消息转换为对应的 EGL 属性。</li> <li>– 如果指定了映射消息，但未指定它的相关映射编辑，则迁移工具将该映射消息转换为对应的 EGL 属性。</li> <li>– 如果未指定用户界面编辑和消息信息以及映射编辑和消息信息，则迁移工具不设置对应的 EGL 属性。应用正常的 EGL 缺省值。</li> </ul> </li> <li>• 在 VisualAge Generator 中，仅对映射编辑指定对齐和十六进制编辑，因此始终使用它们来设置对应的 EGL 属性。</li> </ul>	<p>除了后面第 43 页的『共享数据项的映射项编辑例程』描述的情况以外，对于带关联部件迁移和不带关联部件迁移，迁移工具按相同的方式迁移缺省编辑和消息。</p>
<p><b>潜在问题 1:</b> 仅当在 VisualAge Generator 中存在有冲突的映射编辑和用户界面编辑，并且您确实希望映射记录与用户界面记录具有不同的编辑时，才会发生问题。在将该项添加至文本表单或打印表单之前，不会发生问题。</p> <p><b>注:</b> 如果从未使用 VAGen Web 事务，则在 VisualAge Generator 中应该只存在映射编辑，并且应该不会有问题。</p> <p><b>可能的解决方案:</b> 除了对 dataItem 部件定义添加注释以列示 VAGen 映射项编辑和消息以外，不能对 dataItem 部件定义执行其它操作。如果将该项添加至文本表单或打印表单，则可以覆盖使该项对该特定表单有所不同所需的任何属性。</p> <p><b>潜在问题 2:</b> 如果在 EGL 用户界面记录中使用该项，也会发生问题。该项可能具有一些从 VAGen 映射编辑迁移而来的其它编辑或消息。</p> <p><b>解决方案:</b> 始终查看通过用户界面记录中的类型定义进行定义的字段的编辑。</p>	<p>在带关联部件迁移列中提到的潜在问题也适用。您可以使用相同的解决方案。</p>

## 共享数据项的映射项编辑例程

**VisualAge Generator:** 共享数据项定义可以具有映射项编辑例程，后者是表、函数或者 EZEC10 或 EZEC11。仅当编辑例程是 EZEC10、EZEC11 或表时，才使用编辑消息。

**EGL:** 数据项可以既具有 validatorDataTable 又具有 validatorFunction。数据项还可以既具有 validatorDataTableMsgKey 又具有 validatorFunctionMsgKey。

**迁移所需的关联部件:** 表或函数部件。

表 11. 共享数据项的映射项编辑例程

带关联部件迁移	不带关联部件迁移
<p>第一次迁移共享数据项时，迁移工具执行下列各项中的第一个适用项:</p> <ul style="list-style-type: none"> <li>• 如果 editRoutineName 是 EZEC10 或 EZEC11，则迁移工具将 validatorFunction 属性设置为等效的 EGL 系统库函数。迁移工具还将 validatorFunctionMsgKey 设置为编辑消息（如果有的话）。</li> <li>• 如果 editRoutineName 是函数，则迁移工具设置 validatorFunction 属性。由于在 VisualAge Generator 中不使用 validatorFunctionMsgKey，所以迁移工具将省略该属性。迁移工具还将该函数添加为该数据项部件的关联项，以便在第 3 阶段创建 import 语句。</li> <li>• 如果 editRoutineName 是表，则迁移工具设置 validatorDataTable 属性。迁移工具还将 validatorDataTableMsgKey 设置为编辑消息（如果有的话）。迁移工具还将该表添加为该数据项部件的关联项，以便在第 3 阶段创建 import 语句。</li> <li>• 如果未指定编辑例程但指定了编辑消息，则迁移工具将 validatorDataTableMsgKey 设置为该编辑消息。</li> </ul> <p><b>注：</b>即使在程序上下文中进行迁移，如果在映射中未使用该共享项，或者它在映射中的编辑与共享项定义中指定的编辑不同，则 editRoutineName 可能不可用。</p>	<p>如果没有与 editRoutineName 同名的函数或表可用，则迁移工具执行下列各项中的第一个适用项:</p> <ul style="list-style-type: none"> <li>• 如果 editRoutineName 是 EZEC10 或 EZEC11，则迁移工具将 validatorFunction 属性设置为等效的 EGL 系统库函数名。迁移工具还将 validatorFunctionMsgKey 设置为编辑消息（如果有的话）。</li> <li>• 如果 editRoutineName 的长度超过 7 个字符，则它肯定是函数名，因此迁移工具设置 validatorFunction 属性。由于在 VisualAge Generator 中不使用 validatorFunctionMsgKey，所以迁移工具将省略该属性。</li> <li>• 如果指定了编辑消息，则迁移工具设置 validatorDataTable 和 validatorDataTableMsgKey。</li> <li>• 否则，迁移工具设置 validatorFunction 属性并发出错误消息。</li> </ul> <p>如果未指定编辑例程但指定了编辑消息，则迁移工具将 validatorDataTableMsgKey 设置为该编辑消息。</p>
<p><b>潜在问题：</b>仅当函数与 dataTable 同名（很有可能位于不同的子系统中）时，才会发生问题。在这种情况下，一个子系统使用函数，另一个子系统使用 dataTable。在将该项添加至文本表单之前，不会发生问题。</p> <p><b>可能的解决方案：</b>重命名 dataTable，使得不再存在两个同名的部件。对项定义同时指定 validatorFunction 和 validatorDataTable 属性。如果将该项添加至文本表单，则删除该特定项所不需要的 validatorFunction 或 validatorDataTable 属性。<b>缺点：</b>必须将程序和表单修改为使用新的 dataTable 名称。</p>	<p><b>潜在问题 1：</b>如果迁移工具的猜测不正确，则会发生问题。在将该项添加至文本表单之前，不会发生问题。</p> <p><b>可能的解决方案：</b>更正数据项定义。</p> <p><b>潜在问题 3：</b>带关联部件迁移列中列示的问题也会发生。您可以使用相同的解决方案。</p>

## 共享数据项的填充字符

**VisualAge Generator:** 对于字符、混合或 DBCS，映射编辑的缺省填充字符是 NULL；对于数字，是空白；对于十六进制，是 0。对于字符、混合、DBCS、Unicode 和数字，用户界面编辑的缺省填充字符是空白；对于十六进制，是 0。对于用户界面记录，NULL 不是有效的填充字符。可以对映射编辑和用户界面编辑指定另一个填充字符。

**EGL:** dataItem 部件只有一个缺省的 fillCharacter 属性。尽管在此发行版的 EGL 中不支持用户界面记录，但是迁移工具会将数据项的映射 fillCharacter 属性和用户界面 fillCharacter 属性合并。这将最大程度地保留数据项信息。

**迁移所需的关联部件:** 无。

表 12. 共享数据项的填充字符

带关联部件迁移	不带关联部件迁移
<p>第一次迁移共享数据项时，迁移工具执行下列各项中的第一个适用项：</p> <ul style="list-style-type: none"> <li>• 如果用户界面编辑指定了填充字符，则迁移工具将该字符迁移为 EGL fillCharacter 属性。</li> <li>• 如果映射编辑指定了填充字符，则迁移工具将该字符迁移为 EGL fillCharacter 属性。</li> <li>• 如果用户界面编辑和映射编辑都未指定填充字符，则迁移工具不设置 EGL fillCharacter 属性。</li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题：</b> 仅当将使用类型定义的字段添加至 dataItem 部件，并且已使用其中一种类型的编辑将该部件迁移至另一种类型的用户界面（表单或用户界面记录）时，才会发生问题。</p> <p><b>可能的解决方案：</b> 在将新字段添加至表单或用户界面记录时，始终查看填充字符。</p>	<p>带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。</p>

## 处理记录的不确定情况

### 重新定义的记录

**VisualAge Generator:** 重新定义的记录类型为另一个记录提供另一种数据项布局。重新定义的记录指定它正在重新定义的记录的名称。例如，RecordA 是一个 REDEFINED 记录，它重新定义 RecordB。VisualAge Generator 根据 RecordA 在程序中的使用来确定 RecordA 是否真的是 RecordB 的重新定义。如果 RecordA 被用作被调用参数，则不将 RecordA 看作 RecordB 的重新定义。

**EGL:** RecordA 是 basicRecord。仅在程序定义中提供重新定义信息 - 而不在 RecordA 的定义中提供该信息。

**迁移所需的关联部件：**

- 当迁移重新定义的记录时：无。
- 当迁移程序时：重新定义的记录（RecordA）。

表 13. 重新定义的记录

带关联部件迁移	不带关联部件迁移
<p>当迁移重新定义的记录时，迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 将重新定义的记录（RecordA）迁移为 basicRecord。</li> <li>• 在 RecordA 中包括“VAGen 信息”注释，此注释指示 RecordA 重新定义 RecordB。</li> <li>• 发出一条参考消息，指示 RecordA 重新定义 RecordB。</li> </ul>	<p>当迁移重新定义的记录时，迁移工具执行带关联部件迁移列中提到的操作。</p>

表 13. 重新定义的记录 (续)

带关联部件迁移	不带关联部件迁移
<p>当迁移程序时，如果 RecordA 可用，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 如果 RecordA 在 VisualAge Generator 中被看作 RecordB 的重新定义，则迁移工具在 RecordA 的声明中包括 redefines 属性。</li> <li>• 如果 RecordA 在 VisualAge Generator 中未被看作 RecordB 的重新定义，则迁移工具在 RecordA 的声明中不包括 redefines 属性。</li> </ul>	<p>当迁移程序时，如果 RecordA 不可用，则迁移工具不知道 RecordA 是重新定义的记录。迁移工具在 RecordA 的声明中不包括 redefines 属性。</p>
<p><b>新使用的注意事项：</b> 仅当需要在新程序中使用 RecordA 和 RecordB 时，才会发生问题。每当要将 RecordA 看作 RecordB 的重新定义时，您都必须记得包括 RecordA 的 redefines 属性。</p>	<p><b>潜在问题：</b> 如果 VAGen 程序将 RecordA 用作重新定义，就会发生问题。在迁移刚刚完成时，该程序将不是有效的 EGL 程序，这是因为缺少 RecordA 的定义以及 import 语句。在“问题”视图中将会有有一个错误。如果迁移 RecordA 并对程序添加 import 语句，就将把该程序转换为有效的 EGL 程序。但是，将有两个数据区 - 一个用于 RecordA，另一个用于 RecordB。在验证或生成期间，EGL 不会检测到此更改。程序的运行方式将与在 <b>VisualAge Generator</b> 中的运行方式不同。</p> <p><b>解决方案：</b> 如果迁移其它记录或对程序添加 import 语句，则查看记录定义是否包含“VAGen 信息”注释。如果存在“VAGen 信息”注释（该注释指定 RecordA 是 RecordB 的重新定义，则更新该程序以使 RecordA 的声明包括 redefines 属性。</p> <p><b>注意事项：</b> 带关联部件迁移列中列示的新使用注意事项也适用。</p>

## 记录中的 Level-77 项

**VisualAge Generator:** 工作存储器记录可以具有 Level-77 项。

**EGL:** 记录不能具有 Level-77 项。

**迁移所需的关联部件:**

- 当迁移工作存储器记录时：无。
- 当迁移程序时：主工作存储器记录。
- 当迁移函数时，工作存储器记录。

表 14. 记录中的 Level-77 项

带关联部件迁移	不带关联部件迁移
<p>当迁移任何包含 Level-77 项的工作存储器记录时，迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 将包含 Level-77 项的工作存储器记录分割为两个 basicRecord - 一个用于工作存储器结构，另一个用于 Level-77 项。新的 Level-77 记录名是具有客户提供的后缀的原始工作存储器记录名。</li> <li>• 将新的 Level-77 记录放在具有原始工作存储器记录的那个文件中。</li> <li>• 发出一条参考消息，指示正在创建 Level-77 记录。</li> </ul>	<p>当迁移任何包含 Level-77 项的工作存储器记录时，迁移工具执行带关联部件迁移列中提到的操作。</p>

表 14. 记录中的 Level-77 项 (续)

带关联部件迁移	不带关联部件迁移
当迁移程序时，如果主工作存储器记录可用并包含 Level-77 项，则迁移工具为新的 Level-77 记录添加记录声明到程序中。	当迁移程序时，如果主工作存储器记录不可用，则迁移工具不知道主工作存储器记录是否包含 Level-77 项。迁移工具不包括 Level-77 记录的记录声明。
当迁移函数时，如果工作存储器记录可用，则迁移工具将限定引用更改为函数中的 Level-77 项，以使用新的 Level-77 记录名。	当迁移函数时，如果工作存储器记录不可用，则迁移工具不更改项名的限定。
<p><b>潜在问题:</b> 对于 Level-77 项，如果有两个同名的记录（可能位于不同的子系统中），并且该项在一个记录中是 Level-77 项，而在另一个记录中却不是，则会发生问题。</p> <p><b>可能的解决方案:</b> 将该项移至（新的）公共记录并在所有函数中更改项限定。另外，不要在函数中对该项进行限定。</p> <p><b>新使用的注意事项:</b> 如果将原始工作存储器记录指定为新程序的 inputRecord 属性，则有一个潜在问题。务必考虑是否还需要为新的 Level-77 记录添加声明。</p>	<p><b>潜在问题 1:</b> 如果主工作存储器记录包含 Level-77 项，并且程序使用 Level-77 项，则会发生问题。由于缺少项定义，所以程序的生成将失败。</p> <p><b>解决方案:</b> 对程序添加 Level-77 记录。</p> <p><b>潜在问题 2:</b> 对于函数，如果限定的数据项真的是 VAGen Level-77 项，则会发生问题。</p> <p><b>解决方案:</b> 修改函数，以便为该数据项提供正确的限定符。</p> <p><b>潜在问题 3:</b> 带关联部件迁移列中列示的问题也会发生。您可以使用相同的解决方案。</p> <p><b>注意事项:</b> 带关联部件迁移中列示的新使用注意事项也适用。</p>

## 备用指定记录

**VisualAge Generator:** 一个记录可以将另一个记录指定为备用指定 (altspec) 记录。例如，如果 RecordA 指定 RecordB 这个 altspec，则 RecordB 为 RecordA 提供结构。对于 SQL 记录，RecordB 还为 RecordA 提供 SQL 表和键列表。如果 RecordA 指定了键项，则在确定缺省选择条件时，该项将与 RecordB 的键合并。

**EGL:** 一个记录可以嵌入另一个记录以获取记录结构。只包括记录结构。每个 SQL 记录都必须明确地指定它的整个 SQL 表和键集合。

**迁移所需的关联部件:** 如果 RecordA 是 SQL 记录，则需要被指定为 altspec 记录的记录 (RecordB)。

表 15. 备用指定记录

带关联部件迁移	不带关联部件迁移
<p>如果 RecordA 是指定了 RecordB 的备用指定的 SQL 记录并且 RecordB 可用, 则迁移工具对 RecordA 执行下列操作:</p> <ul style="list-style-type: none"> <li>• 根据 RecordB 中指定的表的列表来创建表名的列表。</li> <li>• 通过合并下列各项来创建键的列表:                             <ul style="list-style-type: none"> <li>– RecordB 中指定了 key=yes 的项 (如果有的话)。</li> <li>– RecordA 中被指定为键项的项 (如果有的话)。</li> </ul> </li> </ul> <p>键的顺序就是项在 RecordB 结构中的出现顺序。</p> <ul style="list-style-type: none"> <li>• 包括指向 RecordB 的嵌入的语句。</li> <li>• 将 RecordA 的缺省选择条件中的任何 !itemColumnName 变量迁移为 RecordB 中的相应 SQL 列名。</li> </ul>	<p>如果 RecordA 是指定了 RecordB 的备用指定的 SQL 记录, 并且 RecordB 不可用, 则迁移工具对 RecordA 执行下列操作:</p> <ul style="list-style-type: none"> <li>• 将 tableNames 属性设置为 ###TABLES_NOT_FOUND###</li> <li>• 将 keyItems 属性设置为 ###KEYS_NOT_FOUND###, 后跟在 RecordA 中被指定为键项的项 (如果有的话)。</li> <li>• 包括指向 RecordB 的嵌入的语句。</li> <li>• 将 RecordA 的缺省选择条件中的任何 !itemColumnName 变量迁移为 !itemColumnName, 并且不进行任何替换。</li> <li>• 发出错误消息, 指示未能确定表和键。</li> <li>• 如果有任何 !itemColumnName 变量, 则发出一条错误消息。</li> </ul>
<p><b>潜在问题:</b> 对于 SQL 记录, 仅当 RecordB 的定义不同时 (通常是在不同的子系统中), 才会发生问题。对于非 SQL 记录, 没有任何问题。</p> <p><b>解决方案:</b> 复制 RecordA 的定义, 以使每个子系统都具有自己的 RecordA 定义。另外, 对 RecordA 指定 <code>containerContextDependent=yes</code>。</p>	<p><b>问题:</b> EGL 对 RecordA 执行的验证在“问题”视图中生成了消息。</p> <p><b>解决方案:</b> 编辑 RecordA 并根据 RecordB 包括适当的表和键信息。并且, 将缺省选择条件中的任何 !itemColumnName 变量替换为 RecordB 中的对应 SQL 列名。</p>

## 具有相同记录名的不同定义

**VisualAge Generator:** 记录根据当前位于工作空间中的项目和包来包含共享数据项。这使不同的子系统或程序能够具有同一个记录名的不同定义。

**EGL:** 为记录定义提供了 `containerContextDependent=yes` 属性。此属性使您能够指定用于类型定义的 `dataItem` 部件基于程序的部件名空间。

**迁移所需的关联部件:** 无。您应该完全了解所有子系统的 VAGen 部件结构才能设置此记录属性。

表 16. 具有相同记录名的不同定义

带关联部件迁移	不带关联部件迁移
<p>迁移工具不为记录定义设置 <code>containerContextDependent=yes</code> 属性。如果需要此功能, 则必须对每个需要它的记录设置此属性。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>

## 处理表的不确定情况

### 保留字和表名

**VisualAge Generator:** VisualAge Generator 没有保留字。符号 # 在 VAGen 表名中是无效的。

**EGL:** EGL 具有保留字。另外, EGL 不允许符号 # 作为部件名的第一个字符。dataTable 名不能是保留字。

迁移所需的关联部件: 无。

表 17. 保留字和表名

带关联部件迁移	不带关联部件迁移
迁移工具不会将表重命名。如果表名与保留字列表匹配, 则第 1 阶段迁移中使用的迁移工具会发出错误消息。如果未更改表名, 则第 2 阶段迁移中使用的迁移工具也会发出错误消息。	迁移工具执行带关联部件迁移列中提到的操作。
<p><b>潜在问题:</b> 仅当 dataTable 名与保留字列表匹配时, 才会发生问题。EGL 验证会导致“问题”视图中出现一条消息。</p> <p><b>解决方案:</b> 在迁移之前在 VisualAge Generator 中重命名表, 或者在迁移之后在 EGL 中重命名 dataTable。如果在 VisualAge Generator 中更改名称, 则务必更改程序、映射、函数以及数据项定义中对该表的所有引用。如果在 EGL 中更改名称, 则必须更改 dataTable 的名称以及对它的所有引用。这包括下列位置中的引用:</p> <ul style="list-style-type: none"> <li>• 程序使用声明语句</li> <li>• 程序和函数中的逻辑语句</li> <li>• 数据项 validatorDataTable 属性</li> <li>• 表单字段 validatorDataTable 属性</li> </ul> <p>如果要保留原始表名来作为所生成的 dataTable 的名称, 则将 alias 属性设置为原始 dataTable 名。如果未指定 alias 属性, 则务必更改对该 dataTable 名的任何非 EGL 引用, 包括 CICS 程序定义。</p>	带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。

## 处理映射组和映射的不确定情况

### 保留字和 formGroup 名

- **VisualAge Generator:** VisualAge Generator 没有保留字。符号 # 在 VAGen 映射组名中是无效的。
- **EGL:** EGL 具有保留字。另外, EGL 不允许符号 # 作为部件名的第一个字符。formGroup 名不能是保留字。
- **迁移所需的关联部件:** 无。

表 18. 保留字和 formGroup 名

带关联部件迁移	不带关联部件迁移
迁移工具不会将 formGroup 重命名。如果映射组名与保留字列表匹配, 则第 1 阶段迁移中使用的迁移工具会发出错误消息。如果未更改映射组名, 则第 2 阶段迁移中使用的迁移工具也会发出错误消息。	迁移工具执行带关联部件迁移列中提到的操作。

表 18. 保留字和 formGroup 名 (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题:</b> 仅当 formGroup 名与保留字列表匹配时, 才会发生问题。EGL 验证会导致“问题”视图中出现一条消息。</p> <p><b>解决方案:</b> 在迁移之前在 VisualAge Generator 中将映射组重命名, 或者在迁移之后在 EGL 中将 formGroup 重命名。如果在 VisualAge Generator 中将映射组重命名, 则务必将所有属于该映射组的映射重命名。并且, 更改所有程序定义中对该映射组的所有引用。如果在 EGL 中将该 formGroup 重命名, 则必须更改该 formGroup 的名称以及对它的所有引用, 包括程序使用声明语句中的引用。如果要保留原始映射组名来作为所生成的 formGroup 的名称, 则将 <i>alias</i> 属性设置为原始映射组 (formGroup) 名。如果未指定 <i>alias</i> 属性, 则务必更改对该 formGroup 名的任何非 EGL 引用, 包括 CICS 程序定义。</p>	<p>带关联部件迁移列中列出的那些问题也会发生。您可以使用相同的解决方案。</p>

## 映射组和 formGroup 需求

**VisualAge Generator:** 仅当存在浮动区指定时, 映射组才是必需的。

**EGL:** 始终需要 formGroup 以包含表单。

**迁移所需的关联部件:** 映射组以及该映射组中的所有映射。

表 19. 映射组和 formGroup 需求

带关联部件迁移	不带关联部件迁移
<p>如果映射组不存在, 则迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>• 为具有相同映射组名的所有映射创建一个 formGroup。</li> <li>• 将同一个 formGroup 的所有表单放到同一个 EGL 文件中。</li> <li>• 如果不是以单文件迁移方式进行迁移, 则将表单嵌套在 formGroup 定义中。</li> <li>• 如果正在以单文件方式进行迁移, 则发出错误消息, 指示需要对 formGroup 进行编辑以嵌套表单。</li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。但是, 如果映射组及其所有映射不在同一个迁移集中, 则可能会出现问 题, 如下所述。</p>
<p><b>潜在问题:</b> 无。应该将映射组的所有映射包括在同一个迁移集中。由于迁移集表示的是生成的内容, 所以, 迁移集应该包括映射组中的所有映射。</p> <p>如果正在以单文件方式进行迁移, 则务必将映射组中的所有映射包括在同一个外部源格式文件中。</p>	<p><b>潜在问题:</b> 如果未将同一个映射组名的所有映射包括在同一个迁移集中 (或者, 对于单文件方式迁移来说, 未将它们包括在同一个外部源格式文件中), 则 formGroup 就不会包括所有表单。</p> <p><b>可能的解决方案 1:</b> 确保迁移集包括所有具有相同映射组名的映射。</p> <p><b>可能的解决方案 2:</b> 将缺少的表单添加到 EGL 文件中, 并将它们嵌套在 formGroup 定义中。</p>

## 浮动区和起始位置

**VisualAge Generator:** VisualAge Generator 允许（但不建议）对具有相同设备大小的不同设备类型使用不同的浮动区大小和起始位置。

**EGL:** EGL formGroup 和打印表单仅指定了设备大小。EGL 文本表单既指定了设备大小也指定了表单大小。EGL 只允许对某一设备大小指定一组边界。

迁移所需的关联部件: 无。

表 20. 浮动区和起始位置

带关联部件迁移	不带关联部件迁移
迁移工具执行下列操作: <ul style="list-style-type: none"> <li>• 如果两个或更多个设备具有相同的设备大小但具有不同的浮动区大小或起始位置，则发出错误消息。</li> <li>• 对指定的每个 VAGen 浮动区包括等效的 EGL 设备大小和边界指定。</li> </ul>	迁移工具执行带关联部件迁移列中提到的操作。
<b>潜在问题:</b> 仅当 VisualAge Generator 中的两个或更多个具有相同设备大小的设备指定了不同的浮动区大小或起始位置时，才会发生问题。EGL 验证会导致“问题”视图中出现一条消息。  <b>可能的解决方案:</b> 查看错误消息。编辑 formGroup 定义以指定此设备大小所需的一个浮动区大小和起始位置。	带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。

## 映射组、映射和设备大小

**VisualAge Generator:** VisualAge Generator 支持深度和宽度为 6x40、12x40、16x64 和 255x160 的显示器映射的设备大小。

**EGL:** 对于文本表单，EGL 支持公共设备大小，但对于 COBOL 生成，不允许设备大小 6x40、12x40、16x64 和 255x160。Java 生成支持这些设备。

迁移所需的关联部件: 无。

表 21. 映射组、映射和设备大小

带关联部件迁移	不带关联部件迁移
当迁移映射组时，迁移工具将原始深度和宽度包括在 ScreenFloatingArea 属性内的 screenSize 属性中。工具还发出警告消息。	迁移工具执行带关联部件迁移列中提到的操作。
当迁移显示器映射时，迁移工具将原始深度和宽度包括在所迁移的文本表单的 screenSizes 属性中。工具还发出警告消息。	迁移工具执行带关联部件迁移列中提到的操作。
<b>潜在问题:</b> EGL COBOL 生成不支持这些设备。在生成期间，将发生错误。  <b>解决方案:</b> 如果为 COBOL 生成，则在 EGL 中编辑 formGroup 和文本表单并除去或更改旧的屏幕大小。	带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。

## 映射名和帮助映射名

**VisualAge Generator:** 映射名是由两部分组成的名称，它由映射组和映射名组成。程序的主映射组和帮助映射组都可以包含同名的映射。例如，对于程序 X，主映射组 GROUPA 和帮助映射组 GROUPH 都可以包含名为 MAP1 的映射。映射名限长 8 个字符。映射名可以与程序名相同。符号 # 在 VAGen 映射组名中是无效的，但在映射名的映射名部分中允许符号 #。

**EGL:** 表单名不包含 formGroup 名。而是，文本表单和打印表单是在 formGroup 部件中定义的。EGL 还要求主 formGroup 和帮助 formGroup 中的所有表单名都是唯一的（在程序的两个 formGroup 中没有重复的表单名）。EGL 不允许表单名与程序名相同。另外，EGL 不允许表单名是保留字或使用符号 # 作为表单名的第一个字符。在定义时，EGL 允许表单名的长度超过 8 个字符。在生成时，如果指定了别名，则使用别名作为表单名。在生成的代码的主 formGroup 和帮助 formGroup 中，允许重复的名称。

**迁移所需的关联部件:** 当迁移映射组时，需要程序及其映射组、帮助映射组以及这两个映射组中的所有映射。

表 22. 映射名和帮助映射名

带关联部件迁移	不带关联部件迁移
<p>根据第一个要迁移主映射组或帮助映射组的程序，迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 对于与保留字匹配的映射名或者将符号 # 用作名称的映射名部分的第一个字符的映射名，对该映射名执行重命名。根据需要，程序的主映射组和帮助映射组中的映射都会被重命名。</li> <li>• 检查程序的帮助映射组中的所有映射的名称，了解是否存在与主映射组重复的名称。</li> <li>• 将程序名与该程序的主映射组和帮助映射组中所有映射的名称作比较。</li> </ul> <p>如果帮助映射组中的某个映射与主映射组中的任何映射都不同名，则迁移工具不更改该帮助映射名。</p> <p>如果帮助映射组中的某个映射与程序的主映射组中的任何映射同名，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 如果帮助映射仅包含常量，则迁移工具执行下列操作： <ul style="list-style-type: none"> <li>– 将该帮助映射重命名为 <code>helpMapName</code> 并加上客户指定的后缀。</li> <li>– 包括具有原始帮助映射名的 <code>alias</code> 属性。</li> <li>– 更改任何映射的 <code>helpForm</code> 属性以指定新的帮助映射名。</li> </ul> </li> <li>• 如果帮助映射组中的映射包含变量，则迁移工具执行下列操作： <ul style="list-style-type: none"> <li>– 发出错误消息。</li> <li>– 不重命名该映射。</li> <li>– 迁移该映射。</li> </ul> </li> </ul> <p>这是因为该映射可以被别的某个程序使用，该程序将该帮助映射组指定为该程序的主映射组。</p> <p>如果帮助映射组中的映射只包含常量字段，并且映射名与程序名相同，则迁移工具将该帮助映射重命名。这里所进行的处理与帮助映射组和主映射组中发生冲突映射名时所执行的重命名处理是一样的。</p> <p>如果帮助映射组中的映射包含任何变量字段并且与程序同名，或者如果主映射组中的映射与程序同名，则迁移工具不会将该映射重命名。还将完成在无法为帮助映射组和主映射组中的冲突映射名执行重命名操作时进行的处理。</p>	<p>当迁移映射组时，如果没有程序可用，则迁移工具不知道两个映射组相关，并且不知道某个映射组是否曾经被指定为帮助映射组。迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 执行由于与保留字匹配而需要进行的任何映射名重命名。</li> <li>• 不检查帮助映射的其它重命名。</li> </ul>

表 22. 映射名和帮助映射名 (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题 1:</b> 如果一个 formGroup 在一个程序中被用作主 formGroup, 而在另一个程序中被用作帮助 formGroup, 则会发生问题。</p> <p><b>可能的解决方案:</b> 将该帮助 formGroup 分为两个 formGroup, 其中一个仅包含帮助表单, 另一个包含带有变量字段的表单。指定仅包含帮助表单的 formGroup 作为原始程序的帮助 formGroup。指定包含带有变量字段的表单的 formGroup 作为第二个程序的主 formGroup, 并指定仅包含帮助表单的 formGroup 作为该程序的帮助 formGroup。</p> <p><b>潜在问题 2:</b> 如果帮助 formGroup 中的一个映射包含变量字段并且与主 formGroup 中的某个映射同名, 则会发生问题。</p> <p><b>可能的解决方案:</b> 与问题 1 的可能的解决方案相同。</p> <p><b>潜在问题 3:</b> 如果同一个帮助 formGroup 由多个程序共享, 则会发生问题。在这种情况下, 迁移工具可能不会将所有需要为各种程序重命名的帮助表单重命名。</p> <p><b>可能的解决方案:</b> 通过对名称添加帮助映射后缀, 将帮助 formGroup 中的所有必需表单重命名。包括 alias 属性以提供原始帮助映射名, 以便在生成期间使用。更改所有 formGroup 中的所有对应文本表单, 以指定新的帮助表单名。</p>	<p><b>潜在问题:</b> 仅当 formGroup 在程序中使用并且在主 formGroup 和帮助 formGroup 中的表单名之间存在冲突时, 才会发生问题。</p> <p><b>可能的解决方案:</b> 带关联部件迁移中显示的解决方案也适用。</p>

## 数字变量字段

**VisualAge Generator:** 映射中的数字字段具有一个长度。长度应该足以容纳所有的位、小数点、符号、货币符号和数字分隔符。但是, 如果该字段在运行时不够长, 则 VisualAge Generator 将省略货币符号和数字分隔符。如果该字段是正数值, 则 VisualAge Generator 还省略符号。如果需要适合所允许的空间, 则 VisualAge Generator 将删除高位。

**EGL:** 表单中的变量字段既指定了类型定义 (它包括位数和小数位数) 也指定了 fieldLen 属性 (此属性指定数据在表单上占用的空间)。如果在运行时 fieldLen 的大小不足以包含所有位和格式字符, 则 EGL 发出运行时消息。

**迁移所需的关联部件:** 无。

表 23. 数字变量字段

带关联部件迁移	不带关联部件迁移
<p>当迁移映射中的数字字段时，迁移工具设置长度和 fieldLen，如下所示：</p> <ul style="list-style-type: none"> <li>迁移工具始终将 fieldLen 设置为在 VisualAge Generator 中对该变量字段指定的长度。</li> <li>迁移工具按如下方式设置类型定义中的长度和小数位数： <ul style="list-style-type: none"> <li>如果变量字段未指定小数位数，则迁移工具将类型定义中的长度设置为 fieldLen。</li> <li>如果变量字段指定了小数位数，则迁移工具将类型定义中的长度设置为 fieldLen - 1 以允许输入小数点。这种技术能够在运行时可能会发生的任何溢出问题。</li> </ul> </li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题：</b> 如果表单上的字段长度在运行时不足以包含所有位、小数点、符号、货币符号和数字分隔字符，则 EGL 将发出运行时错误消息。</p> <p><b>解决方案：</b> 更改表单定义，以使 fieldLen 的大小足以包含在运行时将会出现的最大可能数字以及您指定的所有格式字符。</p>	<p>带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。</p>

## 映射变量字段和编辑例程

**VisualAge Generator:** 映射变量字段可以具有编辑例程，后者是表、函数、EZEC10 或 EZEC11。仅当编辑例程是 EZEC10、EZEC11 或表时，才使用编辑消息。

**EGL:** 表单字段可以既具有 validatorDataTable 又具有 validatorFunction。表单字段还可以既具有 validatorDataTableMsgKey 又具有 validatorFunctionMsgKey。

**迁移所需的关联部件：** 表或函数部件。

表 24. 变量映射字段和编辑例程

带关联部件迁移	不带关联部件迁移
<p>第一次迁移映射时，迁移工具执行下列各项中的第一个适用项：</p> <ul style="list-style-type: none"> <li>如果 editRoutineName 是 EZEC10 或 EZEC11，则迁移工具将 validatorFunction 属性设置为等效的 EGL 系统库函数。迁移工具还将 validatorFunctionMsgKey 设置为编辑消息（如果有的话）。</li> <li>如果 editRoutineName 是函数，则迁移工具设置 validatorFunction 属性。由于在 VisualAge Generator 中不使用 validatorFunctionMsgKey，所以迁移工具将省略该属性。</li> <li>如果 editRoutineName 是表，则迁移工具设置 validatorDataTable 属性。迁移工具还将 validatorDataTableMsgKey 设置为编辑消息（如果有的话）。</li> </ul>	<p>如果没有与 editRoutineName 同名的函数或表可用，则迁移工具执行下列各项中的第一个适用项：</p> <ul style="list-style-type: none"> <li>如果 editRoutineName 是 EZEC10 或 EZEC11，则迁移工具将 validatorFunction 属性设置为等效的 EGL 系统库函数名。迁移工具还将 validatorFunctionMsgKey 设置为编辑消息（如果有的话）。</li> <li>如果 editRoutineName 的长度超过 7 个字符，则它肯定是函数名，因此迁移工具设置 validatorFunction 属性。由于在 VisualAge Generator 中不使用 validatorFunctionMsgKey，所以迁移工具将省略该属性。</li> <li>如果指定了编辑消息，则迁移工具设置 validatorDataTable 和 validatorDataTableMsgKey。</li> <li>如果未指定编辑消息，则迁移工具设置 validatorFunction 属性并发出错误消息。</li> </ul>

表 24. 变量映射字段和编辑例程 (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题:</b> 仅当函数与 dataTable 同名 (很有可能位于不同的子系统中) 并且两个程序共享同一个 formGroup (很有可能位于同一个子系统中), 并且一个程序需要使用函数而另一个程序需要使用 dataTable 时, 才会发生问题。</p> <p><b>可能的解决方案:</b> 查看共享 formGroup 的程序。如果发生这种情况, 则创建单独的 formGroup 以使用 validatorDataTable。缺点: 现在要维护两个 formGroup。通过将完全相同的表单移到公共文件中并接着在每个 formGroup 中指定 use formName 语句以指向公共表单, 可以最大程度地克服这种缺点。</p>	<p><b>潜在问题:</b> 仅当迁移工具的猜测不正确时, 才会发生问题。当迁移工具指定了函数时, 任何使用此表单的程序都可能需要 dataTable。</p> <p><b>可能的解决方案:</b> 查看具有错误消息的映射使用。</p>

## 映射字段和数字硬件属性

**VisualAge Generator:** VisualAge Generator 对字符常量字段、字符变量字段和数字变量字段支持数字硬件属性。数字硬件属性能够防止最终用户在变量字段中输入非数字数据。

**EGL:** EGL 对字符变量字段只支持 *isDecimalDigit* 属性。数字字段具有软编辑, 从而确保只能将有效的数字字符和格式字符 (如符号或小数点) 输入到字段中。

**迁移所需的关联部件:** 无。

表 25. 映射字段和数字硬件属性

带关联部件迁移	不带关联部件迁移
<p>迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>对于映射中任何指定了数字硬件属性的字符变量, 迁移工具包括 <i>isDecimalDigit = yes</i> 属性。</li> <li>对于映射中的任何字符常量, 工具始终省略 <i>isDecimalDigit</i> 属性。</li> <li>对于映射中的任何数字变量字段, 工具始终省略 <i>isDecimalDigit</i> 属性。</li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题:</b> 在运行时, 由于最终用户能够将非数字数据输入到数字字段中, 所以他们会注意到轻微的改变。如果发生这种情况, EGL 将发出运行时错误消息。</p> <p><b>可能的解决方案:</b> 考虑通知最终用户, 当从 VAGen 生成的代码切换到 EGL 生成的代码时, 这是预料之中的差别。</p>	<p>带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。</p>

## 映射数组和属性

**VisualAge Generator:** VisualAge Generator 允许 (但不建议) 对数组元素使用不同的属性。例如, 在 VisualAge Generator 中, 保护、需要输入、在输出时需要填充、数字硬件属性、已修正数据标记和光笔检测对于每个映射数组元素都可能会有变化。

**EGL:** 在 EGL 中，可以被覆盖的数组项属性只有字段表示属性（颜色、突出显示、亮度、保护、已修正和轮廓）以及光标、位置和值。

迁移所需的关联部件：无。

表 26. 映射数组和属性字段

带关联部件迁移	不带关联部件迁移
迁移工具使用第一个数组元素（数组下标为 1）的下列属性来设置等效的 EGL 属性：需要输入、在输出时需要填充、数字硬件属性和光笔检测。EGL 对所有数组元素使用第一个数组元素的属性。	迁移工具执行带关联部件迁移列中提到的操作。
<p><b>潜在问题:</b> 仅当对数组元素使用不同的属性时，才会发生问题。</p> <p><b>可能的解决方案:</b> 将第一个数组元素的属性更改为具有最低限制性的值，并在 validatorFunction 中添加逻辑以验证是否每个数组元素都符合必需的条件。并且，将表单在运行时的任何外观差别通知最终用户。</p>	带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。

## 未命名变量字段

**VisualAge Generator:** VisualAge Generator 允许（但不建议）在映射中使用未命名变量字段。在生成时，将把未命名变量字段转换为常量。程序和函数永远不能引用未命名变量字段。

**EGL:** EGL 不允许在表单上使用未命名变量字段。

迁移所需的关联部件：无。

表 27. 未命名变量字段

带关联部件迁移	不带关联部件迁移
<p>对于映射中的任何未命名变量字段，迁移工具检查是否指定了下列任何一项：</p> <ul style="list-style-type: none"> <li>• 初始值</li> <li>• Protect = yes</li> <li>• Cursor = yes</li> <li>• 轮廓而不是“无轮廓”</li> <li>• 突出显示而不是“不突出显示”</li> </ul> <p>如果指定了上述任何一项，则迁移工具创建具有相应 EGL 属性的常量字段并发出警告消息。</p>	迁移工具执行带关联部件迁移列中提到的操作。
<p>如果未对未命名变量字段指定任何属性，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 不在表单上创建常量字段。</li> <li>• 发出警告消息。</li> </ul>	迁移工具执行带关联部件迁移列中提到的操作。
<b>潜在问题:</b> 无。不能在 VisualAge Generator 中引用该字段。	<b>潜在问题:</b> 无。

## 不受保护映射常量

**VisualAge Generator:** VisualAge Generator 支持在映射中使用不受保护常量。在测试和生成时，不受保护常量被视为就象是保护设置为自动跳过一样。

**EGL:** EGL 不支持在表单上使用不受保护常量。对于文本表单上的常量，EGL 既支持 protect=yes 也支持 protect=skip。对于打印表单，EGL 不支持 protect 属性。

迁移所需的关联部件: 无。

表 28. 不受保护映射常量

带关联部件迁移	不带关联部件迁移
当迁移表单时，对于不受保护的常量字段，迁移工具执行下列操作： <ul style="list-style-type: none"><li>• 如果该表单是文本表单，则迁移工具将 EGL protect 属性设置为 skip 并发出错误消息。</li><li>• 如果该表单是打印表单，则迁移工具将省略 protect 属性并且不发出错误消息。在 EGL 打印表单中不使用 protect 属性。</li></ul>	迁移工具执行带关联部件迁移列中提到的操作。
潜在问题: 无。	潜在问题: 无。

## 定位在“行 = 0，列 = 0”处的字段

**VisualAge Generator:** VisualAge Generator 4.5 允许使用来自旧版本的跨系统产品或 VisualAge Generator 的定位在“行 = 0，列 = 0”处的字段。但是，VisualAge Generator 4.5 未提供在此位置创建字段的方法。您无法为定位在“行 = 0，列 = 0”位置处的字段设置属性信息。

**EGL:** EGL 不支持定位在“行 = 0，列 = 0”处的字段。每个字段都必须包括一个属性字节。

迁移所需的关联部件: 无。

表 29. 定位在“行 = 0, 列 = 0”处的字段

带关联部件迁移	不带关联部件迁移
<p>当迁移表单时，如果某个字段定位在“行 = 0, 列 = 0”处，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 如果该字段是常量字段并且值是第一个字符是空白，则迁移工具执行下列操作：                             <ul style="list-style-type: none"> <li>- 从值中除去第一个字符并将字段长度减小 1。</li> <li>- 将 position 属性设置为 [1,1]。</li> <li>- 包括字段的缺省表示属性。</li> <li>- 发出警告消息。</li> </ul> </li> <li>• 如果该字段是常量字段并且值的第一个字符不是空白，或者如果该字段是变量字段，则迁移工具执行下列操作：                             <ul style="list-style-type: none"> <li>- 不更改值或字段长度。</li> <li>- 将 position 属性设置为 [0,0]。</li> <li>- 包括字段的缺省表示属性。</li> <li>- 发出错误消息。</li> </ul> </li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题 1:</b> 如果该字段不能被更改并且定位在 position=[0,0] 位置处，则在“问题”视图中将有一个错误。</p> <p><b>解决方案 1:</b> 修改该表单并更改字段的位置。可能需要移动其它字段或对常量进行再定位以便为该字段的属性字节腾出空间。并且，查看缺省表示属性以确保使用了正确的颜色和突出显示等等。</p> <p><b>潜在问题 2:</b> 如果将常量字段更改为 position=[1,1]，则会由于缺省表示属性的影响而产生另一种运行时外观。</p> <p><b>解决方案 2:</b> 查看迁移警告消息并确保测试任何被迁移工具调整了字段位置的表单。</p>	<p><b>潜在问题:</b> 带关联部件迁移列中列示的问题也会发生。您可以使用相同的解决方案。</p>

## 处理程序的不确定情况

### 程序名和保留字

**VisualAge Generator:** VisualAge Generator 没有保留字。符号 # 在 VAGen 程序名中是无效的。

**EGL:** EGL 具有保留字。另外，EGL 不允许符号 # 作为部件名的第一个字符。程序名不能是保留字。

迁移所需的关联部件: 无。

表 30. 程序名和保留字

带关联部件迁移	不带关联部件迁移
迁移工具不会将程序重命名。如果程序名与保留字列表匹配，则第 1 阶段迁移中使用的迁移工具会发出错误消息。如果未更改程序名，则第 2 阶段迁移中使用的迁移工具也会发出错误消息。	迁移工具执行带关联部件迁移列中提到的操作。
<p><b>潜在问题:</b> 仅当程序名与保留字列表匹配时，才会发生问题。EGL 验证会导致“问题”视图中出现一条消息。</p> <p><b>解决方案:</b> 将该程序重命名。在进行迁移之后，可以在 VisualAge Generator 或 EGL 中完成此操作。如果在 EGL 中重命名该程序，则必须更改该程序的名称以及对该程序的所有引用，包括 call、transfer 和 show 语句中的引用。如果要保留原始程序名来作为所生成的程序的名称，则将 <i>alias</i> 属性设置为原始程序名。如果未指定 <i>alias</i> 属性，则必须更改对该程序名的任何非 EGL 引用，包括 CICS 程序定义。</p>	带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。

## 程序中的隐式数据项

**VisualAge Generator:** VisualAge Generator 允许（但不建议）使用隐式数据项（未在记录、映射、表、被调用参数列表、函数参数列表或函数本地存储器中显式定义的项）。

**EGL:** EGL 不允许隐式数据项。

**迁移所需的关联部件:** 无。

表 31. 程序中的隐式数据项

带关联部件迁移	不带关联部件迁移
迁移工具不为隐式项创建定义。如果程序允许隐式项，则第 2 阶段迁移中使用的迁移工具会发出警告消息。	迁移工具执行带关联部件迁移列中提到的操作。
<p><b>潜在问题:</b> 仅当程序真正使用了隐式项时，才会发生问题。查看任务列表日志，了解是否有任何程序允许隐式项。如果程序实际上使用了隐式项，则在“问题”视图中将会有错误。</p> <p><b>解决方案:</b> 可以在 VisualAge Generator 或 EGL 中将该隐式项的定义添加到程序中。VAGen 验证将显示隐式项所需的定义。</p>	带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。

## 相关联的程序部件

**VisualAge Generator:** 对于 VisualAge Java 来说，程序的关联项可以位于多个项目和包中，对于 VisualAge Smalltalk 来说，可以位于多个配置映射和应用程序中。

**EGL:** 程序的关联项可以位于多个项目、文件夹、包和文件中。

**迁移所需的关联部件:** 对于程序：所有关联项。

注：有关 import 语句的其它信息，请参阅第 28 页的『EGL 构建路径和 import 语句』。

表 32. 相关联的程序部件

带关联部件迁移	不带关联部件迁移
<p>迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 包括一个 package 语句以指定要将 EGL 文件放到哪个包中。</li> <li>• 对于任何包含当前文件中的任何部件所需的关联项的包（这些包位于不同于当前文件的包中），在 EGL 文件中包括 import 语句。仅当使用第 1 - 3 阶段来进行迁移时，才会包括 import 语句。</li> <li>• 对于程序，迁移工具执行下列操作： <ul style="list-style-type: none"> <li>– 包括程序的主工作存储器记录的声明。如果在 VAGen 主工作存储器记录中有 Level-77 项，则工具还包括新的 Level-77 项记录的声明。</li> <li>– 包括 VAGen 的“表及其它记录”列表中的所有记录的声明，这包括任何 VAGen 重新定义的记录的 redefines 属性（如果适用的话）。</li> <li>– 包括所有 I/O 记录的声明。</li> <li>– 包括被用作 MQ API 调用的参数的记录（在 VisualAge Generator 中被指定为 MQ 消息记录的属性的记录）的声明。</li> <li>– 包括 VAGen 的“表及其它记录”列表中的任何表的使用声明。</li> <li>– 包括程序的映射组和帮助映射组的使用声明。</li> </ul> </li> </ul>	<p>迁移工具使用所有的可用程序关联项。</p>

表 32. 相关联的程序部件 (续)

带关联部件迁移	不带关联部件迁移
<p>潜在问题: 无。</p>	<p><b>潜在问题 1:</b> 仅当缺少部件时, 才会发生问题。如果迁移工具检测到缺少部件, 则它会发出一条警告消息, 该警告消息标识了缺少的部件。迁移工具不作任何关于缺少的部件的假定。这会在已迁移的程序中导致各种问题, 包括:</p> <ul style="list-style-type: none"> <li>• 缺少 import 语句。</li> <li>• 缺少 Level-77 记录声明。</li> <li>• 缺少 VAGen 重新定义的记录的 redefines 属性。</li> <li>• 缺少 I/O 记录声明。</li> <li>• 缺少在 MQ API 调用中被用作参数的记录的声明。</li> </ul> <p>除了缺少 redefines 属性的情况以外, 在“问题”视图中将会有一些错误, 这些错误可以帮助您标识问题。<b>注意: 迁移工具不会检测到所有缺少的部件。</b></p> <p><b>可能的解决方案 1A:</b> 将迁移集更改为包括在 VisualAge Generator 中验证程序所需的所有部件。使用新的迁移集来再次迁移程序, 以便同时迁移程序的所有关联项。</p> <p><b>可能的解决方案 1B:</b> 在 EGL 找到缺少的部件并更正 EGL 程序。</p> <p><b>潜在问题 2:</b> 对于缺少 Level-77 项, 请参阅第 46 页的『记录中的 Level-77 项』。</p> <p><b>潜在问题 3:</b> 对于缺少重新定义的记录, 请参阅第 45 页的『重新定义的记录』。</p>

## 迁移所需的中间变量

**VisualAge Generator:** 某些 VAGen 语句需要中间变量来提供 EGL 中的等效支持。

**EGL:** EGL 提供了系统库函数, 这些系统库函数提供了 VAGen 迁移所需的一些信息。此支持只有在 VisualAge Generator 兼容性方式下才可用。

迁移所需的关联部件: 无。

表 33. 迁移所需的中间变量

带关联部件迁移	不带关联部件迁移
<p>当迁移任何程序时，迁移工具总是包括以下各项的声明：</p> <ul style="list-style-type: none"> <li>• <code>custPrefixEZEREPLY</code></li> <li>• <code>custPrefixEZE_ITEMLEN</code></li> <li>• <code>custPrefixEZE_WAIT_TIME</code></li> </ul> <p>如果未选择 VAGen 迁移首选项不初始化以前的 <code>EZESYS</code> 值，迁移工具还将执行以下操作：</p> <ul style="list-style-type: none"> <li>• 包括 <code>custPrefixEZESYS</code> 的声明。</li> <li>• 包括一个初始化语句来将 <code>custPrefixEZESYS</code> 的值设置为旧的 VAGen <code>EZESYS</code> 值。</li> </ul> <p><code>custPrefix</code> 是用来更改与保留字冲突的部件名的前缀。使用 VAGen 迁移语法首选项来设置它的值。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p>这 4 个变量用来迁移下列内容：</p> <ul style="list-style-type: none"> <li>• 语句中对 <code>EZESYS</code> 的引用，这些语句不包括需要旧的 VAGen 值的 <code>IF</code>、<code>WHILE</code> 和 <code>TEST</code>。</li> <li>• VAGen 服务例程（如果未指定 <code>REPLY</code> 选项的话）。在这种情况下，必须保存并复原 <code>handleSysLibraryErrors</code> 的当前值。</li> <li>• <code>TEST nnn</code>、<code>+nnn</code> 或 <code>-nnn</code> 语句在 EGL 中没有直接的等效项。使用一个 EGL 系统库函数来确定用户输入的数据的长度。</li> <li>• <code>EZEWAIT</code> 函数。在这种情况下，迁移工具将添加用来将时间转换为秒的逻辑。</li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题：</b> 仅当您在迁移时选择 VAGen 迁移首选项不初始化以前的 <code>EZESYS</code> 值并且在 <code>IF</code>、<code>WHILE</code> 或 <code>TEST</code> 之外的语句中使用 <code>EZESYS</code> 时，才会出现问题。在此情况下，已迁移工具在该语句中使用 <code>custPrefixEZESYS</code>，但程序的 <code>custPrefixEZESYS</code> 没有声明和初始化语句。在“问题”视图中会出现错误。</p> <p><b>可能的解决方案：</b> 将 EGL 逻辑更改为对 <code>sysVar.systemType</code> 使用新值。</p> <p><b>可能的解决方案 2：</b> 将 <code>custPrefixEZESYS</code> 的声明和初始化语句添加至需要对 <code>EZESYS</code> 使用以前的 VAGen 值的任何程序。</p>	<p>带关联部件迁移列中列示的问题可能会发生。您可以使用相同的解决方案。</p>

## 处理函数（包括 I/O 语句）的不确定情况

### 用于映射的 DISPLAY 语句

**VisualAge Generator:** `DISPLAY` 既用于显示器映射也用于打印机映射。

**EGL:** 使用两个独立的语句：

- *display form* 用于文本表单。
- *print form* 用于打印表单。

在 VisualAge Generator 兼容性方式下，如果表单是打印表单，则接受 *display form*。

**迁移所需的关联部件：**需要映射以确定设备类型。任何可用映射组中的第一个具有此映射名的映射就是迁移工具所使用的映射。当在程序上下文中进行迁移时，迁移工具只查看程序的主映射组。

表 34. 用于映射的 *Display* 语句

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果有具有此名称的映射可用，则迁移工具转换为下列各项：</p> <ul style="list-style-type: none"> <li>• <i>display textForm</i>（如果该映射是显示器映射的话）</li> <li>• <i>print printForm</i>（如果该映射是打印机映射的话）</li> </ul>	<p>如果没有具有此名称的映射可用，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 转换为 <i>display form</i></li> <li>• 发出警告消息，指示未能确定映射类型</li> </ul>

表 34. 用于映射的 Display 语句 (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题 1:</b> 迁移的第一个程序使用了打印表单, 因此迁移工具已迁移为 print 语句。另一个程序使用同一个函数, 但却是对文本表单使用此函数。</p> <p><b>解决方案 1:</b> 使用 VisualAge Generator 兼容性方式。编辑此函数并将 print 语句更改为 display 语句。</p> <p><b>潜在问题 2:</b> 如果您想要除去对 VisualAge Generator 兼容性方式的使用, 并且两个程序使用同一个函数 (一个程序对文本表单使用此函数, 另一个程序对打印表单使用此函数), 就会发生问题。</p> <p><b>可能的解决方案 2A:</b> 如果特定的目标环境始终使用显示器映射, 而其它环境始终使用打印映射, 则可以将 EGL 函数更改为类似于以下的内容:</p> <pre>if (sysVar.systemType is zoscics)     DISPLAY_FUNCTION(); else     PRINT_FUNCTION(); end</pre> <p>其中, DISPLAY_FUNCTION 和 PRINT_FUNCTION 分别使用 display 和 print 语句。</p> <p><b>可能的解决方案 2B:</b> 假定已将该函数迁移为 display 语句, 则将该函数由:</p> <pre>before-logic display textForm; after-logic</pre> <p>更改为:</p> <pre>before-logic-function(); display textForm; after-logic-function();</pre> <p>通过将 before-logic 和 after-logic 放到单独的函数中, 您就能够将大部分逻辑保留在公共函数中。然后, 可以将经过修改的 display 函数复制一份并将其更改为使用打印映射, 但仍使用公共的 before-logic-function 和 after-logic-function。缺点: 这有可能反过来影响使用原始 DISPLAY 函数的函数。</p>	<p><b>潜在问题:</b> 带关联部件迁移列中列示的潜在问题和可能的解决方案适用。</p>

## I/O 错误例程

**VisualAge Generator:** 执行文件或数据库 I/O 的函数可以指定 I/O 错误例程。I/O 错误例程可以是主函数或非主函数; 语法是相同的。在测试或生成时, VisualAge Generator 确定该 I/O 错误例程是程序的主函数还是非主函数。当将主函数用作 I/O 错误例程时, VisualAge Generator 将该函数堆栈弹回到堆栈顶部, 并在堆栈只包含 (I/O 错误例程) 主函数的情况下让堆栈再次从头开始, 然后调用主函数。当将非主函数用作 I/O 错误例程时, VisualAge Generator 将该非主函数添加到当前函数堆栈, 然后调用该函数。

**EGL:** 使用 try 块和 onException 语句来进行错误处理。onException 语句的语法支持下列各项:

- 使用 `exit stack functionName;` 来转移回到主函数
- 使用 `nonmainfunctionName();` 来调用非主函数
- 使用 `mainfunctionName();` 来调用主函数。VisualAge Generator 不支持此表单。EGL 将该主函数添加至当前函数堆栈，然后调用该主函数。

迁移所需的关联部件：程序及其主函数列表。

表 35. 函数的不确定情况 - 文件和数据库 I/O 错误例程

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果有程序可用，则迁移工具将执行下列操作：</p> <ul style="list-style-type: none"> <li>• 将指定了程序主函数的 I/O 错误例程更改为：</li> </ul> <pre>try   I/O-Statement;   onException exit stack functionName; end</pre> <ul style="list-style-type: none"> <li>• 将指定了非主函数的 I/O 错误例程更改为：</li> </ul> <pre>try   I/O-Statement;   onException functionName(); end</pre>	<p>如果没有程序可用，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 假定 I/O 错误例程中指定的函数是非主函数，并将其更改为：</li> </ul> <pre>try   I/O-Statement;   onException functionName(); end</pre> <ul style="list-style-type: none"> <li>• 不发出警告消息，这是因为可能会发出大量的消息，并且这些消息有可能将被忽略或者隐藏其它严重的错误消息。</li> </ul>
<p><b>潜在问题：</b> 如果在一个程序中使用此函数，而在该程序中，I/O 错误例程作为主函数或非主函数的用法与原始程序不同，就会发生问题。</p> <p><b>注：</b> 在“问题”视图中不会有消息。生成操作不会检测到错误。但是，程序的运行方式将与在 VisualAge Generator 中的运行方式不同。EGL 将把主函数添加到堆栈中，而不是象在 VisualAge Generator 中那样弹出堆栈。</p> <p><b>可能的解决方案：</b> 如果发生这种情况，则创建此 I/O 函数的新版本并使用正确的语法来转移到主函数。注意，此技术有可能反过来影响其它调用此 I/O 函数的函数。</p>	<p><b>潜在问题：</b> 如果在 I/O 错误例程是主函数的程序中使用此函数，则会发生问题。</p> <p><b>注：</b> 在“问题”视图中不会有消息。生成操作不会检测到错误。但是，程序的运行方式将与在 VisualAge Generator 中的运行方式不同。EGL 将把主函数添加到堆栈中，而不是象在 VisualAge Generator 中那样弹出堆栈。</p> <p><b>可能的解决方案：</b> 带关联部件迁移中列示的解决方案适用。</p>

## SQL I/O 语句

**VisualAge Generator:** 对于 SQL I/O，测试和生成根据记录定义以及“执行时语句构建”的使用来根据需要单个 I/O 选项展开到多个 SQL 语句中。测试和生成始终根据 SQL 记录定义来为 I/O 语句创建 tables 子句。

**EGL:** 在 EGL 程序中，必须显式地指定 SQL 语句。如果修改了 SQL 语句，则除 into 子句外的所有 SQL 子句都是必需的。“执行时语句构建”被替换为 prepare 语句，并且后面跟着 open、get 或 execute 语句。

迁移所需的关联部件：SQL 记录以及被指定为备用指定记录的记录（如果有的话）。

表 36. 函数的不确定情况 - SQL I/O

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果该 SQL 记录及其备用指定记录可用，则迁移工具根据记录定义、函数中的 SQL 语句以及“执行时语句构建”的使用来创建相应的 EGL 语句。如果函数中的 SQL 语句已被修改，则迁移工具将执行下列操作：</p> <ul style="list-style-type: none"> <li>• 构建带有所有子句（包括 <i>into</i> 子句）的 EGL SQL 语句。</li> <li>• 根据 SQL 记录或（如果适用的话）它的备用指定记录中的表名来创建任何必需的 <i>tables</i> 子句。</li> <li>• 根据 I/O 对象的记录定义或（如果适用的话）I/O 对象的备用指定记录的记录定义来创建此 SQL I/O 语句所需的任何其它缺少的子句。</li> <li>• 将任何 <i>!itemColumnNames</i> 由项名转换为相应的 SQL 列名。</li> <li>• 不查看 SQL 语句是否包含要求进行特殊处理的 SQL 保留字。请参阅第 177 页的『需要进行特殊处理的 SQL 保留字』以了解保留字列表以及当使用其中一个这些保留字作为表名或列名时必须对 SQL 语句所作的更改。</li> </ul> <p>注：</p> <ul style="list-style-type: none"> <li>• 有关与缺少的 SQL 子句相关的问题的详细信息，请参阅第 68 页的『SQL I/O 和缺少的必需 SQL 子句』。</li> <li>• 有关与 <i>!itemColumnNames</i> 的使用相关的问题的详细信息，请参阅第 70 页的『SQL I/O 和 <i>!itemColumnName</i>』。</li> </ul>	<p>如果该 SQL 记录或其备用指定记录不可用，则迁移工具在创建 EGL SQL 语句时仅有 SQL 语句修改和“执行时语句构建”信息可使用。由于迁移工具没有记录定义可用，所以迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 构建带有所有子句（包括 <i>into</i> 子句）的 EGL SQL 语句。</li> <li>• 在任何 <i>tables</i> 子句中，使用 <i>EZE_UNKNOWN_SQLTABLE</i> 作为表名，并使用 T1 作为表标号。</li> <li>• 对于任何缺少的 SQL 子句，使用 <i>EZE_UNKNOWN_SQL_clausename</i>，其中 <i>clausename</i> 是缺少的 SQL 子句的“外部源格式”关键字（例如，SELECT 或 VALUES）。</li> <li>• 对任何列名变量使用 <i>!itemColumnNames</i>。</li> <li>• 发出错误消息，指示需要查看该函数。</li> <li>• 不查看 SQL 语句是否包含要求进行特殊处理的 SQL 保留字。请参阅第 177 页的『需要进行特殊处理的 SQL 保留字』以了解保留字列表以及当使用其中一个这些保留字作为表名或列名时必须对 SQL 语句所作的更改。</li> </ul> <p>注：</p> <ul style="list-style-type: none"> <li>• 有关与缺少的 SQL 子句相关的问题的详细信息，请参阅第 68 页的『SQL I/O 和缺少的必需 SQL 子句』。</li> <li>• 有关与 <i>!itemColumnNames</i> 的使用相关的问题的详细信息，请参阅第 70 页的『SQL I/O 和 <i>!itemColumnName</i>』。</li> </ul>

表 36. 函数的不确定情况 - SQL I/O (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题 1:</b> 仅当两个记录同名并且具有不同的 SQL 表名或表标号时, 才会发生问题。这种情况可能发生在不同的子系统中, 也可能是在对测试和生成使用不同的表的情况下生成时发生的。</p> <p><b>可能的解决方案 1A:</b> 如果是因为在测试与生成之间更改了表名的限定而导致发生问题, 则更改为使用未限定的表名并在绑定时指定限定信息。</p> <p><b>可能的解决方案 1B:</b> 如果因为在不同的子系统中有不同的表名而导致发生问题, 则创建记录的副本并将其重命名。然后, 创建 I/O 函数的副本以使用新的记录名。更正新的 I/O 函数以使用正确的 tables 子句。<b>缺点:</b> 这有可能反过来影响使用此 I/O 函数的函数。</p> <p><b>可能的解决方案 1C:</b> 如果因为在不同的子系统中有不同的表名而导致发生问题, 则将记录更改为使用 <code>tableNameVariables</code> 属性并将所有为此记录执行 I/O 的函数修改为在调用 I/O 函数之前设置表名变量 - 这可能是在每个程序的主函数中进行的。另外, 也可以在 VisualAge Generator 中对表名主变量进行更改并再次迁移程序、记录和函数。<b>缺点:</b> 由于这样做将把静态 SQL 更改为动态 SQL, 所以可能会对性能产生影响。</p> <p><b>潜在问题 2:</b> 如果任何 SQL 表名或列名是其中一个要求进行特殊处理的 SQL 保留字, 则会发生问题。迁移工具不将这些 SQL 保留字括在双引号中。在“问题”视图中会出现错误。</p> <p><b>解决方案 2A:</b> 编辑函数并将 SQL 表名或列名括在双引号中。请参阅第 177 页的『需要进行特殊处理的 SQL 保留字』以获取 SQL 保留字列表以及必需语法的示例。</p>	<p><b>潜在问题 1:</b> 对于任何经过修改的 SQL 语句或者任何使用“执行时语句构建”的 SQL 语句, 会发生问题。根据是否缺少记录以及 SQL 语句中缺少哪些特定 SQL 子句的不同, “问题”视图可能会包含错误。</p> <p><b>解决方案:</b> 查看迁移日志以获取与缺少的 SQL 子句或表名相关的任何消息。另外, 在工作空间中搜索 EZE_UNKNOWN 的任何出现。根据记录定义确定正确的 tables 子句。要了解有关在 EGL 中重新创建 SQL 子句的信息, 请参阅第 68 页的『SQL I/O 和缺少的必需 SQL 子句』。要了解有关更正任何 <code>!itemColumnName</code> 变量的信息, 请参阅第 70 页的『SQL I/O 和 <code>!itemColumnName</code>』。</p> <p><b>其它潜在问题:</b> 带关联部件迁移中显示的潜在问题和解决方案适用。</p>

## SQL I/O 和缺少的必需 SQL 子句

**VisualAge Generator:** 如果您修改了任何 SQL 子句, VisualAge Generator 4.5 就会将所有 SQL 子句存储下来。但是, 一些更旧版本的跨系统产品和 VisualAge Generator 仅存储您修改的子句。如果来自更旧版本的函数在 VisualAge Generator 4.5 中从未被修改过, 则可能会缺少一些必需的 SQL 子句。

**EGL:** 如果修改了任何 SQL 子句, 则必须指定该 SQL 语句的所有 SQL 子句。

**迁移所需的关联部件:** SQL 记录以及被指定为备用指定记录的记录 (如果有的话)。

表 37. SQL I/O 和缺少的 SQL 子句

带关联部件迁移	不带关联部件迁移
如果 SQL 记录及其备用指定记录可用，并且如果存在任何 SQL 子句，但缺少某些子句，则迁移工具创建缺少的子句，如本表的以下各行所示。根据此函数的第一次迁移，迁移工具使用 SQL 记录及其备用指定记录（如果有的话）来创建缺少的子句。	如果 SQL 记录或其备用指定记录不可用，并且如果存在任何 SQL 子句，但缺少某些子句，则迁移工具创建缺少的子句，如本表的以下各行所示。根据此函数的第一次迁移，如果 SQL 记录或其备用指定记录不可用，则迁移工具故意创建无效的 EGL 语法。
<b>缺少 tables 子句:</b> 迁移工具通过列示记录中的所有 SQL 表和表标签来创建 tables 子句。迁移工具按照 SQL 表名和表名主变量在 VAGen 记录定义中的出现顺序来包括它们。	<b>缺少 tables 子句:</b> 迁移工具将 SQL 表名设置为 EZE_UNKNOWN_SQLTABLE，将表标签设置为 T1，并且发出错误消息。
<b>缺少 SELECT 子句:</b> 迁移工具通过按照项在记录中的出现顺序列示记录中的所有 SQL 列名来创建 select 子句。	<b>缺少 SELECT 子句:</b> 迁移工具将 select 子句的 SQL 列名设置为 EZE_UNKNOWN_SQL_SELECT 并发出错误消息。
<b>缺少 INTO 子句:</b> 迁移工具通过按照项在记录中的出现顺序列示记录中的所有项名来创建 into 子句。	<b>缺少 INTO 子句:</b> 迁移工具将 into 子句的项名设置为 EZE_UNKNOWN_SQL_INT0 并发出错误消息。
<b>缺少 INSERTCOLNAME 子句:</b> 迁移工具通过按照项在记录中的出现顺序列示记录中的 SQL 列名来为 VAGen ADD 函数创建要插入的列名的列表。对于任何被标识为只读的项，迁移工具省略 SQL 列名。	<b>缺少 INSERTCOLNAME 子句:</b> 迁移工具将列表的 SQL 列名设置为 EZE_UNKNOWN_SQL_INSERTCOLNAME 并发出错误消息。
<b>缺少 VALUES 子句:</b> 迁移工具通过按照项在记录中的出现顺序列示记录中的项名为 VAGen ADD 函数创建 values 子句。对于任何被标识为只读的项，迁移工具省略项名。	<b>缺少 VALUES 子句:</b> 迁移工具将 values 子句的项名设置为 EZE_UNKNOWN_SQL_VALUES 并发出错误消息。
<b>缺少 FORUPDATEOF 子句:</b> 迁移工具通过按照项在记录中的出现顺序列示记录中的 SQL 列名来创建 for update of 子句。对于任何包括在 EGL keyItems 属性中的项或任何被标识为只读的项，迁移工具省略 SQL 列名。	<b>缺少 FORUPDATEOF 子句:</b> 迁移工具将 for update of 子句的 SQL 列名设置为 EZE_UNKNOWN_SQL_FORUPDATEOF 并发出错误消息。
<b>缺少 WHERE 子句:</b> WHERE 子句不是必需的。迁移工具从不创建 where 子句。	<b>缺少 WHERE 子句:</b> 迁移工具执行带关联部件迁移列中提到的操作。
<b>缺少 ORDERBY 子句:</b> ORDERBY 子句不是必需的。迁移工具从不创建 order by 子句。	<b>缺少 ORDERBY 子句:</b> 迁移工具执行带关联部件迁移列中提到的操作。
<b>潜在问题:</b> 仅当两个同名的记录（通常位于不同的子系统）具有不同的项名或 SQL 列名时，才会发生问题。  <b>可能的解决方案:</b> 创建该函数的副本以便在第二个子系统中使用，并将新函数修改为使用正确的项名和 SQL 列名。 <b>缺点:</b> 这有可能反过来影响使用此 I/O 函数的函数。	<b>潜在问题 1:</b> 对于任何经过修改的 SQL 语句或者任何使用“执行时语句构建”的 SQL 语句，会发生问题。  <b>解决方案 1A:</b> 查看错误消息列表，获取任何与缺少的 SQL 子句相关的消息。修改 SQL I/O 函数以包括缺少的子句。构建缺少的子句所需的信息位于带关联部件迁移列中的相应行中。  <b>解决方案 1B:</b> 在 VisualAge Generator 中编辑该函数并使用“SQL 编辑器”来进行不重要的更改，如在行尾添加空格。保存 SQL 子句，然后再次迁移该函数。务必包括记录定义，以使迁移工具能够将 SQL 表信息包括在 EGL I/O 语句中。  <b>潜在问题 2:</b> 带关联部件迁移中显示的潜在问题和解决方案适用。

## SQL I/O 和 !itemColumnName

**VisualAge Generator:** 对于 SQL I/O, VisualAge Generator 允许在 SQL 语句的某些子句中使用 !itemColumnName。测试和生成确定与 SQL 行记录中的项名相对应的 SQL 列名。

**EGL:** 不支持使用 !itemColumnName。

**迁移所需的关联部件:** SQL 记录以及被指定为备用指定记录的记录 (如果有的话)。

表 38. 函数的不确定情况 - SQL I/O 和 !itemColumnName

带关联部件迁移	不带关联部件迁移
根据此函数的第一次迁移, 如果该 SQL 记录及其备用指定记录可用, 则迁移工具根据该 SQL 记录或者 (如果适用的话) 它的备用指定记录将任何 !itemColumnNames 转换为相应的 SQL 列名。	如果该 SQL 记录或其备用指定记录不可用, 则迁移工具执行下列操作: <ul style="list-style-type: none"><li>• 对任何列名变量使用 !itemColumnNames。</li><li>• 发出错误消息, 指示需要查看该函数。</li></ul>
<b>潜在问题:</b> 仅当两个同名的记录 (通常位于不同的子系统中) 具有不同的与 !itemColumnName 对应的 SQL 列名时, 才会发生问题。 <b>可能的解决方案:</b> 创建该函数的副本以便在第二个子系统中使用, 并将新函数修改为使用正确的 SQL 列名。 <b>缺点:</b> 这有可能反过来影响使用此 I/O 函数的函数。	<b>潜在问题 1:</b> 对于任何经过修改的 SQL 语句或者任何使用“执行时语句构建”的 SQL 语句, 会发生问题。 <b>解决方案:</b> 查看错误消息列表, 获取任何与 !itemColumnNames 相关的消息。修改 SQL I/O 函数, 使其包括基于 SQL 行记录的正确列名。 <b>潜在问题 2:</b> 带关联部件迁移中显示的潜在问题和解决方案适用。

## 带有多个更新的 SQL I/O

**VisualAge Generator:** 对于 SQL I/O, 如果程序中有多个 UPDATE 或 SETUPD 函数, 则每个 SQL REPLACE 函数都必须指定其对应的 UPDATE 或 SETUPD 函数的名称。对于非 SQL I/O, 不要求这样做。对于非 SQL I/O, 不支持 SETUPD。

**EGL:** 对于 SQL I/O, 如果有多个 get for update 或 open for update 语句, 则每个 SQL replace 语句都必须指定其对应的 get 或 open 语句的名称。每个 get 和 open 语句都指定了 resultSetID。replace 语句指定相应的 get 或 open 语句的 resultSetID。resultSetID 不适用于非 SQL I/O。

**迁移所需的关联部件:** 作为 I/O 对象的记录。

表 39. 函数的不确定情况 - 带有多个更新的 SQL I/O

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果该记录可用，则迁移工具根据记录类型创建相应的 EGL 语句。</p> <p>对于 SQL，迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>当迁移任何 UPDATE 或 SETUPD 函数时，始终包括 resultSetID。该 resultSetID 是使用函数名和客户指定的后缀创建的。</li> <li>当迁移任何指定了相应的 UPDATE 或 SETUPD 函数名的 REPLACE 函数时，包括 resultSetID。该 resultSetID 是使用相应的 UPDATE 或 SETUPD 函数名和客户指定的后缀创建的。</li> </ul> <p>对于非 SQL 来说，当迁移 UPDATE 或 REPLACE 函数时，迁移工具始终省略 resultSetID。对于非 SQL I/O，没有 SETUPD 函数。</p>	<p>当迁移 UPDATE 函数时，如果该记录不可用，则迁移工具将执行下列操作：</p> <ul style="list-style-type: none"> <li>通过检查此函数是否还带有 SQL 子句或任何特定于 SQL 的信息（如“执行时语句构建”、单行选择或带锁定游标），尝试确定此函数是否是用于 SQL I/O 的函数。</li> <li>如果迁移工具可以确定此 UPDATE 语句用于 SQL 记录，则迁移工具将 resultSetID 包括在 get 语句中。</li> <li>否则，迁移工具不包括 resultSetID。迁移工具发出警告消息。</li> </ul> <p>由于 SETUPD 仅对 SQL 有效，所以，当迁移 SETUPD 函数时，迁移工具始终包括 resultSetID。</p> <p>当迁移 REPLACE 函数时，如果该函数指定了相应的 UPDATE 或 SETUPD 函数名，则迁移工具包括 resultSetID。</p>
<p>潜在问题：无。</p>	<p>潜在问题：仅当未经修改的 UPDATE 函数确实引用 SQL 记录并且是在包含多个 get 或 open for update 语句的程序中使用，才会发生问题。在这种情况下，每个 replace 语句都将包括 resultSetID，但是，为 VAGen UPDATE 语句迁移的 get 语句将不包括 resultSetID。程序的生成将失败。</p> <p>解决方案：修改函数，让 get 语句包括 resultSetID。</p>

## 处理其它语句的不确定情况

### 语句中的隐式数据项

**VisualAge Generator:** VisualAge Generator 允许（但不建议）使用隐式数据项（未在记录、映射、表、被调用参数列表、函数参数列表或函数本地存储器存储器中显式定义的项）。

**EGL:** EGL 不允许隐式项。

迁移所需的关联部件：无。

表 40. 语句中的隐式数据项

带关联部件迁移	不带关联部件迁移
请参阅第 60 页的『程序中的隐式数据项』	请参阅第 60 页的『程序中的隐式数据项』

### 语句中的 Level-77 项

**VisualAge Generator:** 只有工作存储器记录可以包含 Level-77 项。程序只能引用主工作存储器记录中的 Level-77 项。

**EGL:** 不允许 Level-77 项。

迁移所需的关联部件：当迁移函数时，需要工作存储器记录。

表 41. 语句中的 Level-77 项

带关联部件迁移	不带关联部件迁移
请参阅第 46 页的『记录中的 Level-77 项』	请参阅第 46 页的『记录中的 Level-77 项』

## 赋值语句

**VisualAge Generator:** 允许对记录和映射以及“对应移动”的结果使用赋值语句。允许对项使用 *MOVE* 语句。

**EGL:** 只能对数据项或者记录的逐字节移动使用赋值语句。不能对映射使用赋值语句。记录和映射的对应移动需要 *move by name* 语句。可以对项使用不带修饰符的 *move* 语句，但最好使用赋值语句。

迁移所需的关联部件: 无。

表 42. 赋值语句

带关联部件迁移	不带关联部件迁移
<p>为了尽可能保留更多的公共代码，如果赋值或 <i>Move</i> 语句的源和目标都是未限定的无下标名称，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>检查函数的参数列表、本地存储器和 I/O 对象，以尝试确定赋值或 <i>MOVE</i> 语句的源或目标是否是项、记录或映射。如果迁移工具可以确定，则它按如下方式进行迁移： <ul style="list-style-type: none"> <li>如果源或目标是项，则迁移到赋值语句。</li> <li>如果源或目标是记录或映射，则迁移到按名称移动语句。</li> </ul> </li> <li>如果迁移工具不能确定部件类型，则它将赋值和 <i>MOVE</i> 语句迁移为不带修饰符的 <i>Move</i> 语句。</li> </ul>	<p>处理方式与带关联部件迁移列中提到的处理方式相同。</p>
<p><b>潜在问题:</b> 无。测试和生成将不带修饰符的 <i>Move</i> 语句转换为 VAGen <i>MOVE</i> 语句。根据移动的实际源和目标的不同，这是项到项的移动或按名称移动（对应移动）。任何程序都可以使用该函数，而不必对其进行修改。</p>	<p><b>潜在问题:</b> 无。在带关联部件迁移列中提到的情况也适用。</p>

## FIND 语句

**VisualAge Generator:** *FIND* 语句中的搜索列是可选的。缺省列是 VAGen 表的第一列。

**EGL:** *FIND* 语句被替换为 *if* 语句。搜索列是必需的。

迁移所需的关联部件: VAGen 表。

表 43. FIND 语句

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果未显式地指定搜索列，并且该表可用，则迁移工具将展开该表以从表的第一列中获取搜索列的名称。</p>	<p>如果未显式地指定搜索列，并且该表不可用，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>将搜索列设置为 EZE_UNKNOWN_SEARCH_COLUMN</li> <li>发出错误消息，指示需要使用正确的列名来修改该函数。</li> </ul>
<p><b>潜在问题：</b>仅当两个 dataTable（这两个 dataTable 可能位于不同的子系统中）具有相同的 dataTable 名但具有不同的搜索列名时，才会引起问题。</p> <p><b>解决方案：</b>对于第二个子系统，添加一个字段来作为 dataTable 中第一列的子结构。这个新字段应该与第一个系统中的搜索列同名。这种技术使您能够共享公共函数，而不必更改第二个系统中的任何代码。</p>	<p><b>潜在问题 1：</b>必须提供搜索列名。在“问题”视图中将会有一个错误。</p> <p><b>解决方案：</b>编辑该函数并为 dataTable 指定正确列名。</p> <p><b>潜在问题 2：</b>带关联部件迁移中显示的潜在问题和解决方案适用。</p>

## RETR 语句

**VisualAge Generator:** RETR 语句的搜索列和返回列是可选的。缺省搜索列是 VAGen 表的第一列。缺省返回列是第二列。

**EGL:** RETR 语句被替换为 if 语句。搜索列和返回列都是必需的。

**迁移所需的关联部件：** VAGen 表。

表 44. RETR 语句

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果未显式地指定搜索列或返回列，并且该表可用，则迁移工具将展开该表以获取下列信息：</p> <ul style="list-style-type: none"> <li>从表的第一列中获取搜索列的名称。</li> <li>从表的第二列中获取返回列的名称。</li> </ul>	<p>如果未显式地指定搜索列或返回列，并且该表不可用，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>将搜索列设置为 EZE_UNKNOWN_SEARCH_COLUMN</li> <li>将返回列设置为 EZE_UNKNOWN_RETURN_COLUMN</li> <li>发出错误消息，指示需要使用正确的列名来修改该函数。</li> </ul>
<p><b>潜在问题：</b>仅当两个 dataTable（这两个 dataTable 可能位于不同的子系统中）具有相同的 dataTable 名但具有不同的搜索列名或返回列名时，才会引起问题。</p> <p><b>解决方案：</b>对于第二个子系统，添加一个字段来作为 dataTable 中第一列的子结构。这个新字段应该与第一个系统中的搜索列同名。使用第一个系统中的返回列名来对 dataTable 的第二列创建子结构。这种技术使您能够共享公共函数，而不必更改第二个系统中的任何代码。</p>	<p><b>潜在问题 1：</b>必须提供搜索列名和返回列名。对于缺少的每个列，在“问题”视图中将有一个错误。</p> <p><b>解决方案：</b>编辑该函数并为 dataTable 指定正确列名。</p> <p><b>潜在问题 2：</b>带关联部件迁移中显示的潜在问题和解决方案适用。</p>

## SET map PAGE 语句

**VisualAge Generator:** SET map PAGE 同时用于显示和打印映射。

**EGL:** 使用两个独立的语句。不指定映射名：

- `clearScreen()` 用于文本（显示）表单
- `pageEject()` 用于打印表单

**迁移所需的关联部件：**需要映射以确定设备类型。任何可用映射组中的第一个具有此映射名的映射就是迁移工具所使用的映射。当在程序上下文中进行迁移时，迁移工具只查看程序的主映射组。

表 45. SET map PAGE 语句

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果该映射可用，则迁移工具将 SET map PAGE 转换为：</p> <ul style="list-style-type: none"> <li>• <code>clearScreen()</code>（对于文本表单）</li> <li>• <code>pageEject()</code>（对于打印表单）</li> </ul> <p>迁移工具还包括一个注释，该注释带有最初的映射名。</p>	<p>如果没有映射可用，则迁移工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 将 SET map PAGE 转换为 EZE_SETPAGE()。</li> <li>• 包括带有最初的映射名的注释。</li> <li>• 发出错误消息，指示无法确定映射类型。</li> </ul>
<p><b>潜在问题：</b>任何程序如果使用的映射类型与迁移函数时确定的映射类型不同，则它们在运行时的行为就可能会不同。这是因为 <code>clearScreen</code> 仅适用于文本表单，而 <code>pageEject</code> 仅适用于打印表单。在“问题”视图中不会出现任何错误。程序的生成不会失败。</p> <p><b>可能的解决方案：</b>如果特定的目标环境能够打印并且其它环境总是使用显示器映射，则将 EGL 函数更改为类似于以下的内容：</p> <pre>if (sysVar.systemType is zosbatch)   pageEject(); else   clearScreen(); end</pre> <p>根据系统的特定详细信息的不同，可以根据事务代码和用户标识等来使用类似的逻辑。</p>	<p><b>潜在问题 1：</b>如果在程序中使用了包含该语句的函数，则 EGL 验证导致“问题”视图中出现一条消息。如果在程序中未使用该函数，则“问题”视图中不会有消息。</p> <p><b>解决方案：</b>编辑该函数并根据映射类型将 EZE_SETPAGE() 更改为 <code>clearScreen()</code> 或 <code>pageEject()</code>。</p> <p><b>潜在问题 2：</b>带关联部件迁移中显示的潜在问题和解决方案适用。</p>

## SET mapItem 属性

**VisualAge Generator：**对于打印机映射中的变量和常量，VisualAge Generator 允许使用诸如保护、突出显示和颜色之类的属性。

**EGL：**除下划线以外，EGL 不支持打印表单属性。

**迁移所需的关联部件：**无。

表 46. SET mapItem 属性

带关联部件迁移	不带关联部件迁移
<p>当迁移打印机映射时，迁移工具将省略 EGL 对打印表单不支持的属性。</p> <p>当迁移函数时，迁移工具将迁移 SET 语句，而不考虑该映射是显示器映射还是打印机映射。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>

表 46. SET mapItem 属性 (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题:</b> 对于文本表单来说, 不会有问题。仅当函数包含用来对打印表单设置诸如颜色、突出显示或保护之类的属性的逻辑时, 才会发生问题。在“问题”视图中会出现错误。</p> <p><b>解决方案:</b> 如果该函数仅用于打印表单, 则修改该函数以除去 SET 语句。如果该函数既用于文本表单也用于打印表单, 则创建该函数的副本以便与打印表单配合使用。修改新函数以除去 SET 语句, 并对任何打印表单使用这个新函数。<b>缺点:</b> 这有可能反过来影响使用带有 SET 语句的函数的函数。</p>	<p><b>潜在问题:</b> 带关联部件迁移中列示的潜在问题和解决方案适用。</p>

## 检查 IN 文字或标量

**VisualAge Generator:** VisualAge Generator 支持对文字或标量中 (IN) 的数据项进行 IF 或 WHILE 语句检查。在这种情况下, VisualAge Generator 设置 EZETST 的值并执行等同性比较。

**EGL:** EGL 不支持检查数据项是否包含在文字或标量中 (IN)。

**迁移所需的关联部件:** 无。

表 47. 检查 IN 文字或标量

带关联部件迁移	不带关联部件迁移
<p>对于检查文字中 (IN) 的数据项的 IF 或 WHILE 语句, 迁移工具执行下列操作来与 VAGen 行为匹配:</p> <ul style="list-style-type: none"> <li>• 添加一个语句以将 sysVar.arrayIndex 初始化为 0。</li> <li>• 更改 if 或 while 语句以比较是否相等 (例如, if a == "b")。</li> <li>• 在 if 或 while 后面立即添加一个语句, 以将 sysVar.arrayIndex 设置为 1。</li> </ul> <p>对于检查数据项是否包含在另一个数据项中 (IN) 的 IF 或 WHILE 语句, 迁移工具不会尝试确定第二个数据项是否是数组或标量。迁移工具迁移为 EGL in 迁移。(例如: if a in b)。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p><b>潜在问题:</b> 如果比较操作是对文字进行的, 则不会有问题。仅当第二个数据项实际上是标量时, 才会发生问题。在这种情况下, 在“问题”视图中将有一个错误。</p> <p><b>解决方案:</b> 修改该函数以便在 if 或 while 语句前将 sysVar.arrayIndex 初始化为 0, 并且在 if 或 while 语句后面立即将 sysVar.arrayIndex 设置为 1。并且, 将 if 或 while 语句更改为使用 == 来进行比较, 而不是使用 in。</p>	<p><b>潜在问题:</b> 带关联部件迁移中列示的潜在问题和解决方案适用。</p>

## 检查 SQL 项和映射项是否为 NULL

**VisualAge Generator:** IF、WHILE 和 TEST 支持检查 SQL 项或映射项是否为 NULL。

**EGL:** 可以检查 SQL 项是否为 NULL。可以检查映射项是否为空白。

**迁移所需的关联部件:** 记录或映射。如果未对该项进行限定，则需要该程序及其所有关联项。

表 48. 检查 SQL 项和映射项是否为 NULL

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，如果对该项进行了限定，则迁移工具将执行下列操作：</p> <ul style="list-style-type: none"> <li>• 检查限定符以确定它是记录还是映射。</li> <li>• 如果限定符是 SQL 记录，则转换为检查 <i>null</i>。</li> <li>• 如果限定符是映射，则转换为检查空白。</li> </ul>	<p>迁移工具尝试确定项的类型，如下所示：</p> <ul style="list-style-type: none"> <li>• 如果该项被限定，并且没有限定符可用，则迁移工具执行下列操作： <ul style="list-style-type: none"> <li>– 检查该限定符是否也是该函数的 I/O 对象。如果是的话，CONVERSE 和 DISPLAY I/O 选项保证该 I/O 对象是映射。CLOSE I/O 选项对记录或映射都有效。其它 I/O 选项保证该 I/O 对象是记录。</li> <li>– 并检查函数的参数列表和本地存储器。如果找到限定符，则该限定符是记录。</li> </ul> </li> <li>• 如果迁移工具可以确定该项位于 SQL 记录中或者位于映射中，则工具迁移为： <ul style="list-style-type: none"> <li>– <i>null</i>（对于 SQL 记录）</li> <li>– 空白（对于映射项）</li> </ul> </li> <li>• 如果迁移工具不能确定该项是否位于 SQL 记录中或者位于映射中，则工具执行下列操作： <ul style="list-style-type: none"> <li>– 转换为 EZE_NULL。</li> <li>– 发出错误消息，指示您应该查看此语句。</li> </ul> </li> </ul>
<p>根据此函数的第一次迁移，如果未对该项进行限定，则迁移工具将执行下列操作：</p> <ul style="list-style-type: none"> <li>• 检查函数的参数列表，了解是否已在那里将该项指定为 SQLITEM 或 MAPITEM 参数。如果是的话，工具将以此为基础进行迁移。</li> <li>• 如果该程序及其关联项可用，则迁移工具使用 VAGen 限定规则来确定哪个记录或映射包含该项，然后以此为基础进行迁移。</li> </ul>	<p>如果未对该项进行限定，则迁移工具检查函数的参数列表以了解是否已在那里将该项指定为 SQLITEM 或 MAPITEM。</p> <p>如果迁移工具可以确定该项位于 SQL 记录中或者位于映射中，则工具迁移为：</p> <ul style="list-style-type: none"> <li>• <i>null</i>（对于 SQL 记录）</li> <li>• 空白（对于映射项）</li> </ul> <p>如果迁移工具不能确定该项是否位于 SQL 记录中或者位于映射中，则工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 转换为 EZE_NULL。</li> <li>• 发出错误消息，指示您应该查看此语句。</li> </ul>
<p><b>潜在问题:</b> 无。</p>	<p><b>潜在问题 1:</b> 如果迁移工具使用 EZE_NULL，就会发生问题。在“问题”视图中会出现错误。</p> <p><b>解决方案:</b> 编辑该函数并将 EZE_NULL 更改为 <i>null</i>（对于 SQL 项）或空白（对于表单变量字段）。</p>

## I/O 错误值 UNQ 和 DUP

**VisualAge Generator:** 对于非 SQL, UNQ 和 DUP 始终是软错误, 对于 SQL, 它们始终是硬错误。始终根据 -803 SQL 代码为 SQL 设置 UNQ 和 DUP。如果对函数指定了 I/O 错误例程, 则在下列情况下, 该错误例程获取控制权:

- 发生任何软错误
- 发生任何硬错误 (如果 EZEFECE = 1 的话)

**EGL:** Duplicate 始终是软错误, 它指示 I/O 已成功。Unique 始终是硬错误, 它指示 I/O 已失败。对于 SQL, 不支持 Duplicate。使用 *try* 块和 *onException* 语句来进行错误处理。如果对 I/O 语句指定了 *onException*, 则在下列情况下, 该 *onException* 语句获取控制权:

- 发生任何软错误
- 发生任何硬错误 (如果 *handleHardIOErrors* = 1 的话)

**迁移所需的关联部件:** 在语句中使用的记录。

表 49. I/O 错误值 UNQ 和 DUP

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移, 如果该记录可用, 则迁移工具将执行下列操作:</p> <ul style="list-style-type: none"> <li>• 如果该记录不是 SQL 记录, 则迁移工具将 DUP 更改为 <i>duplicate</i>, 并将 UNQ 更改为 <i>unique</i>。</li> <li>• 如果该记录是 SQL 记录, 则迁移工具将 DUP 和 UNQ 都更改为 <i>unique</i>。</li> </ul>	<p>如果该记录不可用, 则迁移工具尝试确定记录的类型, 如下所示:</p> <ul style="list-style-type: none"> <li>• 如果该语句与函数的 I/O 对象指定同一个记录, 则迁移工具将进行检查以了解该函数是否还有 SQL 子句或者任何特定于 SQL 的信息, 如“执行时语句构建”、单行选择、带锁定游标或者 UPDATE/SETUPD 函数。如果是的话, 迁移工具就假定该记录是 SQL 记录, 并将 DUP 和 UNQ 转换为 <i>unique</i>。</li> <li>• 在其它情况下 (例如在下列情况下), 迁移工具不能确定记录类型: <ul style="list-style-type: none"> <li>– 如果该记录被用作函数的 I/O 对象, 但该函数没有特定于 SQL 的信息。</li> <li>– 如果该记录未被用作函数的 I/O 对象。</li> </ul> </li> </ul> <p>在前面的情况下, 以及在迁移工具无法确定记录类型的其它情况下, 迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>– 将 UNQ 转换为 <i>unique</i>。</li> <li>– 将 DUP 转换为 <i>EZE_DUPLICATE</i> 并发出错误消息。</li> </ul>

表 49. I/O 错误值 UNQ 和 DUP (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题:</b> 仅当同一个记录名具有不同的定义 (即一个用于 SQL, 另一个用于非 SQL, 它们很有可能位于不同的子系统中) 时, 才会发生问题。在迁移函数时, 如果有非 SQL 记录可用, 则当该函数与 SQL 记录配合使用并检查重复时, 就会发生错误。在迁移函数时, 如果有 SQL 记录可用, 则重复检查传达的其它信息不可用于非 SQL 记录。</p> <p><b>可能的解决方案:</b> 复制该函数并将原始函数用于 SQL, 将新函数用于非 SQL。<b>缺点:</b> 这有可能反过来影响使用已检查了 UNQ 或 DUP 的原始函数的函数。</p> <p><b>SQL 的潜在问题:</b> 无。始终按照相同的方式来设置 DUP 和 UNQ, <i>unique</i> 仍然是硬错误。</p> <p><b>非 SQL 的潜在问题 1:</b> 如果未对程序设置 <code>handleHardIOErrors (EZEFEFC) = 1</code>, 就会发生问题。在这种情况下, 由于 <i>unique</i> 现在是硬错误, 所以 <code>onException</code> 语句不会获得控制权, 程序将结束。</p> <p><b>解决方案:</b> 确保程序指定 <code>handleHardIOErrors= 1</code>;</p> <p><b>非 SQL 的潜在问题 2:</b> 如果显式地测试 <code>hardIOError (HRD)</code>, 也会发生问题。在这种情况下, 由于 <i>unique</i> 现在是硬错误, 所以, 在 EGL 中, <code>hardIOError</code> 在某些情况下的测试结果为真, 即使过去它在 <i>VisualAge Generator</i> 中的测试结果不为真。验证和生成不会检测到错误。但是, 程序的运行方式可能与它过去在 <i>VisualAge Generator</i> 中的运行方式不同。</p> <p><b>可能的解决方案:</b> 可能需要重新安排程序逻辑中的 I/O 错误值测试。</p>	<p><b>潜在问题 1:</b> EZE_DUPLICATE 在 EGL 中无效。</p> <p><b>解决方案:</b> 编辑函数并根据记录类型将 EZE_DUPLICATE 更改为 <i>duplicate</i> 或 <i>unique</i>。</p> <p><b>其它潜在问题:</b> 带关联部件迁移中显示的潜在问题和解决方案也适用。</p>

## I/O 错误值 LOK

**VisualAge Generator:** 对于 OS/400® 来说, LOK 始终是软错误。如果对函数指定了 I/O 错误例程, 则在下列情况下, 该错误例程获取控制权:

- 发生任何软错误
- 发生任何硬错误 (如果 EZEFEFC = 1 的话)

**EGL:** LOK 被替换为 *deadlock*, 但这是硬错误。使用 *try* 块和 *onException* 语句来进行错误处理。如果对 I/O 语句指定了 *onException*, 则在下列情况下, 该 *onException* 语句获取控制权:

- 发生任何软错误
- 发生任何硬错误 (如果 `handleHardIOErrors = 1` 的话)

**迁移所需的关联部件:** 无。

表 50. I/O 错误值 LOK

带关联部件迁移	不带关联部件迁移
迁移工具始终将 <i>LOK</i> 更改为 <i>deadlock</i> 。	迁移工具执行带关联部件迁移列中提到的操作。
<p><b>潜在问题 1:</b> 如果未对程序设置 <code>handleHardIOErrors (EZEFECE) = 1</code>，就会发生问题。在这种情况下，由于 <i>deadlock</i> 是硬错误，所以 <code>onException</code> 语句不会获得控制权，程序将结束。</p> <p><b>解决方案:</b> 确保程序指定 <code>handleHardIOErrors= 1</code>;</p> <p><b>潜在问题 2:</b> 如果显式地测试 <code>hardIOError (HRD)</code>，也会发生问题。在这种情况下，由于 <i>deadlock</i> 是硬错误，所以，在 EGL 中，<code>hardIOError</code> 在某些情况下的测试结果为真，而在那些情况下，它在 <i>VisualAge Generator</i> 中的测试结果不为真。验证和生成不会检测到错误。但是，程序的运行方式可能与它过去在 <i>VisualAge Generator</i> 中的运行方式不同。</p> <p><b>可能的解决方案:</b> 可能需要重新安排程序逻辑中的 I/O 错误值测试。</p>	会发生带关联部件迁移列中列示的潜在问题。您可以使用相同的解决方案。

## XFER

**VisualAge Generator:** XFER 语句既可以与映射配合使用也可以与用户界面记录配合使用，它用来将输出发送给用户，然后，当用户输入输入数据时，它转移至另一个程序或事务。

**EGL:** *show* 语句用来将表单发送给终端上的用户，然后，当用户输入输入数据时，它转移至另一个程序或事务。此发行版的 EGL 不支持 Web 事务、用户界面记录或者替换带有用户界面记录的 XFER。但是，*forward* 语句可以用来替换带有用户界面记录的 XFER。*forward* 语句用来将用户界面记录发送给浏览器窗口中的用户，然后，当用户输入输入数据时，它转移至另一个程序或事务。

**迁移所需的关联部件:** 在 XFER 语句中使用的映射。如果没有具有 XFER 语句上指定的名称的映射可用，则迁移工具按照下表中不带关联部件的迁移列中描述的方式来转换 XFER 语句。

表 51. XFER

带关联部件迁移	不带关联部件迁移
<p>根据此函数的第一次迁移，工具按如下方式进行迁移:</p> <ul style="list-style-type: none"> <li>• 如果第二个自变量是映射，则工具迁移为 <i>show</i> 语句。</li> <li>• 如果第二个自变量是用户界面记录，则工具迁移为 <i>forward</i> 语句。</li> </ul>	<p>如果没有映射或用户界面记录可用，则迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>• 如果 XFER 语句包含 (NONCSP，则工具迁移为 <i>show</i> 语句。</li> <li>• 如果 XFER 的目标是 “ ”，则工具迁移为 <i>forward</i> 语句。</li> <li>• 如果第二个自变量的名称的长度超过 8 个字符，则工具迁移为 <i>forward</i> 语句，这是因为只有用户界面记录可以具有长度超过 8 个字符的名称。</li> <li>• 如果第二个自变量的名称的长度不超过 8 个字符，则工具迁移为 <i>forward</i> 语句，这是因为带有用户界面记录的 XFER 比带有映射的 XFER 更常用。工具还发出警告消息。</li> </ul>

表 51. XFER (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题:</b> 当迁移工具根据映射进行迁移时, 任何使用用户界面记录的程序都会发生问题, 反之亦然。程序的生成将失败。</p> <p><b>解决方案:</b> 如果根据映射来迁移函数, 则创建第二个函数, 这个函数包含用于用户界面记录的 forward 语句。将任何使用用户界面记录的程序更改为使用这个新函数。<b>缺点:</b> 这有可能反过来影响任何调用原始函数的函数。</p>	<p>带关联部件迁移列中列示的那些问题也会发生。您可以使用相同的解决方案。</p>

## 处理 EZE 字的不确定情况

对于某些 EZE 字替换项来说, 必须在 EGL 程序中声明附加的项变量。这个附加的项变量在函数中决不会被声明为本地项变量, 这是因为, 如果堆栈中下至带有本地存储器、参数或返回值的分段转换的位置打开了任何函数, 就不能进行分段转换。通过对程序添加附加的项变量, 可以避免中断任何分段转换。

## EZESYS

**VisualAge Generator:** EZESYS 通常是在带有由 VisualAge Generator 指定的文字值的 IF、WHILE 和 TEST 语句中指定的。但是, 在其它语句中也允许 EZESYS。

**EGL:** EGL 系统变量 *systemType* 的值与 VisualAge Generator 不同。当在除 IF、WHILE 和 TEST 以外的语句中使用 EZESYS 时, 迁移工具不了解程序所期望的值, 因此必须使用最初的 VAGen 值。EGL 系统库函数 *getVGSysType* 提供了旧的 VAGen 值。

迁移所需的关联部件: 无。

表 52. EZESYS

带关联部件迁移	不带关联部件迁移
<p>在迁移任何程序时, 如果未选择 VAGen 迁移首选项不初始化以前的 EZESYS 值, 迁移工具将执行以下操作:</p> <ul style="list-style-type: none"> <li>• 包括 <i>custPrefixEZESYS</i> 的声明。</li> <li>• 包括一个初始化语句来将 <i>custPrefixEZESYS</i> 的值设置为旧的 VAGen EZESYS 值。</li> </ul> <p>如果选择了该首选项, 则迁移工具不会包括声明或初始化语句。</p> <p><i>custPrefix</i> 是用来更改与保留字冲突的部件名的前缀。使用 VAGen 迁移语法首选项来设置它的值。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>

表 52. EZESYS (续)

带关联部件迁移	不带关联部件迁移
<p>根据该函数的第一次迁移，迁移工具将执行下列操作：</p> <ul style="list-style-type: none"> <li>• 如果在 IF、WHILE 或 TEST 语句中使用 EZESYS，则迁移工具将 EZESYS 转换为  <code>sysVar.systemType</code></li> </ul> <p>迁移工具将 EZESYS 值转换为与它们等效的 EGL 值。如果 EZESYS 值没有等效的 EGL 值，则迁移工具“按原样”迁移它。例如，迁移工具将 MVS BATCH 转换为等效的 EGL zosbatch。迁移工具将 OS2 和 NTCICS 迁移为与 VisualAge Generator 中相同的值。有关所转换的值的特定信息，请参阅第 233 页的表 104。</p> <ul style="list-style-type: none"> <li>• 如果在任何其它语句中使用 EZESYS，则迁移工具执行下列操作： <ul style="list-style-type: none"> <li>– 发出一条警告消息，指出这种用法将得到旧的 VAGen EZESYS 值</li> <li>– 使用  <code>&lt;custPrefix&gt;EZESYS</code></li> </ul>                     来替换语句中的 EZESYS。                 </li> </ul>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>

表 52. EZESYS (续)

带关联部件迁移	不带关联部件迁移
<p><b>潜在问题 1:</b> 当您在迁移时选择 VAGen 迁移首选项不初始化以前的 EZESYS 值并且在 IF、WHILE 或 TEST 之外的语句中使用 EZESYS 时，将出现问题。在此情况下，迁移工具在该语句中使用 <i>custPrefixEZESYS</i>，但程序的 <i>custPrefixEZESYS</i> 没有声明和初始化语句。在“问题”视图中会出现错误。</p> <p><b>可能的解决方案 1A:</b> 将 EGL 逻辑更改为对 <i>sysVar.systemType</i> 使用新值。</p> <p><b>可能的解决方案 1B:</b> 将 <i>custPrefixEZESYS</i> 的声明和初始化语句添加至需要对 EZESYS 使用以前的 VAGen 值的任何程序。</p> <p><b>潜在问题 2:</b> 对于按原样迁移的 EZESYS 值以及对于在此发行版中不受支持的 EGL 等效值（例如 <i>imsvs</i> 和 <i>imsbmp</i>），将会发生问题。在“问题”视图中会出现错误。</p> <p><b>可能的解决方案 2:</b> 修改该函数并更改逻辑，以便不检查 <i>sysVar.systemType</i> 是否具有在 EGL 中无效的值。</p> <p><b>潜在问题 3:</b> 如果要在除 if 和 while 以外的语句中使用新的 EGL 值，则会发生问题。</p> <p><b>可能的解决方案 3:</b> 修改该函数并更改逻辑，以使用 <i>sysVar.systemType</i> 而不是 <i>&lt;custPrefix&gt;.EZESYS</i>。</p> <p>。务必在用于进行比较的任何 <i>dataTable</i> 中将旧的 VAGen 值更改为新的 EGL 值。</p>	<p>在带关联部件迁移列中提到的潜在问题也适用。您可以使用相同的解决方案。</p>

## EZEWAIT

**VisualAge Generator:** EZEWAIT 指定等待时间（以百分之一秒计）。

**EGL:** *sysLib.wait*，它是 EZEWAIT 的替换项，用来指定等待时间（以秒计）。

**迁移所需的关联部件:** 无。

表 53. EZEWAIT

带关联部件迁移	不带关联部件迁移
<p>当迁移任何程序时，迁移工具始终包括 <i>&lt;custPrefix&gt;EZE_WAIT_TIME</i> 的声明。</p>	<p>迁移工具执行带关联部件迁移列中提到的操作。</p>
<p>当迁移函数时，如果使用了 EZEWAIT，则迁移工具包括一些逻辑来计算等待时间（以秒计）并将结果存储在 <i>&lt;custPrefix&gt;EZE_WAIT_TIME</i></p>	<p>中。迁移工具执行带关联部件迁移列中提到的操作。</p>

表 53. *EZEWAIT* (续)

带关联部件迁移	不带关联部件迁移
<p>潜在问题: 无。但是, 如果在新程序中使用该函数, 则务必在程序中包含</p> <p>&lt;custPrefix&gt;EZE_WAIT_TIME</p> <p>的声明。</p>	<p>在带关联部件迁移列中提到的潜在问题也适用。</p>



---

## 第 2 部分 从 VisualAge Generator 4.5 on Java 迁移到 EGL



---

## 第 4 章 第 1 阶段 - 从 Java 中抽取

在可以从 VisualAge Generator 中抽取源代码之前，您必须安装在 VisualAge for Java 上运行的第 1 阶段迁移工具。您还必须创建 DB2 迁移数据库，该数据库用来存储正在从 VisualAge Generator 4.5 (VAGen 4.5) 迁移到 EGL 的数据。

---

### 在 VisualAge for Java 中安装第 1 阶段迁移工具

VisualAge Generator 到 EGL 第 1 阶段迁移工具是作为名为 VAGenMigJava.exe 的自解压缩文件交付的。要安装此文件，请执行下列操作：

1. 升级到带有修订包 4 的 VisualAge Generator 4.5。您还必须安装 APAR PQ88461 以升级 VisualAge Generator 的“实用程序”功能部件。您可以与 IBM 支持机构联系以获取 APAR 的修订，也可以访问 VAGen Web 站点（网址为 <http://www.ibm.com/software/awdtools/visgen/support/>）并遵循 Download 部分中的链接。并且，请回顾第 307 页的附录 F，『不正确的外部源格式会导致创建 EGL 时发生问题的情况』以了解在特定情况下可能需要的其它 VisualAge Generator APAR。
2. 在系统上，确定 VisualAge for Java 的安装位置。
3. 关闭 VisualAge for Java。
4. 运行自解压缩文件 VAGenMigJava.exe，此文件位于 Rational Developer 或 WebSphere Studio 安装目录的以下子目录中：

\bin

**注：**如果安装您现在正在使用的产品前您已经安装并保留了先前版本的开发者产品，则您想使用的安装目录可能是先前进行安装时使用的目录。

5. 当图形用户界面提示出现时，浏览安装了 VisualAge for Java 的驱动器和目录（例如，c:\Program Files\IBM\VisualAge for Java）。然后，单击**抽取**。

当自解压缩可执行文件运行时，它将把下列文件解压缩到 VisualAge for Java 安装目录中：

- \ide\vgmigration\MigPreferences.xml
- \ide\vgmigration\VGMigReserved.txt
- \ide\vgmigration\checkStage1.sql
- \ide\vgmigration\createdatabase.sql
- \ide\vgmigration\createtables.sql
- \ide\vgmigration\checkStage1.bat
- \ide\vgmigration\SetupDatabase.bat
- \ide\vgmigration\SetupTables.bat
- \ide\vgmigration\deletemigsets.bat
- \ide\features\com-ibm-vgj-mig\

最后一个目录包含基于 Java 的第 1 阶段迁移工具的功能部件。它还包含基于 Java 的第 1 阶段迁移工具所使用的 .xml 文件以及相应的 .dtd 文件。

## 添加迁移功能部件

为了能够使用第 1 阶段迁移工具，您必须添加“IBM VisualAge Generator EGL 迁移”功能部件。要完成此任务，请执行下列步骤：

1. 启动 VisualAge Generator on Java。
2. 添加“IBM VisualAge Generator EGL 迁移”功能部件，如下所示：
  - a. 在“工作台”窗口中，按 F2 键。
  - b. 在左列中选择**功能部件**，然后在右列中选择**添加功能部件**。单击**确定**。
  - c. 选择 **IBM VisualAge Generator EGL 迁移 - versionNumber**。单击**确定**。这将装入迁移功能部件。
  - d. 单击“工作台”中的**项目**选项卡。在工作空间中，您应该会看到“IBM VisualAge Generator EGL 迁移”项目。

## 创建迁移数据库

请参阅第 309 页的『创建 DB2 迁移数据库』以了解有关创建迁移数据库的信息。您需要使用 SetupDatabase.bat 和 SetupTables.bat 文件，这两个文件位于 VisualAge Java 安装目录的 \ide\vgmigration 子目录中。

---

## 设置第 1 阶段首选项

当您在 VisualAge for Java 上安装第 1 阶段迁移工具时，安装过程在目录 *VisualAge-Java-installation-directory\ide\vgmigration* 中创建了名为 MigPreferences.xml 的样本首选项文件。在修改任何首选项之前，您应该先创建一份 MigPreferences.xml 文件副本以作备份。

您可以使用文本编辑器或第 1 阶段迁移工具附带提供的图形用户界面编辑器来编辑该 MigPreferences.xml 文件。要使用图形用户界面编辑器，请执行下列操作：

1. 启动 VisualAge Generator for Java。
2. 在“工作台”窗口中，单击**项目**选项卡。
3. 浏览到 *IBM VisualAge Generator EGL 迁移* 项目。
4. 展开迁移项目，然后展开 *com.ibm.vgj.mig* 包。
5. 在该包中，选择 **PreferencesUI** 类。
6. 右键单击 **PreferencesUI** 类，然后从上下文菜单中单击**属性**。
7. 选择**程序**选项卡。
8. 在“程序”页上，在**命令行自变量**字段中指定下列内容以指向您想要编辑的 MigPreferences.xml 文件：

```
-p filename
```

其中，*filename* 是 MigPreferences.xml 文件的驱动器、目录和文件名。

9. 单击**确定**以保存属性。
10. 右键单击 **PreferencesUI**，然后单击**运行**或**运行 main...**（也可以单击工具栏中的“跑步的人”图标）。这就打开了第 1 阶段图形用户界面首选项编辑器并装入了程序属性所指向的文件。

注：

- 对于需要驱动器和目录的首选项，可以采用下列两种方法中的任何一种方法来指定信息：
  - 绝对路径。例如：d:\tempMig\MySystem\
  - 相对路径。在这种情况下，路径是相对于工作目录的。例如，..\tempMig\MySystem 将产生以下路径：
 

```
VisualAge-Java-installation-directory\ide\project_resources
  \IBM VisualAge Generator EGL Migration\tempMig\MySystem.
```
- 如果您未对日志文件、调试文件和报告文件指定驱动器和目录，则文件将写入以下工作目录：
 

```
VisualAge-Java-installation-directory\ide\project_resources
  \IBM VisualAge Generator EGL Migration
```

下列各节根据图形用户界面中显示首选项的页面来描述您可以修改的首选项：

- 构建规划
- 映射
- 重命名
- 执行

## “构建规划” 页

“构建规划” 页标识第 1 阶段迁移工具读取或写入迁移规划文件的位置以及要从资源库迁移的项目和版本。

- **迁移说明。**“迁移说明” 标识迁移工具写迁移规划文件的位置，该迁移规划文件是第 1 阶段工具根据资源库过滤器创建的。另外，如果您已经创建了迁移规划文件，则“迁移说明” 标识的是迁移工具读取迁移规划文件的位置。

### 注：

- 迁移规划文件在它们被用来装入迁移数据库之前具有文件扩展名 *.pln*，在它们被成功处理之后具有文件扩展名 *.done*。
- 要了解有关为 `VAGenToEGLMigration` 类（这是实际的第 1 阶段迁移工具）设置 `-o`（覆盖）选项的信息，请参阅第 97 页的『运行第 1 阶段工具』。
- **规划目录。**这是目标目录，第 1 阶段迁移工具将把迁移规划文件放在此目录中，第 1 阶段工具也可以在此目录中找到现有的迁移规划文件。
- **规划文件名。**您正在创建的或者正在用来装入迁移数据库的迁移规划文件的可选文件名。当运行第 1 阶段迁移工具时，此文件名将与您对 `VAGenToEGLMigration` 类指定的 `-o`（覆盖）选项结合使用，如下所示：
  - 如果将 `-o` 选项包括在 `VAGenToEGLMigration` 类的属性中，则第 1 阶段迁移工具根据您在“迁移说明”中指定的文件名来执行下列操作：
    - 如果您未指定规划文件名，则迁移工具在创建新的规划文件之前将删除指定规划目录中的所有 *.pln* 文件。迁移工具为每个迁移集创建一个规划文件。在这种情况下，迁移规划文件名的格式为 *migrationSetName\_version.pln*。
    - 如果您指定了规划文件名，则迁移工具在创建具有指定规划文件名的新 *.pln* 文件之前，**只**从指定的规划目录中删除指定的 *.pln* 文件。在这种情况下，单个规划文件列示所有迁移集。

如果要让第 1 阶段迁移工具根据资源库过滤器来创建迁移规划文件，请使用 `-o` 选项。

- 如果在 VAGenToEGLMigration 类的属性中省略 `-o` 选项，则第 1 阶段迁移工具不创建任何新的迁移规划文件。而是，第 1 阶段迁移工具根据您在“迁移说明”中指定的规划目录和规划文件名来运行：
  - 如果您未指定规划文件名，则迁移工具使用指定的规划目录中的所有 `.pln` 文件来运行。
  - 如果您指定了规划文件名，则迁移工具仅使用指定的规划目录中的那一个 `.pln` 文件来运行。

如果您先前已创建了迁移规划文件，并且现在想要运行第 1 阶段迁移工具以使用这些文件来装入迁移数据库，则省略 `-o` 选项。要了解有关创建您自己的迁移规划文件的详细信息，请参阅第 100 页的『手工创建迁移规划文件』。

- **资源库过滤器。**“资源库过滤器”部分使您能够控制第 1 阶段迁移工具在 Java 资源库中所要考虑的项目和版本。通过限制项目和版本，可以大大提高第 1 阶段迁移工具的性能。您可以指定多个过滤器。第 1 阶段迁移工具按如下方式使用项目过滤器以及版本深度或版本名过滤器：

- 迁移工具将资源库中的每个 VAGen 项目与项目过滤器进行匹配。
  - 如果项目名至少与一个项目过滤器不匹配，则不考虑对该项目进行进一步处理。
  - 如果项目名至少与一个项目过滤器相匹配，则对该项目的版本进行如下处理：
    - 如果您选择了版本深度过滤器，则考虑对项目的最新版本进行进一步处理，并且数目可达版本深度过滤器指定的版本数。缺省的版本深度过滤器是 1。
    - 如果您选择了版本名过滤器，则将项目的每个版本名与版本名过滤器列表进行匹配。如果版本名与任何版本名过滤器匹配，则考虑对该版本进行进一步处理。

**注：**版本深度和版本名是互斥的。缺省情况下，版本名过滤器已包括在 `MigPreferences.xml` 文件中。如果要使用版本深度过滤器，则请选择“版本深度”单选按钮并指定要迁移的版本数。

- 如果项目名和版本名导致要考虑对该项目版本进行进一步处理，则第 1 阶段迁移工具执行下列操作：
  - 如果该项目版本是高级 PLP 项目，则第 1 阶段迁移工具将该项目版本用作创建迁移集的基础。假定版本名与版本过滤器匹配，那么每个版本的高级 PLP 项目都会生成不同的迁移集。
  - 如果该项目版本不是高级 PLP 项目，则不考虑对该项目版本进行进一步处理。该项目版本仍可能会包括在其它迁移集中；并没有专门用于该项目版本的迁移集。

指定“资源库过滤器”信息，如下所示：

- **项目过滤器。**迁移工具将资源库中的项目名与您指定的项目过滤器进行匹配。您可以指定多个项目过滤器。要添加或删除过滤器，请使用“添加”和“除去”按钮。要更新过滤器，请在表中进行改写。过滤器是不区分大小写的。您可以使用通配符，如下所示：
  - 项目过滤器 `*xyz*` 与资源库中名称的任何位置包含字符串“xyz”的任何项目名匹配。
  - 项目过滤器 `xyz*` 与资源库中任何以“xyz”开头的项目名匹配。
  - 项目过滤器 `*xyz` 与资源库中任何以“xyz”结尾的项目名匹配。

- 版本深度过滤器。如果项目名与其中一个项目过滤器匹配，并且您已选择版本深度过滤器，则第 1 阶段迁移工具处理的版本数就是您对版本深度指定的数目。缺省值是 1，在这种情况下，第 1 阶段迁移工具仅处理项目的最新版本。
- 版本名过滤器。如果项目名与其中一个项目过滤器匹配，并且您已选择版本名过滤器，则第 1 阶段迁移工具使用版本名过滤器来确定应该考虑对哪些项目版本（如果有的话）进行迁移。您可以指定多个版本名过滤器。要添加或删除过滤器，请使用“添加”和“除去”按钮。要更新过滤器，请在表中进行改写。过滤器是不区分大小写的。您可以使用通配符，如下所示：
  - 版本名过滤器 \*xyz\* 与版本名中任何位置包含字符串“xyz”的任何项目版本名匹配。
  - 版本名过滤器 xyz\* 与任何以“xyz”开头的项目版本名匹配。
  - 版本名过滤器 \*xyz 与以“xyz”结尾的项目版本名匹配。

## “映射”页

“映射”页使您能够控制部件在 EGL 文件中的放置以及迁移期间创建的一些 EGL 项目、包和文件的名称。

- **文件名。**“文件名”部分使您能够控制迁移期间创建的两个 EGL 文件的名称。
  - 公共部件使您能够指定一个 EGL 文件的名称，该文件包含迁移集作用域中多个唯一的可生成部件的公共部件。请指定不带扩展名或路径的文件名。迁移工具将在每个 EGL 包中创建一个公共部件文件，该文件包含由迁移集中多个可生成部件使用（与它们相关联）的部件，或者包含已被标识为公共项目或包的 VAGen 项目或包中的部件。要了解有关是将部件与程序放在一起还是放入公共部件文件的详细信息，请参阅第 30 页的『将部件放到 EGL 文件中』。
  - 未使用的部件使您能够指定一个 EGL 文件的名称，该文件包含迁移集作用域中未使用的部件。请指定不带扩展名或路径的文件名。迁移工具将在每个 EGL 包中创建一个未使用部件文件，该文件包含未被迁移集中任何可生成部件使用（与它们相关联）的部件（如果相应的 VAGen 项目和包未被标识为公共项目或包的话）。
- **跨映射。**“跨映射”部分使您能够指定一些后缀，当其中一个映射组包含来自多个项目或包的映射时，将使用这些后缀。
  - 项目后缀使您能够指定一个后缀，第 1 阶段迁移工具将该后缀与迁移集名称连接到一起以创建新的 EGL 项目名。仅当映射组及其映射位于迁移集中的多个 VAGen 项目中时，迁移工具才创建这个新的 EGL 项目。这个新项目的名称是 *migrationSetName\_ProjectSuffix*。
  - 包后缀使您能够指定一个后缀，第 1 阶段迁移工具将该后缀与项目名连接到一起以便在 EGL 项目中创建新的 EGL 包名。仅当映射组及其映射位于项目中的多个 VAGen 包中时，迁移工具才创建这个新的 EGL 包。这个新包的名称是 *packageName\_PackageSuffix*。
- **公共标识。**“公共标识”部分使您能够指定带通配符的字符串列表，迁移工具可以使用此字符串列表来确定哪些 VAGen 项目和包包含公共（共享）部件。
  - 项目列表使您能够指定一个字符串列表，该字符串列表标识包含公共部件的项目。迁移工具将这个字符串列表与迁移集中的每个项目名进行匹配以确定该项目是否包含公共部件。如果任何字符串与项目名匹配，则该项目中的所有部件都将认为“被使用”。每个不可生成的部件都将放到程序文件中或由“公共部件”首选项指定的文件中。即使某个部件未被迁移集中的任何可生成部件使用，该部件也不会放到未使用部件文件中。您可以指定多个项目过滤器。要添加或删除过滤

器，请使用“添加”和“除去”按钮。要更新过滤器，请在表中进行改写。过滤器是不区分大小写的。您也可以在字符串的开头或末尾使用 \* 作为通配符。

- 包列表使您能够指定一个字符串列表，该字符串列表标识包含公共部件的包。迁移工具将此字符串列表与迁移集中的每个包名进行匹配以确定该包是否包含公共部件。如果任何字符串与包名匹配，则认为该包中的所有部件都将“被使用”。每个不可生成的部件都将放到程序文件中或由“公共部件”首选项指定的文件中。即使某个部件未被迁移集中的任何可生成部件使用，该部件也不会放到未使用部件文件中。您可以指定多个包过滤器。要添加或除去过滤器，请使用“添加”和“除去”按钮。要更新过滤器，请在表中进行改写。过滤器是不区分大小写的。您也可以在字符串的开头或末尾使用 \* 作为通配符。

## “重命名”页

“重命名”页使您能够指定项目名、包名和版本名的重命名规则。重命名规则部分使您能够控制根据 VAGen 项目名和包名派生的 EGL 项目名和包名。“顺序”列中的数字指示了第 1 阶段迁移工具应用重命名规则的顺序，其中，编号最小的规则将首先被应用。要添加或除去重命名规则，请使用“添加”和“除去”按钮。要更新重命名规则，请改写表中单元格的内容。您可以双击任何列标题以根据该列来对规则进行排序。通过指定下列信息来指定规则：

- 顺序指定规则的应用顺序。
- 源字符串指定 VAGen 名称中要更改的字符。
- 目标字符串指定要在得到的 EGL 名称中使用的字符。
- 字符串上下文指定 VAGen 名称中的位置，在重命名期间，迁移工具应该在该位置查找源字符串。值如下所示：
  - 前面表示如果源字符串出现在项目名、包名或版本名的开头，则应用该规则。
  - 后面表示如果源字符串出现在项目名、包名或版本名的末尾，则应用该规则。
  - 任何表示如果源字符串出现在项目名、包名或版本名中的任何位置，则应用该规则。
  - 标记表示仅当源字符串是项目名、包名或版本名的精确匹配时，才应用该规则。
- 映射上下文指示迁移工具是否将重命名规则应用于项目名、包名或版本名。映射上下文值如下所示：
  - 项目表示重命名规则仅应用于 VAGen 项目名。
  - 包表示重命名规则仅应用于 VAGen 包名。
  - 两者表示重命名规则既应用于 VAGen 项目名也应用于 VAGen 包名。
  - 版本表示重命名规则应用于所有项目名的版本名。如果版本名包含目录名或文件名所不允许的特殊字符，如分号 (;)，则使用版本重命名规则。缺省的 *MigPreferences.xml* 文件包含数个版本重命名规则，它们用来帮助确保版本名不会生成无效的目录名或文件名。在第 1 阶段迁移中，迁移工具使用已重命名的版本来创建迁移规划文件名，在第 3 阶段，迁移工具使用该版本来创建目录名。

## “执行”页

- 执行选项。“执行选项”部分使您能够指定要让第 1 阶段迁移工具执行的操作。
  - 生成报告指定要创建一个迁移报告，该报告显示了每个部件将放到 EGL 项目、包和文件结构中的什么位置中。对于查看您对公共部件文件名和未使用部件文件名、跨映射后缀、项目和包公共标识以及重命名规则指定的首选项的结果来说，

此报告很有用。如果您选择了“生成报告”，则迁移工具将在您对“验证”部分中的“报告文件名”指定的驱动器、目录和文件中创建报告。

- 更新数据库指定要让第 1 阶段迁移工具将迁移规划信息（包括部件的“外部源格式”）存储到迁移数据库中。

可以通过几个步骤来运行第 1 阶段迁移工具，如下所示：

- 步骤 1 - 取消选择“生成报告”和“更新数据库”。这使您能够查看已创建的迁移规划文件并确保“资源库过滤器”具有正确的设置并且将处理您想要处理的项目版本。如果您对选择的项目版本不满意，则可以优化“资源库过滤器”并再次运行此步骤，直到您对迁移工具将要处理的项目版本感到满意为止。
- 步骤 2 - 仅选择“生成报告”。这使您能够查看一个报告，该报告显示在迁移期间将如何把 VAGen 项目、包和部件分配到 EGL 项目、包和文件中。如果您对部件的放置位置不满意，则可以优化“映射”和“重命名规则”并再次运行该报告，直到您对部件的放置位置感到满意为止。
- 步骤 3 - 选择“生成报告”和“更新数据库”。这将提供一个最终报告，该报告记录了迁移数据库中存储的信息。

**注：**

- 生成该报告要花费一些时间。因此，您最好查看 .pln 文件以确保迁移工具将处理您想要处理的项目版本。
  - 如果生成报告，则报告文件将被覆盖。如果要保存先前的报告文件，则必须将报告文件移至另一个目录中，或者将新报告生成到新目录中。因为报告文件链接至其它文件，所以，将报告文件重命名会导致链接失效，从而导致文件不再可供查看。
- **数据库。**“数据库”部分使您能够指定关于迁移数据库的详细信息：
    - **数据库驱动程序。**此值始终应该是 `COM.ibm.db2.jdbc.app.DB2Driver`。
    - **数据库名称。**此值始终应该是下列其中一个值：
      - `jdbc:DB2:databaseName`（如果正在使用本地数据库）。
      - `jdbc:nodeName:databaseName`（如果正在使用远程数据库）。

**注：**在这两种情况下，`databaseName` 都是迁移数据库的名称，迁移工具将把迁移集信息写入该迁移数据库。缺省情况下，`databaseName` 是 `VGMIG`。如果您在创建迁移数据库时将数据库名称更改为除 `VGMIG` 以外的名称，则必须将此首选项指定的数据库名称更改为与使用的名称匹配。

- **模式**是用作数据库表限定符的名称。缺省情况下，模式名是 `MIGSCHEMA`。如果您在创建迁移数据库时将模式名更改为除 `MIGSCHEMA` 以外的名称，则必须将此首选项指定的模式名更改为与使用的名称匹配。
- **用户标识**是连接至迁移数据库时所需的用户标识。如果您未指定用户标识，则迁移工具将尝试使用您的登录用户标识来进行连接。如果此尝试失败，则迁移工具将显示一个对话框窗口来要求输入该信息。
- **密码**是连接至迁移数据库时所需的密码。如果您未指定密码，则迁移工具将尝试使用您的登录密码来进行连接。如果此尝试失败，则迁移工具将显示一个对话框窗口来要求输入该信息。

**注：**在首选项文件中，密码不是以加密形式存储的。如果对此感到担忧，则不要在首选项文件中输入密码。并且等待提示。

- **服务。**“服务”部分使您能够指定有关在第 1 阶段要捕获的记录信息和调试信息的详细信息。您可以指定下列信息：
  - **跟踪级别**使您能够指定要写入调试文件的信息的级别。使用下拉列表来指定下列其中一个值：
    1. 记录致命错误消息。如果出现任何这些消息，则表示可能已更新了迁移数据库，但未将迁移规划文件（.pln 文件）更改为具有 .done 文件扩展名。这使您能够重新处理该 .pln 文件。
    2. 记录警告消息以及致命错误消息。
    3. 记录参考消息以及警告和致命错误消息。
    4. 记录调试信息以及参考、警告和致命错误消息。“调试”是唯一一个会导致迁移工具将信息写入调试文件的跟踪级别。

跟踪级别仅影响日志和调试文件。所有消息都将写至“控制台”窗口。
  - **日志文件名**使您能够对日志文件指定驱动器、目录和文件名。您可以创建具有任何文件扩展名的日志文件，但最好将其作为 .xml 文件来进行查看。如果您省略了日志文件名，则迁移工具将把日志信息写至名为 *miglog.xml* 文件，该文件位于“日志文件名”字段中指定的驱动器和目录中。如果您未指定日志文件驱动器和目录，则迁移工具将把日志文件写至工作目录。
  - **调试文件名**使您能够对 IBM 支持机构可能需要的调试文件指定驱动器、目录和文件名。您可以创建具有任何文件扩展名的调试文件，但最好将其作为 .xml 文件来进行查看。仅当“跟踪级别”首选项设置为调试时，才会将信息写入此文件。如果您省略了调试文件名并指定了调试跟踪级别，则迁移工具将把调试文件信息写入文件 *migdebug.xml*，该文件位于“调试文件名”字段中指定的驱动器和目录中。如果您未指定调试文件驱动器和目录，则迁移工具将把调试文件写至工作目录。
- **验证。**“验证”部分使您能够对验证报告指定驱动器、目录和文件名，该验证报告是当您在“执行选项”部分中选择了“生成报告”首选项时生成的。如果您选择了“生成报告”，则必须输入报告文件名。您始终应该指定 .htm 扩展名。如果您未指定驱动器和目录，则迁移工具将把报告文件写至工作目录。

## 样本 MigPreferences.xml 文件

以下是一个样本 MigPreferences.xml 文件：

```
<preferences>
  <database>
    <driver>COM.ibm.db2.jdbc.app.DB2Driver</driver>
    <uri>jdbc:DB2:VGMIG</uri>
    <schema>MIGSCHEMA</schema>
    <userid></userid>
    <password></password>
  </database>
  <migrationSpec>
    <directory>d:\tempMig\MyMigSet</directory>
    <filename></filename>
  </migrationSpec>
  <repositoryFilters>
    <projectName>MyProject*</projectName>
    <versionName></versionName>
  </repositoryFilters>
  <service>
    <tracelevel>4</tracelevel>
    <debugfile>d:\tempMig\MyMigSet\Stage1\migdebug.xml</debugfile>
    <logfile>d:\tempMig\MyMigSet\Stage1\miglog.xml</logfile>
  </service>
</preferences>
```

```

<eglMapping>
  <renameRule order = "1">
    <fromString> </fromString>
    <toString></toString>
    <stringContext>any</stringContext>
    <mappingContext>both</mappingContext>
  </renameRule>
  <renameRule order = "101">
    <fromString>Project</fromString>
    <toString></toString>
    <stringContext>any</stringContext>
    <mappingContext>project</mappingContext>
  </renameRule>
  <renameRule order = "301">
    <fromString>.pkg</fromString>
    <toString></toString>
    <stringContext>any</stringContext>
    <mappingContext>package</mappingContext>
  </renameRule>
  <renameRule order = "302">
    <fromString>.sql</fromString>
    <toString>sql</toString>
    <stringContext>any</stringContext>
    <mappingContext>package</mappingContext>
  </renameRule>
  <renameRule order = "501">
    <fromString>:</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>
  <renameRule order = "502">
    <fromString>/</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>
  <renameRule order = "503">
    <fromString>\</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>
  <renameRule order = "504">
    <fromString>|</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>
  <renameRule order = "505">
    <fromString>?</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>
  <renameRule order = "506">
    <fromString>*</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>
  <renameRule order = "507">
    <fromString>&lt;</fromString>
    <toString>_</toString>
    <stringContext>any</stringContext>
    <mappingContext>version</mappingContext>
  </renameRule>

```

```

<renameRule order = "508">
  <fromString>&gt;</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order = "509">
  <fromString>&quot;</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order = "510">
  <fromString> </fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<verification>
  <generateReport>true</generateReport>
  <reportName>d:\tempMig\MyMigSet\report\MyReport.htm</reportName>
</verification>
<dbUpdate>true</dbUpdate>
<spanningMapsProjectSuffix>MapsProject</spanningMapsProjectSuffix>
<spanningMapsPackageSuffix>mapspackage</spanningMapsPackageSuffix>
<commonPartsFileName>CommonParts</commonPartsFileName>
<unusedPartsFileName>UnusedParts</unusedPartsFileName>
<commonParts>
  <commonProject>*Common*</commonProject>
  <commonPackage>*common*</commonPackage>
</commonParts>
</eglMapping>
</preferences>

```

---

## 在运行第 1 阶段工具之前 - 提示与技巧

在运行第 1 阶段迁移工具之前，您可能想执行一些操作以提高性能。您可能还想保存现有的工作空间以便在迁移完成后使用。

### 提高性能

性能度量显示在开始时使用干净的工作空间可以显著提高第 1 阶段迁移工具的性能。在我们进行的一系列测试中，与不使用干净的工作空间相比，在开始时使用干净的工作空间能够使第 1 阶段的耗时缩短 25% - 30%。如果现有的工作空间大于 20 兆字节，则开始时使用干净的工作空间有助于提高第 1 阶段工具的性能。

要在开始时使用干净的工作空间，请执行下列操作：

1. 关闭 VisualAge Generator。
2. 如果要保留现有工作空间的备份副本以便在迁移完成后使用，请参阅第 97 页的『保存工作空间』。
3. 从以下 VisualAge Generator 下载站点获取一份干净的工作空间（文件名为 `ide.icx`）：  
<ftp://ftp.software.ibm.com/ps/products/visualagegen/fixes/v4.5/FixPack4/windows>
4. 删除 `features.sav` 和 `projects.sav` 文件。
5. 重新启动 VisualAge Generator。
6. 添加所需的 VisualAge Generator 功能部件。
7. 添加“IBM VisualAge Generator EGL 迁移”功能部件。

## 8. 关闭 VisualAge Generator。

为了缩短第 1 阶段迁移工具在分析要迁移哪些项目和版本时花费的时间，您可以考虑创建一个仅包含要迁移的项目版本的资源库。进行迁移时，如果在 VisualAge Generator 中正在进行维护工作，则使用独立的迁移资源库还具有下列优点：

- 要迁移的项目版本集是固定的。如果使用“版本深度”首选项来控制所要迁移的内容，这一点就尤其重要。
- 您可以将新的迁移资源库中的版本与维护资源库进行比较以确定还需要迁移哪些其它项目版本。

如果您创建了一个特殊的资源库，则可以考虑将其用作本地资源库以提高第 1 阶段迁移性能。

## 保存工作空间

在第 1 阶段处理开始和结束时，第 1 阶段迁移工具将从工作空间中删除所有包含 VAGen 部件的项目。这有助于避免工作空间包含重复的部件，并且能够确保在第 1 阶段仅考虑将迁移集中的部件包括在关联部件列表中。如果您希望保存工作空间，则应该在运行第 1 阶段工具之前执行下列操作：

1. 关闭 VisualAge Generator。
2. 将下列文件的备份副本保存到 `\VisualAgeForJava-installation-directory\ide\program` 中：
  - features.sav
  - projects.sav
  - ide.icx
  - ide.ini - 如果在运行第 1 阶段期间未更改任何首选项，则不必保存此文件
  - hpt.ini - 如果在运行第 1 阶段期间未更改任何首选项，则不必保存此文件
3. 启动 VisualAge Generator。

当完成运行第 1 阶段工具时，请执行下列操作以复原工作空间：

1. 关闭 VisualAge Generator。
2. 在运行第 1 阶段工具之前，先复原已备份的文件。
3. 启动 VisualAge Generator。

---

## 运行第 1 阶段工具

在完成编辑首选项之后，您就准备好运行第 1 阶段迁移工具以从 Java 资源库中抽取源代码了。要完成此任务，请执行下列步骤：

1. 浏览到 *IBM VisualAge Generator EGL* 迁移项目。
2. 展开迁移项目，然后展开 *com.ibm.vgj.mig* 包。
3. 在该包中，选择 **VAGenToEGLMigration** 类。
4. 右键单击 **VAGenToEGLMigration** 类，然后从上下文菜单中单击**属性**。
5. 选择**程序选项卡**。
6. 在“程序”页上，在**命令行自变量**字段中指定下列内容以指向您想要编辑的 `MigPreferences.xml` 文件：

表 54. VAGentoEGLMigration 类的有效命令行选项

选项	选项的含义
-h	显示帮助信息，该信息显示了有效的选项
-p filename	使用“filename”作为首选项文件名。必须对文件名进行完全限定（包括驱动器和目录）。
-o	如果存在迁移文件，则覆盖并重新创建它们。

7. 如果这是您第一次运行第 1 阶段工具，则执行下列操作：
  - a. 在同一个“属性”窗口中，选择**类路径**选项卡。
  - b. 在“类路径”页上，选择**其它目录路径**复选框，然后单击**编辑**按钮以编辑其它目录。
  - c. 选择**添加 Jar/Zip**按钮。
  - d. 在“文件选择”窗口中，浏览到 db2java.zip 文件并选择该文件。
    - 如果安装 DB2 时使用了缺省安装目录，则该文件应该位于 \SQLLIB\java 目录中。

在选择 db2java.zip 文件之后，文件名将出现在“其它目录”窗口中。在“其它目录”窗口中单击**确定**。
  - e. 在“类路径”页上，单击**立即计算**按钮，然后，在得到提示时单击**是**。
8. 单击**确定**以保存属性。
9. 右键单击 **VAGenToEGLMigration**，然后单击**运行或运行 main...**（也可以从工具栏中选择“跑步的人”图标）。这就启动了第 1 阶段迁移工具并打开了“控制台”窗口，该工具在该窗口中报告进度以及任何错误消息。迁移工具还将把消息写至迁移首选项中指定的日志文件。

当第 1 阶段迁移工具完成时，如果您已选择了“更新数据库”首选项，则迁移规划信息（包括具有外部源格式的 VAGen 代码）将存储在迁移数据库中。在查看报告和第 1 阶段消息之后，您可能会决定在 VisualAge Generator 中对代码进行更改并再次运行第 1 阶段。当您对第 1 阶段的结果感到满意并将最终的外部源格式代码存储在迁移数据库中之后，您就已准备好执行第 2 阶段迁移了。要运行第 2 阶段迁移工具，请使用 EGL 开发环境。要了解有关继续执行迁移过程的信息，请参阅第 123 页的第 6 章，『第 2 阶段 - 转换为 EGL 语法』。

## 迁移规划和高级 PLP 项目

迁移规划文件仅仅是一个 XML 文件，它指定了一个或多个迁移集的名称，并且，对于每个迁移集，它指定了构成该迁移集的项目名和版本列表。第 1 阶段迁移工具已被设计成根据项目的 repositoryFilter 首选项以及版本名来自动创建迁移规划文件。第 1 阶段工具使用这些过滤器来确定是否应该查看项目版本以确定该项目版本是否是高级 PLP 项目。第 1 阶段工具将每个高级 PLP 项目版本用作迁移集的基础。

如果生成 VAGen 源代码时使用了 PLP 项目，则这些 PLP 项目就是应该用于迁移的 PLP 项目。这是因为，PLP 项目提供了在生成期间一起使用的部件分组，因此具有一组程序的所有关联部件。

如果当前未使用 PLP 项目，则可以执行下列其中一项操作：

- 创建高级 PLP 项目，该项目指定要成组迁移的项目版本列表。然后，您可以使用第 1 阶段迁移工具来自动创建迁移规划。
- 如果您不愿意创建高级 PLP 项目，则可以使用下列其中一种技术来自行创建迁移规划文件：
  - 如果数据库或其它系统中的信息指定了特定的 Java 项目版本进行生成所需的内容，则您可以编写一个工具来根据数据库自动创建迁移规划文件。
  - 手工创建迁移规划文件。

## 创建高级 PLP 项目

**注：**如果项目名或版本名包含 DBCS 字符，则 VisualAge Generator 不支持 PLP。如果项目名或版本名包含 DBCS 字符，则请参阅第 100 页的『手工创建迁移规划文件』以获取有关如何在不使用 PLP 的情况下创建迁移规划文件的信息。

要创建高级 PLP 项目以便在迁移期间使用，请在 VisualAge Generator 中执行下列操作：

1. 从“工作台”窗口中，选择**项目**选项卡。
2. 创建新的 Java 项目以包含项目列表部件。例如，创建名为 MySubsystem1 的项目。
3. 选择新项目，单击鼠标右键并从上下文菜单中选择**管理 -> 配置 VAGen 必需项目**。
4. 在“配置 VAGen 必需项目”窗口中，选择要包括在迁移集中的每个项目。对于每个项目，可以选择特定版本以将其包括在迁移集中。另外，可以选择**最新编辑**，这将导致在迁移期间每次使用此项目时，迁移工具都自动包括当前位于列表顶部的版本。
5. 在选择了迁移集所需的所有项目版本之后，单击**确定**。
6. 设置高级 PLP 项目的版本并进行发布，例如发布为 MySubsystem1。
7. 测试该 PLP 项目是否正确地装入了您所要的迁移集项目版本，如下所示：
  - a. 从工作空间中删除该高级 PLP 项目。
  - b. 单击**所选 -> 添加 -> 项目**。
  - c. 在“添加项目”窗口中，请执行下列操作：
    - 1) 选择**从资源库中添加项目**。
    - 2) 选择刚刚创建的高级 PLP 项目和已创建的版本。
    - 3) 并且，选择**添加 VAGen 必需项目**。
    - 4) 单击**完成**。
  - d. 这就将把该高级 PLP 项目以及它指定的所有项目版本添加到工作空间中。
  - e. 从“VAGen 部件浏览器”中，选择**工具 -> 显示重复部件**。在列表中不应该有任何部件。如果列表包含部件，则需要更改该高级 PLP 项目，以使重复部件不存在。
  - f. 您可能想要为程序和表运行验证以确保它们在 VAGen 中有效并确保不缺少任何部件。

可以将 PLP 项目链接起来。例如，创建一个 PLP 项目，该项目列示所有公共项目的项目版本。然后，对于每个子系统，为该子系统创建一个高级 PLP 项目（它包含所有特定于该子系统的项目版本）和一个 PLP 项目（它指定所有公共项目版本）。这样，就不需要在每个子系统的高级 PLP 项目中列示每一个公共项目版本了。

当您准备好运行第 1 阶段迁移工具时，请执行下列操作：

- 在设置第 1 阶段首选项时，在**构建规划**页上的“资源库过滤器”部分中，设置“项目”列表，以使列表中的一个过滤器与已创建的高级 PLP 项目匹配。
- 当向第 1 阶段工具指示所要使用的首选项文件时，还需要指定 `-o` 选项。`-o` 选项指示第 1 阶段迁移工具根据高级 PLP 项目来创建迁移规划文件并覆盖任何现有迁移规划文件。

## 手工创建迁移规划文件

如果已有一些外部控件，它们可用来确定在 VisualAge Generator 中进行生成时要将哪些项目版本添加至工作空间，则您可能会决定手工创建迁移规划文件或者开发一个工具来根据外部信息自动创建迁移规划文件。迁移规划文件必须具有 `.pln` 文件扩展名和以下格式：

```
<migrationDefinition>
  <migrationSet name="migrationSet1" version="migrationSet1Version1"
    vgName="migrationSet1" vgVersion="migrationSet1Version1">
    <project name="projectName1" version="projectName1Version1"></project>
    <project name="projectName2" version="projectName2Version1"></project>
    .
    .
    <project name="projectNameN" version="projectNameNVersion1"></project>
  </migrationSet>
  <migrationSet name="migrationSet2" version="1.1"
    vgName="migrationSet2" vgVersion="1.1">
    <project name="projectNameA" version="projectNameAVersion1"></project>
    <project name="projectNameB" version="projectNameBVersion1"></project>
    .
    .
    <project name="projectNameZ" version="projectNameZVersion1"></project>
  </migrationSet>
</migrationDefinition>
```

在上一个示例中，下列陈述适用：

- `migrationSet1` 是一个名称，它可以用来引用必须一起迁移的一组项目。迁移集名称存储在迁移数据库中，在以后的迁移阶段，它的用途如下：
  - 在第 1 阶段迁移中，如果映射组中的映射跨多个项目，则使用带有后缀的迁移集名称来为包含该映射组及其所有映射的新 EGL 项目构建名称。如果您更改了重命名规则，则迁移集名称还用来从迁移数据库中除去信息。
  - 在第 2 阶段迁移中，迁移集名称指定要将迁移数据库中的哪一组项目转换为 EGL。
  - 在第 3 阶段，迁移集名称指定要使用迁移数据库中的哪一组项目来在工作空间或临时目录中创建 EGL 项目、包和文件。如果您选择将第 3 阶段的输出保存至临时目录，则迁移集名称和迁移集版本还用来创建高层目录名。

在迁移期间，迁移集名称仅用来标识一组项目。除了映射跨 VisualAge Generator 中的多个项目的情况以外，在迁移之后就不再使用迁移集名称了。

- `projectName1`、`projectName2`、... 和 `projectNameN` 是要成组迁移的项目。在迁移集中，必须仅将 `projectName` 列示一次。迁移工具将把同一个迁移集下列示的所有项目版本装入工作空间并成组地处理它们。
- `projectName1Version1`、`projectName2Version1`、... 和 `projectNameNVersion1` 是这些项目中的每个项目的对应版本。对于迁移集中的每个项目，只能指定一个版本。

- 您指定的项目名和版本名必须与资源库中的项目名和版本名精确匹配。这些名称是区分大小写的。该信息用来将项目版本添加至工作空间，以便可以对部件进行分析以构建第 1 阶段迁移报告以及装入数据库。

您可以构建仅包含一个迁移集的迁移规划文件。另外，可以通过对每个迁移集重复 `<migrationSet>` 与 `</migrationSet>` 标记之间的信息来构建包含数个迁移集的迁移规划文件。

当您准备好运行第 1 阶段迁移工具时，请执行下列操作：

- 在设置第 1 阶段首选项时，在**构建规划**选项卡上，将**规划目录名**设置为用来存储迁移规划文件的驱动器和目录。如果要让第 1 阶段迁移工具仅运行一个已创建的迁移规划，请指定**规划文件名**。如果要让第 1 阶段迁移工具使用指定**规划目录**中的所有迁移规划文件来运行，则保留**规划文件名**为空白。
- 当向第 1 阶段工具指示所要使用的首选项文件时，务必省略 `-o` 选项。通过省略 `-o` 选项，您向第 1 阶段工具指示要使用现有的迁移规划文件。即，迁移工具不创建任何新的迁移规划文件。



---

## 第 3 部分 从 VisualAge Generator 4.5 on Smalltalk 迁移到 EGL



---

## 第 5 章 第 1 阶段 - 从 Smalltalk 中抽取

在可以从 VisualAge Generator 中抽取信息之前，您必须安装在 VisualAge Smalltalk 上运行的第 1 阶段迁移工具。您还必须创建 DB2 迁移数据库，该数据库用来存储正在从 VisualAge Generator 4.5 (VAGen 4.5) 迁移到 EGL 的数据。

---

### 在 VisualAge Smalltalk 上安装第 1 阶段迁移工具

VisualAge Generator 到 EGL 第 1 阶段迁移工具是作为名为 VAGenMigST.exe 的自解压缩文件交付的。要安装此文件，请执行下列操作：

1. 升级到带有修订包 4 的 VisualAge Generator 4.5。并且，请回顾第 307 页的附录 F，『不正确的外部源格式会导致创建 EGL 时发生问题的情况』以了解在特定情况下可能需要的其它 VisualAge Generator APAR。
2. 在系统上，确定 VisualAge Smalltalk 的安装位置。
3. 关闭 VisualAge Smalltalk。
4. 运行自解压缩 VAGenMigST.exe 文件。此文件位于 Rational Developer 或 WebSphere Studio 安装目录下的以下子目录中：

\bin

**注：**如果安装您现在正在使用的产品前您已经安装并保留了先前版本的开发者产品，则您想使用的安装目录可能是先前进行安装时使用的目录。

5. 当图形用户界面提示出现时，浏览到安装了 VisualAge Smalltalk 的驱动器和目录。然后，单击**抽取**。

当自解压缩可执行文件运行时，它将下列文件解压缩到 VisualAge Smalltalk 安装目录中：

- import\vgMigSt.dat
- feature\vgMigSt.ctl
- image\Messages.properties
- image\MigPreferences.xml
- \image\VGMigReserved.txt
- checkStage1.sql
- createdatabase.sql
- createtables.sql
- checkStage1.bat
- SetupDatabase.bat
- SetupTables.bat
- deletemigsets.bat

### 装入迁移功能部件

为了能够使用第 1 阶段迁移工具，必须装入“VAGen EGL 迁移”功能部件。要完成此任务，请执行下列步骤：

1. 启动 VisualAge Generator on Smalltalk.
2. 通过执行下列操作来装入“VAGen EGL 迁移”功能部件:
  - a. 从“系统抄本”中, 选择**工具 -> 装入 / 卸装功能部件**。
  - b. 在“必需选择”窗口中, 请执行下列操作:
    - 1) 确保选择了**显示其它功能部件复选框**。
    - 2) 在可用的功能部件窗格中, 选择其它: **VAGen EGL 迁移 - versionName**。
    - 3) 选择 **>>** 按钮以将其它: **VAGen EGL 迁移 - versionName** 移到已装入的功能部件窗格中。
    - 4) 单击**确定**。这就将导入“VAGen EGL 迁移”功能部件并将其装入到映像中。
3. 在“系统抄本”中, 您应该会看到一些消息, 这些消息指示已成功地装入了“VAGen EGL 迁移”功能部件。在工具栏上, 您还应该看到 **EGL 迁移工具**。在 VisualAge Organizer 中, 您应该在**应用程序窗格**中看到 HptEglMigrationGuiApp。
4. 在装入“VAGen EGL 迁移”功能部件之后, 将提示您保存映像。单击**是**以便不必再次装入该功能部件。

注: 如果在装入功能部件时出现问题, 请检查 `abt.ini` 文件 (位于 `VisualAge-Smalltalk-installation-directory\image` 目录中)。确保 `abt.ini` 文件在 `[EmLibraryInterface]` 标题下面填写了下列字段:

- `ServerAddress=myserver.somecompay.somewhere.com`。此值应该指向贵公司中运行 EMSRV 的服务器。如果您使用本地库, 请设置 `ServerAddress=127.0.0.1`。
- `DefaultName=path-to-mgr50.dat\mgr50.dat`。此值必须是 Smalltalk 库的名称。

---

## 创建迁移数据库

要了解有关创建迁移数据库的信息, 请参阅第 309 页的『创建 DB2 迁移数据库』。运行自解压缩的 `VAGenMigST.exe` 文件时, 需要使用 `SetupDatabase.bat` 和 `SetupTables.bat` 文件, 这两个文件位于 VisualAge Smalltalk 安装目录中。

---

## 设置第 1 阶段首选项

当您在 VisualAge Smalltalk 上安装第 1 阶段迁移工具时, 安装过程在目录 `VisualAge-Smalltalk-installation-directory\image` 中创建了名为 `MigPreferences.xml` 的样本首选项文件。在修改任何首选项之前, 您应该先创建一份 `MigPreferences.xml` 文件副本以作备份。

基于 Smalltalk 的 VisualAge Generator 到 EGL 迁移工具提供了一个图形用户界面编辑器来帮助您指定第 1 阶段迁移首选项。可以采用下列两种方法中的任何一种方法来启动第 1 阶段首选项编辑器:

- 从“系统抄本”中, 选择 **EGL 迁移工具 -> 首选项编辑器**。这就显示了“EGL 迁移首选项编辑器”。缺省情况下, 首选项编辑器将打开您上次修改的首选项文件 (或者, 如果您此前从未修改过任何首选项, 则首选项编辑器在缺省情况下将打开随第 1 阶段工具一起交付的 `MigPreferences.xml` 文件)。如果您需要指向另一个首选项文件, 请单击**打开...按钮**。



- 如果配置映射名至少与一个配置映射过滤器不匹配，则不考虑对该配置映射进行进一步处理。
- 如果配置映射名至少与一个配置映射过滤器相匹配，则对该配置映射的版本进行如下处理：
  - 如果您指定了任何版本名过滤器，则将把配置映射的每个版本名与版本名过滤器列表进行匹配。如果版本名与任何版本名过滤器匹配，则考虑对该版本进行进一步处理。
  - 如果您指定了版本深度过滤器并且未指定任何版本名过滤器，则将考虑对配置映射的最新版本进行进一步处理，并且数目可达版本深度过滤器指定的版本数。

**注：**版本深度和版本名是互斥的。缺省情况下，版本深度过滤器已包括在 `MigPreferences.xml` 文件中。

- 如果配置映射名和版本名导致要考虑对该配置映射版本进行进一步处理，则第 1 阶段迁移工具执行下列操作：
  - 如果该配置映射版本是高级配置映射，则迁移工具将该配置映射版本用作创建迁移集的基础。假定版本名与版本过滤器匹配，那么每个版本的高级配置映射都会生成不同的迁移集。
  - 如果该配置映射版本不是高级配置映射，则不考虑对该配置映射版本进行进一步处理。可能仍会将该配置映射版本包括在其它迁移集中；情况仅仅是为不专门为该配置映射版本创建迁移集。

指定“资源库过滤器”信息，如下所示：

- **配置映射过滤器。**迁移工具将库中的配置映射名与您指定的配置映射过滤器进行匹配。您可以指定多个配置映射过滤器。要添加、更改或删除过滤器，可以右键单击过滤器并使用上下文菜单中的选项。过滤器是不区分大小写的。可以在过滤器中使用通配符，如下所示：
  - 配置映射过滤器 `*xyz*` 与库中名称的任何位置包含字符串“xyz”的任何配置映射名匹配。
  - 配置映射过滤器 `xyz*` 与库中任何以“xyz”开头的配置映射名匹配。
  - 配置映射过滤器 `*xyz` 与库中任何以“xyz”结尾的配置映射名匹配。
- **版本名过滤器。**如果配置映射名与配置映射过滤器匹配，则迁移工具使用版本名过滤器来确定应该考虑迁移哪些配置映射版本（如果有的话）。您可以指定多个版本名过滤器。要添加、更改或删除过滤器，可以右键单击过滤器并使用上下文菜单中的选项。过滤器是不区分大小写的。可以在过滤器中使用通配符，如下所示：
  - 版本名过滤器 `*xyz*` 与版本名中任何位置包含字符串“xyz”的任何配置映射版本名匹配。
  - 版本名过滤器 `xyz*` 与任何以“xyz”开头的配置映射版本名匹配。
  - 版本名过滤器 `*xyz` 与以“xyz”结尾的配置映射版本名匹配。
  - 如果您保留空的版本名过滤器字段，迁移工具将使用版本深度过滤器。
- **版本深度过滤器。**您可以指定要迁移的先前版本的数目。缺省值是 1，在这种情况下，迁移工具仅处理配置映射的最新版本。如果您指定了任何版本名过滤器，则版本深度过滤器将被忽略。

## “映射” 页

“映射” 页使您能够指定下列内容:

- 公共部件的以及未使用的部件的 EGL 文件名。
- 在构建 EGL 项目名和包名时使用的后缀。
- 用于控制应用程序名到 EGL 包名转换方式的选项。
- 关于哪些 VAGen 配置映射和应用程序包含公共部件的信息。

以下内容对“映射” 页上的首选项进行更详细的描述:

- 文件名。“文件名” 部分使您能够控制迁移期间创建的两个 EGL 文件的名称。
  - 公共部件使您能够指定一个 EGL 文件的名称, 该文件包含迁移集作用域中多个唯一的可生成部件的公共部件。请指定不带扩展名或路径的文件名。迁移工具将在每个 EGL 包中创建一个公共部件文件, 该文件包含由迁移集中多个可生成部件使用(与它们相关联)的部件, 或者包含已被标识为公共配置映射或应用程序的 VAGen 配置映射或应用程序中的部件。要了解有关是将部件与程序一起放在文件中还是放入公共部件文件的详细信息, 请参阅第 30 页的『将部件放到 EGL 文件中』。
  - 未使用的部件使您能够指定一个 EGL 文件的名称, 该文件包含迁移集作用域中未使用的部件。请指定不带扩展名或路径的文件名。迁移工具将在任何 EGL 包中创建一个未使用部件文件, 该文件包含未被迁移集中的任何可生成部件使用(与它们相关联)的部件(如果相应的 VAGen 配置映射和应用程序未被标识为公共配置映射或应用程序的话)。
- 跨映射。“跨映射” 部分使您能够指定一些后缀, 当其中一个映射组包含来自多个配置映射或应用程序的映射时, 将使用这些后缀。
  - 项目后缀使您能够指定一个后缀, 第 1 阶段迁移工具将该后缀与迁移集名称连接到一起以创建新的 EGL 项目名。仅当映射组及其映射分布在迁移集中的多个 VAGen 配置映射中时, 迁移工具才创建这个新的 EGL 项目。在应用任何重命名规则之后, 迁移工具将此后缀与迁移集名称连接到一起。
  - 包后缀使您能够指定一个后缀, 第 1 阶段迁移工具将该后缀与项目名连接到一起以便在该 EGL 项目中创建新的 EGL 包名。仅当映射组及其映射分布在配置映射中的多个 VAGen 应用程序中时, 迁移工具才创建这个新的 EGL 包。在应用任何重命名规则之后, 迁移工具连接此后缀以创建 EGL 项目名。
- EGL 包命名选项。“EGL 包命名选项” 部分使您能够指定有关将 Smalltalk 应用程序名转换为 Java 包名的一般规则。
  - 使用包命名点表示法。如果您选择了此选项, 则迁移工具将在应用程序名中第一个大写字母后的每个大写字母前放置一个点, 从而将 VAGen 应用程序名转换为 EGL 包名。例如, 如果您选择了此选项, 则迁移工具将把 MyOrderEntryApp 更改为 My.Order.Entry.App。
  - 将包名转换为小写包名。如果您选择了此选项, 则迁移工具通过将大写字母更改为小写字母来将 VAGen 应用程序名转换为 EGL 包名。例如, 如果您选择了此选项, 则迁移工具将把 MyOrderEntryApp 更改为 myorderentryapp。

通常, 您应该同时选择这两个选项。如果同时选择了这两个选项, 则迁移工具将 MyOrderEntryApp 更改为 my.order.entry.app。EGL 包命名选项是在任何重命名规则应用之后应用的。

- 公共标识。此部分使您能够指定带通配符的字符串列表，迁移工具可以使用此字符串列表来确定哪些配置映射和应用程序包含公共（共享）部件。要添加或删除公共标识，可以右键单击该字段并使用上下文菜单中的选项。当您添加标识时，编辑器将提示您输入下列信息：
  - 上下文指示该字符串是要与配置映射名匹配、与应用程序名匹配还是同时与这两者匹配。
  - 配置映射使您能够指定一个字符串，该字符串标识包含公共部件的配置映射。迁移工具将此字符串与迁移集中的每个配置映射名进行匹配以确定该配置映射是否包含公共部件。如果配置映射名与任何字符串匹配，则认为该配置映射中的所有部件都将“被使用”。每个不可生成的部件都将放到程序文件中或者由“公共部件文件名”首选项指定的文件中；即使某个部件未被迁移集中的任何可生成部件使用，该部件也不会放到未使用部件文件中。您可以输入多个配置映射字符串。
  - 应用程序使您能够指定一个字符串，该字符串标识包含公共部件的应用程序。迁移工具将此字符串与迁移集中的每个应用程序名进行匹配以确定该应用程序是否包含公共部件。如果该字符串与应用程序名匹配，则认为该应用程序中的所有部件都将“被使用”。每个不可生成的部件都将放到程序文件中或者由“公共部件文件名”首选项指定的文件中；即使某个部件未被迁移集中的任何可生成部件使用，该部件也不会放到未使用部件文件中。您可以输入多个应用程序字符串。
  - 两者使您能够指定一个字符串，迁移工具将该字符串与迁移集中的配置映射名和应用程序名进行匹配。两者等同于对上下文配置映射和上下文应用程序指定同一个字符串。
  - 用于标识公共代码的模式使您能够指定一个字符串，迁移工具应该根据所指定的上下文来对该字符串进行匹配。可以在该字符串的开头或末尾使用 \* 作为通配符。过滤器是不区分大小写的。

## “重命名”页

“重命名”页使您能够控制根据 VAGen 配置映射名、应用程序名和版本名派生的 EGL 项目名、包名和版本名。“顺序”列中的数字指示了迁移工具应用重命名规则的顺序，其中，编号最小的规则将首先被应用。要添加或删除重命名规则，可以单击一个规则并使用上下文菜单中的选项。“添加规则”始终将新规则置于列表末尾。当您添加规则时，编辑器将提示您输入下列信息：

- 源字符串指定 VAGen 名称中要更改的字符。
- 目标字符串指定要在得到的 EGL 名称中使用的字符。
- 字符串上下文指定 VAGen 名称中的位置，在重命名期间，迁移工具应该在该位置查找源字符串。值如下所示：
  - 前面表示如果源字符串出现在配置映射名、应用程序名或版本名的开头，则应用该规则。
  - 后面表示如果源字符串出现在配置映射名、应用程序名或版本名的末尾，则应用该规则。
  - 任何表示如果源字符串出现在配置映射名、应用程序名或版本名中的任何位置，则应用该规则。
  - 标记表示仅当源字符串是配置映射名、应用程序名或版本名的精确匹配时，才应用该规则。

- 映射上下文指示迁移工具是否将重命名规则应用于配置映射名、应用程序名或版本名。映射上下文值如下所示：
  - 配置映射表示重命名规则仅应用于 VAGen 配置映射名。
  - 应用程序表示重命名规则仅应用于 VAGen 应用程序名。
  - 两者表示重命名规则既应用于 VAGen 配置映射名也应用于 VAGen 应用程序名。
  - 版本表示重命名规则应用于所有配置映射的版本名。如果版本名包含目录名或文件名所不允许的特殊字符，如分号 (;)，则使用版本重命名规则。在第 1 阶段迁移中，迁移工具使用已重命名的版本来创建迁移规划文件名，在第 3 阶段，迁移工具使用该版本来创建目录名。

## “执行”页

“执行”页使您能够指定有关迁移数据库位置的信息以及有关在第 1 阶段要捕获的记录信息、调试信息和报告信息的信息。以下内容对您可以在“执行”页上指定的首选项进行更详细的描述：

- 数据库信息。此部分使您能够指定关于迁移数据库的详细信息：
  - *DB* 是一个迁移数据库的名称，迁移工具将把迁移集信息写入该迁移数据库。如果您在创建迁移数据库时将数据库名称更改为除 VGMIG 以外的名称，则必须将此首选项指定的数据库名称更改为与使用的名称匹配。
  - 模式是用作数据库表限定符的名称。如果未指定模式，则迁移工具使用 MIGSCHEMA 来作为缺省模式。如果您在创建迁移数据库时将模式名更改为除 MIGSCHEMA 以外的名称，则必须将此首选项指定的模式名更改为与使用的名称匹配。
  - 用户标识是连接至迁移数据库时所需的用户标识。如果您未指定用户标识，则迁移工具将尝试把 VAGen SQL 首选项中指定的用户标识用作缺省用户标识来进行连接。如果连接失败，则迁移工具将尝试使用您的登录用户标识。如果这两次尝试都失败，则迁移工具将显示一个对话框窗口来要求输入该信息。
  - 密码是连接至迁移数据库时所需的密码。如果您未指定密码，则迁移工具将尝试把 VAGen SQL 首选项中指定的密码用作缺省密码来进行连接。如果连接失败，则迁移工具将尝试使用您的登录密码。如果这两次尝试都失败，则迁移工具将显示一个对话框窗口来要求输入该信息。

**注：**在首选项文件中，密码不是以加密形式存储的。如果对此感到担忧，则不要在首选项文件中输入密码。并且等待提示。

- 服务信息。此部分使您能够指定有关在第 1 阶段要捕获的记录信息和调试信息的详细信息。您可以指定下列信息：
  - 跟踪级别使您能够指定要写入调试文件的信息的级别。您可以指定下列其中一个值：
    - *FATAL* (级别 1) - 记录错误消息。
    - *WARN* (级别 2) - 记录警告消息和错误消息。
    - *INFO* (级别 3) - 记录参考消息、警告消息和错误消息。
    - *DEBUG* (级别 4) - 记录调试信息以及参考消息、警告消息和错误消息。“调试”是唯一一个会导致迁移工具将信息写入调试文件的跟踪级别。
  - 日志文件名使您能够为日志文件指定驱动器、目录和文件名。您可以创建具有任何文件扩展名的日志文件，但最好将其作为 .xml 文件来进行查看。如果您省略了

日志文件名，则将把名为 *miglog.xml* 的文件写至“日志文件名”字段中指定的驱动器和目录。如果您未指定驱动器和目录，则迁移工具将把日志文件写至迁移规划目录。

- 调试文件名使您能够对 IBM 支持机构可能需要的调试文件指定驱动器、目录和文件名。您可以创建具有任何文件扩展名的调试文件，但最好将其作为 *.xml* 文件来进行查看。仅当“跟踪级别”首选项设置为 **DEBUG** 时，才会将信息写入此文件。如果您省略了调试文件名，并指定跟踪级别 **DEBUG**，则将把名为 *migDebug.xml* 的文件写至“调试文件名”字段中指定的驱动器和目录。如果您未指定驱动器和目录，则迁移工具将把调试文件写至迁移规划目录。
- 验证信息。此部分使您能够指定关于可以从第 1 阶段迁移工具输出的报告文件的信息。您可以指定下列信息：
  - 报告文件名使您能够指定要对报告文件使用的驱动器、目录和文件名。此报告包含关于将要如何迁移 VAGen 文件的信息。您始终应该指定 *.htm* 扩展名。如果省略了报告文件名，则将把名为 *reportMigrationReport.xml* 的文件写至“报告文件名”字段中指定的驱动器和目录。如果您未指定驱动器和目录，则迁移工具将把报告文件写至迁移规划目录。

## 样本 MigPreferences.xml 文件

以下是一个样本 MigPreferences.xml 文件：

```
<preferences>
  <database>
    <uri>VGMIG</uri>
    <schema>MIGSCHEMA</schema>
    <userid></userid>
    <password></password>
  </database>
  <migrationSpec>
    <directory>d:\TempMig\Stage1</directory>
    <filename></filename>
  </migrationSpec>
  <service>
    <traceLevel>4</traceLevel>
    <logfile>d:\TempMig\stage1\migLog.xml</logfile>
    <debugfile>d:\TempMig\stage1\migDebug.xml</debugfile>
  </service>
  <repositoryFilters>
    <projectName>MyConfigMap*</projectName>
    <versionNumber>1</versionNumber>
  </repositoryFilters>
  <verification>
    <reportName>d:\TempMig\report\MigrationReport.htm</reportName>
  </verification>
  <eglMapping>
    <commonPartsFileName>CommonParts</commonPartsFileName>
    <unusedPartsFileName>UnusedParts</unusedPartsFileName>
    <spanningMapsProjectSuffix>MapsProject</spanningMapsProjectSuffix>
    <spanningMapsPackageSuffix>mapspackage</spanningMapsPackageSuffix>
    <packageDotNotation>true</packageDotNotation>
    <packageLowercase>true</packageLowercase>
    <commonParts>
      <commonConfigMap>*Common*</commonConfigMap>
      <commonApplication>*Common*</commonApplication>
    </commonParts>
    <renameRule order="1">
      <fromString> </fromString>
      <toString></toString>
      <stringContext>any</stringContext>
      <mappingContext>both</mappingContext>
    </renameRule>
  </eglMapping>
</preferences>
```

```

</renameRule>
<renameRule order="101">
  <fromString>CM</fromString>
  <toString></toString>
  <stringContext>back</stringContext>
  <mappingContext>configMap</mappingContext>
</renameRule>
<renameRule order="301">
  <fromString>App</fromString>
  <toString></toString>
  <stringContext>back</stringContext>
  <mappingContext>application</mappingContext>
</renameRule>
<renameRule order="501">
  <fromString>:</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="502">
  <fromString>/</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="503">
  <fromString>\</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="504">
  <fromString>|</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="505">
  <fromString>?</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="506">
  <fromString>*</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="507">
  <fromString>&lt;</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="508">
  <fromString>&gt;</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>
<renameRule order="509">
  <fromString>&quot;</fromString>
  <toString>_</toString>
  <stringContext>any</stringContext>
  <mappingContext>version</mappingContext>
</renameRule>

```

```

        <renameRule order="510">
            <fromString> </fromString>
            <toString>_</toString>
            <stringContext>any</stringContext>
            <mappingContext>version</mappingContext>
        </renameRule>
    </eglMapping>
</preferences>

```

## 根据首选项派生文件名

第 1 阶段迁移工具以相同的方式为日志文件、调试文件和报告文件派生文件名。下表显示了您在首选项中可能指定的名称以及迁移工具使用的结果驱动器、目录和路径名。在本示例中，迁移规划目录是 d:\myVAGenMig。

表 55. 根据首选项派生的文件名

日志文件名首选项	第 1 阶段迁移工具使用的文件名
保留首选项为空白。	d:\myVAGenMig\migLog.xml 注: <ul style="list-style-type: none"> <li>• 调试文件的缺省文件名是 migDebug.xml。</li> <li>• 报告文件的缺省文件名是 \report\MigrationReport.xml。</li> </ul>
mine.xml	d:\myVAGenMig\mine.xml
logs\mine.xml	d:\myVAGenMig\logs\mine.xml
.mine.xml	VisualAge-Generator-installation-directory\image\mine.xml

## 在运行第 1 阶段工具之前 - 提示与技巧

在运行第 1 阶段迁移工具之前，您可能想执行一些操作以提高性能。您可能还想保存现有的映像以便在迁移完成后使用。

### 提高性能

为了将内存使用量降至最低，最好在运行第 1 阶段迁移工具之前清除（即“擦净”）映像。要清除（即“擦净”）映像，请执行下列操作：

1. 从“系统抄本”中，选择工具 → 打开 **VAGen** 工具工作空间。
2. 在“映像管理”部分中，拖动光标经过：
 

**System abtScrubImage**
3. 然后，单击鼠标右键并选择**执行**以运行“System abtScrubImage”。
4. 如果擦净映像，则可能需要重新装入“VAGen EGL 迁移”功能部件。要了解有关如何装入该功能部件的信息，请参阅第 105 页的『装入迁移功能部件』。另外，要将“EGL 迁移工具”选项添加回到“系统抄本”工具栏中，请执行下列操作：
  - a. 将以下内容输入到“系统抄本”窗口中：
 

```
HptEglMigrationGuiApp loaded
```
  - b. 拖动光标经过您输入的行，然后单击鼠标右键并选择**执行**。

为了缩短第 1 阶段迁移工具在分析要迁移哪些配置映射和版本时花费的时间，您可以考虑创建一个仅包含要迁移的配置映射版本的库。进行迁移时，如果在 VisualAge Generator 中正在进行维护工作，则使用独立的迁移库还具有下列优点：

- 要迁移的配置映射版本集是固定的。如果使用“版本深度”首选项来控制所要迁移的内容，这一点就尤其重要。
- 您可以将新的迁移库中的版本与维护库进行比较以确定还需要迁移哪些其它配置映射版本。

如果您创建了一个特殊的库，则可以考虑将其用作本地库以提高第 1 阶段迁移性能。

## 保存映像

在第 1 阶段处理开始时，第 1 阶段迁移工具将从映像中卸装所有包含 VAGen 部件的应用程序。当第 1 阶段工具完成时，只有最后一个要被处理的迁移集留在映像中。通过卸装所有应用程序，可以确保在第 1 阶段仅考虑将迁移集中的部件包括在关联部件列表中。如果您希望保存映像，则应该在运行第 1 阶段工具之前执行下列操作：

1. 关闭 VisualAge Generator。
2. 将下列文件的备份副本保存到 `\VisualAgeForSmalltalk-installation-directory\image` 中：
  - `abt.icx`
  - `abt.ini` - 如果在运行第 1 阶段期间未更改任何首选项，则不必保存此文件
  - `hpt.ini` - 如果在运行第 1 阶段期间未更改任何首选项，则不必保存此文件
3. 启动 VisualAge Generator。

当完成运行第 1 阶段工具时，请执行下列操作以复原工作空间：

1. 关闭 VisualAge Generator。
2. 在运行第 1 阶段工具之前，先复原已备份的文件。
3. 启动 VisualAge Generator。

---

## 运行第 1 阶段迁移工具

在完成编辑首选项之后，您就准备好运行第 1 阶段迁移工具以从 Smalltalk 库中抽取源代码了。要完成此任务，请执行下列步骤：

1. 使用下列其中一种技术来启动“EGL 迁移驱动程序视图”：
  - a. 如果已通过启动“首选项编辑器”修改了首选项，则通过选择 **EGL 迁移工具 -> 迁移驱动程序**来从“系统抄本”中启动“EGL 迁移驱动程序视图”。
  - b. 如果已通过启动“EGL 迁移驱动程序”修改了首选项，则在保存首选项文件后，您将回到“EGL 迁移驱动程序视图”中。
2. 确保迁移首选项文件的文件名指向用来存储首选项的文件。使用**浏览...**按钮来指向另一个首选项文件。使用**编辑...**按钮来查看首选项或对首选项进行最终的修改。
3. 当您对首选项感到满意时，选择要使用的“执行选项”。“执行选项”控制着第 1 阶段迁移工具的输出，如下所示：
  - 覆盖 *PLN* 按如下方式控制迁移规划文件：
    - 如果您选择了覆盖 *PLN*，则第 1 阶段迁移工具对首选项中指定的规划目录中的文件执行下列操作：
      - 如果首选项文件未对 `.pln` 文件指定文件名，则迁移工具将删除指定规划目录中的所有 `.pln` 文件并创建新文件。
      - 如果首选项文件对 `.pln` 文件指定了文件名，则迁移工具在创建新的 `.pln` 文件之前仅从指定规划目录中删除同名的文件。

- 如果您未选择覆盖 *PLN*，则第 1 阶段迁移工具不创建任何新的迁移规划文件。而是，第 1 阶段迁移工具根据首选项中指定的规划目录和规划文件名来运行：
  - 如果首选项文件未对 *.pln* 文件指定文件名，则迁移工具使用指定规划目录中的所有规划来运行。
  - 如果首选项文件对 *.pln* 文件指定了文件名，则迁移工具仅使用该 *.pln* 文件来运行。
- 生成报告控制着迁移工具是否创建首选项文件的 **Verification** 部分中指定的报告。如果您未选择此选项，则不会创建该报告。如果您选择了此选项，则迁移工具将使用首选项中指定的报告文件名来创建该报告。此报告显示在迁移期间如何将配置映射、应用程序和 **VAGen** 部件分配到 **EGL** 项目、包和文件中。最初，您可以取消选择此选项以便可以查看 *.pln* 文件，从而确保迁移工具正在计划处理您所处理的配置映射版本。如果您对选择的配置映射版本不满意，则可以优化首选项并再次运行“覆盖 *PLN*”。当您对将要处理的配置映射版本感到满意时，请选择“生成报告”选项并再次运行第 1 阶段。

**注:**

- 生成报告要花费一些时间，因此，您最好查看 *.pln* 文件以确保迁移工具将处理您想要处理的配置映射版本。
- 如果您选择了“生成报告”选项，则第 1 阶段迁移工具将从报告目录中自动删除任何现有报告文件。如果要保存先前的报告文件，则必须将报告文件移至另一个目录中，或者将新报告生成到新目录中。因为报告文件链接至其它文件，所以，将报告文件重命名会导致链接失效，从而使文件不再可供查看。
- 更新数据库控制着迁移工具是否用迁移规划信息来更新迁移数据库。如果您未选择此选项，则迁移数据库不会被更新。如果您选择了此选项，则将使用迁移规划信息（包括迁移规划中每个 **VAGen** 部件的外部源格式）来更新首选项中指定的迁移数据库。最初，您可以取消选择此选项以便可以查看报告，从而了解配置映射、应用程序和 **VAGen** 部件将以何种方式放到 **EGL** 项目、包和文件中。如果您对放置方式不满意，则可以优化首选项并再次运行该报告。当您对放置方式感到满意时，可以在选择了“更新数据库”的情况下最后运行一次第 1 阶段，以使迁移工具将结果放到迁移数据库中。

**注:**

- 如果您选择了“更新数据库”选项，并且在数据库中已存在迁移集，则第 1 阶段迁移工具将自动地从数据库中删除关于该迁移集的旧信息并接着为该迁移集添加新信息。您不需要从数据库中清除迁移集。
  - 迁移工具不会自动地清除整个迁移规划。
4. 在选择了“执行选项”之后，单击**确定**以运行第 1 阶段迁移工具。迁移工具将把消息写至迁移首选项中指定的日志文件。此工具还将把相同的消息放入“系统副本”。

此刻，如果您已经运行了第 1 阶段迁移工具并选择了“更新数据库”选项，则迁移规划信息（包括具有外部源格式的 **VAGen** 代码）已存储在迁移数据库中。在查看报告和第 1 阶段消息之后，您可能会决定在 **VisualAge Generator** 中更改代码并再次运行第 1 阶段。当您对第 1 阶段的结果感到满意并将最终的外部源格式代码存储在迁移数据库中之

后，您就已准备好执行第 2 阶段迁移了。要运行第 2 阶段迁移工具，请使用 EGL 开发环境。要了解有关继续执行迁移过程的信息，请参阅第 123 页的第 6 章，『第 2 阶段 - 转换为 EGL 语法』。

---

## 迁移规划和高级配置映射

迁移规划文件仅仅是一个 XML 文件，它指定了一个或多个迁移集的名称，并且，对于每个迁移集，它指定了一个高级配置映射和版本，该高级配置映射和版本为该迁移集指定了应用程序及其版本。高级配置映射还可以将其它配置映射及其版本指定为必需映射。但是，对一个迁移集只能指定一个高级配置映射版本。第 1 阶段迁移工具已被设计成根据配置映射的资源库过滤器首选项以及版本名来自动创建迁移规划文件。第 1 阶段工具使用这些过滤器来选择为了确定哪些版本是高级配置映射版本而应该查看的配置映射版本。第 1 阶段工具将每个高级配置映射版本用作迁移集的基础。

如果在生成 VAGen 源代码时使用了高级配置映射，则这些高级配置映射就是应该用于迁移的高级配置映射。这是因为，每个高级配置映射都提供了在生成期间一起使用的部件分组，因此具有一组程序的所有关联部件。

如果当前未使用任何配置映射，则必须创建要用于迁移的配置映射。在这种情况下，最简单的技术是创建一个配置映射版本，该版本包含要成组迁移的所有应用程序版本，包括公共应用程序版本。要了解详细信息，请参阅第 117 页的『创建高级配置映射』。在创建配置映射之后，您可以使用第 1 阶段迁移工具来自动地创建迁移规划。

即使您当前正在使用配置映射，也可能并没有高级配置映射。例如，您可能有一个用于公共应用程序的配置映射以及另一个用于子系统的配置映射。在生成时，您确定要将每个配置映射的哪个版本装入到映像中。在这种情况下，可以执行下列其中一项操作来指定要成组迁移的内容：

- 创建一个要在迁移期间使用的高级配置映射。这个高级配置映射可以指定应用程序版本列表、配置映射版本列表或者应用程序版本与配置映射版本的组合。例如，高级配置映射可以列示公共配置映射和子系统配置映射，以便在进行迁移时将这两种配置映射看作一个组。要了解详细信息，请参阅第 117 页的『创建高级配置映射』。在创建高级配置映射之后，您可以使用第 1 阶段迁移工具来自动地创建迁移规划。
- 如果您不愿意创建高级配置映射，则可以使用下列其中一种技术来自行创建迁移规划文件：
  - 如果数据库或其它系统中的信息指定了特定的 Smalltalk 配置映射版本进行生成所需的内容，则您可以编写一个工具来根据数据库自动创建迁移规划文件。
  - 手工创建迁移规划文件。

## 创建高级配置映射

要创建高级配置映射以便在迁移期间使用，请在 VisualAge Generator 中执行下列操作：

1. 从 VisualAge Organizer 中，选择**选项**并确保选择了**全部菜单**。
2. 从 VisualAge Organizer 中，选择**工具 -> 配置映射**。
3. 从“配置映射浏览器”中，选择**名称 -> 创建**。
4. 在“需要信息”窗口中，输入配置映射的名称，然后单击**确定**。这就自动创建并选择了该配置映射的新版本。

5. 选择应用程序 -> 添加。然后，选择要迁移的每个应用程序以及该应用程序的版本。只能为每个应用程序指定一个版本。在为需要包括的每个应用程序选择了版本之后，单击**确定**。您指定的应用程序版本将被成组地迁移。这个组确定了迁移工具用来解析不确定情况的部件集。
6. 通过选择版本 -> 设置版本来设置该配置映射的版本。
7. 选择配置映射版本，然后通过选择版本 -> 装入来将其装入到映像中。
8. 在装入新的高级配置映射之后，您可能想要对程序和表运行验证以确保它们在 VAGen 中有效并确保不缺少任何部件。在验证程序时，请包括 /GENMAPS、/GENHELPMAPS 和 /NOGENTABLES。这两个映射选项使您能够确保映射对于使用它们的程序来说是有效的。/NOGENTABLES 使您只需对表执行一次验证，而不是对每个使用该表的程序都重新验证该表。

### 将配置映射链接起来

可以将配置映射链接起来。例如，您可以创建一个配置映射，该配置映射列示每个公共应用程序的版本。然后，对于每个子系统，为该子系统创建一个高级配置映射，该高级配置映射包含每个特定于子系统的应用程序所需的版本。可以将公共应用程序的配置映射包括在子系统配置映射中，如下所示：

1. 从“配置映射浏览器”中，选择子系统配置映射的已打开的版本。
2. 选择表达式 -> 添加。
3. 在“需要信息”窗口中，单击**确定**以接受 *true* 作为表达式。
4. 在“配置表达式”窗格中选择 *true*。然后，选择**必需映射 -> 添加 -> 作为第一个**。然后，选择包含公共应用程序的配置映射版本。通过选择版本 -> 设置版本来设置配置映射的版本。
5. 选择配置映射版本，并通过选择版本 -> 随必需映射装入来将其装入到映像中。

通过使用必需映射，您就可以方便地指定公共应用程序版本，而不必在每个子系统的高级配置映射中显式地列示公共应用程序版本。

### 将配置映射与第 1 阶段工具配合使用

当您准备好运行第 1 阶段迁移工具时，请执行下列操作：

- 当设置第 1 阶段首选项时，在“构建规划”页上的“资源库过滤器”部分中，设置“配置映射”列表，以使列表中的一个过滤器与已创建的高级配置映射匹配。
- 在向第 1 阶段工具指示所要使用的首选项文件时，还需要选择覆盖 *PLN* 选项。此选项指示第 1 阶段迁移工具根据高级配置映射来创建迁移规划文件并覆盖任何现有迁移规划文件。

## 手工创建迁移规划文件

如果已有一些外部控件，它们可用来确定当在 VisualAge Generator 中进行生成时将要哪些配置映射版本装入到映像中，则您可能会决定手工创建迁移规划文件或者开发一个工具来根据外部信息自动地创建迁移规划文件。迁移规划文件必须具有 *.pln* 文件扩展名和以下格式：

```
<migrationDefinition>
  <migrationSet
    name="migrationSet1"
    version="migrationSet1Version1"
    vgName="migrationSet1"
    vgVersion="migrationSet1Version1">
```

```

    <configMap
      name="configurationMap1"
      version="configurationMap1Version1">
    </configMap>
    <configMap
      name="configurationMap2"
      version="configurationMap2Version1">
    </configMap>
      .
      .
      .
    <configMap
      name="configurationMapN"
      version="configurationMapNVersion1">
    </configMap>
  </migrationSet>
</migrationDefinition>

```

在上一个示例中，下列陈述适用：

- **migrationSet1** 是一个名称，它可以用来引用必须一起迁移的一组配置映射。迁移集名称存储在迁移数据库中，在以后的迁移阶段，它的用途如下：
  - 在第 1 阶段迁移中，如果映射组中的映射跨多个配置映射，则使用带有后缀的迁移集名称来为包含该映射组及其所有映射的新 EGL 项目构建名称。
  - 在第 2 阶段迁移中，迁移集名称指定要将迁移数据库中的哪一组配置映射转换为 EGL。
  - 在第 3 阶段，迁移集名称指定要使用迁移数据库中的哪一组配置映射来在工作空间或临时目录中创建 EGL 项目、包和文件。如果您选择将第 3 阶段的输出保存至临时目录，则迁移集名称和迁移集版本还用来创建高层目录名。

在迁移期间，迁移集名称仅用来标识一组配置映射。除了映射跨 VisualAge Generator 中的多个配置映射的情况以外，在迁移之后就不再使用迁移集名称了。

- **configurationMap1**、**configurationMap2**、... 和 **configurationMapN** 是要成组迁移的配置映射。在迁移集中，必须仅将 **configurationMap** 列示一次。
- **configurationMap1Version1**、**configurationMap2Version1**、... 和 **configurationMapNVersion1** 是这些配置映射中的每个配置映射的对应版本。对于迁移集中的每个配置映射，只能指定一个版本。
- 您指定的配置映射名和版本名必须与库中的配置映射名和版本名精确匹配。这些名称是区分大小写的。该信息用来将配置映射版本添加至映像，以便可以对那些部件进行分析以构建第 1 阶段迁移报告以及装入迁移数据库。

当您准备好运行第 1 阶段迁移工具时，请执行下列操作：

- 在设置第 1 阶段首选项时，在**构建规划**选项卡上，将**规划目录名**设置为用来存储迁移规划文件的驱动器和目录。如果要想让第 1 阶段迁移工具仅运行一个已创建的迁移规划，请指定**规划文件名**。如果要想让第 1 阶段迁移工具使用指定**规划目录**中的所有迁移规划文件来运行，则保留**规划文件名**为空白。
- 当向第 1 阶段工具指示要生成哪些输出时，务必取消选择覆盖 *PLN*。这将导致迁移工具使用先前根据**规划文件名**首选项创建的 .pln 文件来运行。



---

## 第 4 部分 第 2 阶段和第 3 阶段 - 常用迁移步骤

无论您是从 VisualAge Generator on Java 迁移还是从 VisualAge Generator on Smalltalk 迁移，其余迁移步骤都是相同的。



---

## 第 6 章 第 2 阶段 - 转换为 EGL 语法

无论是从 Java 迁移还是从 Smalltalk 迁移，第 2 阶段迁移都是相同的。

您必须运行另一个迁移工具才能将源代码从外部源格式语法转换为 EGL 语法。此迁移工具是一个插件，它在您安装 EGL 之后可用。您可以采用批处理方式或交互方式来运行此工具。（可选）您可以指定在第 2 阶段完成后自动运行第 3 阶段。

---

### 设置工作台首选项

在开始进行迁移之前，应该先设置工作台首选项。这包括：

- 启动参数
- 必需的 EGL 首选项
- 建议的首选项
- VAGen 迁移语法首选项
- 其它建议设置

#### 启动参数

为了提高 EGL 开发环境的性能，您应该在初始化文件中设置几个启动参数。无论您使用哪种产品，参数都是相同的。初始化文件总是位于产品安装目录中，但是文件名随您所使用的产品的不同而有所变化。例如，如果您正在使用 Rational Application Developer，则初始化文件的名称是 wsappdev.ini。要设置启动参数，请执行下列操作：

1. 使用文本编辑器来编辑初始化文件。
2. 查找下面这一行：

```
VMArgs=-Xj9
```

3. 将该行更改为下面这三行：

```
VMArgs=-Xj9  
VMArgs=-Xmx700m  
VMArgs=-Xms256m
```

设置 `-Xmx` 将增加可用内存。一般会将它设置为小于实际内存的值。例如，如果实际内存为 1000K，则应将 `-Xmx` 设置为 700。这有助于避免使用虚拟内存。

设置 `-Xms` 将增加在启动产品时使用的内存量。将它设置为小于或等于 `-Xmx` 设置的值。

4. 保存初始化文件。
5. 启动 EGL 开发环境。例如，如果要使用 Rational Application Developer，则启动该产品。

#### 必需的 EGL 首选项

在可以使用迁移工具之前，必须启用 EGL 功能和迁移工具功能。要启用这些功能，请执行下列操作：

- 从“工作台”窗口中，选择窗口 -> 首选项。

- 通过选择**工作台**首选项旁边的 + 来展开该首选项，然后选择**功能**。
- 展开 EGL Developer。然后，选择 **EGL 开发**和 **VisualAge Generator 到 EGL 迁移**。您不需要选择其它迁移工具。
- 选择**确定**。

在进行迁移前，还必须设置 VisualAge Generator 兼容性方式。通过设置 VisualAge Generator 兼容性方式，可以避免“问题”视图中出现大量消息。要设置此首选项，请执行下列操作：

- 从“工作台”窗口中，选择**窗口 -> 首选项**。
- 选择 **EGL**。
- 选择 *VisualAge Generator 兼容性*首选项。
- 选择**确定**。
- 当提示您对工作空间执行完全重建时，请选择**确定**。

## 建议的首选项

以下是建议的首选项，它们可以帮助您解决“问题”视图中的任何 EGL 验证消息。从“工作台”窗口中，选择**窗口 -> 首选项**，然后选择下列首选项页，如下所示：

- **EGL -> 编辑器**。选择**显示行号**。
- **工作台 -> 编辑器 -> 文本编辑器**。选择**显示行号**。
- “工作台”页。决定是选择还是取消选择**自动构建**。如果您选择了此选项，则每当您保存文件时，EGL 都将重建工作空间中的所有内容并运行验证。选择此选项的优点是，您可以立即得到有关所作的更改的反馈。缺点是，根据工作空间中的部件数目的不同，重建操作可能要花费一些时间。如果未选择此选项，则您保存文件时，EGL 不会重建任何内容。取消选择此选项的优点是，在修改许多文件时，不需要多次执行重建。但是，您必须记得重建项目（**项目 -> 构建项目**或**项目 -> 全部构建**）才能看到对“问题”视图中的消息所作的任何更改的结果。当您处理迁移日志中的消息列表时，您可能想取消选择**自动构建**。这使您能够控制重建的发生时间。当您正在进行正常的代码开发时，您可能想选择此选项。第 3 阶段迁移工具始终关闭**自动构建**。
- **工作台 -> 本地历史记录**。迁移工具生成的 .egl 文件可能相当大。因此，您应该将**最大文件大小 (MB)**更改为较大的值（例如 5）。此外，您可能想根据备份需求来更改文件保留天数和每个文件的条目数。
- **工作台 -> 透视图**。您可能想将 EGL 透视图或 Web 透视图设置为缺省透视图。如果不需要开发 Web 应用程序，请使用 EGL 透视图。如果您计划开发 Web 应用程序，请使用 Web 透视图。要设置缺省透视图，请选择要使用的透视图，然后选择**设置为缺省**。
- **日志记录**。选择**常规**选项卡，然后将**缺省日志记录级别**设置为 **SEVERE**。（可选）将缺省日志记录级别设置为 **INFO** 或 **WARNING**，但是不要任其设置为 **NONE**。

## VAGen 迁移首选项

下面列示的首选项控制着整个迁移过程。除非另有指示，否则这些首选项既用于第 1 - 3 阶段迁移的第 2 阶段也用于单文件迁移。要设置这些首选项，请从“工作台”窗口中选择**窗口 -> 首选项 -> VAGen 迁移**。

- 单文件方式首选项

- 将各部件分别保存到不同的 *EGL* 文件中。此首选项仅用于单文件迁移。如果您选择了此首选项，则每个程序、映射组和表都将放到它自己的文件中；其它部件将放到您在“导入 VAGen 外部源格式文件”向导中指定的文件中。这符合 *EGL* 的有关每个文件对应一个可生成部件的要求。如果您未选择此首选项，则所有部件都将放到您在“导入 VAGen 外部源格式文件”向导中指定的 *EGL* 文件中。请参阅第 19 页的『单文件迁移概述』以了解有关单文件方式下的部件放置算法的特定信息。
- 重命名用户出口信息
  - 重命名用户出口。VAGen 迁移工具通过对部件名或部件引用添加前缀来对数据项、记录、函数和映射进行简单的重命名。（可选）您可以编写用户出口例程以进行更复杂的重命名操作。例如，在迁移期间，您可能想将连字符 (-) 更改为下划线 (\_)。如果您提供用户出口例程以便对部件进行重命名，则选择“重命名用户出口”首选项。如果您选择了此选项，则必须提供其它有关“重命名用户出口”例程的信息。请参阅白皮书“Using the Rename User Exit in the VisualAge Generator to EGL Migration Tool”以了解有关如何创建“重命名用户出口”的详细信息并获取样本代码。
  - JAR 文件位置。指定系统中包含“重命名用户出口”例程的 .jar 文件的位置。
  - 包名。指定该 .jar 文件中包含“重命名用户出口”例程的包的名称。
  - 类名。指定该包中包含重命名逻辑的类的名称。这个类必须包含方法 `renameUserExit(String s, Connection c)`。
  - 使用数据库。如果“重命名用户出口”使用数据库来提供旧部件名与新部件名之间的关系，则选择此选项。
- 尽量少用 VisualAge Generator 兼容性方式。这一组首选项可以让您尽量减少自动使用迁移技术，这些技术需要 VisualAge Generator 兼容性方式。在选择这些首选项应特别谨慎，原因是它们可能导致“EGL 问题”列表上出现更多错误消息或者导致运行时行为发生更改。
  - 不初始化以前的 EZESYS 值。如果选择此首选项，则迁移工具不会在每个语句中包括用于额外变量的声明和初始化语句，该额外变量将包含 EZESYS 的以前的 VAGen 值。迁移工具在转换包含 EZESYS 的语句 (IF、WHILE 或 TEST 除外) 时仍将使用额外变量。如果选择此首选项并且存在包含 EZESYS 的语句 (IF、WHILE 或 TEST 除外)，“问题”列表上将出现错误。如果计划将所有语句转换为使用新的 EGL sysVar.systemType 值或者如果知道未曾使用 EZESYS，则考虑选择此首选项。有关其它详细信息，请参阅第 80 页的『EZESYS』。
  - 不包括表的 deleteAfterUse。如果选择此首选项，则迁移工具将总是省略每个程序中表使用声明语句的 deleteAfterUse 属性。迁移工具将在 deleteAfterUse 属性被省略的情况下发出警告消息。如果直接从 VisualAge Generator 版本 4.5 修订包 4 迁移并且所有生产程序都是使用该版本和修订包生成的，则考虑设置此首选项。如果从 CSP (跨系统产品) 或 VisualAge Generator 的较早发行版迁移，则一定要对表使用声明中省略了 deleteAfterUse 属性的任何程序进行完整彻底的测试。
  - 不对项或变量使用 evensql=y。如果选择此首选项，在迁移记录中的 PACK 数据项部件或非共享项时，迁移工具将总是使用奇数精度 (或者，如果该项具有最大长度，则使用 18)。迁移工具将在 VAGen 项指定偶数精度的情况下发出警告消息。如果您确定 SQL 表不会对可能在 SQL WHERE 子句或 EGL prepare 语句中引用的列使用偶数精度，则考虑设置此首选项。或者您可以选择此首选项，进行迁移，然后查看迁移工具对其发出警告消息的所有项定义。使用未在 SQL 表定义中指定的精度可能会导致数据库访问性能降低。

- 不设置兼容性方式。如果选择此首选项，则每次迁移工具转换生成选项部件时，该工具总是会省略 `vagCompatibility="YES"` 构建描述符选项。仅当执行下列所有操作时，才应选择此首选项：
  - 选择“尽量减少使用 VisualAge Generator 兼容性方式”部分中的所有其它 3 个首选项。
  - 在 VisualAge Generator 兼容性方式被禁用时，确保所有已迁移部件名符合 EGL 部件命名约定。可指定“重命名用户出口”并编写出口以在迁移时重命名 VAGen 部件来实现这一点。
  - 在启动迁移之前禁用 VisualAge Generator 兼容性的 EGL 首选项。

## VAGen 迁移语法首选项

下面列示的首选项控制着从 VAGen 语法到 EGL 语法的迁移。除非另有指示，否则这些首选项既用于第 1 - 3 阶段迁移的第 2 阶段也用于单文件迁移。要设置这些首选项，请从“工作台”窗口中选择窗口 -> 首选项 -> VAGen 迁移 -> VAGen 迁移语法首选项。

### • 重命名首选项

- **重命名前缀。**每当 VAGen 数据项、记录、函数或映射名是 EGL 或 SQL 保留字或者以符号 # 开头时，迁移工具都使用此前缀。此工具将对 VAGen 部件名添加“重命名前缀”以创建有效的 EGL 部件名。例如，`date` 是 EGL 保留字。如果存在名为 `DATE` 的函数，并且使用缺省重命名前缀 `VAGen_`，则迁移工具将把函数 `DATE` 更改为 `VAGen_DATE`。此工具还将把对该函数的所有引用由 `DATE` 更改为 `VAGen_DATE`。您可以将重命名前缀设置为除空白或 `EZE` 以外的任意值。并且，重命名前缀不能以符号 # 开头。请您确保选择的内容不会导致与任何部件名发生冲突。
- **Level-77 后缀。**每当 VAGen 工作存储器记录包含 Level-77 项时，迁移工具都将使用此后缀。EGL 不支持 Level-77 项。迁移工具将 VAGen 工作存储器记录分割成两个 EGL 记录。第一个记录与原始 VAGen 工作存储器记录同名并包含所有非 Level-77 项。第二个记录包含所有 Level-77 项。迁移工具通过将原始工作存储器记录名与 Level-77 后缀连接到一起来对第二个记录命名。例如，如果原始工作存储器记录名为 `MYRECORD`，并且您使用缺省 Level-77 后缀 `_Level77Items`，则包含 Level-77 项的 EGL 记录将被命名为 `MYRECORD_Level77Items`。您可以将 Level-77 后缀设置为除空白以外的任意值。但是，请确保选择的内容不会导致与任何部件名发生冲突。
- **帮助映射后缀。**每当程序的主映射组和帮助映射组包含同名的映射时，迁移工具就使用此后缀。EGL 要求程序的这两个 `formGroup` 中的所有表单都具有唯一的名称。迁移工具将把帮助组中那些与主映射组中的映射名有冲突的映射重命名。考虑以下示例。程序的主映射组是 `MAPGP1`，并且它包含 `MAP1`。同一程序的帮助映射组是 `MAPGP2`，并且它包含 `MAP1`。通过使用缺省帮助映射后缀 `_helpmap`，迁移工具将 `MAPGP2` 中的 `MAP1` 重命名为 `MAP1_helpmap`。您可以将帮助映射后缀设置为除空白以外的任意值。但是，请确保选择的内容不会导致与任何部件名发生冲突。

### • SQL 首选项

- **结果集后缀。**在 VisualAge Generator 中，当 SQL `REPLACE` 函数需要引用相应的 `UPDATE` 或 `SETUPD` 函数时，`REPLACE` 函数必须指定函数名。在 EGL 中，在一个函数中支持多个 I/O。并且，使用结果集标识来唯一地标识 `get` 或 `open` 语句。迁移工具通过将函数名与结果集后缀连接到一起来创建结果集标识。例如，

如果函数名为 MY-SETUPD，并且您使用缺省结果集后缀 \_RSI01，则包括在该函数的 open 语句中的结果集标识将是 MY-SETUPD\_RSI01。您可以将结果集后缀设置为除空白以外的任意值。但是，请确保选择的内容不会导致与任何部件名发生冲突。

- 准备后缀。在 VisualAge Generator 中，如果需要在运行时准备 SQL I/O 语句，则请选择“执行时语句构建”。对应的 EGL 语句是 *prepare* 语句。prepare 语句包含一个 *prepare* 语句标识，因此，其它 I/O 语句（如 *close*、*get*、*execute* 和 *open*）可以指定与它们相关联的 *prepare* 语句。迁移工具通过将函数名与准备后缀连接到一起来创建 *prepare* 语句标识。例如，如果存在名为 MY-EXEC-TIME-BUILD 的函数，并且您使用缺省准备后缀 \_PREP01，则包含在 *prepare* 语句中的 *prepare* 语句标识是 MY-EXEC-TIME-BUILD\_PREP01。您可以将准备后缀设置为除空白以外的任意值。但是，请确保选择的内容不会导致与任何部件名发生冲突。
  - 省略列名。在 VisualAge Generator 中，始终对 SQL 记录中的字段指定 SQL 列名。在 EGL 中，如果 SQL 列名与字段名相同，则可以省略该列名。如果您选择了“省略列名”，则每当 SQL 列名与字段名相同时，迁移工具将省略 EGL column 属性。通过省略列名，可以使 EGL 源代码更加清晰明了。
  - 省略 *isNullable* 属性。在 VisualAge Generator 中，对于 SQL 记录中的每个字段，始终包括空指示符变量。为了保留 VisualAge Generator 的行为，对于 SQL 记录中的每个字段，迁移工具始终包括 *isNullable* 属性。但是，在 EGL 中，如果列在 SQL 中被定义为非空列，则可以省略空指示符变量。如果您选择了“省略 *isNullable* 属性”，则迁移工具将对 SQL 记录中的每个字段省略 *isNullable* 属性。通过省略 *isNullable* 属性，可以提高运行时性能。但是，仅当您确定所有 SQL 列都被定义为非空列时，才应该选择“省略 *isNullable* 属性”。如果任何 SQL 列可以为空，在迁移期间就不应该选择“省略 *isNullable* 属性”。在进行迁移后，如果您想提高性能，则可以对 SQL 记录中的所选字段除去 *isNullable* 属性。
  - 省略 *isReadOnly* 属性。在 VisualAge Generator 中，必须对 SQL 记录中的每个字段显式地设置只读属性。如果对 SQL 记录指定了多个表，则“只读”必须始终为 *yes*。缺省情况下，迁移工具对 SQL 记录中的每个字段都包括 *isReadOnly* 属性。但是，如果只对 SQL 记录指定了一个 SQL 表，则 EGL *isReadOnly* 属性将缺省为 *no*，如果对 SQL 记录指定了多个 SQL 表，则该属性将缺省为 *yes*。如果您选择了“省略 *isReadOnly* 属性”，则仅当对 SQL 记录指定了一个表并且 VisualAge Generator 的“只读”属性设置为 *yes* 时，迁移工具才包括 *isReadOnly*。通过省略 *isReadOnly* 属性，可以使 EGL 源代码更加清晰明了。
- 其它转换选项
    - 将共享数据项转换为基本项定义。如果选择此首选项，则每当在记录、表、函数本地存储器、函数参数列表或程序参数列表中使用共享数据项时，迁移工具都将把共享项转换为基本定义。如果当前组织机构标准不鼓励在新应用程序中使用共享数据项，则此选项使您在进行迁移时能够从现有应用程序中除去共享数据项的使用。
    - 将逗号小数点更改为句点小数点。如果工作站指定了通常使用逗号来指示小数点位置的语言环境，则迁移工具将把数字文字自动转换为使用点。例如，对于法语和德语语言环境，迁移工具自动地将逗号转换为点。但是，某些客户指定英语作为他们的语言环境并将通常的小数点设置重设为逗号。如果您使用这种技术，则必须选择“将逗号小数点更改为句点小数点”首选项。这将确保迁移工具将把数字文字中的逗号转换为点。EGL 只支持将点用作小数点位置指示符。

## 其它建议设置

建议进行下列附加设置:

- (可选) 关闭“欢迎”视图。
- 如果您尚未使用 EGL 透视图, 则通过选择窗口 -> 打开透视图 -> 其它 -> **EGL** -> **确定**来切换到该透视图。此外, 如果您计划开发新的 EGL pageHandler, 则应该切换到 Web 透视图。可以通过右键单击窗口右上角选项卡上的透视图图标并接着选择**关闭**来关闭其它透视图。
- 如果“导航器”视图尚未包括在您正在使用的透视图, 则您可能想添加此视图。要添加“导航器”视图, 请选择窗口 -> 显示视图 -> 其它。从“显示视图”窗口中, 展开**基本**, 然后选择**导航器**。

**注:** 某些活动(如删除 EGL 包)在“导航器”视图中并不完全受支持。如果您正在重构项目或包, 请始终使用“项目资源管理器”视图。

- “问题”视图将显示任何验证错误。因此, 您可能想将“问题”视图设置为位于前台。在“工作台”窗口的右下部分中, 选择“问题”选项卡以将“问题”视图置于前台。
- 在“问题”视图中, 选择右上角的**过滤器...**图标。在**过滤器**窗口中, 请执行下列操作:
  - 选择**仅对选择的资源**单选按钮。这将把“问题”视图中的错误消息限制为仅包括当前所选文件的消息。当存在错误时, 这样做可以帮助您将注意力每次集中在一个文件上。“问题”视图的标题栏提供了与过滤器匹配的消息的计数以及工作空间中所有项目的消息总数。
  - 取消选择**将可视项限制为**。这使您能够查看所有消息。
- 当打开了多个文件进行编辑时, 您可以对“导航器”视图进行配置, 以便当您每次在“导航器”视图中选择打开的文件时, 该视图都会自动地将该文件显示到前台(使该文件的编辑器会话成为活动编辑器)。要完成此任务, 请选择“导航器”视图工具栏中的**与编辑器链接**图标。您也可能想选择“项目资源管理器”视图工具栏中的**与编辑器链接**图标。

---

## 设置第 2 阶段 VAGen 迁移文件

可以通过向导来调用执行第 2 阶段迁移的工具。为了为第 2 阶段作准备, 请创建一个项目, 在该项目中, 您可以选择保存第 2 阶段首选项以备后用。要创建第 2 阶段迁移首选项, 请执行下列操作:

1. 启动 EGL 开发环境。务必设置工作台首选项, 如第 123 页的『设置工作台首选项』一节所述。
2. 第 2 阶段向导将询问数据库驱动程序位置。您可以将类路径变量设置为包含此值以使向导能够自动选用此值。执行此操作的最简单方法如下所示:
  - a. 在窗口 -> 首选项下面, 选择 **Java** -> **构建路径** -> **类路径变量**。
  - b. 单击**新建**按钮。
  - c. 对于**名称**, 输入以下内容: **DB2\_DRIVER\_PATH**
  - d. 对于**路径**, 单击**文件**按钮并浏览到 db2java.zip 文件。(这就是您在第 1 阶段使用的那个 db2java.zip 文件。缺省情况下, 此文件是 \SQLLIB\java\db2java.zip。)
  - e. 在“新建变量条目”窗口中单击**确定**, 然后在“首选项”窗口中单击**确定**。

3. 如果您选择保存第 2 阶段首选项文件，则创建一个简单的可以包含该首选项文件的项目。如果要以批处理方式运行第 2 阶段，这样做就很有用。通过选择文件 -> 新建 -> 项目来为此任务启动向导。展开简单并选择项目，接着选择下一步。
4. 对项目指定名称。例如，VAGENMIG。然后，单击完成。
5. 在“项目资源管理器”视图中，在“其它项目”文件夹中列示了一个简单项目。在“导航器”视图中，将有一个简单项目按字母顺序与所有其它项目列示在一起。
6. 右键单击该项目，然后从上下文菜单中选择新建 -> 其它。
7. 展开 VAGen 迁移到 EGL，然后选择 VAGen 迁移文件。单击下一步。
8. 输入适当的第 2 阶段首选项：
  - a. 在向导的第一页上，按照下表的描述编辑首选项。在您通过 Tab 键离开字段之前，迁移工具不会对任何“数据库信息”字段进行验证。这样可以防止在您输入信息时多次尝试连接至数据库。

表 56. 在向导第一页上输入的首选项

首选项	含义	值
装入现有文件	这允许您选择先前保存的第 2 阶段首选项文件。单击选择文件按钮以选择现有的 .vgmig 文件。单击装入文件按钮以从该文件中检索首选项并将那些首选项显示在向导中。	(可选) 选择并装入现有的 .vgmig 文件。
数据库驱动程序位置	这是 DB2 驱动程序的位置。	<i>path_to_db2java.zip</i> \db2java.zip
数据库驱动程序	这是 DB2 驱动程序的名称。	此值必须总是 COM.ibm.db2.jdbc.app.DB2Driver。此值对于本地数据库或者以本地方式进行编目的远程数据库都适用。
数据库名称	这是在第 1 阶段迁移中使用的 DB2 数据库的名称。	此值应该具有以下格式： <ul style="list-style-type: none"> <li>• jdbc:DB2:databaseName</li> </ul> databaseName 是在第 1 阶段迁移中使用的 DB2 数据库的名称。对于第 1 阶段来说，缺省值是 VGMIG。
数据库模式	这是在第 1 阶段迁移中使用的 DB2 数据库模式的名称。	此值是在第 1 阶段迁移中使用的 DB2 模式的名称。对于第 1 阶段来说，缺省值是 MIGSCHEMA。
数据库用户标识	这是在第 1 阶段迁移中使用的数据库用户标识。	请使用您在第 1 阶段使用的数据库用户标识。(缺省值是您的登录标识。)
数据库密码	这是在第 1 阶段迁移中使用的数据库密码。	请使用您在第 1 阶段使用的数据库密码。(缺省值是您的登录密码。)
日志文件位置	这是日志文件将写至的位置。	输入文件系统中的有效位置(驱动器和目录)。
日志文件名	这是要将第 2 阶段消息写至的日志文件的名称。	输入有效的文件名。

- b. 在向导的第二页上，按照下表的描述来编辑首选项：

表 57. 在向导第二页上输入的首选项

首选项	含义	值
Java 或 COBOL 单选按钮	此选项确定迁移工具是否创建包含 Java 源文件夹的项目。	如果您正在使用 Rational Web Developer 或 Rational Application Developer, 则始终应该选择 Java 单选按钮。  对于其它产品, 如果您仅计划生成 COBOL, 则可以选择 COBOL。
迁移余下的 VAGen 部件	此选项确定是否将未被迁移集中任何可生成部件引用的部件转换为 EGL。	选取此框以将未引用的部件转换为 EGL。通常, 您应该选择“迁移余下的部件”。如果您未选择“迁移余下的部件”, 则不会将迁移集中未使用的控制部件以及任何其它部件迁移为 EGL 源代码。
导入到工作空间中	此选项确定在第 2 阶段完成后是否自动启动第 3 阶段 (将 EGL 导入到当前工作空间中的文件中)。注: 如果选取此框, 则必须选择此复选框下面的其中一个单选按钮以指定要导入的版本 (最新版本或最旧版本 - 请参阅下面的描述), 这是因为工作空间每次只能包含一个项目版本。	选择此框以便在将部件转换为 EGL 之后直接导入 EGL 文件。如果要到以后 (在第 3 阶段) 才导入文件, 请保留此框处于未选取状态。 注: • 如果您选择了此选项, 则不需要单独地运行第 3 阶段。在第 2 阶段 (转换) 完成后, 迁移工具将直接地自动启动第 3 阶段 (导入) 并完成迁移过程。
最新版本	此选项指定应该导入所需迁移集的最新版本。	仅当选取了导入到工作空间中复选框时, 才可以选择此选项。
最旧版本	此选项指定应该导入所需迁移集的最旧版本。	仅当选取了导入到工作空间中复选框时, 才可以选择此选项。
覆盖现有文件	第 3 阶段 (导入过程) 使用由第 2 阶段生成的 EGL 来创建并导入第 1 阶段报告中指定的 EGL 文件。如果在工作空间中已存在与第 3 阶段将要导入的 EGL 文件同名的 EGL 文件, 则此选项确定是否覆盖那些文件。	仅当选取了导入到工作空间中复选框时, 才可以选择此选项。覆盖现有文件使您能够指定要让第 3 阶段迁移工具如何处理以下情况: 当前正在迁移的迁移集包含一些部件, 那些部件应该放到已存在于工作空间中的文件中。如果您选择了覆盖现有文件, 则第 3 阶段迁移工具将替换现有文件并且仅包括当前迁移集所包含的那些部件。如果您未选择覆盖现有文件, 则第 3 阶段迁移工具将把任何新部件合并到现有文件中。新部件是按部件类型的字母顺序放置的。请参阅第 34 页的『覆盖与合并文件』以获取有关此选项的作用的更完整讨论。

表 57. 在向导第二页上输入的首选项 (续)

首选项	含义	值
将迁移的文件保存至临时目录	此选项用来将 EGL 文件保存到文件系统中的某个位置。这使您能够同时访问多个项目版本的 EGL 文件。(在工作空间中, 您每次只能看到一个版本。) 您可以将 EGL 文件从这里直接移到资源库中。	如果您计划迁移多个迁移集版本, 则执行下列操作: <ul style="list-style-type: none"> <li>• 选择此框, 以便可以将每个版本写入不同的子目录。</li> <li>• 指定一个文件夹, 版本的子目录将放到该文件夹下。</li> <li>• 不要选择<b>立即迁移</b>。由于资源需求方面的原因, 迁移至临时目录的操作只应该以批处理方式完成。如果您选择了<b>立即迁移</b>, 则迁移工具将要求您确认确实要以联机方式运行。</li> <li>• 选择<b>将当前配置保存至文件</b>。您还必须指定项目和文件名, 当前配置将作为 .vgmig 文件被保存在该位置中。无论是以联机方式进行迁移还是以批处理方式进行迁移, 只要选择了<b>将迁移的文件保存至临时目录</b>, .vgmig 文件就是必需的。如果以批处理方式运行第 2 阶段, 请指向已保存的 .vgmig 文件以指定迁移首选项。</li> </ul>
文件夹	这是用来保存 EGL 文件的目录。每个迁移集版本都将变为您对“文件夹”指定的目录下的一个子目录。	指定文件系统中的现有目录。
立即迁移	此选项指定您想要立即运行第 2 阶段, 而不是仅仅设置首选项文件以便将来进行迁移。	您必须选择 <b>立即迁移</b> 才能以联机方式运行第 2 阶段, 您也可以选择 <b>将当前配置保存至文件</b> 以保存首选项, 以便将来可以以批处理方式运行第 2 阶段。如果要保留首选项的副本以供将来参考, 则可以同时选择这两个选项。如果已经选择了 <b>将迁移的文件保存至临时目录</b> , 则不要选择 <b>立即迁移</b> 。保存至临时目录的操作只能以批处理方式完成。选择 <b>立即迁移</b> 表示您想要以联机方式进行迁移。
将当前配置保存至文件	此选项使您能够将指定的首选项保存至文件。以后, 可以通过下列其中一种方法来运行第 2 阶段: <ul style="list-style-type: none"> <li>• 在联机方式下, 右键单击已保存的 .vgmig 文件并从上下文菜单中选择<b>启动迁移</b>。</li> <li>• 在批处理方式下, 使用 -importFile 选项来指定已保存的 .vgmig 文件。要了解详细信息, 请参阅第 133 页的『以批处理方式运行第 2 阶段』。</li> </ul>	您必须选择 <b>立即迁移</b> 才能以联机方式运行第 2 阶段, 您也可以选择 <b>将当前配置保存至文件</b> 以保存首选项, 以便将来可以运行第 2 阶段。如果要保留首选项的副本以供将来参考, 则可以同时选择这两个选项。  如果选择了此选项, 则还必须指定“路径”和“文件名”, 当前配置将作为 .vgmig 文件被保存到该位置中。以后, 在运行第 2 阶段时, 请指向已保存的 .vgmig 文件以指定迁移首选项。
路径	此选项指定一个项目, 文件应该被保存到该项目中。	\projectName, 其中 projectName 是要包含所保存的文件的项目的名称。
文件名	此选项指定要将首选项保存至的文件的名称。	fileName, 其中 fileName 是文件的期望名称, 并且不带文件扩展名。将会自动地追加扩展名 .vgmig。

- c. 在向导的第三页上, 向导列示了您指定的数据库中的迁移集。请选择要迁移的迁移集。如果您未选择任何迁移集, 则迁移工具将迁移该数据库中的所有迁移集。单击**完成**。

您选择的复选框组合确定了向导执行的操作:

- 如果选择了将当前配置保存至文件, 则当您单击“完成”按钮之后, 所有选项都将被保存到指定的文件中。
- 如果选择了立即迁移, 则当您单击“完成”按钮后, 将运行第 2 阶段迁移。
- 如果还选择了导入到工作空间中 / 或将迁移的文件保存至临时目录, 则在第 2 阶段完成后, 第 3 阶段将自动启动。

下面是第 2 阶段首选项文件 (stage2.vgmig) 的一个示例:

```
databaseDriverLocation=d:\SQLLIB\java\java\db2java.zip
databaseDriver=COM.ibm.db2.jdbc.app.DB2Driver
databaseName=jdbc:DB2:VGMIG
databaseSchema=MIGSCHEMA
databaseUserid=myuserid
databasePassword=encoded:AAEDAwQFBwYKCwo+Pw==
configurationPlan=MyMigrationSetA,1.1
migrateRemainingParts=yes
workspaceImport=latest
overrideExistingFiles=yes
tempDirectory=
logFileLocation=D:\tempmig\MyMigrationSetA\stage2\MyMigrationSetA.log
migrateNow=yes
projectType=COBOL
```

---

## 运行第 2 阶段

可以按批处理方式运行第 2 阶段迁移工具, 也可以从 EGL 开发环境中的用户界面运行它。

### 从用户界面中运行第 2 阶段

第 128 页的『设置第 2 阶段 VAGen 迁移文件』中描述的向导提供了用于将首选项保存在 .vgmig 文件中的选项。如果您在该向导中选择了立即迁移框, 则当您在向导中单击“完成”按钮时, 第 2 阶段迁移将启动。如果您在该向导中未选择立即迁移框, 则当您准备好使用已保存的 .vgmig 文件来运行第 2 阶段迁移时, 请执行下列操作:

1. 从“导航器”视图中, 通过右键单击项目名称旁边的 + 符号来展开包含第 2 阶段首选项文件的项目。
2. 右键单击 .vgmig 首选项文件以打开上下文菜单。
3. 单击启动迁移。

第 2 阶段迁移将启动并将指定的迁移集的外部源格式文件转换为 EGL 源代码, 然后将 EGL 源代码存储在迁移数据库中。如果您选择了导入到工作空间中或将迁移的文件保存至临时目录, 则在第 2 阶段完成之后, 第 3 阶段将自动启动。在第 3 阶段完成后, 迁移工具将自动启动工作空间刷新步骤。这个刷新步骤会花费一些时间, 当存在大量部件时尤其如此。当刷新步骤完成时, 会出现一个弹出窗口, 该窗口告诉您迁移已完成。请您务必等待该弹出窗口出现。

当迁移和刷新步骤完成后, 将有下列输出:

- 第 2 阶段迁移日志文件。日志文件位于作为“日志文件”位置指定的目录中。日志文件包含关于已迁移的部件的信息以及在第 2 阶段迁移期间发出的任何参考消息、警告消息或错误消息。

- 迁移集的任务列表日志文件。这个任务列表文件是在第 3 阶段开始时创建的，它包含由第 2 阶段生成的消息的合并列表，那些消息可能要求您执行其它任务以完成迁移。在“每个可生成部件及其关联项的消息都是成组列示的”这一方面，任务列表与 VisualAge Generator 生成消息有点类似。如果某个部件具有消息，并且它是数个程序的关联项，则那些消息对每个程序都列示一次。任务列表与 VisualAge Generator 生成消息的区别在于，未使用的不可生成部件的消息按项目、包和文件名列示在任务列表末尾。任务列表与第 2 阶段迁移日志文件放在同一个目录中。
- 如果您选择了**导入到工作空间中**，则第 3 阶段将自动启动并创建迁移集所需的 EGL 项目、源文件夹、包和 EGL 文件。第 3 阶段工具还将项目导入到工作空间中并重建该项目，以便能够运行 EGL 验证。
- 如果您选择了**将迁移的文件保存至临时目录**，则第 3 阶段将自动启动并创建迁移集所需的 EGL 项目、源文件夹、包和 EGL 文件。第 3 阶段工具将把这些项目放入指定的临时目录。当在临时目录中创建项目时，第 3 阶段迁移工具将在项目名后面追加 VAGen 版本名。这使您能够一次迁移多个项目版本以便以后将它们导入到工作空间中。

## 以批处理方式运行第 2 阶段

第 2 阶段向导使您能够为执行立即迁移而选择一个或多个迁移集。它还使您能够将信息保存在文件中，以便将来以批处理方式迁移该信息。要为将来的迁移创建文件，请执行下列操作：

1. 执行第 128 页的『设置第 2 阶段 VAGen 迁移文件』中描述的步骤，但是，还执行下列操作：
  - 选择将**当前配置保存至文件**并指定路径和文件名。自动创建的文件具有后缀 `.vgmig`，这个文件就是您以批处理方式运行第 2 阶段或第 3 阶段时需要到 `-importFile` 指定的文件。
  - 确保取消选择**立即迁移**。取消选择此选项表示您想要为以后执行的迁移保存信息。
  - 您可以定义多个 `.vgmig` 文件以便将来一整批地进行迁移。
2. 创建具有 `.bat` 文件扩展名的文件。

对于 Windows 环境来说，`.bat` 文件的内容应该是：

```
set path=InstallDirectory\eclipse\jre\bin;%path%
set classpath=InstallDirectory\eclipse\startup.jar;
               InstallDirectory\egl\eclipse\plugins\
               com.ibm.etools.egl.vagenmigration_<i>version</i>\runtime\eglMigration.jar;
cd InstallDirectory
java com.ibm.etools.egl.internal.vagenmigration.batch.VGMIG
   -importFile Path\vgmigFileName.vgmig
   -data Path\workspace
   >Path\LogName.log
```

对于 Linux 环境来说，`.bat` 文件的内容应该是：

```
# !/bin/bash
export PATH=InstallDirectory/eclipse/jre/bin/:$PATH
export CLASSPATH=InstallDirectory/eclipse/startup.jar;
                 InstallDirectory/egl/eclipse/plugins/
                 com.ibm.etools.egl.vagenmigration_<i>version</i>/runtime/eglMigration.jar:$CLASSPATH
cd InstallDirectory/
```

```
java com.ibm.etools.egl.internal.vagenmigration.batch.VGMIG
-importFile /Path/vgmigFileName.vgmig
-data /Path/workspace
>/Path/LogName.log
```

此外，对于 Linux 环境来说，必须更改该 .bat 文件的许可权以使其可以执行。要使该 .bat 文件可以执行，请打开“命令终端”并输入以下命令：

```
chmod u+x mybatchfile.bat
```

*mybatchfile.bat*

已创建的 .bat 文件的名称。

注：

- 下列语句必须写在一行上：
  - 对于 Windows 环境来说，是 **set classpath** 语句。
  - 对于 Linux 环境来说，是 **export classpath** 语句。
  - **java** 语句。
- 对于要以批处理方式迁移的每个 .vgmig 文件，将 **java** 语句重复一次。但是，在创建任何 .vgmig 文件时，如果选择了导入到工作空间中，则确保没有任何 .vgmig 文件导致相同的 EGL 项目名。如果尝试为同一个 .bat 文件中的同一个 EGL 项目迁移多个 .vgmig 文件，则 EGL 项目将仅反映最后一个要迁移的 .vgmig 文件。
- *InstallDirectory* 是安装了 Rational Developer 或 WebSphere Studio 产品的驱动器和目录。必须将 *InstallDirectory* 包括在 path 语句、classpath 语句和 cd（切换目录）语句中。

注：如果安装您现在正在使用的产品前您已经安装并保留了先前版本的开发者产品，则您想使用的安装目录可能是先前进行安装时使用的目录。

- *version* 是插件版本号（例如，6.0 或 6.0.0.1）。通常，您应该使用所看到的 *com.ibm.etools.egl.vagenmigration\_version* 插件的最高版本号。
- *Path\vgmigFileName.vgmig* 是指一个 .vgmig 文件的驱动器、目录和文件名，该 .vgmig 文件指定了要从迁移数据库中迁移的迁移集。该目录必须包含工作空间名。这就是您在步骤 1 中保存的 .vgmig 文件（例如，d:\myworkspace\mySimpleProject\myMigrationInformation.vgmig）。
- *Path\workspace* 是用来放置 EGL 文件的驱动器、目录和工作空间名。（例如，工作空间可以是 d:\myworkspace。）迁移集使用的任何 EGL 项目和包都是由迁移工具自动创建的。*-data* 参数是可选的；仅当需要指定不同于缺省值的 VAGen 迁移语法首选项时，*-data* 才是必需的。如果要设置任何 VAGen 迁移语法首选项，则在运行 .bat 文件之前，必须在 *-data* 参数指定的工作空间中指定它们。要了解有关如何设置首选项的信息，请参阅第 126 页的『VAGen 迁移语法首选项』。
- *Path\LogName.log* 指向要为 java 命令创建的日志文件的驱动器、目录和文件名。此日志文件列示任何与 java 命令本身相关的问题。由第 2 阶段或第 3 阶段生成的任何日志消息都将放到一个日志文件中，该日志文件是在向导的第一页上指定的，以后，它将被保存到 .vgmig 文件中。如果在同一个 .bat 文件中包括了多个 java 命令，则确保为每个 java 命令指定不同的日志文件名。

对于 Windows 环境来说，java 命令示例可能与下面的内容相似，但是该命令的所有内容应该位于一行上：

```
java com.ibm.etools.egl.internal.vagemigration.batch.VGMIG
  -importFile d:\myworkspace\mySimpleProject\myMigrationInformation.vgmig
  -data d:\myworkspace\
  >d:\migrationLogs\myMigrationInformationPiped.log
```

对于 Linux 环境来说，java 命令示例可能与下面的内容相似，但是该命令的所有内容应该位于一行上：

```
java com.ibm.etools.egl.internal.vagemigration.batch.VGMIG
  -importFile /d:/myworkspace/mySimpleProject/myMigrationInformation.vgmig
  -data /d:/myworkspace
  >/d:/migrationLogs/myMigrationInformationPiped.log
```

3. 关闭 EGL 开发环境。
4. 打开“命令提示符”窗口，转至包含该 .bat 文件的目录并运行该 .bat 文件。

**注：**您可以安全地忽略以下消息：*PolicyClassLoader* 找不到策略 *com.ibm.jxesupport.JxeClassLoaderPolicy*。

5. 此过程完成后，EGL 项目、源文件夹、包和文件将分别存储在您为它们指定的目录中。与每个 java 命令对应的日志文件包含已迁移的部件列表以及任何错误消息。这些消息就是当您使用用户界面来运行第 2 阶段时写入日志文件的消息。同样，任务列表文件包含当您使用用户界面来运行第 2 阶段时写至此文件的消息。
6. 启动 EGL 开发环境。
7. 您应该会看到工作空间中的 EGL 项目、源文件夹、包和文件。



## 第 7 章 第 3 阶段 - 导入

迁移的第 3 阶段也是通过 EGL 提供的插件运行的。在此阶段，您运行另一个迁移工具，该迁移工具根据第 2 阶段存储在迁移数据库中的 EGL 语法来构建 EGL 文件。

### 运行第 3 阶段工具

从 EGL 开发环境的“工作台”窗口中，执行下列操作：

1. 从“文件”菜单中，选择导入。
2. 选择从数据库进行 VAGen 迁移并单击下一步。
3. 为此阶段的迁移指定首选项：
  - a. 在向导的第一页上，按照下表的描述编辑首选项。在您通过 Tab 键离开字段之前，迁移工具不会对任何“数据库信息”字段进行验证。这样可以防止在您输入信息时多次尝试连接至数据库。

表 58. 在向导第一页上输入的首选项

首选项	含义	值
装入现有文件	这允许您选择先前保存的第 3 阶段首选项文件。单击“选择文件”按钮以选择现有的 .vgmig 文件。单击“装入文件”按钮以从该文件中检索首选项并将那些首选项显示在向导中。	(可选) 选择并装入现有的 .vgmig 文件。
数据库驱动程序位置	这是 DB2 驱动程序的位置。	<i>path_to_db2java.zip</i> \db2java.zip
数据库驱动程序	这是 DB2 驱动程序的名称。	此值必须总是 COM.ibm.db2.jdbc.app.DB2Driver。此值对于本地数据库或者以本地方式进行编目的远程数据库都适用。
数据库名称	这是在第 1 阶段迁移中使用的 DB2 数据库的名称。	此值应该具有以下格式： • jdbc:DB2: <i>databaseName</i> <i>databaseName</i> 是在第 1 阶段使用的 DB2 数据库的名称。对于第 1 阶段来说，缺省值是 VGMIG。
数据库模式	这是在第 1 阶段迁移中使用的 DB2 数据库模式的名称。	此值是在第 1 阶段使用的 DB2 模式的名称。对于第 1 阶段来说，缺省值是 MIGSCHEMA。
数据库用户标识	这是在第 1 阶段迁移中使用的数据库用户标识。	请使用您在第 1 阶段使用的数据库用户标识。(缺省值是您的登录标识。)
数据库密码	这是在第 1 阶段迁移中使用的数据库密码。	请使用您在第 1 阶段使用的数据库密码。(缺省值是您的登录密码。)
日志文件位置	这是日志文件将写至的位置。	输入文件系统中的有效位置(驱动器和目录)。
日志文件名	这是要将第 3 阶段消息写至的日志文件的名称。	输入有效的文件名。

- b. 在向导的第二页上，按照下表的描述来编辑第二页上的首选项：

表 59. 在向导第二页上输入的首选项

首选项	含义	值
Java 或 COBOL 单选按钮	此选项确定迁移工具是否创建包含 Java 源文件夹的项目。	如果您正在使用 Rational Web Developer 或 Rational Application Developer, 则始终应该选择 Java 单选按钮。  对于其它产品, 如果您仅计划生成 COBOL, 则可以选择 COBOL。
最新版本	此选项指定应该导入所需迁移集的最新版本。	选择其中一个单选按钮。
最旧版本	此选项指定应该导入所需迁移集的最旧版本。	选择其中一个单选按钮。
覆盖现有文件	第 3 阶段 (导入过程) 使用由第 2 阶段生成的 EGL 来创建并导入第 1 阶段报告中指定的 EGL 文件。如果在工作空间中已存在与第 3 阶段将要导入的 EGL 文件同名的 EGL 文件, 则此选项确定是否覆盖那些文件。	覆盖现有文件使您能够指定要让第 3 阶段迁移工具如何处理以下情况: 当前正在迁移的迁移集包含一些部件, 那些部件应该放到已存在于工作空间中的文件中。如果您选择了覆盖现有文件, 则第 3 阶段迁移工具将替换现有文件并且仅包括当前迁移集所包含的那些部件。如果您未选择覆盖现有文件, 则第 3 阶段迁移工具将把任何新部件合并到现有文件中。新部件是按部件类型的字母顺序放置的。请参阅第 34 页的『覆盖与合并文件』以获取有关此选项的作用的更完整讨论。
将迁移的文件保存至临时目录	此选项用来将 EGL 文件保存到文件系统中的某个位置。这使您能够同时访问多个项目版本的 EGL 文件。(在工作空间中, 您每次只能看到一个版本。) 您可以将 EGL 文件从这里直接移到资源库中。	如果您计划迁移多个迁移集版本, 则执行下列操作: <ul style="list-style-type: none"> <li>• 选择此框, 以便可以将每个版本写入不同的子目录。</li> <li>• 指定一个文件夹, 版本的子目录将放到该文件夹下。</li> <li>• 不要选择立即迁移。由于资源需求方面的原因, 迁移至临时目录的操作只应该以批处理方式完成。如果您选择了立即迁移, 则迁移工具将要求您确认确实要以联机方式运行。</li> <li>• 选择将当前配置保存至文件。您还必须指定项目和文件名, 当前配置将作为 .vgmig 文件被保存在该位置中。无论是以联机方式进行迁移还是以批处理方式进行迁移, 只要选择了将迁移的文件保存至临时目录, .vgmig 文件就是必需的。如果以批处理方式运行第 3 阶段, 请指向已保存的 .vgmig 文件以指定迁移首选项。</li> </ul>
文件夹	这是用来保存 EGL 文件的目录。每个迁移集版本都将变为您对“文件夹”指定的目录下的一个子目录。	指定文件系统中的现有目录。
立即迁移	此选项指定您想要立即运行第 3 阶段, 而不是仅仅设置首选项文件以便将来进行迁移。	您必须选择立即迁移才能以联机方式运行第 3 阶段, 您也可以选择将当前配置保存至文件以保存首选项, 以便将来可以以批处理方式运行第 3 阶段。如果要保留首选项的副本以供将来参考, 则可以同时选择这两个选项。如果已经选择了将迁移的文件保存至临时目录, 则不要选择立即迁移。保存至临时目录的操作只能以批处理方式完成。选择立即迁移表示您想要以联机方式进行迁移。

表 59. 在向导第二页上输入的首选项 (续)

首选项	含义	值
将当前配置保存至文件	<p>此选项使您能够将指定的首选项保存至文件。以后，可以通过下列其中一种方法来运行第 3 阶段：</p> <ul style="list-style-type: none"> <li>在联机方式下，右键单击已保存的 .vgmig 文件并从上下文菜单中选择 <b>启动迁移</b>。</li> <li>在批处理方式下，使用 <code>-importFile</code> 选项来指定已保存的 .vgmig 文件。要了解详细信息，请参阅第 133 页的『以批处理方式运行第 2 阶段』。</li> </ul>	<p>您必须选择 <b>立即迁移</b> 才能以联机方式运行第 3 阶段，您也可以选择 <b>将当前配置保存至文件</b> 以保存首选项，以便将来可以运行第 3 阶段。如果要保留首选项的副本以供将来参考，则可以同时选择这两个选项。</p> <p>如果选择了此选项，则还必须指定“路径”和“文件名”，当前配置将作为 .vgmig 文件被保存到该位置中。以后，在运行第 3 阶段时，请指向已保存的 .vgmig 文件以指定迁移首选项。</p>
路径	指定一个项目，文件应该被保存到该项目中。	<code>\projectName</code> ，其中 <i>projectName</i> 是要包含所保存的文件的项目的名称。
文件名	此选项指定要将首选项保存至的文件名称。	<i>fileName</i> ，其中 <i>fileName</i> 是文件的期望名称，并且不带文件扩展名。将会自动地追加扩展名 <code>.vgmig</code> 。

c. 在向导的第三页上，选择要导入的迁移集。

**注：**“VAGen 迁移 - 从数据库导入”向导仅列示已被迁移到 EGL 的迁移集。这将确保您在运行第 3 阶段之前运行第 2 阶段以将迁移集转换为 EGL 源代码并将 EGL 存储到迁移数据库中。

4. 单击完成。

5. 迁移工具将根据您选择的迁移集来创建 EGL 项目、EGL 源文件夹和 EGL 包。此工具从迁移数据库中抽取 EGL 源代码并根据迁移集创建 EGL 文件。迁移工具还包括 `import` 语句并更新项目的 EGL 构建路径，以便可以解析部件引用。

## 以批处理方式运行第 3 阶段

以批处理方式运行第 2 阶段与第 3 阶段的唯一差别是：用来创建 .vgmig 文件的向导是不同的。请参阅第 137 页的『运行第 3 阶段工具』以了解有关设置 .vgmig 文件以便仅运行第 3 阶段的详细信息。请参阅第 133 页的『以批处理方式运行第 2 阶段』以了解有关包括在 .bat 文件中的命令的详细信息以及可以对批处理方式指定的选项的描述。

## 使用已写至临时目录的迁移集

如果将第 3 阶段的输出定向到一个临时目录，则迁移工具将为每个迁移集版本创建一个子目录。子目录名的格式为 `migrationSetName_versionName`。可以使用下列任何一种技术来将项目导入到工作空间中：

- 如果在子目录中只有几个项目，则您可以使现有工作空间指向每个项目。这种技术并不会将项目复制到工作空间中；它仅仅是使项目可供工作空间使用。当您修改或删除项目中的文件时，将在工作空间所指向的文件系统目录中进行更改。

1. 从现有工作空间中，选择 **窗口 -> 首选项 -> 工作台** 并取消选择 **自动构建**。这将避免在导入每个项目时多次执行重建。

2. 从“导航器”或“项目资源管理器”视图中，选择**新建 -> EGL 项目**（或者**EGL Web 项目**）。
  3. 对**项目位置**选择“浏览”按钮并指向第一个项目。在**项目名字**段中输入（或者复制并粘贴）项目名。您可能还想选择使用**EGL 首选项**中指定的**构建描述符**。选择**完成**。
  4. 对该迁移集子目录中的每个项目重复步骤 2 和 3。
  5. 选择**项目 -> 全部构建**以使用新项目来重建工作空间。
- 如果在子目录中有许多项目，则您可能会发现对该子目录启动工作空间更为方便，如下所示：
    1. 启动 EGL 开发环境。
    2. 当提示您输入工作空间名时，请指向包含迁移集版本的子目录并选择**确定**。
    3. 修改工作台首选项以执行下列操作：
      - 打开用于 EGL 开发和“VisualAge 生成到 EGL 迁移”的工作台功能。
      - 确保针对 VisualAge Generator 兼容性方式设置了 EGL 首选项。
      - 设置您通常为新工作空间设置的任何其它首选项。
    4. 如果“问题”视图包含指示无法构建项目的错误，请使用“导航器”视图来找到任何已关闭的项目。已关闭的项目的项目名左边没有加号 (+)。通过从上下文菜单中选择**打开项目**来选择已关闭的项目。这时，那些项目应该会显示在“项目导航器”视图中。

---

## 第 8 章 以单文件方式运行迁移

另一种使用第 1 - 3 阶段来运行迁移的方法是以单文件方式运行迁移。此过程允许将一个外部源格式文件直接迁移为 EGL 文件。要以此方式运行迁移，必须先将 VisualAge Generator 部件导出到外部源格式文件中，然后将该外部源格式文件导入到 EGL 中。在导入过程中，根据首选项的不同，将把外部源格式文件迁移到一个或多个 EGL 文件中。

要从 VisualAge Generator 中导出部件，请执行下列操作：

1. 启用 VisualAge Generator on Java (或 VisualAge Generator on Smalltalk) 并打开 VAGen 部件浏览器。
2. 选择要导出的部件，然后右键单击选择。
3. 从上下文菜单中，选择**导入 / 导出** -> **VAGen 导出** (或 **VAGen 带关联项导出**)。
4. 在框中输入外部源格式文件的名称，然后单击**保存**按钮。(如果输入现有文件的名称，则将询问是要将部件添加到该文件中还是要覆盖该文件。请选择适合于您的选项。)

要为单文件方式作准备，执行以下操作：

1. 启动 EGL 开发环境并指向您所在的工作空间。(例如，d:\workspaces\myworkspace)。
2. 设置迁移首选项。请参阅第 126 页的『VAGen 迁移语法首选项』以了解有关如何执行此操作的信息。
3. 从“工作台”窗口中，选择**窗口** -> **首选项** -> **VAGen 迁移**。通常，您始终应该确保将各部件分别保存到不同的 **EGL 文件** 中首选项是选中的。如果您选择了此首选项，则每个程序、映射组和表都将放到它自己的文件中。这符合 EGL 的有关每个文件对应一个可生成部件的要求。如果您未选择**将各部件分别保存到不同的 EGL 文件中**，则所有部件都将放到同一个 EGL 文件中。请参阅第 19 页的『单文件迁移概述』以了解有关单文件方式下的部件放置算法的特定信息。
4. 创建一个新的 EGL 项目 (例如，MyProject)。此外，如果您计划开发新的 EGL pageHandler，则您可能想创建 EGL Web 项目。
5. 在 EGL 项目的 EGLSource 目录下，创建一个新的 EGL 包 (例如，my.pkg)。
6. 请参阅第 141 页的『使用用户界面来运行单文件迁移』一节以了解有关以联机方式运行的详细信息，或者参阅第 142 页的『使用批处理方式来运行单文件迁移』以了解有关创建批处理命令文件以通过一个命令文件处理多个外部源格式文件的详细信息。

---

### 使用用户界面来运行单文件迁移

要将“外部源格式”文件导入到 EGL 中，执行以下操作：

1. 从“导航器”视图中，选择一个 EGL 包，生成的 EGL 文件将放到该包中。
2. 右键单击该包。从上下文菜单中，选择**导入**。
3. 选择 **VAGen 外部源格式文件**，然后选择**下一步**。
4. 在**输入文件名**字段中，输入要导入的外部源格式文件的名称。
5. 在**源文件夹**字段中，输入项目和源文件夹的名称，EGL 文件将放到该文件夹中。(例如，MyProject\EGLSource。)

6. 在**包名字段**中，输入一个包的名称，EGL 文件将放到该包中。迁移工具还使用您对 EGL 文件中的 `package` 语句指定的包名（例如，`my.pkg`）。
7. 在**EGL 文件名字段**中，输入将要根据外部源格式文件创建的 EGL 文件的名称。缺省情况下，EGL 文件将与该外部源格式文件同名，但具有 `.egl` 文件扩展名。请参阅第 19 页的『单文件迁移概述』以了解有关迁移工具如何使用将各部件分别保存到不同的 EGL 文件中首选项以及外部源格式文件中的部件类型来确定单文件方式迁移期间所要创建的文件的信息。
8. 在**日志文件位置**字段中，输入驱动器和目录，迁移日志文件将放到该位置。在**日志文件名字段**中，为迁移日志文件输入名称。缺省情况下，日志文件名与您指定的外部源格式文件名相匹配。迁移日志文件包含迁移期间写出的任何消息。
9. 单击**完成**。如果您在 EGL 文件名字段中指定的文件名已存在于您在“容器”字段中指定的容器中，则将提示您是要对该文件进行追加还是要进行覆盖。
10. 当迁移完成时，您应该会注意到以下情况：
  - 应该会有一个或多个 EGL 文件列示在您指定的项目、源文件夹和包中。请参阅第 19 页的『单文件迁移概述』以了解有关迁移工具如何使用将各部件分别保存到不同的 EGL 文件中首选项以及外部源格式文件中的部件类型来确定单文件方式迁移期间所要创建的文件的信息。
  - 任何错误消息都会显示在弹出窗口中。如果您未指定日志文件位置，则可以使用**保存至文件**按钮来将消息保存在一个文件中。务必关闭该弹出窗口。
11. 选择项目，然后选择项目 **→ 构建项目**。这将导致运行验证以使“问题”视图反映项目中所有文件的最新消息。

---

## 使用批处理方式来运行单文件迁移

用户界面使您能够每次迁移一个外部源格式文件。对于批处理方式来说，可以在一个命令文件中迁移多个外部源格式文件。要使用批处理方式，请执行下列操作：

1. 创建具有 `.bat` 文件扩展名的文件。对于 Windows 环境来说，`.bat` 文件的内容应该是：

```
set path=InstallDirectory\eclipse\jre\bin;%path%
set classpath=InstallDirectory\eclipse\startup.jar;
               InstallDirectory\egl\eclipse\plugins\
               com.ibm.etools.egl.vagenmigration_<i>version</i>\runtime\eglMigration.jar;
cd InstallDirectory
java com.ibm.etools.egl.internal.vagenmigration.batch.VGMIG
   -importFile Path\ExternalSourceFormatFile.esf
   -eglFile Path\EGLFile.egl
   -data Path\workspace
   -package packageName
   -overwrite
   >Path\LogName.log
```

对于 Linux 环境来说，`.bat` 文件的内容应该是：

```
# !/bin/bash
export PATH=InstallDirectory/eclipse/jre/bin/:$PATH
export CLASSPATH=InstallDirectory/eclipse/startup.jar:
                 InstallDirectory/egl/eclipse/plugins/
                 com.ibm.etools.egl.vagenmigration_<i>version</i>/runtime/eglMigration.jar:$CLASSPATH
cd InstallDirectory/
java com.ibm.etools.egl.internal.vagenmigration.batch.VGMIG
   -importFile Path/ExternalSourceFormatFile.esf
   -eglFile Path/EGLFile.egl
```

```
-data /Path/workspace  
-package packageName  
-overwrite  
>/Path/LogName.log
```

此外，对于 Linux 环境来说，必须更改该 .bat 文件的许可权以使其可以执行。要使该 .bat 文件可以执行，请打开“命令终端”并输入以下命令，其中 *mybatchfile.bat* 是您创建的 .bat 文件的名称：

```
chmod u+x mybatchfile.bat
```

#### 注：

- 下列语句必须写在一行上：
  - 对于 Windows 环境来说，是 **set classpath** 语句。
  - 对于 Linux 环境来说，是 **export classpath** 语句。
  - **java** 语句。
- 对于要迁移的每个外部源格式文件，将 **java** 语句重复一次。
- *InstallDirectory* 是安装了 Rational Developer 或 WebSphere Studio 产品的驱动器和目录。必须将 *InstallDirectory* 包括在 path 语句、classpath 语句和 cd（切换目录）语句中。

**注：**如果安装您现在正在使用的产品前您已经安装并保留了先前版本的开发者产品，则您想使用的安装目录可能是先前进行安装时使用的目录。

- *version* 是插件版本号（例如，6.0 或 6.0.0.1）。通常，您应该使用所看到的 *com.ibm.etools.egl.vagenmigration\_version* 插件的最高版本号。
- *Path\ExternalSourceFormatFile.esf* 是指要迁移的外部源格式文件的驱动器、目录和文件名（例如，d:\temp\VAGenFiles\PROG1.esf）。
- *Path\EGFile.egl* 是指要创建的 EGL 文件的驱动器、目录和文件名。目录必须包含用于存放 EGL 源文件的工作空间、EGL 源文件夹和包（例如，d:\myworkspace\MyProject\EGLSource\my\pkg\prog1.egl）。*EGFile.egl* 的使用方式与您使用“导入 VAGen 外部源格式文件”向导时指定的 EGL 文件名字段的使用方式相同。请参阅第 19 页的『单文件迁移概述』以了解有关迁移工具如何使用将各部件分别保存到不同的 EGL 文件中首选项以及外部源格式文件中的部件类型来确定单文件方式迁移期间所要创建的文件的信息。
- *Path\workspace* 是工作空间的驱动器和目录（例如，d:\workspaces\myworkspace）。如果您未指定 *-data* 选项，则您在 VAGen 迁移语法首选项中指定的任何内容都将被忽略，并且迁移工具将使用缺省的 VAGen 迁移语法首选项。如果您想要指定 VAGen 迁移语法首选项，则必须指定 *-data* 选项并指向设置了那些首选项的工作空间。
- *packageName* 是要与 EGL 文件相关联的包的名称（例如，my.pkg）。在迁移工具创建的 .egl 文件的 package 语句中也使用该包名。
- *-overwrite* 参数是可选的。此参数告知迁移工具是否覆盖指定目录中具有指定名称的现有 EGL 文件。
- *Path\LogName* 是指要为迁移相应外部源格式文件创建的日志文件的位置和文件名。虽然将迁移消息发送至日志文件是可选项，但强烈建议您这样做。

对于 Windows 环境来说，java 命令示例可能与下面的内容相似，但是该命令的所有内容应该位于一行上：

```
java com.ibm.etools.egl.internal.vagemigration.batch.VGMIG
  -importFile d:\temp\VAGenFiles\prog1.esf
  -eglFile d:\workspaces\myworkspace\MyEGLProject\EGLSource\my\pkg\prog1.egl
  -data d:\workspaces\myworkspace
  -package my.pkg -overwrite >d:\temp\EGLLogs\prog1.log
```

对于 Linux 环境来说, java 命令示例可能与下面的内容相似, 但是该命令的所有内容应该位于一行上:

```
java com.ibm.etools.egl.internal.vagemigration.batch.VGMIG
  -importFile /d:/temp/VAGenFiles/prog1.esf
  -eglFile /d:/workspaces/myworkspace\MyEGLProject/EGLSource/my/pkg/prog1.egl
  -data /d:/workspaces/myworkspace
  -package my.pkg -overwrite >/d:/temp/EGLLogs/prog1.log
```

2. 关闭 EGL 开发环境。
3. 打开命令提示符窗口, 转至包含该 .bat 文件的目录并运行该 .bat 文件。

**注:** 您可以安全地忽略以下消息: *PolicyClassLoader* 找不到策略 *com.ibm.jxesupport.JxeClassLoaderPolicy*。

4. 此过程完成后, EGL 文件和日志文件将分别存储在您为它们指定的目录中。日志文件包含已迁移的部件列表以及任何错误消息。这些消息就是您以联机方式使用“导入”向导时列示在弹出窗口中的那些消息。
5. 启动 EGL 开发环境。
6. 选择已导入了外部源格式文件的项目, 然后单击鼠标右键并选择**刷新**。这将根据文件系统的内容来刷新项目, 以使 EGL 能够识别在批处理方式迁移期间创建的、追加的或覆盖的 EGL 文件。还会导致运行验证以使“问题”视图反映项目中所有文件的最新消息。然后, 您就可以展开已创建的包以查看 EGL 文件了。

---

## 第 5 部分 完成迁移



---

## 第 9 章 完成迁移

在使用第 1 – 3 阶段迁移或单文件迁移迁移了源代码之后，您还应该完成一些其它任务。这包括：

- 导出首选项。
- 为源代码资源库中的 EGL 项目和包保存一条基线。
- 用于完成单文件迁移的预备步骤。
- 查看 EGL 源代码。
- 查看 EGL 构建描述符部件。
- 查看 EGL 链接选项部件。
- 查看 EGL 资源关联部件。
- 为调试作准备。
- 对 COBOL 生成进行生成和测试。
- 对 Java 生成进行生成和测试。
- 查看标准。

---

### 导出首选项

在试验项目中使用 EGL 之后，您可能已经设置了除第 123 页的『必需的首选项』、第 124 页的『建议的首选项』和第 126 页的『VAGen 迁移语法首选项』描述的必需首选项和建议首选项以外的其它首选项。例如，您可能设置了与所选源代码资源库和库管理过程相关的其它首选项。您可以将首选项导出到一个文件中，以使其它开发者可以导入这些首选项以便以此为基础设置他们自己的首选项。导出技术也是一种方便的将首选项从一工作空间移至另一工作空间的方法。要导出首选项，请执行下列操作：

- 选择窗口 -> 首选项。
- 选择“首选项”窗口左下角的**导出**。
- 指向要用来保存首选项的文件。

其他开发者可以通过执行下列操作来将首选项导入到工作空间中：

- 选择窗口 -> 首选项。
- 选择 EGL 功能，如第 123 页的『必需的首选项』所述。
- 选择窗口 -> 首选项。
- 选择“首选项”窗口左下角的**导入**。
- 指向已用来保存首选项的文件。

**注：** 这种技术对透视图设置及视图设置不起作用。此技术仅更改首选项。

---

## 为 EGL 项目和包保存一条基线

在尝试解决“问题”视图中的任何消息或修改任何已迁移的 EGL 代码之前，您可能想为源代码资源库中的 EGL 项目和包创建一个版本。通过在迁移后立即存储 EGL 项目和报告并设置它们的版本，就有了一条基线，您就可以确切地了解迁移工具生成了哪些源代码。此基线还使您能够对必须手工进行的任何代码更改进行跟踪。这在试验项目期间特别有用，它提供了一种捕获所有更改的方法，从而使您能够记录必须进行的更改的类型。在迁移其它子系统时，此项记录对您会有所帮助。

---

## 用于完成单文件迁移的预备步骤

单文件迁移并不执行第 1 – 3 阶段迁移所执行的所有操作。您必须手工执行下列步骤：

- 将任何表单嵌套在它们的对应 formGroup 中。如果将两个 formGroup 迁移至同一个包并且两个 formGroup 包含相同的表单名（例如 MAP1），则必须完成此步骤。
- 解决同一个 EGL 包中的任何重复部件。如果将两个程序及其关联项迁移至同一个 EGL 包并且这两个程序共享一些公共部件，就会发生存在重复部件的情况。可以将公共部件定义分到中央公共文件中，也可以从其中一个文件中除去重复的部件。如果所有文件位于同一个包中，则不需要修改 EGL 构建路径属性或添加 import 语句。
- 更新当前项目的 EGL 构建路径属性，以使其列示当前项目在解析任何部件名时需要引用的所有项目。要更新 EGL 构建路径，在“项目导航器”视图中，选择当前项目，右键单击选择，然后选择属性。在“EGL 构建路径”页上的“项目”选项卡上，选择当前项目需要引用的其它项目。务必在 EGL 构建路径中包括任何这样的项目：它们包含当前项目中的文件需要导入的包。例如，如果 FileA 位于 ProjectB 中，并且 FileA 需要导入 packageC，则务必将 packageC 所在的项目包括在 ProjectB 的 EGL 构建路径中。
- 在 EGL 文件中添加 import 语句以指向文件需要引用的公共包。对 import 语句指定的包必须存在于对当前 EGL 文件所在的项目的 EGL 构建路径指定的项目中。例如，如果 FileA 位于 ProjectB 中，则 FileA 中的 import 语句只能引用对 ProjectB 的 EGL 构建路径指定的项中的包。

---

## 第 1 — 3 阶段迁移与单文件迁移的公共步骤

### 查看 EGL 源代码

无论您是使用第 1 – 3 阶段迁移还是单文件迁移，都需要执行下列步骤：

- 查看并解决迁移日志或任务列表日志中的错误。这些错误反映了迁移工具无法解决的不确定情况。修改 EGL 源代码以解决这些错误。例如，如果使用了 VAGen RETR 语句并且未显式地指定搜索列，并且该表在迁移期间不可用，则 EGL 语法将包括 EZE\_UNKNOWN\_SEARCH\_COLUMN。您必须根据 dataTable 定义使用正确的搜索列名来更新 EGL 源代码。要获取有关解析当迁移工具故意创建无效 EGL 语法时所使用的特定字符串的帮助，请参阅第 299 页的附录 D，『“问题”视图中的消息』。
- 如果程序名、dataTable 名或 formGroup 名是保留字，则必须更改部件名。如果生成 COBOL，则您可能想将部件的 alias 属性设置为指定原始部件名。这样，您就不必更改对程序、dataTable 或 formGroup 的任何外部引用。
- 查看并解决“问题”视图中的任何其它错误。有关解决“问题”视图中作为迁移过程结果而产生的公共消息的帮助，请参阅第 303 页的附录 E，『“问题”视图中的 IWN.xxx 消息』。

- 确定是否需要为任何记录或函数设置 *containerContextDependent* 属性。要了解更多详细信息，请参阅第 29 页的『*containerContextDependent* 属性』。

## 查看 EGL 构建描述符部件

迁移工具将 VAGen 生成选项部件转换为 EGL 构建描述符部件。但是，某些 VAGen 选项没有 EGL 等效项。另外，EGL 具有几个可能需要您设置的新构建描述符选项。在“问题”视图中，您可能会看到由这些更改中的任何一项更改导致的错误。有关解决“问题”视图中作为迁移过程结果而产生的公共消息的帮助，请参阅第 303 页的附录 E，『“问题”视图中的 IWN.xxx 消息』。您可能需要使用文本编辑器来解决一些问题。无论您是使用第 1 – 3 阶段迁移还是单文件迁移，都需要执行下列步骤：

- 查看一般构建描述符选项。
- 查看 COBOL 生成构建描述符选项。
- 查看 Java 生成构建描述符选项。
- 建立调试构建描述符部件。

## 查看一般构建描述符选项

无论您是计划生成 COBOL 还是 Java，都需要查看下列构建描述符选项：

- 如果为一种本地语言生成，并且该本地语言使用的小数点是逗号，则必须包括 *decimalSymbol* 构建描述符选项。
- 如果使用 VAGen EZESYS 特殊函数字来确定运行时环境，则您可能想设置 EGL 构建描述符选项 *eliminateSystemDependentCode*。有关此选项的更多信息，请参阅联机帮助。
- 要了解有关主构建描述符的信息，请参阅联机帮助。这种技术替换了 VAGen 的“缺省生成选项部件”首选项。迁移工具自动地将任何包含 *NOOVERRIDE* 属性的生成选项部件分割为两个构建描述符部件。其中一个构建描述符部件名为 *xxxxx*，另一个名为 *xxxxx\_NOOVERRIDE*，其中，*xxxxx* 是原始 VAGen 生成选项部件名。名为 *xxxxx* 的部件包含所有未指定 *NOOVERRIDE* 属性的 VAGen 生成选项的 EGL 替换项。名为 *xxxxx\_NOOVERRIDE* 的部件包含所有指定了 *NOOVERRIDE* 属性的 VAGen 生成选项的 EGL 替换项。如果您决定使用主构建描述符，则必须执行这里描述的分割为两个部件的操作。
- 如果您已使用 VAGen /OPTIONS 生成选项来对生成选项部件进行链接，请使用 *nextBuildDescriptor* 选项来查看 EGL 构建描述符部件的链接方式。您可能需要对这种链接方式进行修改以获取与 VisualAge Generator 中的构建描述符选项集相同的构建描述符选项集。
- 迁移工具在创建 EGL 项目时并不创建缺省构建描述符。您可以为文件、包、EGL 源文件夹、项目或工作台级别建立缺省构建描述符。与可生成部件最接近的缺省构建描述符就是所使用的构建描述符。例如，可以只为一个文件指定缺省构建描述符并为工作台指定另一个缺省构建描述符。在这种情况下，如果生成该文件包含的程序，则将使用该文件的缺省构建描述符。当生成任何其它程序时，将使用工作台缺省构建描述符。
  - 要为特定的文件、包、EGL 源文件夹或项目设置首选项，请选择资源（文件、包、文件夹或项目），然后单击鼠标右键并从上下文菜单中选择属性。在左窗格中选择 *EGL 缺省构建描述符*。选择要在此资源中的所有可生成部件中用作缺省构建描述符的构建描述符。假定没有更接近的 EGL 缺省构建描述符，目标系统构建描述符就是每当生成此资源中的任何内容时将要使用的缺省构建描述符。当使用调试工具时，使用的缺省构建描述符是 *调试构建描述符*。

- 要为构建描述符部件设置工作台首选项，请选择窗口 -> 首选项 -> **EGL -> 缺省构建描述符**。如果未显式地覆盖此首选项，则它将应用于所有项目、包、源文件夹和文件。您可以既设置用于生成的目标系统构建描述符又设置用来与调试工具配合使用的调试构建描述符。
- 如果控制部件（构建描述符部件、链接选项部件、资源关联部件、绑定控制部件和链接编辑部件）并不是都在同一个文件中，则对于包含要从当前文件中引用的其它构建部件的文件，必须将当前文件修改为包含用于那些文件的 `import` 语句。例如，如果 `buildDescriptorPartA` 引用另一个文件中的 `linkageTableB`，则包含 `buildDescriptorPartA` 的文件必须包含一个 `import` 语句以导入包含 `linkageTableB` 的文件。使用 EGL 构建部件编辑器来添加 `import` 语句。

## 查看 COBOL 生成构建描述符选项

如果您计划生成 COBOL，则需要查看或设置下列构建描述符选项：

- 对于 VisualAge Generator 来说，COBOL 生成的输出将被传送到主机以运行准备步骤。EGL 使用构建服务器来处理准备步骤。对于 EGL z/OS 和 iSeries 构建服务器，必须指定一个端口号来传送生成输出。请与 Enterprise Developer 构建服务器的安装和配置人员联系以确定远程构建服务器侦听构建请求时使用的端口号。
- 如果为 zOS 环境生成 COBOL，则还必须执行下列操作：
  - 建立一个绑定控制部件以用作模板。（请参阅第 151 页的『建立一个绑定控制部件以用作模板』。）
  - 建立一个特定于程序的绑定控制部件。（请参阅第 152 页的『建立特定于程序的绑定控制部件』。）
  - 查看链接编辑命令。（请参阅第 152 页的『查看链接编辑命令』。）

## 查看 Java 生成构建描述符选项

如果您计划生成 Java，则需要查看或设置下列构建描述符选项：

- 务必添加 `genProject` 构建描述符选项以指定 Java 生成的输出的存放位置。没有任何 VAGen 生成选项会迁移为 EGL `genProject` 构建描述符选项。
- 某些 EGL 构建描述符选项的行为与它们的对应 VAGen 生成选项的行为略有不同。请参阅联机帮助以了解有关下列构建描述符选项的信息，以确定是否需要为您的环境设置或修改它们：
  - `genProperties`，迁移工具根据 VAGen `/genproperties` 选项设置此选项。
  - `enableJavaWrapperGen`，迁移工具根据 VAGen `/system=JAVAWRAPPER` 选项设置此选项。
- 有些新的 EGL 构建描述符选项没有对应的 VAGen 生成选项。请参阅联机帮助以了解有关下列构建描述符选项的信息，以确定是否需要为您的环境设置它们：
  - `dateMask`
  - `cicsj2cTimeout`
  - `sessionBeanID`
  - `sqlJDBCdriverClass`
  - `sqlJNDIName`
  - `sqlValidationConnectionURL`

## 建立调试构建描述符部件

请创建一个构建描述符部件并使其包含您在调试期间要使用的构建描述符选项。请参阅联机帮助以获取有关创建调试构建描述符部件的指南。

## 建立一个绑定控制部件以用作模板

VisualAge Generator 使用绑定控制模板来创建缺省绑定控制命令。缺省 VAGen 模板绑定一个 DB2 规划，但您可能已对该模板作了修改以使其绑定一个包，或者已进行了其它更改以便与您所在组织的标准相符。VAGen 模板存储在工作空间外部名为 EFK2MBDA.tpl 和 EGK2MBDD.tpl 的文件中。仅当需要为特定程序执行特殊的绑定时，绑定控制部件才是必需的。

EGL 不使用绑定控制模板。而是，EGL 需要绑定控制部件。如果绑定包，则通过创建一个包含要用于所有绑定的模板的 EGL 绑定控制部件并将此部件存储在工作空间中，可以获得类似于 VisualAge Generator 模板的效果。注意：如果绑定规划，则本节描述的技术不起作用。如果绑定规划，则请参阅第 152 页的『建立特定于程序的绑定控制部件』。

如果您已修改 VAGen 绑定控制模板以便为每个程序绑定一个包，则可以采用该模板以将其用作 EGL 绑定控制部件。建议您将此绑定控制部件与其它控制部件放在同一个文件中。例如，您可能有一个与以下内容相似的 VAGen 绑定控制模板：

```
DSN SYSTEM(%MYDB2SUBSYSTEM%)
BIND PACKAGE(%MYCOLLECTIONNAME%) -
MEMBER(%EZEMBR%) -
.
.
```

在上一个示例中，MYDB2SUBSYSTEM 和 MYCOLLECTIONNAME 是 VAGen 生成选项中设置的符号参数，EZEMBR 是使用当前正在生成的程序的名称自动设置的。

对于绑定包，需要创建的 EGL 绑定控制部件与 VAGen 模板非常相似，但需要另外 3 行以及对 EZEMBR 符号参数进行更改。相应的 EGL 绑定控制部件与以下内容相似：

```
TSOLIB ACTIVATE DA('%DSNLOAD%')
ALLOC FI(DBRMLIB) SHR DA('%EZEPID%.%SYSTEM%.DBRMLIB' +
'%ELA%.SELADBRM')
DSN SYSTEM(%MYDB2SUBSYSTEM%)
BIND PACKAGE(%MYCOLLECTIONNAME%) -
    MEMBER(%EZEALIAS%) -
.
.
```

DSNLOAD、EZEPID 和 ELA 具有的含义与它们在 VisualAge Generator 中具有的含义相同。当在绑定控制部件中需要所生成的程序的名称时，EZEALIAS 是 EZEMBR 的 EGL 替换项。SYSTEM 是 EZEENV 的 EGL 替换项。如果在 EGL 构建服务器上使用不同的数据集命名约定，则可能需要修改绑定控制部件的前 3 行。请与 Enterprise Developer 构建服务器的安装和配置人员联系以根据您所在组织的命名约定确定前 3 行所需的内容。如果在 VisualAge Generator 中未设置 projectID 构建描述符选项以及 DSNLOAD 和 ELA 符号参数，则您可能还需要修改 EGL 构建描述符选项以设置这些值。请参阅第 273 页的『符号参数』以了解对符号参数名称所作的更改。并且，请参阅联机帮助以获取更多关于对 EGL 绑定控制部件使用模板以及设置 EGL 符号参数值的信息。

除了创建 EGL 绑定控制部件以用作模板之外，还必须将构建描述符部件修改为包括 *bind* 构建描述符选项以指向绑定控制部件。建议您将 *bind* 构建描述符选项添加到其中一个现有的公共构建描述符部件中，以便最大程度地减少所需修改的构建描述符部件的数目。

## 建立特定于程序的绑定控制部件

如果在 VisualAge Generator 中绑定规划，则通常每个程序都需要不同的绑定命令。在这种情况下，需要特定于程序的绑定命令来为该程序以及位于同一运行单元中的所有其它程序绑定规划。执行此操作的典型方法是创建名为 *xxxx.BND* 的绑定控制部件，其中 *xxxx* 是程序的名称。然后，设置 VAGen 生成选项 */BIND=BND* 以指定在搜索特定于程序的绑定命令时要用 VisualAge Generator 使用的后缀。在一些罕见的情况下，一个程序所需的内容与模板提供的内容不同，如果在这种情况下绑定包，则也可以使用 *.BND* 后缀。

EGL *bind* 构建描述符选项不允许指定后缀。而是，*bind* 构建描述符选项必须指定特定绑定控制部件的名称。在 EGL 中，如果未指定 *bind* 构建描述符选项，则 EGL 将查找与程序同名的绑定控制部件。通常，最简单的技术是绑定包并遵循第 151 页的『建立一个绑定控制部件以用作模板』描述的过程。但是，如果要绑定规划，或者在绑定控制部件模板未提供程序所需内容的情况下，可以创建特定于程序的绑定控制部件。

如果 VAGen 特定于程序的绑定控制部件使用缺省的 *.BND* 后缀，则迁移工具将自动除去 *.BND* 后缀并添加 EGL 绑定控制部件所需的其它语句。假定命名约定是 *programName.BND*，并且始终有特定于程序的绑定命令部件，则不需要为此程序指定 EGL *bind* 构建描述符选项。但是，如果正在使用 EGL *bind* 构建描述符来为大部分程序指定用作模板的绑定控制部件，并且需要为某个程序提供特定于程序的绑定控制部件，则必须为这个特定的程序创建一个构建描述符部件并将 *bind* 构建描述符选项设置为指向该特定于程序的绑定控制部件。否则，EGL 将使用用作模板的绑定控制部件，这是因为，它就是正常的 *bind* 构建描述符选项所指定的绑定控制部件。

## 查看链接编辑命令

VisualAge Generator 提供了缺省的链接编辑命令，那些命令基于目标环境和数据库访问。但是，在某些情况下，程序可能需要特定的链接编辑命令（例如，以便为 MVS Batch 环境链接 *PLI* 程序）。执行此操作的典型方法是创建名为 *xxxx.LKG* 的链接编辑部件，其中 *xxxx* 是程序的名称。然后，设置 VAGen 生成选项 */LINKEDIT=LKG* 以指定在搜索特定于程序的链接编辑命令时要用 VisualAge Generator 使用的后缀。

EGL *linkEdit* 构建描述符选项不允许指定后缀。而是，*linkEdit* 构建描述符选项必须指定特定链接编辑部件的名称。在 EGL 中，如果您未指定 *linkEdit* 构建描述符选项，则 EGL 将查找与程序同名的链接编辑部件。如果 EGL 找不到与程序同名的链接编辑部件，则 EGL 根据目标环境和数据库访问来创建缺省链接编辑命令，这与 VisualAge Generator 执行的操作类似。因此，仅当创建了与程序不同名的链接编辑部件时，才需要指定 *linkEdit* 构建描述符选项。如果为若干 COBOL 环境生成同一程序，可能需要这样做。

如果 VAGen 特定于程序的链接编辑部件使用缺省的 *.LKG* 后缀，则迁移工具将自动除去 *.LKG* 后缀。假定命名约定是 *programName.LKG*，则不需要为此程序指定 EGL *linkedit* 构建描述符选项。EGL 在尝试创建缺省链接编辑命令之前将能够首先找到特定于程序的部件。

## 查看 EGL 链接选项部件

迁移工具将 VAGen 链接表部件转换为 EGL 链接选项部件。但是，某些 VAGen 选项没有 EGL 等效项。另外，EGL 具有几个可能需要您设置的新链接选项。在“问题”视图中，您可能会看到由这些更改中的任何一项更改导致的错误。有关解决“问题”视图中作为迁移过程结果而产生的公共消息的帮助，请参阅第 303 页的附录 E，『“问题”视图中的 IWN.xxx 消息』。并且，请参阅联机帮助以了解有关环境所支持的链接选项的详细信息。无论您是使用第 1 – 3 阶段迁移还是单文件迁移，都需要执行下列步骤：

- 查看并解决迁移日志中的以及“问题”视图中的消息。您可能需要使用文本编辑器来解决一些问题。
- 对于 callLink 来说，请考虑下列各项：
  - 并非 VisualAge Generator 中的所有链接类型在 EGL 中都受支持。例如，CSOCALL 不再受支持。迁移工具将把 CSOCALL 转换为 remoteCall。但是，必须对 EGL remoteCall 指定的属性与必须对 CSOCALL 指定的属性不同。
  - 并非 VisualAge Generator 中的所有 remoteComType 值在 EGL 中都受支持。例如，DCE 和 DCESECURE 不再受支持。迁移工具将按原样转换这些不受支持的值，这将产生无效的 EGL 链接选项部件。这将确保在“问题”视图中有一个错误，从而提示您必须修改链接选项部件以指定要与 EGL 配合使用的选项。
  - 如果更改为使用 remoteComType=CICSECI，则必须添加 *ctgPort* 和 *ctgLocation* 属性。这要求为了进行远程 CICS 事务调用而配置和设置 CICS 事务网关服务器。
  - 如果更改为使用 remoteComType=CICSSSL，则必须添加 *ctgKeyStore* 和 *ctgKeyStorePassword* 属性。另外，如果尚未将 *ctgPort* 和 *ctgLocation* 属性包括在 VAGen 链接表中，则必须为 EGL remoteComType=CICSSSL 包括它们。
  - 在 EGL 中不支持 conversionTable=BINARY。迁移工具将按原样转换此值，以使 EGL 链接部件包含一个位置标志符。但是，您必须修改该值。
- 对于 fileLink 来说，在 EGL 中不支持 conversionTable=BINARY。迁移工具将按原样转换此值，以使 EGL 链接部件包含一个位置标志符。但是，您必须修改该值。
- 对于 asynchLink (VAGen crtlink) 来说，在 EGL 中不支持 conversionTable=BINARY。迁移工具将按原样转换此值，以使 EGL 链接部件包含一个位置标志符。但是，您必须修改该值。
- EGL 的“转移目标程序”链接信息等同于 VAGen dxfrlink 条目。如果为 Java 生成并使用 VAGen XFER 语句，则可能需要添加 EGL 的“转移至事务”条目。要了解有关这个新链接条目的信息，请参阅联机帮助。

## 查看 EGL 资源关联部件

迁移工具将 VAGen 资源关联部件转换为 EGL 资源关联部件。但是，某些 VAGen 选项没有 EGL 等效项。另外，EGL 具有几个可能需要您设置的新资源关联选项。在“问题”视图中，您可能会看到由这些更改中的任何一项更改导致的错误。有关解决“问题”视图中作为迁移过程结果而产生的公共消息的帮助，请参阅第 303 页的附录 E，『“问题”视图中的 IWN.xxx 消息』。并且，请参阅联机帮助以了解有关环境所支持的资源关联选项的详细信息。无论您是使用第 1 – 3 阶段迁移还是单文件迁移，都需要执行下列步骤：

- 查看并解决迁移日志中的以及“问题”视图中的消息。您可能需要使用文本编辑器来解决一些问题。

- 并非 VisualAge Generator 中的所有文件类型在 EGL 中都受支持。例如，BTRIEVE 和 MFCOBOL 不再受支持。迁移工具将按原样转换这些不受支持的选项，以使资源关联部件包含一个位置标志符。这将确保在“问题”视图中有一个错误，从而提示您必须修改资源关联部件以指定要与 EGL 配合使用的选项。根据所选 EGL 文件类型选项的不同，可能还必须为资源关联条目设置其它属性。
- 查看 *FormFeedOnClose* 和 *text* 属性的联机帮助以确定是否需要为您的环境设置这些值。在 VisualAge Generator 中，只能在工作站环境的资源关联文件中指定等效的选项（分别为 */noff* 和 */text*）。因此，由于迁移工具只处理资源关联部件，所以此工具不设置这些选项。

## 为调试作准备

您应该执行下列操作来为调试作准备:

- 从“工作台”窗口中，选择窗口 -> 首选项 -> EGL -> 调试。请参阅联机帮助以确定需要为环境设置这些首选项中的哪些首选项（如果有的话）。
- 并且，查看 **EGL-> 缺省构建描述符** 首选项。您可能想为整个工作空间设置缺省调试构建描述符。另外，可以为项目、EGL 源文件夹、包或文件设置缺省调试构建描述符。
- 如果正在从调试器中调用远程 CICS 系统上生成的 EGL 程序或非 EGL 程序，则需要配置并使用 CICS 事务网关服务器。不再支持使用 CICS 客户机或 CICS 事务网关来直接调用 CICS。

## 对 COBOL 生成进行生成和测试

您应该执行下列操作来为 COBOL 生成作准备:

- 现在，您可能想为源代码资源库中的 EGL 项目和包创建另一个版本。此操作可以提供另一条代码基线，该代码基线反映了您在生成之前手工进行的更改。
- 如果正在为 zOS 生成，则确保已安装 Enterprise Developer Server for zOS V5.0 并应用所有最新的 PTF。
- 如果正在为 iSeries 生成，则确保已在主机环境中安装了用于 iSeries 的 EGL 运行时并应用了所有最新的 PTF。
- 确保已在主机环境中安装并配置了 EGL 构建服务器。在 VisualAge Generator 中，对 zOS 和 iSeries 的准备过程进行的定制是通过在工作站上更改准备模板完成的。在 EGL 中，此定制是在主机上完成的。有关是否仍需要进行定制以及如何为每种目标主机环境进行定制的更多信息，请参阅联机帮助。
- 请与 Enterprise Developer 构建服务器的安装和配置人员联系。您务必了解包含生成输出和准备输出的主机数据集的任何命名约定更改。例如，在 VisualAge Generator 中，当为 MVS Batch 进行生成时，数据集的缺省名称是 *xxxx.MVSBATCH.yyyy*，其中 *xxxx* 是您在 */projectid* 生成选项中指定的高级限定符，*yyyy* 是代码类型（例如，EZESRC 表示 COBOL 源代码）。对于 EGL 来说，由于目标环境名已更改，所以相应的数据集名称是 *xxxx.ZOSBATCH.yyyy*。这表示可能需要在主机上定义新的一组数据集。
- 生成程序和 dataTable。当生成程序时，使用下列构建描述符选项:
  - `genFormGroup="YES"`
  - `genHelpFormGroup="YES"`
  - `genDataTables="NO"`

这使您能够与使用 formGroup 的程序一起生成那些 formGroup，但只将 formGroup 生成一次，而不必考虑使用该 formGroup 的程序的数目。请解决生成期间捕获到的任何验证错误。

- 对生成的代码进行测试。
- 现在，您可能想为源代码资源库中的 EGL 项目和包创建另一个版本。此操作可以提供另一条代码基线，该代码基线反映了由生成和测试期间找到的问题导致的更改。

## 对 Java 生成进行生成和测试

您应该执行下列操作来为 Java 生成作准备：

- 现在，您可能想为源代码资源库中的 EGL 项目和包创建另一个版本。此操作可以提供另一条代码基线，该代码基线反映了您在生成之前手工进行的更改。
- 生成程序和 dataTable。当生成程序时，使用下列构建描述符选项：
  - genFormGroup="YES"
  - genHelpFormGroup="YES"
  - genDataTables="NO"

这使您能够与使用 formGroup 的程序一起生成那些 formGroup，但只将 formGroup 生成一次，而不必考虑使用该 formGroup 的程序的数目。请解决生成期间捕获到的任何验证错误。

- 如果您修改了 VAGen 产品消息文本，则可以对 EGL 消息文本作类似的修改。
- 对生成的代码进行测试。
- 现在，您可能想为源代码资源库中的 EGL 项目和包创建另一个版本。此操作可以提供另一条代码基线，该代码基线反映了由生成和测试期间找到的问题导致的更改。

## 查看标准

您可能想查看当前编码标准和设置一些要对所编写的任何新代码使用的新标准。例如，如果生成 COBOL，则您可能要考虑的一些标准包括：

- 在部件名中使用下划线而不是连字符。COBOL 程序不允许在名称中使用下划线。但是，COBOL 生成将自动地把下划线更改为连字符，以使生成的 COBOL 仍然可读。仅当在将下划线更改为连字符后出现重复的部件名时，才会指定别名。
- 为了提高所生成的 COBOL 代码的可读性，应该避免使用别名。可以通过使用下列命名约定来做到这一点：
  - 记录名和函数名的长度应该不超过 18 个字符。此项限制与 VisualAge Generator 中的限制相同。
  - 数据项名的长度应该不超过 27 个字符。此长度比 VisualAgeGenerator 的建议限制（30 个字符）和最大长度（32 个字符）略短。
- 现在，程序名和 dataTable 名的长度可以是 8 个字符。

## 除去对 VisualAge Generator 兼容性方式的使用

VAGen 兼容性方式支持许多行为，这些行为使您能够方便地保留 VAGen 程序的行为。没有必要关闭 VAGen 兼容性方式。但是，当您仅仅使用了这些兼容性行为中的少数几个行为时，您可能想除去对 VAGen 兼容性方式的使用。以下列表描述了当 VAGen 兼容性方式处于打开状态时的 EGL 行为，此列表还提供了当您不想使用该行为时对您可能有帮助的信息。

- EGL 允许在部件名中使用连字符 (-) 以及本地语言字符 @ 和 #。如果您打算除去对 VAGen 兼容性方式的使用，则应该创建一个“重命名用户出口”以便在第 2 阶段迁移期间使用。创建“重命名用户出口”以便能够除去对连字符、@ 和 # 的使用。例如，如果您在 VAGen 部件名中从不使用下划线，则可以创建一个“重命名用户出口”以通过将连字符更改为下划线来将部件重命名。请参阅第 124 页的『VAGen 迁移首选项』以了解有关如何为第 2 阶段迁移指定“重命名用户出口”的详细信息。如果您关闭了 VAGen 兼容性方式，则对于任何包含连字符、@ 或 # 的部件名或变量名，在“问题”列表中都将有一个错误。您可以通过更改名称来更正问题。
- EGL 允许使用基本数据类型 *numc* 和 *pacf*。*numc* 与 *num* 类似，但是，*numc* 使用 C 作为正号指示符。*pacf* 与十进制 (VAGen PACK) 类似，但是，*pacf* 使用 F 作为正号指示符。如果您关闭了 VAGen 兼容性方式，则对于任何指定了基本类型 *numc* 或 *pacf* 的 *dataItem* 定义或变量声明，在“问题”列表中都将有一个错误。通过将基本类型分别更改为 *num* 或十进制，并进行新的正号指示符所要求进行的任何相关程序更改，或许可以更正问题。
- 对于一维的结构字段数组，EGL 将下标缺省为 1。一维结构字段数组是 VAGen 数组的 EGL 等效项，或者是记录、映射或表中多次出现的项的 EGL 等效项。如果您关闭了 VAGen 兼容性方式，则对于指定了需要下标的结构字段数组的任何语句，在“问题”列表中都将有一个错误。为了更正问题，必须将该语句修改为显式地指定下标 1。
- EGL 允许在 *dataTable* 的使用声明中使用 *deleteAfterUse* 属性。*deleteAfterUse* 属性替换了 VAGen 的 *Keep After Use* 属性。如果您关闭了 VAGen 兼容性方式，则对于每个包含指定了 *deleteAfterUse* 的使用声明的程序，在“问题”列表中都将有一个错误。您可以通过除去 *deleteAfterUse* 属性来更正问题，在这种情况下，EGL 将把表看作与指定了 *Keep After Use = yes* 的 VAGen 表类似。

如果已经使用 VisualAge Generator 版本 4.5 修订包 4 生成了生产程序，则除去 EGL *deleteAfterUse* 属性不会有任何影响。如果从 CSP (跨系统产品) 或较早版本的 VisualAge Generator 迁移，则应全面彻底地测试更改为除去 *deleteAfterUse* 属性的任何程序。如果指定 VAGen 迁移首选项不包括表的 *deleteAfterUse*，则迁移工具将自动省略 *deleteAfterUse* 属性并对受影响的程序和表发出警告消息。

- EGL 允许在 SQL 记录中使用 *sqlDataCode* 属性。迁移工具将保留“VAGen SQL 数据代码”属性以便为十六进制项指定 SQL 类型 (例如 SQL 时间戳记)，或者，如果基本类型由于共享数据项未包括在迁移集中而未知时，也保留该属性。通过始终将数据项部件包括在迁移集中 (这样可以避免迁移工具由于基本类型未知而包括 *sqlDataCode*)，可以最大程度地降低迁移到 EGL *sqlDataCode* 属性的可能性。但是，对于十六进制项来说，迁移工具仍必须包括 *sqlDataCode*。如果您关闭了 VAGen 兼容性方式，则对于任何指定了 *sqlDataCode* 属性的 SQL 字段，在“问题”列表中都将有一个错误。您可以通过将该字段更改为使用其中一种新 EGL 数据类型并接着进行任何相关程序更改以使用这种新数据类型来更正问题。对于变长字段来说，包括 *sqlVariableLen = yes* 属性。
- EGL 支持 call 语句选项 *externallyDefined* 和 *noRefresh*。这些选项替换了 VAGen call 语句中的 *NONCSP* 和 *NOMAPS* 选项。如果您关闭了 VAGen 兼容性方式，则对于任何指定了 *externallyDefined* 或 *noRefresh* 的 call 语句，在“问题”列表中都将有一个错误。可以通过从 call 语句中除去这些选项并在被调用程序的链接选项条目中指定对应的 EGL 替换选项来更正问题。在构建描述符选项中，务必指向该链接选项部件。

- EGL 支持 `transfer` 和 `show` 语句的 `externallyDefined` 选项。这个选项替换了 VAGen `DXFR` 和 `XFER` 语句中的 `NONCSP` 选项。如果您关闭了 VAGen 兼容性方式，则对于任何指定了 `externallyDefined` 的 `transfer` 或 `show` 语句，在“问题”列表中都将有一个错误。您可以通过从 `transfer` 或 `show` 语句中除去此选项并在转移源程序和转移目标程序的链接选项条目中指定对应的 EGL 替换选项来更正此问题。在构建描述符选项中，务必指向该链接选项部件。在链接选项部件中，根据 `transfer` 语句类型的不同，使用具有转移目标程序形式或转移目标事务条目形式的转移链接条目。对于 `show` 语句来说，转移链接条目必须具有转移目标事务条目的格式。
- EGL 按照 `print printForm` 语句的实现方式实现 `display printForm` 语句。您可以通过将映射包括在迁移集中来最大程度地减少对 `display printForm` 的使用。这使迁移工具能够正确地迁移为用于文本映射的 `display` 语句以及用于打印机映射的 `print` 语句。如果您关闭了 VAGen 兼容性方式，则对于每个 `display printForm` 语句，在“问题”列表中都将有一个错误。您可以通过将该语句更改为 `print` 语句来更正问题。
- 仅当在屏幕上显示未被赋予值的字段时，EGL 才使用表单字段的 `value` 属性。`value` 属性并不会为存储器中的表单字段设置初始值。迁移工具在迁移映射时将包括 `value` 属性。如果您关闭了 VAGen 兼容性方式，则 EGL 将使用 `value` 属性来为存储器中的表单字段设置初始值。在“问题”列表中可能有错误也可能没有错误。例如，在 VisualAge Generator 中，数字映射变量字段可以具有初始值“MM/DD/YYYY”以便为“映射编辑器预览”方式提供一个值，并且，如果程序在执行 `DISPLAY` 或 `CONVERSE I/O` 选项前未赋予该字段任何数据，则该数字映射变量字段还可以向最终用户提供输出。在本示例中，如果您关闭了 VAGen 兼容性方式，则由于值与 `num` 基本类型不兼容，所以在“问题”列表中将有消息。但是，如果初始值是数字（例如 5），在“问题”列表就不会有任何消息，但是程序的行为可能与它在 VisualAge Generator 中的行为不同。如果您关闭了 VAGen 兼容性方式，则除了更正“问题”列表中的消息以外，您还应该在 EGL 表单中搜索 `value` 属性，然后确定为了防止行为发生任何变化而需要进行的程序更改。这可能包括除去或更改表单字段的 `value` 属性。
- EGL 允许使用 `VGLib.handleSysLibraryErrors` 系统变量。这个系统变量替换了 VAGen `EZEREPLY`，后者控制调用或 `EZE` 函数调用语句的 `EZERT8`（EGL `sysVar.errorCode`）的设置。如果您关闭了 VAGen 兼容性方式，则对于每个指定了 `VGLib.handleSysLibraryErrors` 的语句，在“问题”列表中都将有一个错误。您可以通过除去对 `VGLib.handleSysLibraryErrors` 的使用来更正问题。如果 `VGLib.handleSysLibraryErrors` 设置为 1，则应该将调用或函数调用嵌套在 `try ... onError ... end` 块中。如果 `VGLib.handleSysLibraryErrors` 设置为 0，则不要添加 `try ... onError ... end` 块。
- 如果您使用的 EGL 发行版支持 DLI，则 EGL 允许使用 `dliVar.handleHardDliErrors` 系统变量。这个系统变量替换了 VAGen `EZEDLERR`。在 VAGen 中，`EZEDLERR` 和 `EZEFEC`（EGL `sysVar.handleHardIOErrors`）共同用来控制错误处理。如果 `EZEDLERR` 和 `EZEFEC` 都为 0，或者未指定错误例程，则当发生硬 I/O 错误时，程序将结束。如果 `EZEDLERR` 或 `EZEFEC` 设置为 1，并且指定了错误例程，则当发生硬 I/O 错误时，该错误例程将获取控制权。通常，`EZEDLERR` 和 `EZEFEC` 是在程序开头设置的，并且从不更改。如果您关闭了 VAGen 兼容性方式，则对于每个指定了 `dliVar.handleHardDliErrors` 的语句，在“问题”列表中都将有一个错误。您需要查看程序逻辑以确定是否可以安全地将 `dliVar.handleHardDliErrors` 与 `sysVar.handleHardIOErrors` 合并。如果在程序中未使用 `sysVar.handleHardIOErrors`，或者只是在程序的开头设置了它，则将 `dliVar.handleHardDliErrors` 与 `sysVar.handleHardIOErrors` 合并的操作就会相对比较简单。

- EGL 允许使用 *VGLib.getVAGSysType* 系统函数。迁移工具将声明一个名为 *customerPrefixEZESYS* 的变量并在每个程序中将其初始化为 *VGLib.getVAGSysType* 的结果。*customerPrefix* 是您在第 2 阶段迁移期间指定的重命名前缀。*VGLib.getVAGSysType* 提供了原始的 VAGen EZESYS 系统值（例如 MVSCICS 或 OS400），当迁移除 IF、WHILE 或 TEST 以外的语句时，可以使用这些系统值。如果关闭 VAGen 兼容性，对于每个程序，“问题”列表上会出现针对初始化 *customerPrefixEZESYS* 的语句的错误。可以通过将程序更改为除去 *customerPrefixEZESYS* 声明和初始化语句来更正此问题。当您保存程序时，对于每个使用 *customerPrefixEZESYS* 的语句，“问题”列表中还会有一个错误。为了更正此错误，可以将该语句更改为使用 EGL *sysVar.systemType* 系统变量，该系统变量提供了新的 EGL 系统值。根据 *customerPrefixEZESYS* 使用方式的不同，可能还需要将数据库、文件或 *dataTable* 中的值由旧的 VAGen 系统值更改为新的 EGL 系统值。请参阅第 80 页的『EZESYS』和第 236 页的 EZESYS state conditions。

如果指定 VAGen 迁移首选项不初始化以前的 *EZESYS* 值，则迁移工具将自动省略每个程序中的变量声明和初始化语句。

- EGL 允许使用 *VGLib.connectionService* 系统函数。这个系统函数替换了 VAGen EZECONCT 系统函数，后者根据您指定的自变量以及运行时环境来提供各种 SQL 连接服务。如果您关闭了 VAGen 兼容性方式，则对于每个使用 *VGLib.connectionService* 的语句，在“问题”列表中都将有一个错误。可以通过更改为使用其中一个新的 EGL 专用系统函数（例如 *sysLib.connect*、*sysLib.disconnect*、*sysLib.disconnectAll* 或 *sysLib.queryDatabase*）来更正错误。您所使用的 EGL 系统函数取决于您对 VAGen EZECONCT 系统函数指定的自变量以及运行时环境。您还应该检查对 *VGVar.sqlIsolationLevel* 的任何使用，这是因为，它可能会影响到您对 EGL 系统函数的选择或者您需要对该系统函数指定的自变量。
- EGL 允许使用 *VGVar.segmentedMode* 系统变量。这个系统变量替换了 VAGen EZESEGM，后者使您在运行时能够在 CICS 主事务程序中在分段方式与非分段方式之间进行切换。如果您关闭了 VAGen 兼容性方式，则对于每个使用 *VGVar.segmentedMode* 的语句，在“问题”列表中都将有一个错误。在许多情况下，由于很少使用 EZESEGM，所以在“问题”列表中不会有任何错误。如果存在错误，则除去对 *VGVar.segmentedMode* 的使用可能要求重构程序以便不需要在分段方式与非分段方式之间进行切换。
- EGL 允许使用 *VGVar.sqlIsolationLevel* 系统变量。这个系统变量替换了 VAGen EZESQISL，后者用来为 VSE 运行时环境控制旧发行版的跨系统产品和 VisualAge Generator 中的 SQL 隔离级别以及 VisualAge Generator 4.5 中的 SQL 隔离级别，以便访问 ODBC 数据库。如果您关闭了 VAGen 兼容性方式，则对于每个使用 *VGVar.sqlIsolationLevel* 的语句，在“问题”列表中都将有一个错误。在许多情况下，由于很少使用 EZESQISL，所以在“问题”列表中不会有任何错误。如果有错误，并且未使用 *VGVar.sqlIsolationLevel* 来控制程序逻辑，则您可能能够彻底地将其除去。但是，如果使用了 *VGVar.sqlIsolationLevel* 来控制某些程序逻辑，则可以将 *VGVar.sqlIsolationLevel* 替换为程序中声明的新变量。务必检查对 *VGLib.connectionService* 的任何使用，这是因为该系统函数的行为可能依赖于 *VGVar.sqlIsolationLevel* 的值。
- 如果从数据库中检索到的值为空，则 EGL 会根据基本类型将未包含在 SQL 记录中的 SQL 主变量初始化为空白或零。仅当为字段设置了 *isNullable=yes* 属性时，EGL 才初始化 SQL 记录中的主变量。如果您关闭了 VAGen 兼容性方式，则在“问题”列表中不会有错误。但是，由于当从数据库中检索到的值为空时字段不会被初始

化，因此，程序的运行方式与在 VisualAge Generator 中的运行方式可能不同。通常，您必须仔细地评估每个程序以确定是否可以安全地关闭 VAGen 兼容性方式。如果您知道对 SQL 定义的所有数据库列都被定义为 NOT NULL，则可以安全地关闭 VAGen 兼容性方式。此外，如果在从 SQL 数据库中检索数据之前始终将主变量初始化为空白或零，则可以安全地关闭 VAGen 兼容性方式，这是因为，在进行数据库访问之前，任何包含空数据的列都已被初始化。务必对未包含在 SQL 记录中的主变量进行初始化。

- EGL 通过将精度增加 1 来支持十进制字段（VAGen PACK 字段）具有偶数精度，但是，它对 SQL WHERE 子句和 EGL prepare 语句中的主变量引用不执行此操作。如果您关闭了 VAGen 兼容性方式，则在“问题”列表中不会有错误。但是，程序的运行方式可能与它在 VisualAge Generator 中的运行方式不同。确切地说，具有偶数精度的十进制字段对于数据库中包含的数据来说可能太小了。通常，您需要仔细地评估每个数据项以确定是否可以安全地关闭 VAGen 兼容性方式。在 VisualAge Generator 中，您可以使用“引用”工具来在所有数据项和记录部件中搜索文本字符串 *evensql = Y*（在 = 号前后有一个空格）。此项搜索可以帮助您确定是否有指定了偶数精度的项或记录。在 EGL 中，可以通过搜索 *decimal(2, decimal(4, .... decimal(18* 来搜索具有偶数精度的十进制字段。如果未使用具有偶数精度的十进制字段，则可以安全地关闭 VAGen 兼容性方式。如果确实使用了具有偶数精度的十进制字段，则需要考虑更改为下一个更大的奇数精度将对 SQL 访问性能产生的影响。如果 EGL 主变量与 SQL 列定义的精度精确匹配，则 SQL 能够为十进制字段提供较好的性能。

如果指定 VAGen 迁移首选项不要对项或变量使用 *evensql=y*，迁移工具将自动使用奇数精度（或者，如果该项具有最大长度，则使用 18）并且对受影响的数据项部件或非共享记录项发出警告消息。

如果您确实关闭了 VAGen 兼容性方式首选项，则务必从每个构建描述符部件中除去 *vagCompatibility="YES"*。如果指定 VAGen 迁移首选项不设置兼容性方式，则迁移工具将自动省略每个构建描述符部件中的 *vagCompatibility="YES"*。



---

## 第 6 部分 语言和运行时差别

在 VisualAge Generator 与 EGL 之间存在各种语言和运行时差别。



---

## 第 10 章 语言和运行时差别

---

### 语言差别

要了解有关 EGL 在哪些方面不能完全代替 VisualAge Generator Developer 的信息，请参阅第 6 页的『确定是否可以迁移到 EGL』。

要了解有关无法精确地迁移至 EGL 语言的 VAGen 语言元素或迁移策略的详细信息，请参阅第 41 页的第 3 章，『处理不确定情况』。

要了解有关每个 VAGen 语言元素如何迁移至 EGL 的详细信息，请参阅第 179 页的附录 B，『VisualAge Generator 与 EGL 语言元素的关系』。

---

### 运行时差别

在将源代码迁移至 EGL 之后，您应该生成并彻底测试代码以确保它的运行方式与在 VisualAge Generator 中的运行方式相同。根据目标环境的不同，特定的运行时差别也会有所变化，如下所示：

- 一般差别。
- 在调试方面的差别。
- 在生成的 COBOL 方面的差别。
- 在生成的 Java 方面的差别。
- 在分布式 CICS 与本机工作站环境之间的差别。
- 在生成的 C++ 与生成的 Java 之间的差别。

#### 一般差别

存在下列运行时行为差别时，在迁移日志或“问题”视图中不会有任何消息。这些问题可能在调试期间发生，也可能在运行生成的 Java 或 COBOL 代码时发生：

- 下列各项适用于文本程序：
  - 如果表单字段的长度不足以包含所有的位以及格式信息（符号、小数点、货币符号和数字分隔符），则会发生运行时错误。
  - 始终考虑非缺省填充字符，即使程序不发出 SET formItem FULL 语句亦如此。
  - 表单上的数组始终使用第一个数组元素的验证属性和定义格式的属性。这可能与 VisualAge Generator 的行为略有不同，后者允许数组元素使用的属性有所不同。要了解详细信息，请参阅第 56 页的『映射数组和属性』。
  - 如果任何映射包含位于 0 行 0 列处的字段，则务必测试使用相应表单的程序，以了解其外观或行为是否存在任何差别。要了解详细信息，请参阅第 58 页的『定位在“行 = 0，列 = 0”处的字段』。
- 当迁移程序时，如果 VAGen REDEFINED 记录不可用，则迁移工具不将 EGL *redefines* 属性包括在数据声明中。这将产生两个单独的记录区域，而不是象在 VisualAge Generator 中那样产生一个具有两个定义的区域。未初始化的或无效的数据会导致错误（包括异常终止）。要了解详细信息，请参阅第 45 页的『重新定义的记录』。

- 在 EGL 中，发生硬 I/O 错误的情况要比 VisualAge Generator 中的情况多：
  - 在 VisualAge Generator 中，非 SQL 记录的 UNQ 是软错误，因此不设置 HRD I/O 错误状态。在 EGL 中，*unique* 是硬错误，因此 *hardIOError* 的测试结果也为真。要了解详细信息，请参阅第 77 页的『I/O 错误值 UNQ 和 DUP』。
  - 对于 iSeries 来说，将把 VAGen I/O 错误值 LOK 迁移为 EGL *deadlock* I/O 错误状态。在 VisualAge Generator 中，LOK 是软错误，因此不设置 HRD I/O 错误状态。在 EGL 中，*deadlock* 是硬错误，因此 *hardIOError* 的测试结果也为真。要了解详细信息，请参阅第 78 页的『I/O 错误值 LOK』。
- 在迁移函数时，如果没有 I/O 错误例程可用，则迁移工具假定 I/O 错误例程不是主函数并将其转换为函数调用。在 VisualAge Generator 中，如果 I/O 错误例程是主函数，并且在运行时发生错误，则 VisualAge Generator 将清除函数的当前执行堆栈并开始一个新堆栈，该堆栈仅带有被指定为 I/O 错误例程的主函数。这还将清除该执行堆栈的任何存储器。在 EGL 中，由于迁移工具已将错误例程转换为函数调用，所以，如果在运行时发生错误，EGL 就会将主函数添加至当前执行堆栈，而不是清除堆栈并开始一个仅带有主函数的新堆栈。这有可能在函数带有本地存储器或参数列表时导致无限循环或使用大量的资源。要了解详细信息，请参阅第 65 页的『I/O 错误例程』。

## 在调试方面的差别

在调试方面存在着一些差别，这些差别可能会对测试产生影响。因为在下列情况中，调试提供的支持与生成的 COBOL 中所提供的支持有所不同，所以，如果为 COBOL 环境生成，您就尤其需要了解这些差别：

- 映射的差别如下所示：
  - 文本表单不支持闪烁。
  - 只有字符字段才支持 *isDecimalDigit* 属性。它是作为软件编辑而不是硬件属性实现的。数字字段也具有软件编辑。要了解详细信息，请参阅第 56 页的『映射字段和数字硬件属性』。
- 对于定义了备用索引记录的带索引记录，DUP I/O 错误值的设置与 VisualAge Generator 中的设置不同。对于 VisualAge Generator 来说，对于后面跟有 SCAN 或 SCANBACK I/O 选项的 SET record SCAN，不会为 SET record SCAN 语句设置 DUP I/O 错误值。对于除了所检索到的最后一个具有重复键的记录外的每一个具有重复键的记录，设置 DUP I/O 错误值。对于 Java 生成来说，后面跟有 *get next* 或 *get previous* 语句的 *set record position* 语句将导致对该 *set record position* 而不是对检索到的第一个具有重复键的记录设置 *duplicate* I/O 错误状态。其余具有重复键的记录将导致设置 *duplicate* I/O 错误状态，这与 VisualAge Generator 中的情况相同。对除了第一个和最后一个具有重复键的记录以外的所有记录设置 EGL *duplicate* 状态。有关带索引记录和备用索引记录以及它们与 *set record position*、*get next* 和 *get previous* 的配合使用的更多信息，请参阅联机帮助。
- SQL 的差别如下所示：
  - 在 EGL 中不支持 ODBC。如果使用除 DB2 以外的 SQL 数据库管理器，则必须获取用于该数据库管理器的 JDBC 驱动程序。
  - JDBC 不支持两阶段落实。因此，存在下列差别：
    - 对 SQL 管理器和 MQ 系列管理器进行单独的调用来进行提交和回滚。因此，即使发生问题，也有可能提交或回滚资源，而不必对其它资源进行相应的提交或回滚。

- EZECONCT (EGL VGLib.connectionService)。在 VisualAge Generator 中, *unit of work* 自变量的 R 选项将连接更改为另一个数据库, 而不结束当前连接。这允许您在同一个工作单元内更新多个数据库。在 EGL 中, *unit of work* 自变量的 R 选项以及 D1C、D2A、D2C 和 D2E 选项全都被看作您指定的是 D1E。D1E 是一阶段落实, 但不自动释放数据库连接。必须显式地请求 DISC、DCURRENT 或 DALL 选项才能与数据库断开连接。要了解详细信息, 请参阅 VGLib.connectionService 的联机帮助。
- EZESQISL (EGL VGVar.sqlIsolationLevel)。在 VisualAge Generator 中, 值 1 表示您需要游标稳定性。在 EGL 中, 值 1 表示您需要可序列化事务。
- 不支持 EZESQRRM (EGL VGVar.sqlerrmc)。
- 不支持 EZESQWN6 (EGL VGVar.sqlwarn[7])。
- EZESQLCA (EGL sysVar.sqlca) 字段受限制。它们未包含 EZESQRRM 和 EZESQWN6 的值。

## 在生成的 COBOL 方面的差别

对于生成的 COBOL 代码, 存在下列差别:

- EGL 生成的 COBOL 文本和基本程序与 VisualAge Generator 程序完全兼容。对于下列任何一种情况, 不必重新生成或重新编译 VAGen 程序:
  - 除 Web 事务以外的 VAGen 程序使用 CALL、DXFR 或 XFER 来作为转移至 EGL 文本程序或基本程序的方法。
  - EGL 程序使用 *call*、*transfer to program*、*transfer to transaction* 或 *show* 来作为转移至 VAGen 程序的方法。

在 EGL 程序与 VAGen 程序之间进行调用或转移时的限制与在两个 VAGen 程序之间进行调用或转移时的限制相似。例如, VAGen 被调用程序不能使用 DXFR 或 XFER 语句来转移至其它程序。同样, EGL 被调用程序不能使用 *transfer to program*、*transfer to transaction* 或 *show* 来转移至其它程序。

- 映射的差别如下所示:
  - 只有字符字段才支持 *isDecimalDigit* 属性。它是作为数字硬件属性实现的。数字字段具有软件编辑。要了解详细信息, 请参阅第 56 页的『映射字段和数字硬件属性』。
  - 不再支持下列设备大小: 6x40、12x40、16x64 和 255x60。要了解详细信息, 请参阅第 51 页的『映射组、映射和设备大小』。

## 在生成的 Java 方面的差别

对于生成的 Java 代码, 存在下列差别:

- EGL 生成的 Java 文本和基本程序与 VisualAge Generator 程序完全兼容。对于下列任何一种情况, 不必重新生成 VAGen 程序:
  - 除 Web 事务以外的 VAGen 程序使用 CALL、DXFR 或 XFER 来作为转移至 EGL 文本程序或基本程序的方法。
  - EGL 程序使用 *call*、*transfer to program*、*transfer to transaction* 或 *show* 来作为转移至 VAGen 程序的方法。

在 EGL 程序与 VAGen 程序之间进行调用或转移时的限制与在两个 VAGen 程序之间进行调用或转移时的限制相似。例如, VAGen 被调用程序不能使用 DXFR 或

XFER 语句来转移至其它程序。同样, EGL 被调用程序不能使用 *transfer to program*、*transfer to transaction* 或 *show* 来转移至其它程序。

## 主机与工作站环境之间的差别

如果从为主机环境(如 CICS)生成更改成为工作站环境(如本机 AIX)生成,则需要考虑下列各项:

- 主机环境的整理顺序为 EBCDIC。Java 生成的整理顺序为 UNICODE。因此,您需要查看下列各项:
  - 范围匹配有效表
  - 为键的高位值和低位值编写的特定值
  - 针对两个文件的匹配的键的比较
- 最终用户需要知道下列各项:
  - 功能键的映射,原因是工作站键盘通常没有针对 PF13 - PF24 和 PA1 - PA3 的键。
  - 在切换进/出 DBCS 方式时不必输入 SO/SI 字符。
- 通过 CALL 从 AIX 本机传递至 CICS 的所有记录的总长度不能超出 32567。这是 CICS 限制。如果选择是为 CICS 生成的并且使用了缺省值 parmform=COMMPTR,则所有记录的总长度可能会超出此限制。如果选择使用了 parmform=COMMDATA,则所有记录的总长度在此限制内。
- 还应查看下列各节中的信息:
  - 『在分布式 CICS 与本机工作站环境之间的差别』
  - 第 169 页的『在生成的 C++ 与生成的 Java 之间的差别』

## 在分布式 CICS 与本机工作站环境之间的差别

要在分布式环境中运行生成的 EGL 代码,必须更改为作为本机进程运行,而不是使用“在 Transaction Series (TX Series 或 CICS) 下运行”选项。以下列表概述了从 CICS 环境移至本机环境时遇到的差别或所需进行的更改。此列表使用 VAGen 术语,但您必须在对应的 EGL 语言元素中进行更改。要确定对应的 EGL 语言元素,请参阅第 179 页的附录 B,『VisualAge Generator 与 EGL 语言元素的关系』。

- 一般差别如下所示:
  - 服务器上的同一地址空间内只能有一个用户运行。用户在他们自己的工作站上运行。
  - 不支持客户机工作单元。
  - 已从 C++ 生成更改为 Java 生成。您务必回顾第 169 页的『在生成的 C++ 与生成的 Java 之间的差别』一节。
  - 当从 CICS 迁移至本机环境时,务必测试性能和可伸缩性。
  - CICS 环境与本机环境之间的通信协议是不同的。您必须确定计划使用的协议,然后相应地更改 EGL 链接选项部件和资源关联部件。
- 在本机环境中,不支持下列特定于 CICS 的特殊函数和服务例程:
  - 用于写 CICS 日志条目的 AUDIT (EGL sysLib.audit)。对于本机环境,您可以创建自己的名为 AUDIT 的非 EGL 程序来将类似的信息写入文件。
  - 用于删除临时存储器队列的 EZEPURGE (EGL sysLib.purge)。必须除去对 sysLib.purge 的引用。另外,当在 ZOSCICS 环境中运行时,可以检查

sysVar.systemType 并且仅使用 sysLib.purge。如果使用此技术，则务必包括 EGL 构建描述符选项 eliminateSystemDependentCode="YES"。

- 用于设置远程文件位置、远程程序或远程事务启动位置（远程事务是使用 CREATX（EGL sysLib.startTransaction）启动的）的 EZELOC（EGL sysLib.remoteSystemID）。
- 在本机环境中，不支持特定于 CICS 的资源关联。必须将资源关联部件更改为使用 EGL 本机环境所支持的选项。以下是本机环境的生成所不支持的特定于 CICS 的资源关联：
  - CICS 假脱机文件。
  - 瞬时数据队列，包括触发器级别为 1 的瞬时数据队列。
  - 临时存储器队列。
  - 本地 VSAM 文件，AIX 环境除外。
- 在本机环境中，不支持 CICS 提供的下列功能部件：
  - 安全服务。
  - 数据库连接和保留时间。
  - CICS 文件管理，包括可恢复文件的使用。
  - 真实的分段支持。
  - 程序管理。
- 在程序之间转移时的区别如下所示：
  - 对于除 Web 事务以外的主程序，CICS 中的 XFER 语句转移至下一个事务标识。对于本机环境，XFER 语句转移至一个程序名。因此，必须将所有 EGL *transfer to transaction* 和 *show* 语句修改为指定程序名。
  - 在 CICS 环境中，支持面向非 VAGen 程序的 XFER 或 DXFR。对于本机环境，不支持面向非 EGL 程序的 *transfer to program*、*transfer to transaction* 和 *show*。
- 提交差别和回滚差别如下所示：
  - CICS 支持两阶段落实。本机环境仅支持单阶段落实。
  - 对于 CICS，可以将文件定义为可恢复文件。对于本机环境，这是不可能的。
  - 消息队列的提交或回滚是与 CICS 环境中的其它资源同时进行的。本机环境仅支持单阶段落实，因此，如果在提交或回滚期间发生问题，消息队列就可能不能与 SQL 资源同时被提交或回滚。
- CALL CREATX（EGL sysLib.startTransaction）的差别如下所示：
  - CALL CREATX 在 CICS 中启动另一个事务并使用参数 *prid* 和 *recip*。EGL sysLib.startTransaction 为本机环境启动另一个程序并忽略参数 *prid* 和 *recip*。如果为本机环境进行生成，则必须至少将 sysLib.startTransaction 更改为指定一个程序名。
  - CICS 既支持本地 CALL CREATX 也支持远程 CALL CREATX。EGL sysLib.startTransaction 仅支持启动本地程序。
- 使用 EZECONCT（EGL VGLib.connectionService）的 SQL 连接服务。在 VisualAge Generator 中，对于 CICS 环境，EZECONCT 忽略密码。在 EGL 中，对于本机环境，VGLib.connectionService 使用密码。请参阅第 169 页的『在生成的 C++ 与生成的 Java 之间的差别』以了解由于更改为 Java 生成而造成的其它差别。
- EZE 特殊数据字差别如下所示：

- EZEAPP (EGL sysVar.transferName)。在 VisualAge Generator 中，对于 CICS 环境，当 EZEAPP 与 XFER 语句配合使用时，EZEAPP 包含要启动的新事务的名称。在 EGL 中，对于本机环境，当 sysVar.transferName 与 transfer to transaction 或 show 语句配合使用时，sysVar.transferName 包含要启动的新程序的名称。
- EZEDEST (EGL sysVar.resourceAssociation)。在 VisualAge Generator 中，对于 CICS 环境，当程序运行时，EZEDEST 包含与记录相关联的系统资源名。在 EGL 中，对于本机环境，sysVar.resourceAssociation 也包含与记录相关联的系统资源名。但是，根据系统和文件类型的不同，信息的格式将有所变化。因此，由于正在同时更改运行时环境和文件类型，所以您必须查看对 sysVar.resourceAssociation 的任何使用以确保程序提供的信息对于本机环境和文件类型来说是正确的。
- EZEDESTP (EGL converseVar.printerAssociation)。在 VisualAge Generator 中，对于 CICS 环境，EZEDESTP 包含与打印文件相关联的目标。在 EGL 中，对于本机环境，converseVar.printerAssociation 也包含与打印文件相关联的文件名。但是，根据系统和文件类型的不同，信息的格式将有所变化。因此，由于正在同时更改运行时环境和文件类型，所以您必须查看对 converseVar.printerAssociation 的任何使用以确保程序提供的信息对于本机环境和文件类型来说是正确的。
- EZELTERM (EGL sysVar.terminalID)。在 VisualAge Generator 中，对于 CICS 环境，EZELTERM 包含 CICS 终端标识，并且等同于 EZEUSR。在 EGL 中，对于本机环境，sysVar.terminalID 是根据 Java 虚拟机系统属性 user.name 初始化的。如果无法检索此属性，则 sysVar.terminalID 包含空白。sysVar.terminalID 并不等同于 sysVar.sessionID (后者是 EZEUSR 的 EGL 替换项)。
- EZERCODE (EGL sysVar.returnValue)。在 VisualAge Generator 中，对于 CICS 环境，不会将 EZERCODE 中的值传递给系统或调用程序。在 EGL 中，对于本机环境，EZERCODE 中的值将被忽略。
- EZERT8 (EGL sysVar.errorCode)。在 VisualAge Generator 中，对于 CICS 环境，EZERT8 具有两种格式中的其中一种格式：
  - *RSnnnnnn*，其中 *nnnnnn* 是 VAGen 返回码，它基于文件访问以及所发生的问题。
  - *nnnnnnnn*，其中，前两个字符是 CICS EXEC 接口块中 EIBFN 的第一个字节的十六进制表示。其余 6 个字符包含 CICS EXEC 接口块中 EIBRCODE 的字节 0-2 的十六进制表示。

在 EGL 中，对于本机环境，返回码信息随文件类型的不同而有所变化。您应该查看 sysVar.errorCode 的使用以确保正在检查的值对于环境和文件类型来说是正确的。

- EZESEGTR (EGL sysVar.transactionID)。在 VisualAge Generator 中，对于 CICS 环境，EZESEGTR 被初始化为当前事务标识，并且还用来设置在 CONVERSE 之后生效的新事务标识。EZESEGTR 可以用来控制程序逻辑。在 EGL 中，对于本机环境，EZESEGTR 被忽略，并且不能用来控制程序逻辑。
- EZEUSR (EGL sysVar.sessionID)。在 VisualAge Generator 中，对于 CICS 环境，EZEUSR 包含 CICS 终端标识，并且等同于 EZELTERM。在 EGL 中，对于本机环境，sysVar.sessionID 是根据 Java 虚拟机系统属性 user.name 初始化的。如果无法检索此属性，则 sysVar.userID 包含空白。
- EZEUSRID (EGL sysVar.userID)。在 VisualAge Generator 中，对于 CICS 环境，如果用户已登录到系统，则 EZEUSRID 包含 CICS 用户标识，否则它包含空白。在 EGL 中，对于本机环境，sysVar.userID 是根据 Java 虚拟机系统属性 user.name 初始化的。如果无法检索此属性，则 sysVar.userID 包含空白。

## 在生成的 C++ 与生成的 Java 之间的差别

如果从生成的 C++ 更改为生成的 Java, 则存在下列差别:

- 生成的 Java 不能与 VAGen 生成的 C++ 程序一起工作。为 Java 生成的 EGL 程序不能转移至为 C++ 生成的 VAGen 程序, 也不能从后者转移至前者。为 Java 生成的 EGL 程序可以调用为 C++ 生成的 VAGen 被调用批处理程序。
- 在 VAGen 生成的 C++ 程序中, 二进制数据是以 Intel 格式 (逆字节顺序) 存储的。在 EGL 生成的 Java 程序中, 二进制数据不是以 Intel 格式存储的。EGL 转换表对生成的 Java 程序与生成的 C++ 程序之间的调用所传递的二进制数据进行转换。但是, 在生成的 Java 程序中使用由 C++ 程序所写的文件之前, 必须编写一个程序来转换那些文件中的任何二进制数据。
- 对于生成的 Java 程序, `transfer` 或 `show` 语句的名称解析规则如下所示:
  - 如果 `transfer` 或 `show` 语句显式地指定包名和程序名, 则这就是所使用的程序 (例如: `transfer to program mypackage.program1`)。
  - 如果 `transfer` 或 `show` 语句仅显式地指定程序名, 则将使用下列各项中的第一个适用项来解析程序名:
    - 包含该 `transfer` 或 `show` 语句的文件显式地导入包和程序 (例如: `import mypackage.program1` 和 `transfer to program program1`)。
    - 转移目标程序与该 `transfer` 或 `show` 语句位于同一个包中。
    - 包含 `transfer` 或 `show` 语句的文件导入一个包, 该包包含转移目标程序 (例如 `import mypackage.*` 和 `transfer to program program1`)。
    - 在生成时使用的链接选项部件包含转移源程序的转移链接条目, 该条目指定了转移目标程序以及包含该程序的包。对于 `show` 语句来说, 转移链接条目必须具有转移目标事务条目的格式。
  - 如果 `transfer` 或 `show` 语句使用了 `sysVar.transferName`, 则必须使用链接选项部件来指定包含转移目标程序的包。
- 一般差别如下所示:
  - 当使用 VisualAge Generator 生成的 C++ 代码时, 资源关联是在运行时完成的。在 EGL 中, 您可以选择在生成时指定资源关联信息并将其生成到属性文件中。将资源关联构建描述符设置为指向资源关联部件。并且, 将 `genProperties` 构建描述符设置为 `GLOBAL` 或 `PROGRAM`。有关这些构建描述符选项的详细信息, 请参阅联机帮助。
- 映射的差别如下所示:
  - 文本表单不支持闪烁。
  - 只有字符字段才支持 `isDecimalDigit` 属性。它是作为软件编辑而不是硬件属性实现的。数字字段也具有软件编辑。要了解详细信息, 请参阅第 56 页的『映射字段和数字硬件属性』。
- 对于定义了备用索引记录的带索引记录, `DUP I/O` 错误值的设置与 VisualAge Generator 中的设置不同。对于 VisualAge Generator 来说, 对于后面跟有 `SCAN` 或 `SCANBACK I/O` 选项的 `SET record SCAN`, 不会为 `SET record SCAN` 语句设置 `DUP I/O` 错误值。对于除了所检索到的最后一个具有重复键的记录外的每一个具有重复键的记录, 设置 `DUP I/O` 错误值。对于 Java 生成来说, 后面跟有 `get next` 或 `get previous` 语句的 `set record position` 语句将导致对该 `set record position` 而不是对检索到的第一个具有重复键的记录设置 `duplicate I/O` 错误状态。其余具有重复键的记录将导致设置 `duplicate I/O` 错误状态, 这与 VisualAge Generator 中的情况相同。对除

了第一个和最后一个具有重复键的记录以外的所有记录设置 EGL duplicate 状态。有关带索引记录和备用索引记录以及它们与 *set record position*、*get next* 和 *get previous* 的配合使用的更多信息，请参阅联机帮助。

- SQL 的差别如下所示：
  - 在 EGL 中不支持 ODBC。如果使用除 DB2 以外的 SQL 数据库管理器，则必须获取用于该数据库管理器的 JDBC 驱动程序。
  - JDBC 不支持两阶段落实。因此，存在下列差别：
    - 对 SQL 管理器和 MQ 系列管理器进行单独的调用来进行提交和回滚。因此，即使发生问题，也有可能提交或回滚资源，而不必对其它资源进行相应的提交或回滚。
    - EZECONCT (EGL VGLib.connectionService)。在 VisualAge Generator 中，*unit of work* 自变量的 R 选项将连接更改为另一个数据库，而不结束当前连接。这允许您在同一个工作单元内更新多个数据库。在 EGL 中，*unit of work* 自变量的 R 选项以及 D1C、D2A、D2C 和 D2E 选项全都被看作您指定的是 D1E。D1E 是一阶段落实，但不自动释放数据库连接。必须显式地请求 DISC、DCURRENT 或 DALL 选项才能与数据库断开连接。要了解详细信息，请参阅 VGLib.connectionService 的联机帮助。
    - EZESQISL (EGL VGVar.sqlIsolationLevel)。在 VisualAge Generator 中，值 1 表示您需要游标稳定性。在 EGL 中，值 1 表示您需要可序列化事务。
    - 不支持 EZESQRRM (EGL VGVar.sqlerrmc)。
    - 不支持 EZESQWN6 (EGL VGVar.sqlwarn[7])。
    - EZESQLCA (EGL sysVar.sqlca) 字段受限制。它们未包含 EZESQRRM 和 EZESQWN6 的值。
- EZE 特殊数据字差别如下所示：
  - EZECONVT (EGL sysVar.callConversionTable)。在 VisualAge Generator 中，对于 C++ 生成，转换表名具有 ELAxxyyy 格式，其中 xx 指示系统，yyy 指示语言。在 EGL 中，对于 Java 生成，EGL 提供的转换表名具有 CSOBxxxx 格式，其中 CSO 是固定的前缀，B 指示目标系统的字节顺序，xxxx 指示目标系统的代码页。B 的有效值是 X (表示 Unix 系统)、I (表示 Intel 系统) 和 E (表示 EBCDIC 系统)。EGL 自动将 EGL 表名转换为 CSO 表名，因此您不需要更改任何代码。但是，EGL 未向您提供创建您自己的 CSO 转换表的功能。
  - EZERCODE (EGL sysVar.returnValue)。在 VisualAge Generator 中，对于 C++ 生成，将把 EZERCODE 传递回给系统或调用程序。如果该程序异常结束，则传递回 VAGen 返回码，而不是传递回 EZERCODE 中的值。在 EGL 中，对于 Java 生成，将忽略 EZERCODE。

---

## 第 7 部分 附录



---

## 附录 A. 保留字

---

### EGL 保留字

在 EGL 中有大量的保留字。保留字不能被用作部件名。如果部件名是 EGL 保留字，则迁移工具将把函数、数据项、记录和映射重命名。迁移工具不会将表、映射组或程序重命名。EGL 保留字如下所示：

表 60. EGL 保留字

字母	保留字
A	absolute, add, all, any, as
B	bigInt, bin, bind, blob, boolean, by, byName, byPosition
C	call, case, char, clob, close, const, continue, converse, current
D	dataItem, dataTable, date, dbChar, decimal, decrement, delete, display, dliCall
E	else, embed, end, escape, execute, exit, externallyDefined
F	false, field, first, float, for, forEach, form, formGroup, forUpdate, forward, freeSql, from, function
G	get, goto
H	handler, hex, hold
I	if, import, in, inOut, insert, int, interval, into, is, isa
L	label, languageBundle, last, library, like
M	matches, mbChar, money, move
N	new, next, no, noRefresh, not, nullable, num, number, numc
O	onEvent, onException, open, openUI, otherwise, out
P	pacf, package, pageHandler, passing, prepare, previous, print, private, program, psb
R	record, ref, relative, replace, return, returning, returns
S	scroll, self, set, show, singleRow, smallFloat, smallInt, sql, sqlCondition, stack, string
T	this, time, timeStamp, to, transaction, transfer, true, try, type
U	unicode, update, url, use, using, usingKeys
W	when, while, with, withinParent
Y	yes

注：EGL 部件名不能以 EZE 或符号 # 开头。

---

### EGL 枚举字

VAGen 的每个属性都具有固定的有效值列表。EGL 提供了一个枚举列表，该列表包含相应属性的有效值。例如，EGL 将 align 属性的有效值存储在名为 AlignKind 的 EGL 枚举中。此外，EGL 允许对属性值使用变量和表达式。当 EGL 解析属性值时，EGL 首先查找具有该名称的变量。如果没有具有该名称的变量，则 EGL 将检查与该属性相对应的枚举以验证属性值。例如，在表单中，如果变量字段具有属性 align=center，则 EGL 首先在该表单中查找名为 center 的变量。如果在该表单中没有名为 center 的变量，则

EGL 接着使用 `AlignKind.center`。表 61 显示了 EGL 枚举名、该枚举中的有效值列表以及一些部件，在那些部件中，属性的使用方式会导致在变量名与枚举值之间发生冲突。

此外，还有若干个可以使用 EGL 库常量和变量来进行设置的属性。表 62 列示了一些 EGL 属性，这些属性是使用 EGL 库常量和变量来进行引用的。但是，迁移工具仅使用表中标有星号 (\*) 的 EGL 库变量和常量。

为了避免您在编写新代码时发生冲突，可以执行下列其中一种操作：

- 在定义变量时，如果该变量的名称可能与有效值列表中的任何值相匹配，则应避免使用该变量名。
- 始终对属性枚举值进行限定（例如，始终指定 `align = AlignKind.center`）。
- 每当发生冲突时，必须对属性枚举值进行限定（例如，每当在特定表单中存在名为 `center` 的字段时，指定 `align = AlignKind.center`）。

迁移工具根据预期的使用频率、迁移工具实际使用的属性以及迁移性能考虑事项来结合使用这三种技术。为了最大程度地降低对重命名部件和变量的需求，迁移工具将执行下列操作：

- 由于 `YES` 和 `NO` 是 EGL 保留字，所以迁移工具始终将名为 `YES` 或 `NO` 的部件或变量重命名。
- 始终将名为 `PFn` 的部件或变量重命名，其中 `n` 介于 1 与 24 之间。实际上，迁移工具将这些值视为保留字。这可以改善程序和表单的 `helpKey` 和 `validationBypassKeys` 属性的外观，并且不会影响迁移性能。
- 在 `FormGroup` 中，迁移工具执行下列操作：
  - 始终对 `deviceType` 属性值进行完全限定。这样，由于不需要查看 `FormGroup` 中所有表单中的所有字段名，因此提高了性能。
  - 由于映射组没有等效的 `VAGen` 属性，所以迁移工具从不使用 `helpKey` 或 `validationBypassKeys` 属性。
- 在一个表单中，仅当该表单中的一个或多个字段与其中一个可以在该表单中使用的枚举值、EGL 常量或 EGL 变量同名时，迁移工具才对该表单中的属性进行限定。如果该表单上的任何字段发生冲突，则迁移工具使用相应的枚举或库名来对该表单中的所有字段的所有属性值进行完全限定。但是，为了确定是否有任何字段发生冲突，迁移工具将忽略 `IndexOrientationKind` 值（`across` 和 `down`），这是因为迁移工具从不使用此属性。
- 由于在部件名与枚举值之间不可能发生冲突，因此，从不对其它部件类型中的属性值进行限定。

**注：**表 2 和表 3 仅供参考。因此，这些表包括库、`ConsoleUI` 和 `PageHandler` 部件，迁移工具从不创建这些部件。

表 61. EGL 枚举

EGL 枚举	对变量名施加了限制的有效值列表	可能会发生冲突的部件
<code>AlignKind</code>	<code>center, left, none, right</code>	表单
<code>CallingConventionKind</code>	<code>I4GL, Library</code>	
<code>CaseFormatKind</code>	<code>defaultCase, lower, upper</code>	<code>ConsoleUI</code>

表 61. EGL 枚举 (续)

EGL 枚举	对变量名施加了限制的有效值列表	可能会发生冲突的部件
ColorKind	black, blue, cyan, defaultColor, green, magenta, red, yellow, white 注意: black 仅对 ConsoleUI 才有效。	表单 ConsoleUI
DataSource	databaseConnection, reportData, sqlStatement	报告
DeviceTypeKind	doubleByte, singleByte	FormGroup
DisplayUseKind	button, hyperlink, input, output, secret, table	任何记录都可以与 PageHandler 配合使用。
EventKind	afterDelete, afterField, afterInsert, afterOpenUI, afterRow, beforeDelete, beforeField, beforeInsert, beforeOpenUI, beforeRow, menuAction, onKey	ConsoleUI
ExportFormat	html, pdf, text	ConsoleUI
HighlightKind	blink, defaultHighlight, noHighlight, reverse, underline	表单 ConsoleUI
IndexOrientationKind	across, down	表单
IntensityKind	bold, defaultHighlight, dim, invisible, normalIntensity	表单 ConsoleUI
LineWrapKind	character, compressed, word	ConsoleUI
OutlineKind	bottom, left, right, top 注: sysLib.box 是一个常量, 它等同于 [left,right,top,bottom]。sysLib.noOutline 是一个常量, 它表示没有轮廓。	表单
PfKeyKind	pfn, 其中 (1 <= n <=24)	表单 FormGroup 程序
ProtectKind	skip, no, yes	表单
SelectTypeKind	index, value	任何记录都可以与 PageHandler 配合使用。
SignKind	leading, none, parens, trailing	表单 任何记录都可以与 PageHandler 配合使用。

表 62. 使用 sysLib 常量和 sysVar 变量的 EGL 枚举

EGL 属性	EGL 常量和变量 (* 表示值由迁移工具使用)	可能会发生冲突的部件
dateFormat	strLib.defaultDateFormat* strLib.eurDateFormat strLib.isoDateFormat strLib.jisDateFormat strLib.usaDateFormat VGVar.systemGregorianDateFormat* VGVar.systemJulianDateFormat* 注: 迁移工具仅在 DataItem 部件中才使用 defaultDateFormat。	表单 ConsoleUI

表 62. 使用 sysLib 常量和 sysVar 变量的 EGL 枚举 (续)

EGL 属性	EGL 常量和变量 (* 表示值由迁移工具使用)	可能会发生冲突的部件
fillCharacter	strLib.nullFill*	表单
outline	sysLib.box* sysLib.noOutline 注: outlineKind 是其它轮廓值的枚举。	表单
timeFormat	strLib.defaultTimeFormat strLib.eurTimeFormat strLib.isoTimeFormat strLib.jisTimeFormat strLib.usaTimeFormat	任何记录都可以与 PageHandler 配合使用。
timeStampFormat	strLib.db2TimeStampFormat strLib.odbcTimeStampFormat	ConsoleUI

## SQL 保留字

有许多 SQL 保留字不能在 SQL 子句中使用。如果部件名是 SQL 保留字，则迁移工具将把函数、数据项、记录和映射重命名。迁移工具不会将表、映射组或程序重命名。SQL 保留字如下所示：

字母	保留字
A	absolute, action, add, alias, all, allocate, alter, and, any, are, as, asc, assertion, at, authorization, avg
B	begin, between, bigint, binaryLargeObject, bit, bit_length, blob, boolean, both, by
C	call, cascade, cascaded, case, cast, catalog, char, char_length, character, character_length, characterLargeObject, characterVarying, charLargeObject, charVarying, check, clob, close, coalesce, collate, collation, column, comment, commit, connect, connection, constraint, constraints, continue, convert, copy, corresponding, count, create, cross, current, current_date, current_time, current_timestamp, current_user, cursor
D	data, database, date, dateTime, day, deallocate, dec, decimal, declare, default, deferrable, deferred, delete, desc, describe, diagnostics, disconnect, distinct, domain, double, doublePrecision, drop
E	else, end, endExec, escape, except, exception, exec, execute, exists, explain, external, extract
F	false, fetch, first, float, for, foreign, found, from, full
G	get, getCurrentConnection, global, go, goto, grant, group
H	having, hour
I	identity, image, immediate, in, index, indicator, initially, inner, input, insensitive, insert, int, integer, intersect, into, is, isolation
J	join
K	key
L	language, last, leading, left, level, like, local, long, longint, lower, ltrim
M	match, max, min, minute, module, month
N	national, nationalCharacter, nationalCharacterLargeObject, nationalCharacterVarying, nationalCharLargeObject, nationalCharVarying, natural, nchar, ncharVarying, nlob, next, no, not, null, nullIf, number, numeric
O	octet_length, of, on, only, open, option, or, order, outer, output, overlaps
P	pad, partial, position, prepare, preserve, primary, prior, privileges, procedure, public
R	raw, read, real, references, relative, restrict, revoke, right, rollback, rows, rtrim, runtimeStatistics

字母	保留字
S	schema, scroll, second, section, select, session, session_user, set, signal, size, smallint, some, space, sql, sqlcode, sqlerror, sqlstate, substr, substring, sum, system_user
T	table, tablespace, temporary, terminate, then, time, timestamp, timezone_hour, timezone_minute, tinyint, to, trailing, transaction, translate, translation, trim, true
U	uncatalog, union, unique, unknown, update, upper, usage, user, using
V	values, varbinary, varchar, varchar2, varying, view
W	when, whenever, where, with, work, write
Y	year
Z	zone

## 需要进行特殊处理的 SQL 保留字

在 EGL 中，如果将下列 SQL 保留字用作 SQL 表名或列名，则需要对它们进行特殊的处理：

call, from, group, having, insert, order, select, set, union, update, values, where

要使用这些 SQL 保留字，请执行下列操作：

- 要对 sqlRecord 中的项指定 column 属性，请指定：

```
column = "\"reservedWord\""
```

例如：

```
column = "\"FROM\""
```

- 要对 sqlRecord 指定 tableNames 属性，请指定：

```
tableNames = [{"\"reservedWord2\""}]
```

例如：

```
tableNames = [{"\"ORDER\""}]
```

- 要将其中一个保留字用作记录的 defaultSelectCondition 中的 SQL 列名，请指定：

```
defaultSelectCondition = #sqlCondition{ "reservedWord" = ... }
```

例如：

```
defaultSelectCondition = #sqlCondition{ "ORDER" = ... }
```

- 要将其中一个保留字用作 SQL I/O 语句中的 SQL 列名，请指定：

```
... #sql{ select "reservedWord" from "reservedWord2" .... } ...
```

例如：

```
... #sql{ select "FROM" from "ORDER" .... } ...
```

## Java 保留字

Java 具有不能用作包名的保留字。如果正在生成 Java，则您可能想避免使用下列名称：

字母	保留字
A	abstract
B	boolean, break, byte

字母	保留字
C	case, catch, char, class, const, continue
D	default, do, double
E	else, extends
F	false, final, finally, float, for
G	goto
I	if, implements, import, instanceof, int, interface
L	long
N	native, new, null
P	package, private, protected, public
R	return
S	short, static, super, switch, synchronized
T	this, throw, throws, transient, true
V	void, volatile
W	while

---

## 附录 B. VisualAge Generator 与 EGL 语言元素的关系

此附录中的表具有 3 列:

- VisualAge Generator 4.5 - 此列显示 VAGen 语言元素。在与部件类型相关的各节中, 表的组织以及使用的术语与 VAGen 用户界面相对应。语句、EZE 字和服务例程的表是根据语句、EZE 字或服务例程的类型组织的。
- 迁移工具生成的 EGL - 此列显示对应的 EGL 语言元素。此列仅显示迁移所需的信息, 而未打算提供完整的 EGL 语法。对于某些 EGL 语言元素来说, 可能还有其它属性、值和选项可用。例如, EGL set 语句提供了在 VisualAge Generator 中不可用的其它选项。在这些表中仅列示了与 VAGen 语言元素相对应的 set 语句选项。您可以将此列用作在 EGL 文档中查找更详细信息的指南。
- 迁移工具注意事项 - 此列包含有关迁移工具如何处理从 VisualAge Generator 到 EGL 的转换的附加信息。此列还提供了对有关不确定情况的章节的引用, 并且, 在有必要的时候, 还提供了有关带关联部件迁移和不带关联部件迁移的详细信息以及迁移 VAGen 语言元素时可能会发生的潜在问题。

对于每种部件类型, 各节的第一个表中的第一行提供了:

- VisualAge Generator 4.5 - 可以在 VisualAge Generator 的各个窗口中对该部件类型指定的信息的概述。
- EGL - 对应的 EGL 部件类型的整体 EGL 语法 (使用迁移工具所使用的语法)。其它语法变体也是有可能的。例如, 当迁移 VAGen 表时, 迁移工具始终将 dataTable 内容放在 dataTable 结构后面, 因此这就是在『表』这一节中显示的语法。EGL 语法还允许将 dataTable 内容放在 dataTable 结构前面。

在表中使用以下语法:

- | - 几个选项的选项。无论是在 VisualAge Generator 4.5 列中还是在 EGL 列中, 选项的顺序都是相同的。
- 符号列表 - 较长的选项或值列表的选项。无论是在 VisualAge Generator 4.5 列中还是在 EGL 列中, 选项的顺序都是相同的。
- 斜体 - 在从 VisualAge Generator 迁移时由迁移工具填写的值, 或者是您在编写新的 EGL 语句时填写的值。
- 粗体 - 必须按显示的样子指定的 EGL 关键字和符号。
- { } - 括住可以重复 0 到 n 次的信息。
- { } - 括住 EGL 属性列表; 各属性之间始终由逗号分隔。
- [ ] - 括住可选信息。
- [ ] - 括住 EGL 值列表; 各个值之间始终由逗号分隔。

---

### 一般语约定

VisualAge Generator 与 EGL 的整体语法有一些差别。

表 63. 一般语法约定

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>注释是以下列格式指定的:</p> <ul style="list-style-type: none"> <li>• 程序、表和记录的 prolog。</li> <li>• 项和函数的描述。</li> <li>• 函数中的注释由下列各项指示: <ul style="list-style-type: none"> <li>- 分号 (;)。同一行上位于分号后面的所有内容都被视为注释。</li> <li>- /*。同一行上位于 /* 后面的所有内容都被视为注释。</li> </ul> </li> </ul>	<p>注释是以下列格式指定的:</p> <ul style="list-style-type: none"> <li>• // 指示一行注释。同一行上位于 // 后面的所有内容都被视为注释。注释只有一行。</li> <li>• /* 注释 */。位于 /* 后面的所有内容都被视为注释, 直到下一个 */ 为止。注释可以跨多行。</li> </ul>	<p>迁移工具按照以下方式进行转换:</p> <ul style="list-style-type: none"> <li>• 将 prolog 和部件描述转换为 EGL // 行注释。</li> <li>• 将函数中的注释转换为 /* 注释 */。</li> </ul>
<p>根据语言环境的不同, 小数点可以是句点或逗号。</p>	<p>在开发期间, 小数点始终是句点。根据运行时语言环境的不同, 生成和运行时使用句点或逗号。</p>	<p>在迁移期间, 如果语言环境在缺省情况下使用逗号作为小数点, 或者您选择了迁移语法首选项“将逗号小数点更改为句点小数点”, 则迁移工具将把逗号更改为句点。</p>
<p>属性是在一个专用编辑器中使用复选框和下拉列表等控件输入的。</p>	<p>属性是在文本编辑器中输入的, 并且必须用逗号将它们隔开。</p>	<p>没有特殊的注意事项。</p>
<p>值列表是在专用的编辑器中输入的。例如, SQL 记录的表名是在“SQL 行属性”窗口中输入的。</p>	<p>必须将值列表括在方括号 [ ] 中。</p>	<p>没有特殊的注意事项。</p>
<p>引用其它部件的属性值是在专用的编辑器中输入的。例如, 映射变量字段的编辑例程是在“变量字段属性”窗口的“编辑”页上输入的。</p>	<p>必须将引用其它部件的属性值引在双引号中。</p>	<p>没有特殊的注意事项。</p>

## 数据项

『数据项』一节由下列各表组成:

- 数据项 - 一般语法、数据类型、长度、小数位数和描述 (第 181 页的表 64)
- 缺省映射属性和用户界面属性 - 一般信息 (第 183 页的表 65)
- 缺省映射属性和用户界面属性 - 一般编辑 (第 183 页的表 66)
- 缺省映射属性和用户界面属性 - 数字编辑 (第 185 页的表 67)
- 缺省映射属性和用户界面属性 - 错误消息 (第 186 页的表 68)
- 用户界面属性 - 标签和帮助 (第 186 页的表 69)

**注:** 在 EGL 中, 一个 dataItem 部件只有一组编辑属性和消息属性。尽管此发行版的 EGL 不支持用户界面记录, 但是迁移工具会将数据项的映射属性和用户界面属性合并到一起。迁移工具还将标签属性和帮助用户界面属性转换为它们的 EGL 等效属性。这样可以最大程度地保留数据项信息。如果您开发任何新的 EGL 表单或 pageHandler, 则可以使用此信息。

表 64. 数据项 - 一般语法、数据类型、长度、小数位数和描述

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 数据项部件:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• 基本信息: <ul style="list-style-type: none"> <li>- 数据类型</li> <li>- 长度</li> <li>- 小数位数</li> <li>- 描述</li> </ul> </li> <li>• 缺省映射属性</li> <li>• 用户界面 (UI) 属性</li> </ul>	<p>EGL 语法示例:</p> <pre>// Description DataItem itemName   dataType(lengthInformation)   { [ { formattingProperties } ]     [ { validationProperties } ]     [ { pageItemProperties } ]   } end</pre>	<p>迁移工具使用 VAGen 数据类型、长度和小数位数来确定 EGL dataType 和 lengthInformation.</p> <p>迁移工具将 VAGen 缺省映射属性和用户界面属性合并成一组 EGL 定义格式的属 性、验证属性和页项属性。</p>
<p>字符项类型:</p> <ul style="list-style-type: none"> <li>• Char</li> <li>• Hex</li> <li>• DBCS</li> <li>• 混合</li> <li>• Unicode (仅限于 VisualAge for Java)</li> </ul> <p>长度是字符数。在记录编辑器中, 还可以显示字节数。</p>	<p>对应的字符项类型:</p> <ul style="list-style-type: none"> <li>• char</li> <li>• hex</li> <li>• dbchar</li> <li>• mbchar</li> <li>• unicode</li> </ul> <p>长度是字符数。</p>	<p>迁移工具将字符数据项转换为对应的类型 and 长度。</p>
<p>数字字符 (分区十进制) 类型:</p> <ul style="list-style-type: none"> <li>• Num</li> <li>• Numc</li> </ul> <p>长度是总位数, 最大值为 18。小数位数是小数点右边的位数。在记录编辑器中, 还可以显示字节数。</p>	<p>对应的数字类型:</p> <ul style="list-style-type: none"> <li>• num</li> <li>• numc</li> </ul> <p>精度是总位数。小数位是小数点右边的位数。</p> <p>对于调试和 Java 生成来说, num 字段的最大精度是 32, 对于 COBOL 生成来说, 是 31。numc 字段的最大精度是 18。</p>	<p>迁移工具转换为对应的类型、精度和小数位。如果小数位数为 0, 则迁移工具省略小数位。</p>

表 64. 数据项 - 一般语法、数据类型、长度、小数位数和描述 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>压缩十进制类型:</p> <ul style="list-style-type: none"> <li>• Pacf</li> <li>• Pack</li> </ul> <p>长度是总位数, 最大值为 18。小数位数是小数点右边的位数。Pacf 的长度必须是奇数或 18。Pack 的长度可以是奇数或偶数。除长度 18 以外, 可以将偶数长度记录在数据项定义中, 但是, 对于测试、生成以及在数据项编辑器和记录编辑器中, 偶数长度被看作下一个更长的奇数长度。只有 SQL 记录编辑器显示偶数长度, 并且只有 SQL 记录支持将偶数长度用于测试和生成。偶数长度只在 SQL <i>where</i> 子句以及在使用执行时语句构建的 SQL 函数中使用。在记录编辑器中, 还可以显示字节数。</p>	<p>对应的数字类型:</p> <ul style="list-style-type: none"> <li>• pacf</li> <li>• decimal</li> </ul> <p>精度是总位数。小数位是小数点右边的位数。</p> <p>pacf 字段的最大精度是 18。对于调试和 Java 生成来说, 数字字段的最大精度是 32, 对于 COBOL 生成来说, 是 31。pacf 的长度必须是奇数或 18。</p> <p>decimal 的长度可以是奇数或偶数。dataItem 部件定义和所有记录类型都支持偶数长度。</p> <p>在进行测试和生成时, 如果使用 VisualAge Generator 兼容性方式, 则 EGL 对具有偶数精度的十进制项执行下列操作:</p> <ul style="list-style-type: none"> <li>• 在所有记录中, 将精度增加 1。</li> <li>• 在 SQL <i>where</i> 子句或 <i>prepare</i> 语句中, EGL 使用一个具有偶数精度的临时变量。</li> </ul>	<p>迁移工具转换为对应的类型、精度和小数位。如果小数位数为 0, 则迁移工具省略小数位。对于 Pack 项来说, 如果 dataItem 部件定义中记录的是偶数长度, 则迁移工具将其迁移为偶数长度。</p> <p>如果指定 VAGen 迁移首选项不要对项或变量使用 <i>evensql=y</i>, 迁移工具将自动使用奇数精度 (或者, 如果该项具有最大长度, 则使用 18) 并且对受影响的数据项部件或非共享记录项发出警告消息。</p>
<p>二进制项类型:</p> <ul style="list-style-type: none"> <li>• Bin, 长度为 4, 没有小数位</li> <li>• Bin, 长度为 9, 没有小数位</li> <li>• Bin, 长度为 18, 没有小数位</li> <li>• Bin, 长度为 4、9 或 18 并具有小数位</li> </ul>	<p>对应的二进制类型:</p> <ul style="list-style-type: none"> <li>• smallint (没有精度或小数位)</li> <li>• int (没有精度或小数位)</li> <li>• bigint (没有精度或小数位)</li> <li>• 具有精度和小数位的 bin</li> </ul>	<p>迁移工具根据长度和小数位将二进制数据项转换为对应的类型。仅当指定了小数位数 (小数位) 时, 才使用 bin 类型。</p>
<p>描述</p>	<p>无。</p>	<p>迁移工具将项描述转换为位于 dataItem 定义前面的注释。</p>

表 65. 缺省映射属性和用户界面属性 - 一般信息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>可以同时为数据项指定缺省映射属性和用户界面 (UI) 属性。这些属性包括:</p> <ul style="list-style-type: none"> <li>• 格式编辑</li> <li>• 验证编辑</li> <li>• 错误消息</li> </ul> <p>用户界面属性还包括标签和帮助文本。</p> <p>在 VisualAge Generator 中显式地设置一些属性将自动导致设置其它属性。例如, 如果设置数字分隔符, 则还将显式地设置填充字符、需要输入、对齐、货币符号和符号。</p>	<p>DataItem 部件可以具有下列属性:</p> <ul style="list-style-type: none"> <li>• 定义格式的属性</li> <li>• 验证属性</li> <li>• 页项属性</li> </ul> <p>某些属性的类别已更改为与 VisualAge Generator 不同。例如, 错误消息与验证属性分组在一起。页项属性包括用户界面标签和帮助文本。下列各表中的 EGL 列显示了 EGL 属性的类别。</p>	<p>迁移工具将缺省映射属性与用户界面属性合并到一起, 并使用用户界面属性具有优先权。验证编辑及其相关错误消息是成对迁移的。迁移工具仅迁移 VisualAge Generator 中显式设置的属性。此工具不会自动地为 EGL 属性插入缺省值。请参阅第 42 页的『共享编辑和消息』中有关合并映射和用户界面编辑的信息以了解详细信息和潜在问题。</p> <p>另请参阅第 43 页的『共享数据项的映射项编辑例程』中有关共享数据项的映射项编辑的信息以了解详细信息和潜在问题。</p>

表 66. 缺省映射属性和用户界面属性 - 一般编辑

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>编辑类型 (仅适用于用户界面) - 值:</p> <ul style="list-style-type: none"> <li>• 无</li> <li>• 布尔</li> <li>• 日期</li> <li>• 时间</li> </ul>	<p>EGL 支持多个属性:</p> <ul style="list-style-type: none"> <li>• 无</li> <li>• isBoolean = yes</li> <li>• dateFormat = defaultDateFormat</li> <li>• timeFormat = "HH:mm:ss"</li> </ul> <p>(定义格式的属性)</p>	<p>没有特殊的注意事项。</p>
<p>编辑函数 (仅适用于用户界面)</p>	<p>validatorFunction (验证属性)</p>	<p>没有特殊的注意事项。</p>
<p>编辑表 (仅适用于用户界面)</p>	<p>validatorDataTable (验证属性)</p>	<p>没有特殊的注意事项。</p>
<p>对 Web 运行编辑函数 (仅适用于用户界面)</p>	<p>不受支持。</p>	<p>迁移工具将其转换为注释。此注释包括 EGL runValidatorFromProgram 属性, 后者预计将成为最终的替换项。EGL 属性与 VAGen 属性相反。迁移工具在创建注释时将把 yes 转换为 no 并将 no 转换为 yes。</p>

表 66. 缺省映射属性和用户界面属性 - 一般编辑 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
编辑例程 (仅适用于映射)	validatorFunction 或 validatorDataTable (验证属性)	如果未指定用户界面编辑函数和编辑表, 则迁移工具执行下列操作: <ul style="list-style-type: none"> <li>• 如果映射编辑例程是 EZEC10 或 EZEC11, 则设置 validatorFunction 属性。</li> <li>• 如果编辑例程是函数, 则设置 validatorFunction 属性。</li> <li>• 如果编辑例程是表, 则设置 validatorDataTable 属性。</li> </ul> 如果指定了用户界面编辑函数或编辑表, 则迁移工具不迁移映射编辑例程。  如果编辑例程在迁移期间不可用, 则有一些特殊的注意事项。请参阅第 43 页的『共享数据项的映射项编辑例程』中有关映射编辑例程的信息以了解其它详细信息和潜在问题。
对齐 - 左   右   无 (仅适用于映射) 注: <ul style="list-style-type: none"> <li>• 对于映射项来说, 数字字段的缺省值是“右”, 所有其它字段的缺省值是“左”。</li> <li>• 对于用户界面项来说, 不支持对齐。</li> </ul>	align = left   right   none (定义格式的属性) 注: <ul style="list-style-type: none"> <li>• 对于表单字段来说, 数字字段的缺省值是 right, 所有其它字段的缺省值是 left。</li> <li>• 对于页项来说, 不支持 align。</li> </ul>	没有特殊的注意事项。
日期编辑掩码 (仅适用于映射)  有效值如下所示: <ul style="list-style-type: none"> <li>• SYSGREGRN</li> <li>• SYSJULIAN</li> <li>• dateEditPattern</li> </ul>	dateFormat = value  有效值如下所示: <ul style="list-style-type: none"> <li>• systemGregorianCalendarFormat</li> <li>• systemJulianDateFormat</li> <li>• "dateEditPattern"</li> </ul> (定义格式的属性) 注: 在 dateEditPattern 中, 迁移工具将内容转换为下列 EGL 表示法: <ul style="list-style-type: none"> <li>• yy 或 yyyy 表示年份。</li> <li>• MM 表示月份。</li> <li>• dd 表示一个月中的第几天。</li> <li>• DDD 表示一年中的第几天。</li> </ul>	如果用户界面编辑类型未指定日期, 则迁移工具根据 VisualAge Generator 中指定的日期编辑掩码 (如果有的话) 来设置 dateFormat。如果用户界面编辑类型指定了日期, 则迁移工具不迁移映射日期编辑掩码。
最小输入	minimumInput (验证属性)	没有特殊的注意事项。

表 66. 缺省映射属性和用户界面属性 - 一般编辑 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
填充字符 注: <ul style="list-style-type: none"> <li>对于字符、混合和数字字段来说, 用户界面记录中使用的项的缺省填充字符是空白。对于十六进制字段来说, 缺省填充字符是零。对于 DBCS 和 Unicode 字段来说, 空白是必需的填充字符。NULL 不是有效的填充字符。</li> <li>对于字符、DBCS 或混合字段来说, 映射中使用的项的缺省填充字符是 NULL。对于数字字段来说, 缺省填充字符是空白, 对于十六进制字段来说, 缺省填充字符是零。</li> </ul>	fillCharacter (定义格式的属性) 注: <ul style="list-style-type: none"> <li>除非在特定的页面或表单中被覆盖, 否则同一个缺省 fillCharacter 既用于页项也用于表单字段。</li> <li>strLib.nullFill 是 EGL 的 NULL 填充字符常量。此外, 也可以使用 "" (两个连续的双引号)。</li> </ul>	由于在 EGL 中只有一个缺省填充字符, 所以有一些特殊的注意事项。请参阅第 44 页的『共享数据项的填充字符』中有关不确定数据项和填充字符的信息以了解详细信息和潜在问题。
折换	upperCase (定义格式的属性)	没有特殊的注意事项。
十六进制编辑 (仅适用于映射)	isHexDigit (验证属性)	没有特殊的注意事项。
需要输入	inputRequired (验证属性)	没有特殊的注意事项。
检查 SO/SI 空格	needsSOSI (验证属性)	没有特殊的注意事项。

表 67. 缺省映射属性和用户界面属性 - 数字编辑

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
最小值和最大值 注: 如果指定了最小值或最大值, 则必须同时指定这两者。	validValues = <code>[[minimumValue, maximumValue]]</code> (验证属性) 注: 在 validValues 属性中, 可以列示多对值和单个的值。	迁移工具将最小值和最大值组合成 EGL validValues 属性。
符号 - 无   前   后 注: 缺省值为“无”。	sign = none   leading   trailing (定义格式的属性) 注: 缺省值为 none。	没有特殊的注意事项。
货币 (既适用于映射也适用于用户界面) 货币符号 (仅适用于用户界面)	currency = yes   no   currencySymbol = "symbol" (定义格式的属性) 注: <ul style="list-style-type: none"> <li>currencySymbol 也适用于表单。</li> <li>如果 currency = yes, 但是未指定 currencySymbol, 则在运行时使用的实际货币符号的设置方式与 VisualAge Generator 中的设置方式相同。</li> </ul>	迁移工具将迁移下列各项中的第一个适用项: <ul style="list-style-type: none"> <li>如果指定了用户界面货币符号, 则迁移工具迁移为 currency = yes 和 currencySymbol = "symbol"。</li> <li>如果用户界面货币编辑设置为 yes 或 no, 则工具将 currency 属性分别设置为 yes 或 no。</li> <li>如果映射货币编辑设置为 yes 或 no, 则工具将 currency 属性分别设置为 yes 或 no。</li> </ul>
分隔符	numericSeparator (定义格式的属性)	没有特殊的注意事项。
零编辑	zeroFormat (定义格式的属性)	没有特殊的注意事项。

表 68. 缺省映射属性和用户界面属性 - 错误消息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
编辑表 (仅适用于用户界面)	validatorDataTableMsgKey (验证属性)	没有特殊的注意事项。
EZE 函数 (仅适用于用户界面)	validatorFunctionMsgKey (验证属性)	没有特殊的注意事项。
编辑例程 (仅适用于映射)	validatorDataTableMsgKey 或 validatorFunctionMsgKey (验证属性)	<p>仅当未指定用户界面编辑表、用户界面编辑函数、用户界面编辑表消息和用户界面 EZE 函数消息时，迁移工具才迁移映射编辑例程消息。如果迁移工具迁移映射编辑例程消息，则该工具执行下列操作：</p> <ul style="list-style-type: none"> <li>• 如果编辑例程是 EZEC10 或 EZEC11，则设置 validatorFunctionMsgKey。</li> <li>• 如果编辑例程是表，则设置 validatorDataTableMsgKey。</li> <li>• 如果编辑例程是函数，则不迁移编辑例程消息，这是因为，在 VisualAge Generator 中，在这种情况下不使用该消息。</li> </ul> <p>有一些特殊的注意事项。请参阅第 43 页的『共享数据项的映射项编辑例程』中有关不确定数据项和映射编辑例程的信息以了解其它详细信息和潜在问题。</p>
最小输入	minimumInputMsgKey (验证属性)	没有特殊的注意事项。
需要输入	inputRequiredMsgKey (验证属性)	没有特殊的注意事项。
数据类型	typeChkMsgKey (验证属性)	没有特殊的注意事项。
数字范围	validValuesMsgKey (验证属性)	没有特殊的注意事项。

表 69. 用户界面属性 - 标签和帮助

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
用户界面标签	displayName (页项属性)	没有特殊的注意事项。
帮助文本	help (页项属性)	没有特殊的注意事项。

## 记录

『记录』一节由下列各表组成：

- 记录 - 一般语法、记录类型、属性和 prolog (第 187 页的表 70)
- 记录 - 大部分记录类型的记录结构 (第 188 页的表 71)
- 记录 - SQL 属性和 SQL 记录结构 (第 190 页的表 72)

**注：**迁移工具始终将 VAGen 记录转换为 EGL 固定记录部件。

表 70. 记录 - 一般语法、记录类型、属性和 prolog

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 记录部件:</p> <ul style="list-style-type: none"> <li>recordName</li> <li>基本信息                             <ul style="list-style-type: none"> <li>记录类型</li> <li>记录结构 (项列表)</li> </ul> </li> <li>属性 (根据记录类型的不同而有所变化)</li> <li>Prolog</li> </ul> <p>可以通过指定备用指定记录或者通过包括项列表来给定记录结构。</p>	<p>EGL 记录示例:</p> <pre> //*** Record=recordName*** // prolog //***** Record recordName type recordType { [ recordProperties ] } recordStructure end // end recordName                     </pre> <p>注: 可以通过指定嵌入的语句或者通过包括项列表来给定记录结构。</p>	<p>没有特殊的注意事项。</p>
<p>记录类型:</p> <ul style="list-style-type: none"> <li>工作存储器记录</li> <li>重新定义的记录</li> <li>串行记录</li> <li>带索引记录</li> <li>相对记录</li> <li>消息队列记录</li> <li>SQL 行记录</li> <li>用户界面记录</li> <li>DL/I 段记录</li> </ul>	<p>EGL 记录类型:</p> <ul style="list-style-type: none"> <li>basicRecord</li> <li>basicRecord</li> <li>serialRecord</li> <li>indexedRecord</li> <li>relativeRecord</li> <li>mqRecord</li> <li>sqlRecord</li> <li>在此发行版中不受支持。</li> <li>在此发行版中不受支持。</li> </ul>	<p>迁移工具将重新定义的记录迁移为 basicRecord。迁移工具将包括带有记录定义的注释以提供重新定义的记录的名称。对于重新定义的记录, 有一些特殊的注意事项。请参阅第 45 页的『重新定义的记录』中的信息以了解详细信息和潜在问题。</p>
<p>工作存储器记录属性:</p> <ul style="list-style-type: none"> <li>备用指定</li> </ul>	<p>basicRecord 属性:</p> <ul style="list-style-type: none"> <li>嵌入的语句</li> </ul>	<p>迁移工具将备用指定迁移为嵌入的语句。</p>
<p>重新定义的记录属性:</p> <ul style="list-style-type: none"> <li>重新定义</li> </ul> <p>注: “重新定义”属性指定用于提供物理存储器的另一个记录的名称。当前记录提供同一物理存储器的另一种数据项布局。</p>	<p>basicRecord 属性:</p> <ul style="list-style-type: none"> <li>无。仅在使用该记录的程序中才指定重新定义信息。同一个记录既可以用作另一个记录的重新定义也可以用作正常的记录。</li> </ul>	<p>迁移工具将包括带有记录定义的注释以提供重新定义的记录的名称。</p> <p>在使用该记录的程序中, 迁移工具还在该记录的声明语句中包括 redefines 属性。</p> <p>根据该记录在程序中的使用方式的不同以及根据该记录在迁移期间是否可用, 有一些特殊的注意事项。请参阅第 45 页的『重新定义的记录』中的信息以了解详细信息和潜在问题。</p>
<p>串行记录属性:</p> <ul style="list-style-type: none"> <li>文件名</li> <li>备用指定</li> <li>可变长度项</li> <li>Occurrences 项</li> </ul>	<p>serialRecord 属性:</p> <ul style="list-style-type: none"> <li>fileName</li> <li>嵌入的语句</li> <li>lengthItem</li> <li>numElementsItem</li> </ul>	<p>迁移工具将备用指定迁移为嵌入的语句。</p>

表 70. 记录 - 一般语法、记录类型、属性和 prolog (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
带索引记录属性: <ul style="list-style-type: none"> <li>• 文件名</li> <li>• 记录标识</li> <li>• 备用指定</li> <li>• 可变长度项</li> <li>• Occurrences 项</li> </ul>	indexedRecord 属性: <ul style="list-style-type: none"> <li>• fileName</li> <li>• keyItem</li> <li>• 嵌入的语句</li> <li>• lengthItem</li> <li>• numElementsItem</li> </ul>	迁移工具将备用指定迁移为嵌入的语句。
相对记录属性: <ul style="list-style-type: none"> <li>• 文件名</li> <li>• 记录标识</li> <li>• 备用指定</li> </ul>	relativeRecord 属性: <ul style="list-style-type: none"> <li>• fileName</li> <li>• keyItem</li> <li>• 嵌入的语句</li> </ul>	迁移工具将备用指定迁移为嵌入的语句。
消息队列记录属性: <ul style="list-style-type: none"> <li>• 文件名</li> <li>• 备用指定</li> <li>• 将消息包括在事务中</li> <li>• 打开队列以便在进行输入时独占使用</li> <li>• 记录长度项</li> <li>• Occurrences 项</li> <li>• 队列描述符记录</li> <li>• 打开选项记录</li> <li>• 消息描述符记录</li> <li>• 获取选项记录</li> <li>• 放置选项记录</li> </ul>	mqRecord 属性: <ul style="list-style-type: none"> <li>• queueName</li> <li>• 嵌入的语句</li> <li>• includeMsgInTransaction</li> <li>• openQueueExclusive</li> <li>• lengthItem</li> <li>• numElementsItem</li> <li>• queueDescriptorRecord</li> <li>• openOptionsRecord</li> <li>• msgDescriptorRecord</li> <li>• getOptionsRecord</li> <li>• putOptionsRecord</li> </ul>	迁移工具将备用指定迁移为嵌入的语句。
SQL 行记录属性: <ul style="list-style-type: none"> <li>• 请参阅第 190 页的表 72。</li> </ul>	SQL 行记录属性: <ul style="list-style-type: none"> <li>• 请参阅第 190 页的表 72。</li> </ul>	没有特殊的注意事项。
Prolog	无。	迁移工具将 prolog 转换为位于记录定义前面的注释。

表 71. 记录 - 大部分记录类型的记录结构

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
记录结构 - 变体 1: 备用指定。如果 RecordA 指定 RecordB 的备用指定, 则 RecordB 为 RecordA 提供所有的项。在 RecordA 中没有项结构。  如果 RecordB 包含 Level-77 项, 则 RecordA 将仅包含 RecordB 中的非 Level-77 项。	记录结构 - 变体 1: EGL 嵌入的语句指定了为当前记录提供字段结构的记录。RecordA 嵌入了 RecordB。例如: embed RecordB;	迁移工具将备用指定迁移为嵌入的语句。  对于工作存储器记录中的 Level-77 项来说, 有一些特殊的注意事项。请参阅第 46 页的『记录中的 Level-77 项』中的信息以了解详细信息和潜在问题。

表 71. 记录 - 大部分记录类型的记录结构 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>记录结构 - 带有共享项的变体 2:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• Occurs</li> <li>• 共享</li> <li>• levelNumber</li> </ul> <p>注: levelNumber 是隐藏的, 但是它基于记录中的数据项层次结构。</p> <p>注: 类型、长度、小数位数和描述在记录编辑器中是可视的, 但是并不存储在记录中。</p>	<p>记录结构 - 带有 EGL 类型定义的变体 2 的示例:</p> <pre>levelNumber itemName       itemName [occurs];</pre> <p>注: 类型、长度、小数位数和描述在编辑器中是不可视的。</p>	<p>如果您选择了迁移语法首选项将共享数据项转换为基本项定义, 并且数据项部件可用, 则迁移工具将把共享项转换为使用基本定义进行定义的 EGL 变量, 该基本定义基于对该数据项部件指定的类型、长度和小数位数。类型、长度和小数位数信息的迁移方式与第 181 页的表 64 中描述的迁移方式相同。</p> <p>如果您未选择迁移语法首选项将共享数据项转换为基本项定义, 或者数据项部件不可用, 则迁移工具将把共享项转换为使用类型定义进行定义的 EGL 变量。对于迁移来说, 类型定义总是与项名相同。</p> <p>如果 occurs 为 1, 迁移工具将省略 occurs 信息。</p> <p>对于工作存储器记录中的 Level-77 项来说, 有一些特殊的注意事项。请参阅第 46 页的『记录中的 Level-77 项』中的信息以了解详细信息和潜在问题。</p>
<p>记录结构 - 带有非共享项的变体 2:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• Occurs</li> <li>• 类型</li> <li>• 长度</li> <li>• 小数位数</li> <li>• 非共享</li> <li>• 描述</li> <li>• levelNumber 是隐藏的, 但是它基于记录中的数据项层次结构。</li> </ul> <p>注: 类型、长度、小数位数和描述与项一起存储在记录中。</p>	<p>记录结构 - 带有 EGL 基本类型的变体 2 的示例:</p> <pre>levelNumber itemName       dataType(lengthInformation)       [occurs];       // Description</pre> <p>注: 类型、长度、小数位数和描述在编辑器中是可视的。</p>	<p>迁移工具将非共享项迁移为使用基本类型定义的 EGL 变量。类型、长度和小数位数信息的迁移与第 181 页的表 64 “数据项 - 一般语法、数据类型、长度、小数位数和描述”中描述的情况相同。</p> <p>如果 occurs 为 1, 迁移工具将省略 occurs 信息。</p> <p>对于工作存储器记录中的 Level-77 项来说, 有一些特殊的注意事项。请参阅第 46 页的『记录中的 Level-77 项』中的信息以了解详细信息和潜在问题。</p>

表 72. 记录 - SQL 属性和 SQL 记录结构

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>SQL 记录属性:</p> <ul style="list-style-type: none"> <li>• 缺省键项</li> <li>• 备用指定</li> <li>• SQL 表: <ul style="list-style-type: none"> <li>- 标签</li> <li>- 名称</li> </ul> </li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• 如果记录未指定备用指定, 则键项就是记录结构中指定了 key=yes 的项。缺省键项将被忽略。</li> <li>• 如果记录指定了备用指定, 则键项就是当前记录中的缺省键项与备用指定记录中指定了 key=yes 的项的合并。键是按照各个项在记录结构中的出现顺序合并的。如果在备用指定记录中还将当前记录中的缺省键项指定为 key=yes, 则在合并的键列表中仅包括该项一次。</li> <li>• SQL 表名既可以是实际表名 ( 正常情况 ) 也可以是在运行时将被替换的表名主变量。表名主变量以分号 (;) 开头。</li> </ul>	<p>sqlRecord 属性:</p> <ul style="list-style-type: none"> <li>• keyItems</li> <li>• 嵌入的语句</li> <li>• tableNames 和 / 或 tableNameVariables</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• keyItems 是记录的所有键的列表。未对记录结构中的项指定 key=yes。</li> <li>• 当表名不是主变量时, tableNames 属性是表名和表标签的列表。当表名是在运行时将被替换的主变量时, tableNameVariables 属性是表名和表标签的列表。tableNameVariables 属性中的表名不以分号开头。可以在同一个记录定义中同时使用 tableNames 和 tableNameVariables。</li> </ul>	<p>迁移工具按如下方式构建 keyItems 属性:</p> <ul style="list-style-type: none"> <li>• 如果未包括 VAGen 备用指定, 则迁移工具将使用记录结构中任何指定了 key=yes 的项, 并且不包括 VAGen 缺省键项。</li> <li>• 如果包括了 VAGen 备用指定, 则迁移工具将把备用指定记录中任何指定了 key=yes 的项与当前记录中的缺省键项合并。键是按照各个项在记录结构中的出现顺序列示的。如果当前记录中的缺省键项与备用指定记录中的其中一个键项相同, 则在 keyItems 列表中仅包括该项一次。</li> </ul> <p>迁移工具按照以下方式构建 tableNames 和 tableNameVariables 列表:</p> <ul style="list-style-type: none"> <li>• 当表名不是主变量时, 根据表名和表标签构建 tableNames。</li> <li>• 当表名是主变量时, 根据表名和表标签构建 tableNameVariables。</li> </ul> <p>有一些特殊的注意事项。请参阅第 47 页的『备用指定记录』中有关 SQL 备用指定的信息以了解详细信息和潜在问题。</p>
<p>SQL 缺省条件:</p> <ul style="list-style-type: none"> <li>• whereClauseText</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• SQL 缺省条件使您能够指定一个 where 子句, 最常见的情况是, 当在 SQL 行记录中使用了多个表时, 对连接条件指定该子句。语法是 SQL 语法。</li> <li>• 允许使用 !itemColumnName 变量。这些变量指定 SQL 行记录中的项的名称。在测试或生成时, VisualAge Generator 会将其替换为对应的 SQL 列名。</li> </ul>	<p>缺省选择条件示例:</p> <pre>defaultSelectCondition =   #sqlCondition{     whereClauseText   }</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• defaultSelectCondition 的用途与在 VisualAge Generator 中的用途相同。</li> <li>• 不支持 !itemColumnName 变量。必须使用实际的 SQL 列名。</li> </ul>	<p>迁移工具将任何 !itemColumnName 变量转换为它们对应 SQL 列名。</p> <p>有一些特殊的注意事项。请参阅第 47 页的『备用指定记录』中有关 SQL 备用指定的信息以了解详细信息和潜在问题。</p>
<p>记录结构 - 变体 1: 备用指定。如果 RecordA 指定 RecordB 的备用指定, 则 RecordB 为 RecordA 提供所有的项。在 RecordA 中没有项结构。</p>	<p>记录结构 - 变体 1: EGL 嵌入的语句指定了为当前记录提供字段结构的记录。RecordA 嵌入了 RecordB。例如:</p> <pre>embed RecordB;</pre>	<p>迁移工具将备用指定迁移为嵌入的语句。</p>

表 72. 记录 - SQL 属性和 SQL 记录结构 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>记录结构 - 带有共享项的变体 2:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• 只读</li> <li>• 键</li> <li>• SQL 列名</li> <li>• SQL 代码</li> <li>• 共享</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• 类型、长度、小数位数和描述在记录编辑器中是可视的, 但是并不存储在记录中。</li> <li>• 在 SQL 记录中, 从来不会使用级别号。</li> <li>• 对于 pack 和 binary 字段来说, 不将 SQL 代码包括在外部源格式文件中。如果未将 char、dbchar 或 unicode 字段的 SQL 代码包括在外部源格式文件中, 则将把该字段看作固定长度的字段。仅当记录是从旧版本的 VisualAge Generator 迁移而来并且您从未使用 VisualAge Generator 4.5 对其进行修改时, 才会发生这种情况。</li> </ul>	<p>记录结构 - 带有 EGL 类型定义的变体 2 的示例:</p> <pre> levelNumber itemName itemName { [sqlDataCode=sqlCodeNumber]   [column="SQLColumnName"]   [ isReadOnly=yes ]   [ isNullable = yes ]   [ sqlVariableLen = yes ] }; </pre> <p>注:</p> <ul style="list-style-type: none"> <li>• 类型、长度、小数位数和描述在编辑器中是不可视的。</li> <li>• 级别号在 SQL 记录中是可选的。如果包括了级别号, 则 SQL 记录就成为固定记录。如果未包括级别号, 则 SQL 记录就不是固定记录。非固定记录允许使用新的 EGL 数据类型 (BLOB、CLOB 和字符串)。但是, 非固定记录的其它行为与 VAGen 行为不兼容。</li> </ul>	<p>如果您选择了迁移语法首选项将共享数据项转换为基本项定义, 并且数据项部件可用, 则迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>• 将共享项转换为 EGL 变量, 该变量是使用基于对该数据项部件指定的类型、长度和小数位数的基本定义来定义的。</li> <li>• 对于 hex 项来说, 包括 sqlDataCode 属性。</li> <li>• 对于 char、dbchar 或 unicode 字段来说, 如果 VAGen SQL 数据代码指示该项具有可变长度, 则设置 sqlVariableLen=yes 属性。如果 VAGen SQL 数据代码指示该项具有固定长度, 则迁移工具将省略 sqlVariableLen 属性。</li> </ul> <p>如果您未选择迁移语法首选项将共享数据项转换为基本项定义, 或者数据项部件不可用, 则迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>• 将共享项转换为使用类型定义来定义的 EGL 变量。对于迁移来说, 类型定义总是与项名相同。</li> <li>• 如果外部源格式文件包含 sqlDataCode 属性, 并且该属性不是 VAGen 二进制字段或压缩字段的其中一个值, 则包括该属性。</li> <li>• 如果 VAGen SQL 数据代码指示该项具有可变长度, 则设置 sqlVariableLen=yes 属性。如果 VAGen SQL 数据代码指示该项具有固定长度, 则迁移工具将省略 sqlVariableLen 属性。</li> </ul> <p>迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>• 在 sqlRecord 的 EGL keyItems 属性中包括任何 key=yes 项。</li> <li>• 始终对 sqlRecord 中的项添加级别号, 以使该记录成为固定记录。这将保留 VAGen 的行为。</li> </ul>

表 72. 记录 - SQL 属性和 SQL 记录结构 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>记录结构 - 带有非共享项的变体 2:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• 类型</li> <li>• 长度</li> <li>• 小数位数</li> <li>• 只读</li> <li>• 键</li> <li>• SQL 列名</li> <li>• SQL 代码</li> <li>• 非共享</li> <li>• 描述</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• 类型、长度、小数位数和描述与项一起存储在记录中。</li> <li>• 在 SQL 记录中, 从来不会使用级别号。</li> <li>• 对于 pack 和 binary 字段来说, 不将 SQL 代码包括在外部源格式文件中。如果未将 char、dbchar 或 unicode 字段的 SQL 代码包括在外部源格式文件中, 则将把该字段看作固定长度的字段。仅当记录是从旧版本的 VisualAge Generator 迁移而来并且您从未使用 VisualAge Generator 4.5 对其进行修改时, 才会发生这种情况。</li> </ul>	<p>记录结构 - 带有 EGL 基本类型的变体 2 的示例:</p> <pre>levelNumber itemName dataType(lengthInformation) // Description { [sqlDataCode=sqlCodeNumber]   [ column="SQLColumnName" ]   [ isReadOnly=yes ]   [ isNullable = yes ]   [ sqlVariableLen = yes ] };</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• 类型、长度、小数位数和描述在编辑器中是可视的。</li> <li>• 级别号在 SQL 记录中是可选的。如果包括了级别号, 则 SQL 记录就成为固定记录。如果未包括级别号, 则 SQL 记录就不是固定记录。非固定记录允许使用新的 EGL 数据类型 (BLOB、CLOB 和字符串)。但是, 非固定记录的其它行为与 VAGen 行为不兼容。</li> </ul>	<p>迁移工具将非共享项迁移为使用基本类型定义的 EGL 变量。类型、长度和小数位数信息的迁移与第 181 页的表 64 中描述的情况相似。</p> <p>仅对于 hex 项, 迁移工具才包括 sqlDataCode 属性。</p> <p>对于 char、dbchar 或 unicode 数据项来说, 如果 VAGen SQL 数据代码指示该项具有可变长度, 则迁移工具设置 sqlVariableLen=yes。如果 VAGen SQL 数据代码指示该项具有固定长度, 则迁移工具将省略 sqlVariableLen 属性。</p> <p>迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>• 在 sqlRecord 的 EGL keyItems 属性中包括任何 key=yes 项。</li> <li>• 始终对 sqlRecord 中的项添加级别号, 以使该记录成为固定记录。这将保留 VAGen 的行为。</li> </ul>
<p>VAGen 数据类型 - Char</p> <ul style="list-style-type: none"> <li>• 数据代码 - 453</li> <li>• 数据代码 - 449 或 457</li> </ul>	<p>EGL 数据类型:</p> <ul style="list-style-type: none"> <li>• char; 省略 sqlVariableLen</li> <li>• varchar, sqlVariableLen = yes</li> </ul>	<p>没有特殊的注意事项。</p>
<p>VAGen 数据类型 - DBCS</p> <ul style="list-style-type: none"> <li>• 数据代码 - 469</li> <li>• 数据代码 - 465 或 473</li> </ul>	<p>EGL 数据类型:</p> <ul style="list-style-type: none"> <li>• dbchar; 省略 sqlVariableLen</li> <li>• vardbchar, sqlVariableLen = yes</li> </ul>	<p>没有特殊的注意事项。</p>
<p>VAGen 数据类型 - Unicode</p> <ul style="list-style-type: none"> <li>• 数据代码 - 469</li> <li>• 数据代码 - 465 或 473</li> </ul>	<p>EGL 数据类型:</p> <ul style="list-style-type: none"> <li>• unicode; 省略 sqlVariableLen</li> <li>• varunicode, sqlVariableLen = yes</li> </ul>	<p>没有特殊的注意事项。</p>

表 72. 记录 - SQL 属性和 SQL 记录结构 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
SQL 列名 注: SQL 列名是必需的。	column = "SQLColumnName" 注: column 属性是可选的。如果 column 属性被省略, 则它将缺省为字段名。	迁移工具根据省略列名首选项进行迁移。 • 如果您选择了此首选项, 则迁移工具执行下列操作: – 如果 SQL 列名与字段名相同, 则省略 column 属性。 – 如果 SQL 列名与字段名不同, 则包括 column 属性。 • 如果未选择此首选项, 则迁移工具始终包括 column 属性。
没有对应的属性。 注: 对于 SQL 项来说, VisualAge Generator 始终包括空指示符变量。	isNullable = yes   no 注: isNullable 的缺省值为 no。	迁移工具根据省略 isNullable 属性首选项进行迁移。 • 如果您选择了此首选项, 则迁移工具不包括 isNullable 属性, 该属性将缺省为 no。 • 如果您未选择此首选项, 则迁移工具始终包括 isNullable=yes。这将保留 VAGen 的行为。
只读 注: 始终显式地设置“只读”。如果对 SQL 记录指定了多个表, 则“只读”必须始终为 yes。	isReadOnly = yes   no 注: 如果对 SQL 记录指定了多个表, 则 ReadOnly 缺省为 yes。如果只对 SQL 记录指定了一个表, 则 ReadOnly 缺省为 no。	迁移工具根据省略 isReadOnly 属性首选项进行迁移。 • 如果您选择了此首选项, 则仅当对记录指定了一个表并且 VAGen 只读属性设置为 yes 时, 迁移工具才包括 isReadOnly 属性。 • 如果您未选择此首选项, 则每当 VAGen 只读属性设置为 yes 时, 迁移工具都包括 isReadOnly 属性。

## 表

『VAGen 表』一节由下列各表组成:

- VAGen 表 - 一般语法、表类型、属性和 prolog (第 194 页的表 73)
- VAGen 表 - 表结构 (第 194 页的表 74)
- VAGen 表 - 表内容 (第 195 页的表 75)

表 73. 表 - 一般语法、表类型、属性和 prolog

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 表部件:</p> <ul style="list-style-type: none"> <li>• tableName</li> <li>• 基本信息                             <ul style="list-style-type: none"> <li>- 表类型</li> <li>- 表结构 (项列表)</li> </ul> </li> <li>• 属性</li> <li>• Prolog</li> <li>• 表内容</li> </ul>	<p>EGL 语法示例:</p> <pre> //*** DataTable=tableName*** // prolog //***** DataTable tableName   type tableType   { [otherTableProperties]     [alias =       "originalTableName" ] }   tableStructure   [{ contents =     [{rowContents}]] end // end tableName                     </pre>	<p>即使表与 EGL 保留字列表冲突, 迁移工具也不会将表重命名。迁移工具不设置 alias 属性。如果您必须将该表重命名, 则可以使用 alias 属性来指定 VAGen 表的原始名称。要了解详细信息, 请参阅第 48 页的『保留字和表名』中有关表名的信息。</p>
<p>表类型:</p> <ul style="list-style-type: none"> <li>• 未指定</li> <li>• 匹配无效</li> <li>• 匹配有效</li> <li>• 范围匹配有效</li> <li>• 消息</li> </ul>	<p>DataTable 类型:</p> <ul style="list-style-type: none"> <li>• basicTable</li> <li>• matchInvalidTable</li> <li>• matchValidTable</li> <li>• rangeChkTable</li> <li>• msgTable</li> </ul>	<p>没有特殊的注意事项。</p>
<p>属性 - 运行时属性:</p> <ul style="list-style-type: none"> <li>• 驻留</li> <li>• 共享</li> </ul>	<p>DataTable 属性:</p> <ul style="list-style-type: none"> <li>• resident</li> <li>• shared</li> </ul>	<p>没有特殊的注意事项。</p>
<p>属性 - 折换表内容</p>	<p>无。如果要折换表内容, 则必须输入大写的內容。</p>	<p>如果 VAGen 表指定应该折换表内容, 则迁移工具将确保将表内容中的字符数据、十六进制数据和混合数据转换为大写。</p>
<p>Prolog</p>	<p>无。</p>	<p>迁移工具将 prolog 转换为位于 DataTable 定义前面的注释。</p>

表 74. 表 - 表结构

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 表结构 - 带有共享项:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• 共享</li> <li>• levelNumber</li> </ul> <p>注: levelNumber 是隐藏的, 但是它基于记录中的数据项层次结构。</p> <p>注: 类型、长度、小数位数和描述在表编辑器中是可视的, 但是并不存储在表中。</p>	<p>DataTable 结构 - 带有 EGL 类型定义:</p> <pre> levelNumber itemName   itemName ;                     </pre> <p>注: 类型、长度、小数位数和描述在编辑器中是不可视的。</p>	<p>如果您选择了迁移语法首选项将共享数据项转换为基本项定义, 并且数据项部件可用, 则迁移工具将把共享项转换为使用基本定义进行定义的 EGL 变量, 该基本定义基于对该数据项部件指定的类型、长度和小数位数。类型、长度和小数位数信息的迁移方式与第 181 页的表 64 中描述的迁移方式相同。</p> <p>如果您未选择迁移语法首选项将共享数据项转换为基本项定义, 或者数据项部件不可用, 则迁移工具将把共享项转换为使用类型定义进行定义的 EGL 变量。对于迁移来说, 类型定义总是与项名相同。</p>

表 74. 表 - 表结构 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 表结构 - 带有非共享项:</p> <ul style="list-style-type: none"> <li>• itemName</li> <li>• 类型</li> <li>• 长度</li> <li>• 小数位数</li> <li>• 非共享</li> <li>• 描述</li> <li>• levelNumber</li> </ul> <p>注: levelNumber 是隐藏的, 但是它基于表中的数据项层次结构。</p> <p>注: 类型、长度、小数位数和描述与项一起存储在表中。</p>	<p>DataTable 结构 - 带有 EGL 基本类型:</p> <pre>levelNumber itemName     dataType(lengthInformation) ;     // Description</pre> <p>注: 类型、长度、小数位数和描述在编辑器中是可视的。</p>	<p>迁移工具将非共享项迁移为使用基本类型定义的 EGL 变量。类型、长度和小数位数的迁移与第 181 页的表 64 中描述的情况相同。</p>

表 75. 表 - 表内容

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>表内容:</p> <ul style="list-style-type: none"> <li>• 表内容是在定义格式的编辑器中输入的。表内容是为表结构中的顶级(父代)项输入的。</li> <li>• 不将字符和十六进制数据引在引号中。</li> </ul>	<p>DataTable 内容:</p> <ul style="list-style-type: none"> <li>• 每行的内容都用方括号括起来。有一对外部方括号, 它将整个行集括起来。</li> <li>• 必须用逗号将行内容中的各个值分隔开。</li> <li>• 包含十六进制数据的字符数据必须用双引号引起来。</li> </ul> <p>示例:</p> <pre>contents = [ [ rowContents ]              { , [ rowContents ] }            ]</pre> <p>其中, rowContents = value { , value }</p>	<p>如果 VAGen 表指定应该折换表内容, 则迁移工具将确保将表内容中的字符数据、十六进制数据和混合数据转换为大写。</p> <p>迁移工具还用双引号将字符数据(包括十六进制数据)引起来。</p>

## 映射组

『映射组』一节组织成下列各表:

- 映射组 - 一般信息(第 196 页的表 76)
- 映射组 - 一般语法和浮动区(第 196 页的表 77)
- 映射组 - 设备名、类型和大小(第 197 页的表 78)

表 76. 映射组 - 一般信息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>仅当存在浮动区时，映射组部件才是必需的。</p> <p>如果不存在映射组部件，VisualAge Generator 将自动生成具有相同映射组名的所有映射，就好象的确存在映射组部件一样。</p>	<p>formGroup 是必需的。</p>	<p>如果在迁移集中不存在 formGroup 部件，则迁移工具将创建一个 formGroup 部件。</p>
<p>映射名由映射组名和映射名组成。</p>	<p>表单名不包含 formGroup 名。</p> <p>可以在 formGroup 中定义（嵌套）表单。</p> <p>并且，表单也可以位于 formGroup 外部。在这种情况下，formGroup 必须包括一个 use 语句来指定表单名，并包括一个 import 语句来导入该表单所在的包。通过使用这种技术，您可以定义一个公共表单（例如弹出列表表单）并在许多不同的 formGroup 中使用该表单。</p>	<p>迁移工具将所有映射都迁移为表单。此工具不会尝试标识跨多个映射组的完全相同的公共映射定义。</p> <p>如果以单文件方式进行迁移，则迁移工具对 formGroup 中的每个表单都包括一个 use 语句。您应该移动表单以使它们嵌套在它们的 formGroup 内。</p> <p>如果使用第 1 - 3 阶段迁移来进行迁移，则迁移工具将自动地把所有表单都嵌套在 formGroup 部件中。</p>
<p>当程序指定了映射组时，该程序可以仅仅通过引用映射名来使用该映射组中的任何映射。</p>	<p>当程序包括了 use 语句以指示所使用的 formGroup 时，该程序可以仅仅通过引用表单名来引用该 formGroup 中的任何映射。</p>	<p>没有特殊的注意事项。</p>

表 77. 映射组 - 一般语法和浮动区

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>映射组部件可以包含：</p> <ul style="list-style-type: none"> <li>• 映射组名</li> <li>• 浮动区信息 <ul style="list-style-type: none"> <li>- 设备名</li> <li>- 设备大小</li> <li>- 大小 <ul style="list-style-type: none"> <li>- 行数</li> <li>- 列</li> </ul> </li> <li>- 位置 <ul style="list-style-type: none"> <li>- 起始行</li> <li>- 起始列</li> </ul> </li> </ul> </li> </ul>	<p>formGroup 可以包含：</p> <ul style="list-style-type: none"> <li>• formGroup 名</li> <li>• formGroup 属性</li> <li>• 屏幕浮动区信息</li> <li>• 打印浮动区信息</li> <li>• 包括在 formGroup 中的表单的 use 语句。</li> </ul> <p>以下是一个 formGroup 格式示例：</p> <pre> <b>FormGroup</b> groupName {   [ <b>alias</b>="generationName"   [ <b>screenFloatingArea</b>     {screenFloatingAreaInformation}]   [<b>printFloatingArea</b>     {printFloatingAreaInformation}]   <b>Form</b> formName <b>type</b> textForm     {formProperties}     [variableFields]     [constantFields]   <b>end</b> // end formName   <b>use</b> formName2; <b>end</b> // end groupName </pre>	<p>迁移工具使用 VAGen 设备类型来确定浮动区信息是用于显示器映射（screenFloatingArea）还是用于打印机映射（printFloatingArea）。</p> <p>要了解有关设置 deviceType 的信息，请参阅第 197 页的表 78。</p>

表 77. 映射组 - 一般语法和浮动区 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
无。	alias	即使映射组与 EGL 保留字列表有冲突，迁移工具也不会将该映射组重命名。有一些特殊的注意事项。请参阅第 49 页的『保留字和 formGroup 名』以了解详细信息和潜在问题。
<p>浮动区信息包含下列内容:</p> <ul style="list-style-type: none"> <li>• 设备名</li> <li>• 设备大小 (行数 x 列数)</li> <li>• 浮动区指定                             <ul style="list-style-type: none"> <li>- 大小                                     <ul style="list-style-type: none"> <li>- 行数</li> <li>- 列</li> </ul> </li> <li>- 位置                                     <ul style="list-style-type: none"> <li>- 起始行</li> <li>- 起始列</li> </ul> </li> </ul> </li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• 在 VisualAge Generator 中，您定义浮动区的大小和起始位置。</li> <li>• 允许 (但不建议) 具有相同大小的设备使用不同的浮动区指定。</li> </ul>	<p>浮动区信息包含下列内容:</p> <ul style="list-style-type: none"> <li>• 设备大小</li> <li>• 边界信息</li> </ul> <p>打印浮动区信息还包括设备类型。</p> <p>以下是用于文本表单的屏幕浮动区的示例:</p> <pre>screenFloatingArea {   screenSize=[lines,columns]   topMargin=nn,   bottomMargin=nn,   leftMargin=nn,   rightMargin=nn }</pre> <p>以下是用于打印表单的打印浮动区的示例:</p> <pre>printFloatingArea {   deviceType=singleByte,   pageSize=[lines,columns]   topMargin=nn,   bottomMargin=nn,   leftMargin=nn,   rightMargin=nn }</pre> <p>注: 对于 screenSize 或 pageSize 来说，仅允许一个浮动区指定。</p>	<p>迁移工具使用 VAGen 设备类型来确定浮动区指定是用于显示器映射 (screenFloatingArea) 还是用于打印映射 (printFloatingArea)。</p> <p>迁移工具按如下方式计算边界信息:</p> <ul style="list-style-type: none"> <li>• 将 topMargin 设置为 VAGen floatingAreaStartingLine - 1。</li> <li>• 将 bottomMargin 设置为 VAGen deviceRows - (floatingAreaStartingLine + floatingAreaLines) + 1。</li> <li>• 将 leftMargin 设置为 VAGen floatingAreaStartingColumn - 1。</li> <li>• 将 rightMargin 设置为 VAGen deviceColumns - (floatingAreaStartingColumn + floatingAreaColumns) + 1。</li> </ul> <p>要了解有关设置 deviceType 的信息，请参阅第 197 页的表 78。</p>
<p>打印机类型可以是下列其中一项:</p> <ul style="list-style-type: none"> <li>• 打印机</li> <li>• DBCS 打印机</li> </ul>	<p>deviceType=singleByte   doubleByte</p> <p>注: 只能对打印表单指定 deviceType 属性。</p>	<p>迁移工具根据 VAGen 打印机类型来设置 EGL deviceType 属性。</p> <p>迁移工具始终使用 DeviceTypeKind 来限定 deviceType 值 (例如 deviceType = DeviceTypeKind.doubleByte)。这将使得名称不会与 formGroup 中的表单上的任何变量字段发生冲突。</p>

表 78. 映射组 - 设备名、类型和大小

VisualAge Generator 设备名	设备大小 (行数 x 列数)	设备类型	迁移工具注意事项
3643-2	6 x 40	显示器	COBOL 生成功能不支持此设备大小。
3277-1	12 x 40	显示器	COBOL 生成功能不支持此设备大小。
3643-4	16 x 64	显示器	COBOL 生成功能不支持此设备大小。
3278-1, 3278-1B, ANY-1D	12 x 80	显示器	没有特殊的注意事项。

表 78. 映射组 - 设备名、类型和大小 (续)

VisualAge Generator 设备名	设备大小 (行数 x 列数)	设备类型	迁移工具注意事项
3278-2, 3278-2B, ANY-2D	24 x 80	显示器	没有特殊的注意事项。
3278-3, 3278-3B, ANY-3D	32 x 80	显示器	没有特殊的注意事项。
3278-4, 3278-4B, ANY-4D	43 x 80	显示器	没有特殊的注意事项。
3278-5, 3278-5B, ANY-5D	27 x 132	显示器	没有特殊的注意事项。
ANY-D (配置为 62x160 的 3290)	255 x 160	显示器	COBOL 生成功能不支持此设备大小。
5550D	24 x 80	DBCS 显示器	没有特殊的注意事项。
3767, PRINT-B, PRINTER	255 x 132	打印机	对于 printFloatingArea, EGL deviceType=singleByte
5550P	255 x 158	DBCS 打印机	对于 printFloatingArea, EGL deviceType=doubleByte

## 映射

『映射』一节由下列各表组成:

- 映射 - 一般信息 (第 198 页的表 79)
- 显示器映射 - 一般语法、映射类型和属性 (第 199 页的表 80)
- 打印机映射 - 一般语法、映射类型和属性 (第 201 页的表 81)
- 映射常量和变量字段 - 一般信息 (第 202 页的表 82)
- 映射常量和变量字段 - 一般语法、数据类型、长度、小数位数和描述 (第 204 页的表 83)
- 映射常量和变量字段 - 属性 (第 206 页的表 84)
- 映射变量字段 - 一般编辑 (第 208 页的表 85)
- 映射变量字段 - 数字编辑 (第 209 页的表 86)
- 映射变量字段 - 错误消息 (第 210 页的表 87)

表 79. 映射 - 一般信息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
映射分为两种类型: <ul style="list-style-type: none"> <li>• 显示器映射</li> <li>• 打印机映射</li> </ul>	表单分为两种类型: <ul style="list-style-type: none"> <li>• 文本表单</li> <li>• 打印表单</li> </ul>	没有特殊的注意事项。

表 79. 映射 - 一般信息 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
映射名由映射组名和映射名组成。	<p>表单名不包含 formGroup 名。</p> <p>可以在 formGroup 中定义（嵌套）表单。</p> <p>并且，表单也可以位于 formGroup 外部。在这种情况下，formGroup 必须包括一个 use 语句来指定表单名，并包括一个 import 语句来导入该表单所在的包。通过使用这种技术，您可以定义一个公共表单（例如弹出列表表单）并在许多不同的 formGroup 中使用该表单。</p>	<p>迁移工具将所有映射都迁移为表单。此工具不会尝试标识跨多个映射组的完全相同的公共映射定义。</p> <p>如果以单文件方式进行迁移，则迁移工具对 formGroup 中的每个表单都包括一个 use 语句。您应该移动表单以使它们嵌套在它们的 formGroup 内。</p> <p>如果使用第 1 - 3 阶段迁移来进行迁移，则迁移工具将自动地把所有表单都嵌套在 formGroup 部件中。</p>
当程序指定了映射组时，该程序可以仅仅通过引用映射名来使用该映射组中的任何映射。	当程序包括了 use 语句以指示所使用的 formGroup 时，该程序可以仅仅通过引用表单名来引用该 formGroup 中的任何映射。	没有特殊的注意事项。

表 80. 显示器映射 - 一般语法、映射类型和属性

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>显示器映射可以包含：</p> <ul style="list-style-type: none"> <li>• 映射组名和映射名</li> <li>• 映射属性 <ul style="list-style-type: none"> <li>- 一般属性 <ul style="list-style-type: none"> <li>- 帮助映射名</li> <li>- 帮助键</li> <li>- 旁路键</li> <li>- 变量字段折换</li> </ul> </li> <li>- 布局属性 <ul style="list-style-type: none"> <li>- 映射大小</li> <li>- 起始位置</li> <li>- 浮动映射</li> </ul> </li> <li>- 设备 <ul style="list-style-type: none"> <li>- 类型（显示或打印）</li> <li>- 支持的设备</li> </ul> </li> </ul> </li> <li>• 常量字段</li> <li>• 变量字段</li> <li>• 变量字段的字段编辑顺序</li> </ul>	<p>文本表单部件可以包含：</p> <ul style="list-style-type: none"> <li>• 表单名</li> <li>• 表单类型</li> <li>• 表单属性</li> <li>• 常量字段</li> <li>• 变量字段</li> <li>• 变量字段的验证顺序</li> </ul> <p>以下是由迁移工具创建的文本表单的格式示例：</p> <pre> Form mapName type textForm { screenSizes=[sizeList],   formSize=[24,80], position=[1,1],   helpForm="helpFormName",   helpKey=pf1,   validationBypassKeys=[pf3],   msgField="VAGEN_EZEMSG"} [ variableFields ] [ constantFields ] end // end mapName                     </pre>	<p>迁移工具使用 VAGen 设备类型来确定该映射是显示器映射（文本表单）还是打印机映射（打印表单）。</p> <p>要了解有关确定设备是显示器还是打印机的信息，请参阅第 197 页的表 78。</p>
帮助映射名	helpForm	没有特殊的注意事项。
帮助键	helpKey	没有特殊的注意事项。
旁路键	validationBypassKeys	没有特殊的注意事项。
最多可以为映射指定 5 个旁路键。	最多可以为表单指定 5 个 validationBypassKeys。	

表 80. 显示器映射 - 一般语法、映射类型和属性 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
变量字段折换	对表单不受支持。表单上的每个 char 或 mbchar 变量字段都必须指定是否将用户输入的数据自动转换为大写数据。	迁移工具执行下列操作: <ul style="list-style-type: none"> <li>• 如果对整个映射指定了变量字段折换, 则迁移工具将为每个字符字段和混合字段包括 <code>upperCase=yes</code>。</li> <li>• 如果未对整个映射指定变量字段折换, 则迁移工具使用对每个字符字段或混合字段指定的折换信息来确定是否对该字段设置 <code>upperCase</code> 属性。</li> </ul>
映射大小 - 行数和列数	<code>formSize = [Lines, Columns]</code>	没有特殊的注意事项。
起始位置 - 如果映射是浮动映射的话, 则需要将行列指定为 NEXT,SAME。	<code>position = [Line, Column ]</code> 如果省略了位置信息, 则该表单是浮动表单。	如果选择了浮动映射, 则迁移工具将省略位置信息。
浮动映射	无。如果省略了位置信息, 则该表单是浮动表单。	如果选择了浮动映射, 则迁移工具将省略位置信息。
设备类型 - 显示器或 DBCS 显示器	<code>type textForm</code>	迁移工具使用设备类型信息来确定是将该映射迁移为文本表单还是迁移为打印表单。
支持的设备 注: 支持的设备显示了设备类型、行数和列数	<code>screenSizes = [[Lines, Columns], [Lines, Columns]]</code> 注: 对于要让表单支持的每种屏幕大小, 包括 <code>[Lines, Columns]</code> 对。	迁移工具使用设备类型信息来确定相应的 <code>screenSizes</code> 属性。如果几个 VAGen 设备具有相同的屏幕大小, 则迁移工具只将该屏幕大小包括一次。  由于并非 VAGen 支持的所有设备在 EGL 中都受 COBOL 生成功能支持, 所以有一些特殊的注意事项。要了解详细信息, 请参阅第 51 页的『映射组、映射和设备大小』。
无。在 VisualAge Generator 中, 消息字段始终名为 EZEMSG。	<code>msgField</code> 这是要包含任何 EGL 错误消息的字段的名称。	如果映射中的任何位置包含 EZEMSG, 则迁移工具设置 <code>msgField</code> 属性。
无。	<code>alias</code>	如果映射由于与 EGL 保留字冲突或者由于映射名以符号 # 开头而必须被重命名, 则迁移工具包括 <code>alias</code> 属性。对于程序的帮助映射组中的映射, 如果该映射由于与该程序的主映射组中的映射名冲突而必须被重命名, 则迁移工具也包括 <code>alias</code> 属性。  有一些特殊的注意事项。要了解详细信息, 请参阅第 52 页的『映射名和帮助映射名』。

表 81. 打印机映射 - 一般语法、映射类型和属性

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>打印机映射可以包含:</p> <ul style="list-style-type: none"> <li>• 映射组名和映射名</li> <li>• 映射属性 <ul style="list-style-type: none"> <li>- 一般属性 <ul style="list-style-type: none"> <li>- 帮助映射名</li> <li>- 帮助键</li> <li>- 旁路键</li> <li>- 变量字段折换</li> <li>- SO/SI 占据位置</li> </ul> </li> <li>- 布局属性 <ul style="list-style-type: none"> <li>- 映射大小</li> <li>- 起始位置</li> <li>- 浮动映射</li> </ul> </li> <li>- 设备 <ul style="list-style-type: none"> <li>- 类型 (显示或打印)</li> <li>- 支持的设备</li> </ul> </li> </ul> </li> <li>• 常量字段</li> <li>• 变量字段</li> <li>• 变量字段的字段编辑顺序</li> </ul>	<p>打印表单可以包含:</p> <ul style="list-style-type: none"> <li>• 表单名</li> <li>• 表单属性</li> <li>• 常量字段</li> <li>• 变量字段</li> </ul> <p>以下是由迁移工具创建的文本表单的格式示例:</p> <pre> Form mapName type printForm { formSize=[255,158], position=[1,1],   addSpaceForSOSI=yes } [ variableFields ] [ constantFields ] end // end mapName                     </pre>	<p>迁移工具使用 VAGen 设备类型来确定该映射是显示器映射 (文本表单) 还是打印机映射 (打印表单)。</p> <p>对于打印表单, 迁移工具总是省略下列属性:</p> <ul style="list-style-type: none"> <li>• 一般属性 <ul style="list-style-type: none"> <li>- 帮助映射名</li> <li>- 帮助键</li> <li>- 旁路键</li> <li>- 变量字段折换</li> </ul> </li> <li>• 设备 <ul style="list-style-type: none"> <li>- 支持的设备</li> </ul> </li> <li>• 变量字段的字段编辑顺序</li> </ul> <p>要了解有关确定设备是显示器还是打印机的信息, 请参阅第 196 页的表 77。</p>
帮助映射名	不适用于打印表单。	对于打印表单, 迁移工具省略此属性。
帮助键	不适用于打印表单。	对于打印表单, 迁移工具省略此属性。
旁路键	不适用于打印表单。	对于打印表单, 迁移工具省略此属性。
变量字段折换	不适用于打印表单。	对于打印表单, 迁移工具省略此属性。
SO/SI 占据位置	addSpaceForSOSI	没有特殊的注意事项。
映射大小 - 行数和列数	formSize = [Lines, Columns]	没有特殊的注意事项。
起始位置 - 如果映射是浮动映射的话, 则需要将行列指定为 NEXT,SAME。	position = [Line, Column ] 如果省略了位置信息, 则该表单是浮动表单。	如果选择了浮动映射, 则迁移工具将省略位置信息。
浮动映射	无。 如果省略了位置信息, 则该表单是浮动表单。	如果选择了浮动映射, 则迁移工具将省略位置信息。
设备类型 - 打印机或 DBCS 打印机	type printForm	迁移工具使用设备类型信息来确定是将该映射迁移为文本表单还是迁移为打印表单。
支持的设备	不适用于打印表单。	对于打印表单, 迁移工具省略此属性。
无。在 VisualAge Generator 中, 消息字段始终名为 EZEMSG。	msgField 这是要包含任何 EGL 错误消息的字段的名称。	如果映射中的任何位置包含 EZEMSG, 则迁移工具设置 msgField 属性。

表 81. 打印机映射 - 一般语法、映射类型和属性 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
无。	alias	<p>如果映射由于与 EGL 保留字冲突或者由于映射名以符号 # 开头而必须被重命名, 则迁移工具包括 <i>alias</i> 属性。对于程序的帮助映射组中的映射, 如果该映射由于与该程序的主映射组中的映射名冲突而必须被重命名, 则迁移工具也包括 <i>alias</i> 属性。</p> <p>有一些特殊的注意事项。请参阅第 52 页的『映射名和帮助映射名』</p>

表 82. 映射常量和变量字段 - 一般信息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>必须将映射中的所有位置计为下列其中一项:</p> <ul style="list-style-type: none"> <li>• 1 个变量字段</li> <li>• 1 个常量字段</li> <li>• 位于常量字段或变量字段开头的 1 个属性字节</li> </ul>	<p>不必对表单上的所有位置进行这种计算。不需要指定具有缺省属性 (noHighLight、normalIntensity、protect=skip、defaultColor、no outlining 和 no cursor) 的空白常量。</p>	<p>迁移工具省略具有缺省属性的空白常量。</p>
<p>可以对显示器映射中的常量字段指定实际上并不适用于常量的属性。例如:</p> <ul style="list-style-type: none"> <li>• 不受保护</li> <li>• 需要输入</li> <li>• 在输入时需要填充</li> <li>• 数字属性</li> <li>• 已修正数据标记</li> </ul>	<p>文本表单上的常量字段不能指定对常量无意义的属性。</p>	<p>对于文本表单上的常量, 迁移工具将省略不受支持的属性。</p>
<p>打印机映射中的常量字段可以具有实际上并不适用于打印机的属性。例如:</p> <ul style="list-style-type: none"> <li>• 颜色</li> <li>• 亮度</li> <li>• 除下划线以外的突出显示</li> <li>• 保护</li> <li>• 初始光标字段</li> <li>• 光笔检测</li> </ul>	<p>打印表单上的常量字段不能指定对常量无意义的属性。</p>	<p>对于打印表单上的常量, 迁移工具将省略不受支持的属性。</p>

表 82. 映射常量和变量字段 - 一般信息 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>打印机映射中的变量字段可以指定实际上并不适用于打印机的属性。例如:</p> <ul style="list-style-type: none"> <li>• 颜色</li> <li>• 亮度</li> <li>• 除下划线以外的突出显示</li> <li>• 保护</li> <li>• 初始光标字段</li> <li>• 需要输入</li> <li>• 在输入时需要填充</li> <li>• 数字属性</li> <li>• 已修正数据标记</li> <li>• 光笔检测</li> </ul>	<p>打印表单上的变量字段不能指定对打印表单无意义的属性。</p>	<p>对于打印表单上的变量字段，迁移工具将省略不受支持的属性。</p>
<p>打印机映射中的变量字段可以指定实际上并不适用于打印机的编辑。例如:</p> <ul style="list-style-type: none"> <li>• 最小输入</li> <li>• 折换</li> <li>• 十六进制编辑</li> <li>• 需要输入</li> <li>• 最小值</li> <li>• 最大值</li> <li>• 编辑消息</li> </ul>	<p>打印表单上的变量字段不能指定对打印表单无意义的属性。</p>	<p>对于打印表单上的变量字段，迁移工具将省略不受支持的属性。</p>

表 83. 映射常量和变量字段 - 一般语法、数据类型、长度、小数位数和描述

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>映射中的变量具有下列信息:</p> <ul style="list-style-type: none"> <li>• 名称</li> <li>• 基于您放在该映射中的内容的信息: <ul style="list-style-type: none"> <li>- 数据类型</li> <li>- 位置</li> </ul> </li> <li>• 基本信息: <ul style="list-style-type: none"> <li>- 描述</li> <li>- 初始值</li> <li>- 以字节计的长度</li> <li>- 数组下标</li> <li>- 数字编辑</li> </ul> </li> <li>• 属性</li> <li>• 编辑, 包括小数位数</li> <li>• 错误消息</li> </ul> <p>注: <i>position</i> 是属性字节的位置。以字节计的长度是以字节计的字段长度 (不包括属性字节)。以字节计的长度也用作数据值的长度。</p>	<p>映射中的变量字段的信息包括:</p> <ul style="list-style-type: none"> <li>• 名称</li> <li>• 字符字段的类型和长度 (以字符计)</li> <li>• 数字字段的类型、精度和小数位</li> <li>• 位置</li> <li>• 以字节计的字段长度</li> <li>• 表示属性</li> <li>• 定义格式的屬性</li> <li>• 验证属性</li> <li>• 值</li> </ul> <p>通常, 以下情况成立:</p> <ul style="list-style-type: none"> <li>• VAGen 属性与 EGL 表示属性对应。</li> <li>• VAGen 编辑和消息与 EGL 定义格式的屬性或验证属性对应。</li> <li>• 但是, 某些 VAGen 属性和编辑被合并成单个 EGL 属性, 或者被移到另一个类别。</li> </ul> <p>以下是 EGL 变量字段的一个示例:</p> <pre> itemName dataType(lengthInformation) // description { position=[row,column], fieldLen=length validationOrder=n [presentationProperties] [formattingProperties] [value="initialValue"] } [arrayInformation]                     </pre> <p>注: <i>position</i> 是属性字节的位置。<i>fieldLen</i> 是以字节计的字段长度 (不包括属性字节)。<i>dataType(lengthInformation)</i> 中给出的基本类型信息是数据值的长度。</p>	<p>迁移工具将 EGL <i>fieldLen</i> 属性设置为以字节计的 VAGen 长度。此工具按如下方式设置 <i>dataType</i> 的 <i>lengthInformation</i>:</p> <ul style="list-style-type: none"> <li>• 对于 <i>char</i>、<i>dbchar</i> 和 <i>mbchar</i> 字段, 迁移工具将 <i>lengthInformation</i> 设置为字符数而不是字节数。</li> <li>• 对于指定了数字编辑的 VAGen <i>char</i> 字段, 迁移工具执行下列操作: <ul style="list-style-type: none"> <li>- 将该字段转换为 EGL <i>num</i> 类型。</li> <li>- 将精度设置为以字节计的 VAGen 长度。然后, 如果在 VisualAge Generator 中对该字段指定了小数位数, 则将精度减小 1。</li> <li>- 将小数位设置为 VisualAge Generator 中指定的小数位数。</li> </ul> </li> </ul> <p>有一些特殊的注意事项。要了解详细信息, 请参阅第 54 页的『数字变量字段』。</p>

表 83. 映射常量和变量字段 - 一般语法、数据类型、长度、小数位数和描述 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>映射中的常量具有下列信息:</p> <ul style="list-style-type: none"> <li>• 基于您放在该映射中的内容的信息: <ul style="list-style-type: none"> <li>- 数据类型</li> <li>- 位置</li> </ul> </li> <li>• 基本信息: <ul style="list-style-type: none"> <li>- 初始值</li> <li>- 以字节计的长度</li> </ul> </li> <li>• 属性</li> </ul> <p>注: <i>position</i> 是属性字节的位置。以字节计的长度是以字节计的字段长度 (不包括属性字节)。</p>	<p>映射中的常量字段的信息包括:</p> <ul style="list-style-type: none"> <li>• 位置</li> <li>• 字段长度</li> <li>• 表示属性</li> <li>• 值</li> </ul> <p>通常, 以下情况成立:</p> <ul style="list-style-type: none"> <li>• VAGen 属性与 EGL 表示属性对应。</li> <li>• EGL 常量字段不支持仅适用于输入编辑的属性。</li> </ul> <p>常量的数据类型是根据 <i>value</i> 属性确定的。</p> <p>以下是 EGL 常量字段的一个示例:</p> <pre>{ position=[row,column],   fieldLen=length,   [presentationProperties]   [value="initialValue"] }</pre> <p>注: <i>position</i> 是属性字节的位置。<i>fieldLen</i> 是以字节计的字段长度 (不包括属性字节)。</p>	<p>迁移工具将 EGL <i>fieldLen</i> 属性设置为 VisualAge Generator 长度。</p>
<p>数据类型:</p> <ul style="list-style-type: none"> <li>• 字符常量</li> <li>• 字符变量</li> <li>• DBCS 常量</li> <li>• DBCS 变量</li> <li>• 混合常量</li> <li>• 混合变量</li> <li>• 选择了数字编辑的字符变量</li> </ul> <p>注: 类型是根据您放在映射中的字段的类型以及是否选择了“数字编辑”框确定的。</p>	<p>EGL 数据类型:</p> <ul style="list-style-type: none"> <li>• 无</li> <li>• char</li> <li>• 无</li> <li>• dbchar</li> <li>• 无</li> <li>• mbchar</li> <li>• num</li> </ul>	<p>没有特殊的注意事项。</p>
<p>描述</p>	<p>无。</p>	<p>迁移工具将描述转换为位于数据类型和长度信息后面的注释。</p>
<p>初始值</p>	<p><i>value</i></p> <p>注:</p> <ul style="list-style-type: none"> <li>• 在 VisualAge Generator 兼容性方式下, 仅当在屏幕上显示未被赋予值的字段时, 才使用 <i>value</i> 属性。<i>value</i> 属性不用来设置存储器中的字段的初始值。</li> <li>• 当未指定 VisualAge Generator 兼容性方式时, <i>value</i> 属性在程序启动时为程序中的字段提供初始值。</li> </ul>	<p>没有特殊的注意事项。</p>

表 83. 映射常量和变量字段 - 一般语法、数据类型、长度、小数位数和描述 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
长度	EGL 变量具有下列属性: <ul style="list-style-type: none"> <li>length, 这是字段中的字符数或位数。</li> <li>fieldLen, 这是字段在映射中占用的空间 (不包括属性字节)。</li> </ul>	迁移工具使用 VAGen 长度来设置 EGL length 和 EGL fieldLen 属性。对于数字字段, 有一些特殊的注意事项。要了解详细信息, 请参阅第 54 页的『数字变量字段』。
数组下标 注: <ul style="list-style-type: none"> <li>数组大小是根据变量字段的最大数组下标确定的。</li> <li>可以覆盖数组元素的一些属性, 如光标位置、颜色、突出显示、亮度和保护。</li> <li>也可以覆盖数组元素的初始值。</li> </ul>	<pre>itemName datatype(lengthInfo) [ arraySize ] { properties for index 1 } itemName[n] { properties for index n }</pre> 注: <ul style="list-style-type: none"> <li>数组大小是紧接着 datatype 和长度信息指定的。</li> <li>可以覆盖光标位置和表示属性, 如颜色、突出显示、亮度和保护。</li> <li>也可以覆盖 value 属性。</li> <li>可以通过执行下列操作来指定每个元素的位置:                             <ul style="list-style-type: none"> <li>通过 <i>position=[row,column]</i> 来指定每个元素的确切位置。</li> <li>对下标 1 指定以下附加属性: <i>columns</i>、<i>linesBetweenRows</i>、<i>spacesBetweenColumns</i> 和 <i>indexOrientation</i>。</li> </ul> </li> </ul>	迁移工具始终显式地对每个数组元素设置位置。
字段编辑顺序 注: <ul style="list-style-type: none"> <li>字段编辑顺序是从“定义”下拉列表中指定的。</li> <li>缺省字段编辑顺序基于变量字段在映射中的位置, 即先从左到右, 然后从上到下。</li> <li>某些版本的跨系统产品和 VisualAge Generator 不在“外部源格式”中记录字段编辑顺序。</li> </ul>	validationOrder 注: 缺省 validationOrder 基于变量字段在映射中的位置, 即先从左到右, 然后从上到下。	如果未将 validationOrder 包括在映射的“外部源格式”中, 则迁移工具将省略它。

表 84. 映射常量和变量字段 - 属性

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
亮度: <ul style="list-style-type: none"> <li>正常</li> <li>暗</li> <li>亮</li> </ul>	亮度: <ul style="list-style-type: none"> <li>normalIntensity</li> <li>invisible</li> <li>bold</li> </ul> (表示属性)	没有特殊的注意事项。

表 84. 映射常量和变量字段 - 属性 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>突出显示:</p> <ul style="list-style-type: none"> <li>不突出显示</li> <li>闪烁</li> <li>反相显示</li> <li>下划线</li> </ul>	<p>突出显示:</p> <ul style="list-style-type: none"> <li>noHighlight</li> <li>blink</li> <li>reverse</li> <li>underline</li> </ul> <p>(表示属性)</p>	没有特殊的注意事项。
<p>保护:</p> <ul style="list-style-type: none"> <li>不受保护</li> <li>受保护</li> <li>自动跳过</li> </ul>	<p>保护:</p> <ul style="list-style-type: none"> <li>no</li> <li>yes</li> <li>skip</li> </ul> <p>(表示属性)</p>	没有特殊的注意事项。
<p>颜色:</p> <ul style="list-style-type: none"> <li>单色</li> <li>蓝色</li> <li>红色</li> <li>粉红色</li> <li>绿色</li> <li>青绿色</li> <li>黄色</li> <li>白色</li> </ul>	<p>颜色:</p> <ul style="list-style-type: none"> <li>defaultColor</li> <li>blue</li> <li>red</li> <li>magenta</li> <li>green</li> <li>cyan</li> <li>yellow</li> <li>white</li> </ul> <p>(表示属性)</p>	没有特殊的注意事项。
初始光标字段	<p>cursor = yes   no</p> <p>(表单字段属性)</p>	没有特殊的注意事项。
需要输入	inputRequired (验证属性)	<p>迁移工具按如下方式将 VAGen “需要输入” 属性与 “需要输入” 编辑合并:</p> <ul style="list-style-type: none"> <li>如果选择了 “需要输入” 属性或 “需要输入” 编辑, 则迁移工具包括 inputRequired。</li> <li>如果两项都未选择, 则迁移工具省略 inputRequired。</li> </ul>
在输入时需要填充	fill (验证属性)	没有特殊的注意事项。
<p>数字属性</p> <p>注: CHA 字段 (包括选择了数字编辑的 CHA 字段) 支持此属性。</p>	<p>isDecimalDigit (验证属性)</p> <p>注: 只有 char 字段才支持此属性。</p>	<p>如果选择了数字属性, 则迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>对 char 字段包括 isDecimalDigit。</li> <li>对数字字段省略 isDecimalDigit。EGL 为数字字段提供了软件编辑以便与 VAGen 兼容。</li> </ul> <p>要了解其它详细信息, 请参阅第 56 页的『映射字段和数字硬件属性』。</p>
已修正数据标记	modified (表示属性)	没有特殊的注意事项。

表 84. 映射常量和变量字段 - 属性 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
光笔检测	detectable (表示属性)	没有特殊的注意事项。
轮廓: <ul style="list-style-type: none"> <li>• left</li> <li>• right</li> <li>• over</li> <li>• under</li> <li>• box</li> </ul>	轮廓: <ul style="list-style-type: none"> <li>• left</li> <li>• right</li> <li>• top</li> <li>• bottom</li> <li>• box</li> </ul> (表示属性)	没有特殊的注意事项。

表 85. 映射变量字段 - 一般编辑

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
编辑例程	validatorFunction 或 validatorDataTable (验证属性)	迁移工具执行下列操作: <ul style="list-style-type: none"> <li>• 如果映射编辑例程是 EZEC10 或 EZEC11, 则设置 validatorFunction 属性。</li> <li>• 如果编辑例程是函数, 则设置 validatorFunction 属性。</li> <li>• 如果编辑例程是表, 则设置 validatorDataTable 属性。</li> </ul> 注: 如果编辑例程在迁移期间不可用, 则有一些特殊的注意事项。请参阅第 55 页的『映射变量字段和编辑例程』以了解其它详细信息和潜在问题。
对齐 - 左   右   无 注: 对于映射项, 数字字段的缺省值是“右”, 所有其它字段的缺省值是“左”。	align = left   right   none (定义格式的属性) 注: 对于表单字段, 数字字段的缺省值是 right, 所有其它字段的缺省值是 left。	没有特殊的注意事项。
日期编辑掩码 值如下所示: <ul style="list-style-type: none"> <li>• SYSGREGRN</li> <li>• SYSJULIAN</li> <li>• dateEditPattern</li> </ul>	dateFormat = value 值如下所示: <ul style="list-style-type: none"> <li>• systemGregorianCalendar</li> <li>• systemJulianDateFormat</li> <li>• "dateEditPattern"</li> </ul> (定义格式的属性) 注: 在 dateEditPattern 中, 迁移工具将内容转换为下列 EGL 表示法: <ul style="list-style-type: none"> <li>• yy 或 yyyy 表示年份。</li> <li>• MM 表示月份。</li> <li>• dd 表示一个月中的第几天。</li> <li>• DDD 表示一年中的第几天。</li> </ul>	没有特殊的注意事项。
最小输入	minimumInput (验证属性)	没有特殊的注意事项。

表 85. 映射变量字段 - 一般编辑 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
填充字符 注: 对于字符、DBCS 或混合字段, 映射中使用的项的缺省填充字符是 NULL; 对于数字字段, 是空白; 对于十六进制字段, 是 0。	fillCharacter (定义格式的属性) 注: <ul style="list-style-type: none"> <li>对于字符、DBCS 或混合字段, 映射中使用的项的缺省填充字符是 NULL; 对于数字字段, 是空白; 对于十六进制字段, 是 0。</li> <li>strLib.nullFill 是 EGL 的 NULL 填充字符常量。此外, 也可以使用 "" (两个连续的引号)。</li> </ul>	没有特殊的注意事项。
折换	upperCase (定义格式的属性)	迁移工具执行下列操作: <ul style="list-style-type: none"> <li>如果对整个映射指定了变量字段折换, 则迁移工具将为每个字符字段和混合字段包括 upperCase=yes。</li> <li>如果未对整个映射指定变量字段折换, 则迁移工具使用对每个字符字段或混合字段指定的折换信息来确定是否对该字段设置 upperCase 属性。</li> </ul>
十六进制编辑	isHexEdit (验证属性)	没有特殊的注意事项。
需要输入	inputRequired (验证属性)	迁移工具按如下方式将 VAGen “需要输入” 属性与 “需要输入” 编辑合并: <ul style="list-style-type: none"> <li>如果选择了 “需要输入” 属性或 “需要输入” 编辑, 则迁移工具包括 inputRequired。</li> <li>如果两项都未选择, 则迁移工具省略 inputRequired。</li> </ul>
检查 SO/SI 空格	needsSOSI (验证属性)	没有特殊的注意事项。

表 86. 映射变量字段 - 数字编辑

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
最小值和最大值 注: 如果指定了最小值或最大值, 则必须同时指定这两者。	validValues = [[minimumValue, maximumValue]] (验证属性) 注: 在 validValues 属性中, 可以列示多对值和单个的值。	迁移工具将最小值和最大值组合成 EGL validValues 属性。
符号: <ul style="list-style-type: none"> <li>无</li> <li>前</li> <li>后</li> </ul> 注: 缺省值为 “无”。	符号: <ul style="list-style-type: none"> <li>none</li> <li>leading</li> <li>trailing</li> </ul> (验证属性) 注: 缺省值为 none。	没有特殊的注意事项。

表 86. 映射变量字段 - 数字编辑 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
货币	currency = yes   no  currencySymbol = "symbol"  (定义格式的属性) 注: 如果 currency = yes, 但是未指定 currencySymbol, 则在运行时使用的实际货币符号的设置方式与 VisualAge Generator 中的设置方式相同。	迁移工具仅将 currency 设置为 yes 或 no。对于表单变量字段来说, 由于在 VisualAge Generator 中没有等效的信息, 因此此工具从来不设置 currencySymbol="symbol"。
分隔符	numericSeparator (定义格式的属性)	没有特殊的注意事项。
零编辑	zeroFormat (定义格式的属性)	没有特殊的注意事项。

表 87. 映射变量字段 - 错误消息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
编辑例程	validatorFunctionMsgKey 或者 validatorDataTableMsgKey (验证属性)	迁移工具按如下方式迁移编辑例程消息: <ul style="list-style-type: none"> <li>• 如果编辑例程是 EZEC10 或 EZEC11, 则设置 validatorFunctionMsgKey。</li> <li>• 如果编辑例程是表, 则设置 validatorDataTableMsgKey。</li> <li>• 如果编辑例程是函数, 则不迁移编辑例程消息, 这是因为, 在 VisualAge Generator 中, 在这种情况下不使用该消息。</li> </ul> 请参阅第 55 页的『映射变量字段和编辑例程』以了解其它详细信息和潜在问题。
最小输入	minimumInputMsgKey (验证属性)	没有特殊的注意事项。
需要输入	inputRequiredMsgKey (验证属性)	没有特殊的注意事项。
数据类型	typeChkMsgKey (验证属性)	没有特殊的注意事项。
数字范围	validValuesMsgKey (验证属性)	没有特殊的注意事项。

## 程序

『程序』一节由下列各表组成:

- 程序 - 一般语法、程序类型、被调用参数和 prolog (第 211 页的表 88)
- 程序 - 程序说明、属性和“表及其它记录”列表 (第 212 页的表 89)
- 程序 - 主函数和流语句 (第 214 页的表 90)

表 88. 程序 - 一般语法、程序类型、被调用参数和 prolog

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>程序部件:</p> <ul style="list-style-type: none"> <li>• programName</li> <li>• 程序类型</li> <li>• 说明 (随程序类型的不同而有所变化): <ul style="list-style-type: none"> <li>- 工作存储器记录</li> <li>- PSB</li> <li>- Firstmap</li> <li>- 第一个用户界面记录</li> <li>- 映射组</li> <li>- 帮助映射组</li> </ul> </li> <li>• 表及其它记录</li> <li>• 被调用参数</li> <li>• Prolog</li> <li>• 属性 (根据程序类型的不同而有所变化)</li> <li>• 结构图 <ul style="list-style-type: none"> <li>- 主函数</li> <li>- 流语句 (在结构图中是隐藏的, 但可以对任何主函数指定它)</li> </ul> </li> </ul>	<p>EGL 语法样本:</p> <pre> /***/ Program=programName // prolog //***** <b>Program</b> programName   <b>type</b> eglProgramType   //vagenProgramType   [ ( calledParameters ) ]   {   [alias= "originalProgramName"]   <b>includeReferencedFunctions</b>     =yes,   <b>allowUnqualifiedItemReferences</b>     =yes,   <b>localSQLScope</b>=yes,   <b>throwNrfEofExceptions</b>=yes,   <b>handHardIOErrors</b>=no   [ propertiesBasedOnType ]   }   [ dataDeclarations ]   [ useDeclarations ]   <b>function main ( )</b>   { <b>functionLabel</b>:     <b>functionName</b>( ) ;     [{<b>functionFlowStatements</b>}]}   <b>end</b> // end main <b>end</b> // end programName </pre>	<p>即使程序名与 EGL 保留字列表有冲突, 迁移工具也不会将该程序重命名。迁移工具不设置 alias 属性。如果您必须将该程序重命名, 则可以使用 alias 属性来指定 VAGen 程序的原始名称。请参阅第 59 页的「程序名和保留字」。</p> <p>迁移工具将 VAGen 程序类型作为注释包括在程序定义中。</p> <p>迁移工具按照以下方式迁移“表及其它记录”列表:</p> <ul style="list-style-type: none"> <li>• 将记录迁移为 dataDeclarations。</li> <li>• 将表迁移为 useDeclarations。</li> </ul> <p>迁移工具始终包括下列属性以保留 VAGen 行为:</p> <ul style="list-style-type: none"> <li>• includeReferencedFunctions</li> <li>• allowUnqualifiedItemReferences</li> <li>• localSQLScope</li> <li>• throwNrfEofExceptions</li> <li>• handleHardIOErrors</li> </ul>
<p>程序类型:</p> <ul style="list-style-type: none"> <li>• 主事务</li> <li>• 被调用事务</li> <li>• 主批处理</li> <li>• 被调用批处理</li> <li>• Web 事务</li> </ul> <p>注: 请参阅下面有关“主事务执行方式值”的那一行以了解其它详细信息。</p>	<p>EGL 程序类型:</p> <ul style="list-style-type: none"> <li>• textUIProgram</li> <li>• textUIProgram</li> <li>• basicProgram</li> <li>• basicProgram</li> <li>• 在此发行版中不受支持。</li> </ul>	<p>迁移工具将 VAGen 程序类型作为注释包括在程序定义中。请参阅第 212 页的表 89 以了解有关分段值如何与 EGL 属性对应的信息。</p>

表 88. 程序 - 一般语法、程序类型、被调用参数和 *prolog* (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>被调用参数 注:</p> <ul style="list-style-type: none"> <li>被调用参数是在一个特殊窗口中输入的。</li> <li>参数类型指示了该参数是项、记录还是映射。</li> <li>参数名总是另一个 VAGen 部件的名称。没有等效的 EGL 类型定义或基本类型。</li> </ul>	<p>EGL 被调用参数示例: ( <i>parameterName typeInfo</i> { , <i>parameterName typeInfo</i> } )</p> <ul style="list-style-type: none"> <li>必须用逗号将各个参数分隔开。</li> <li>参数可以是 <i>dataItem</i>、记录或表单。它们并不直接地与 VAGen 参数类型相对应。</li> <li>EGL <i>typeInfo</i>: <ul style="list-style-type: none"> <li>对于 <i>dataItem</i> 来说, 可以是基本类型</li> <li>对于 <i>dataItem</i>、记录或表单来说, 可以是类型定义。</li> </ul> </li> </ul>	<p>迁移工具包括原始 VAGen 参数类型作为注释。</p> <p>如果您选择了迁移语法首选项将共享数据项转换为基本项定义, 并且数据项部件可用, 则迁移工具将把共享项转换为使用基本定义进行声明的 EGL 参数, 该基本定义基于对该数据项部件指定的类型、长度和小数位数。类型、长度和小数位数信息的迁移方式与第 181 页的表 64 中描述的迁移方式相同。</p> <p>如果您未选择迁移语法首选项将共享数据项转换为基本项定义, 或者数据项部件不可用, 则迁移工具将把共享项转换为使用类型定义进行声明的 EGL 参数。对于迁移来说, 类型定义总是与项名相同。</p> <p>有一些特殊的注意事项。请参阅第 45 页的『重新定义的记录』以了解详细信息和潜在问题。</p>
Prolog	无。	迁移工具将 <i>prolog</i> 转换为位于程序定义前面的注释。

表 89. 程序 - 程序说明、属性和“表及其它记录”列表

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>一般信息:</p> <ul style="list-style-type: none"> <li>某些 VAGen 说明和属性信息将迁移为 EGL 属性、数据声明或使用声明。</li> </ul>	<p>一般信息:</p> <ul style="list-style-type: none"> <li>下列各行指示了相应的 EGL 语言元素是程序属性、数据声明还是使用声明。</li> </ul>	没有特殊的注意事项。
<p>主事务执行方式值:</p> <ul style="list-style-type: none"> <li>非分段</li> <li>分段</li> <li>单个分段</li> </ul> <p>注: 被调用事务总是以非分段方式运行。</p>	<p>分段 - 值:</p> <ul style="list-style-type: none"> <li><i>segmented = no</i></li> <li><i>segmented = yes</i></li> <li><i>segmented = yes</i></li> </ul> <p>(程序属性)</p> <p>注: 不对被调用程序指定 <i>segmented</i> 属性。</p>	如果外部源格式文件未包含分段信息, 则迁移工具在缺省情况下迁移为 <i>segmented = no</i> 。

表 89. 程序 - 程序说明、属性和“表及其它记录”列表 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>工作存储器记录 (说明)</p> <ul style="list-style-type: none"> <li>既可以主程序指定工作存储器记录也可以对被调用程序指定工作存储器记录。有时, 将工作存储器记录称为程序的主工作存储器记录。</li> <li>总是初始化主工作存储器记录。</li> </ul>	<p>inputRecord (程序属性)</p> <ul style="list-style-type: none"> <li>只能对主程序指定 inputRecord 属性。</li> <li>总是初始化 inputRecord。</li> <li>数据声明也是必需的。</li> </ul>	<p>对于主程序, 迁移工具将主工作存储器记录转换为 inputRecord 属性。</p> <p>对于主程序和被调用程序, 迁移工具还包括主工作存储器记录的数据声明。对于被调用程序中的数据声明, 迁移工具包括 <i>initialized = yes</i> 属性。</p> <p>如果主工作存储器记录包含 Level-77 项, 则迁移工具包括 Level-77 记录的数据声明语句。</p> <p>请参阅第 46 页的『记录中的 Level-77 项』以了解详细信息和潜在问题。</p>
PSB (说明)	在此发行版中不受支持。	迁移工具将注释掉 PSB 的使用声明。如果程序未使用 DL/I, 则您可能能够在此发行版中生成和运行它。
Firstmap (说明)	inputForm (程序属性)	没有特殊的注意事项。
第一个用户界面记录 (说明)	在此发行版中不受支持。	没有特殊的注意事项。
映射组 (说明)	<b>use formGroup</b> (使用声明)	没有特殊的注意事项。
帮助映射组 (说明)	<b>use formGroup { helpGroup=yes }</b> (使用声明)	没有特殊的注意事项。
消息表前缀 (程序属性)	msgTablePrefix (程序属性)	没有特殊的注意事项。
允许隐式数据项 (程序属性)	不受支持。	迁移工具不创建隐式定义。请参阅第 60 页的『程序中的隐式数据项』以了解详细信息和潜在问题。
<p>键指定:</p> <ul style="list-style-type: none"> <li>帮助键 (1 个键)</li> <li>旁路键 (可达 5 个键)</li> <li>F1-12 = F13-24</li> </ul> <p>(程序属性)</p> <p>注: 只要对程序指定一次键指定, 该键指定就将应用于映射组和帮助映射组。</p>	<p>EGL 键指定示例:</p> <pre><b>use formGroup</b> { [ <b>helpGroup = yes</b> ]   <b>helpKey = pfNumber,</b>   <b>validationBypassKeys =</b>     [ <b>pfNumberList</b> ],   <b>pfKeyEquate = yes   no } ;</b></pre> <p>(程序的 formGroup 和帮助 formGroup 的使用声明属性。)</p> <p>注:</p> <ul style="list-style-type: none"> <li>必须用逗号将 validationBypassKeys 列表中的各个值分隔开。</li> <li>不对程序的帮助 formGroup 指定 validationBypassKeys 属性。</li> </ul>	<p>迁移工具将键指定信息的 EGL 等效项包括在 formGroup 和帮助 formGroup 的 <i>use declaration</i> 语句中。迁移工具在帮助 formGroup 的使用声明中省略 validationBypassKeys 属性。</p>

表 89. 程序 - 程序说明、属性和“表及其它记录”列表 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>表及其它记录:</p> <ul style="list-style-type: none"> <li>记录</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>重新定义信息存储在 VAGen 重新定义的记录中, 而不是存储在程序中。</li> <li>永远不会将用作 I/O 对象的记录包括在“表及其它记录”列表中。</li> </ul>	<p>EGL 其它记录示例:</p> <pre>recordName recordName   [{ redefines = "otherRecord"}];</pre> <p>(数据声明)</p> <p>注:</p> <ul style="list-style-type: none"> <li>如果该记录与另一个记录对同一物理存储器提供不同的记录布局, 则必须对程序的数据声明指定 redefines 属性。</li> <li>数据声明对于程序中使用的所有记录 (包括 I/O 记录) 来说都是必需的。</li> </ul>	<p>迁移工具总是使用同一个记录名来作为类型定义。</p> <p>如果在程序中将一个 VAGen 记录用作重新定义的记录, 则迁移工具在数据声明语句中包括 redefines 属性。请参阅第 45 页的『重新定义的记录』以了解详细信息和潜在问题。</p> <p>迁移工具还包括被程序用作 I/O 对象的所有记录的数据声明。</p> <p>迁移工具包括这样的记录的数据声明: 那些记录被指定为被程序用作 I/O 对象的任何 MQ 消息记录的属性。</p>
<p>表及其它记录:</p> <ul style="list-style-type: none"> <li>表</li> <li>可以对每个表指定 Keep After Use。</li> </ul>	<p>EGL use 声明示例:</p> <pre>use tableName   [{deleteAfterUse = yes}];</pre> <p>(使用声明)</p>	<p>迁移工具将“表及其它记录”列表中的表转换为 use 声明。</p> <p>DeleteAfterUse 与 VAGen Keep After Use 的含义相反。迁移工具会将 yes 和 no 互换。</p> <p>如果指定 VAGen 迁移首选项不包括表的 deleteAfterUse, 则迁移工具将自动省略 deleteAfterUse 属性并对受影响的程序和表发出警告消息。</p>

表 90. 程序 - 主函数和流语句

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 程序将主函数指定为 VAGen 结构图中的顶级函数。仅当展开结构图时, 所有其它函数才会出现。</p> <p>每个主函数都可以具有流语句。这些语句不会出现在结构图中, 但可以通过该图来访问它们。</p>	<p>EGL 程序仅指定一个主函数。这个函数总是命名为 main。</p> <p>没有流语句。</p> <p>下面是一个程序主函数语法示例:</p> <pre>function main ( )   { functionLabel:     functionName ( ) ;   [ { functionFlowStatements } ] } end // end main</pre>	<p>迁移工具构建 EGL 主函数。对于每个 VAGen 主函数, 此工具在主函数内包括下列内容:</p> <ul style="list-style-type: none"> <li>函数标号, 以便可以在 EGL exit stack functionLabel 语句中引用该 VAGen 主函数。此工具总是将 functionLabel 设置为 functionName。</li> <li>函数调用语句, 以调用 VAGen 主 functionName。</li> <li>VAGen 主函数的流语句 (如果有的话)。</li> </ul> <p>请参阅下列各项以获取有关迁移流语句的详细信息:</p> <ul style="list-style-type: none"> <li>请参阅第 227 页的『语句』</li> <li>请参阅第 239 页的『EZE 字』</li> <li>请参阅第 248 页的『服务例程』</li> </ul>

---

## 函数

下列各表将 VAGen 函数部件与 EGL 函数部件进行比较，并描述了迁移工具处理转换的方式。

『函数』一节由下列各表组成：

- 函数 - 一般语法、描述、参数、返回值和本地存储器（第 216 页的表 91）
- 函数 - EXECUTE I/O 选项（第 218 页的表 92）
- 函数 - 映射和用户界面记录的 I/O 选项（第 218 页的表 93）
- 函数 - 记录的 I/O 错误例程（第 219 页的表 94）
- 函数 - 串行记录、带索引记录、相对记录和消息队列记录的 I/O 选项（第 220 页的表 95）
- 函数 - 不带“执行时语句构建”的缺省（未修改的）SQL 语句的 I/O 选项（第 220 页的表 96）
- 函数 - 不带“执行时语句构建”的已修改的 SQL 语句的 I/O 选项（第 222 页的表 97）
- 函数 - 带“执行时语句构建”的 SQL 语句的 I/O 选项（第 225 页的表 98）

**注：**此发行版的 EGL 不支持 Web 事务、用户界面记录或对 CONVERSE 用户界面记录进行替换。但是，预计用于映射和用户界面记录的 EGL converse 语句是相同的。迁移工具在转换 CONVERSE I/O 选项时并不考虑该 I/O 对象是映射还是用户界面记录。这样可以最大程度地保留逻辑。

表 91. 函数 - 一般语法、描述、参数、返回值和本地存储器

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>函数部件可以包含:</p> <ul style="list-style-type: none"> <li>• 函数名</li> <li>• I/O 选项</li> <li>• I/O 对象</li> <li>• 属性:                             <ul style="list-style-type: none"> <li>- 错误例程</li> <li>- 描述</li> </ul> </li> <li>• 函数返回值</li> <li>• 函数参数</li> <li>• 函数本地存储器</li> <li>• SQL 语句</li> <li>• I/O 选项前面的语句</li> <li>• I/O 选项后面的语句</li> <li>• DLI 调用</li> </ul>	<p>函数部件可以包含:</p> <ul style="list-style-type: none"> <li>• functionName</li> <li>• functionParameterList</li> <li>• returnItemType</li> <li>• dataDeclarations</li> <li>• I/O 语句前面的语句</li> <li>• I/O 语句</li> <li>• I/O 语句后面的语句</li> </ul> <p>以下是由迁移工具创建的函数调用语句的格式示例:</p> <pre>// Description Function functionName (functionParameterList) [ returns( returnItemType ) ] [ dataDeclarations ] [ beforeStatements ] [ I/O Statement ] [ afterStatements ] end // end functionName</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• 使用 VAGen I/O 选项、I/O 对象、错误例程和 SQL 语句来创建 EGL I/O 语句。</li> <li>• 在此发行版中, 不支持 DLI 调用。</li> </ul>	<p>迁移工具使用 I/O 选项、I/O 对象和错误例程来构建 EGL I/O 语句。请参阅第 219 页的表 94。</p> <p>有关迁移工具如何处理描述、函数返回值、函数参数和函数本地存储器的详细信息, 请参阅本表中的下列各行。</p> <p>要了解有关 I/O 语句前后的语句的迁移方式的详细信息, 请参阅下列内容:</p> <ul style="list-style-type: none"> <li>• 对于语句, 请参阅第 227 页的『语句』</li> <li>• 对于 EZE 字, 请参阅第 239 页的『EZE 字』</li> <li>• 对于服务例程, 请参阅第 248 页的『服务例程』</li> </ul> <p>要了解有关 I/O 选项和错误例程的详细信息, 请参阅下列内容:</p> <ul style="list-style-type: none"> <li>• 请参阅第 218 页的表 92 以了解 EXECUTE I/O 选项。</li> <li>• 请参阅第 218 页的表 93 以了解映射和用户界面记录的 I/O 选项。</li> <li>• 请参阅第 219 页的表 94 以了解记录的 I/O 错误例程。</li> <li>• 请参阅第 220 页的表 95 以了解串行记录、带索引记录、相对记录和消息队列记录的 I/O 选项。</li> </ul> <p>有关 SQL 语句的详细信息, 请参阅本节中的下列各表:</p> <ul style="list-style-type: none"> <li>• 请参阅第 220 页的表 96 以了解未修改的 SQL 语句的 I/O 选项。</li> <li>• 请参阅第 222 页的表 97 以了解不带“执行时语句构建”的已修改的 SQL 语句的 I/O 选项。</li> <li>• 请参阅第 225 页的表 98 以了解带有“执行时语句构建”的 SQL 语句的 I/O 选项。</li> </ul>
描述	无	迁移工具将函数描述转换为位于函数定义前面的注释。

表 91. 函数 - 一般语法、描述、参数、返回值和本地存储器 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>函数参数:</p> <ul style="list-style-type: none"> <li>函数参数是在一个特殊窗口中输入的。</li> <li>用作函数参数的项可以是共享的, 也可以是非共享的。非共享函数的定义存储在函数中。</li> </ul>	<p>函数参数:</p> <ul style="list-style-type: none"> <li>必须用逗号将各个参数分隔开。</li> <li>每个参数都具有类型信息。</li> <li>(可选) 每个参数都具有参数类型信息。</li> </ul> <p>以下是一个函数参数格式示例:</p> <pre>( parameterName typeInfo   [ parameterType ]   { , parameterName typeInfo     [ parameterType ] } )</pre> <p>以下是一个特定的函数参数示例:</p> <pre>(parmSharedItem parmSharedItem   field,   parmNonSharedItem char(10)   nullable,   parmRecord parmRecord)</pre>	<p>函数参数:</p> <p>迁移工具按如下方式设置类型信息:</p> <ul style="list-style-type: none"> <li>对于记录, typeInfo 是指定同一个记录名的类型定义。</li> <li>如果项类型是其中一种 VAGen Any* 类型, 则 typeInfo 是对应的 EGL 特殊项类型。</li> <li>如果该项是共享数据项, 则迁移工具执行下列操作:           <ul style="list-style-type: none"> <li>如果您选择了迁移语法首选项将共享数据项转换为基本项定义, 并且数据项部件可用, 则迁移工具将把共享项转换为使用基本定义进行声明的 EGL 函数参数, 该基本定义基于对该数据项部件指定的类型、长度和小数位数。类型、长度和小数位数信息的迁移方式与第 181 页的表 64 中描述的迁移方式相同。</li> <li>如果您未选择迁移语法首选项将共享数据项转换为基本项定义, 或者数据项部件不可用, 则迁移工具将把共享项转换为使用类型定义进行声明的 EGL 函数参数。对于迁移来说, 类型定义总是与项名相同。</li> </ul> </li> <li>如果项是非共享的数据项, 则根据项类型、长度和小数位数来迁移 typeInfo, 并遵循第 181 页的表 64 中描述的规则。</li> </ul>
<p>函数参数:</p> <ul style="list-style-type: none"> <li>函数参数类型:           <ul style="list-style-type: none"> <li>记录</li> <li>项</li> <li>映射项</li> <li>SQL 项</li> </ul> </li> </ul>	<p>函数参数:</p> <ul style="list-style-type: none"> <li>函数参数类型:           <ul style="list-style-type: none"> <li>无</li> <li>无</li> <li>字段</li> <li>可空</li> </ul> </li> </ul>	
<p>函数参数:</p> <ul style="list-style-type: none"> <li>特殊项类型, 未指定长度:           <ul style="list-style-type: none"> <li>AnyChar</li> <li>AnyDBCS</li> <li>AnyMix</li> <li>AnyHex</li> <li>AnyUnicode</li> <li>AnyNumeric</li> </ul> </li> </ul>	<p>函数参数:</p> <ul style="list-style-type: none"> <li>特殊项类型, 未指定长度:           <ul style="list-style-type: none"> <li>char</li> <li>dbchar</li> <li>mbchar</li> <li>hex</li> <li>unicode</li> <li>number</li> </ul> </li> </ul>	
<p>函数返回值:</p> <ul style="list-style-type: none"> <li>数据类型</li> <li>长度</li> <li>小数位数</li> <li>描述</li> </ul>	<p>EGL 返回值:</p> <ul style="list-style-type: none"> <li>以下是一个 <i>returns</i> 语句格式示例:</li> </ul> <pre>returns( returnItemType )   // Description</pre>	<p>如果函数包括返回值, 则迁移工具根据第 181 页的表 64 中描述的规则来迁移数据类型、长度和小数位数。</p>

表 91. 函数 - 一般语法、描述、参数、返回值和本地存储器 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>函数本地存储器:</p> <ul style="list-style-type: none"> <li>函数本地存储器是在一个特殊窗口中输入的。</li> <li>用作函数本地存储器的项可以是共享的,也可以是非共享的。非共享项的定义存储在函数中。</li> </ul>	<p>函数变量声明:</p> <ul style="list-style-type: none"> <li>函数变量声明必须包括变量名及其相关类型信息。</li> <li>以下是一个函数变量声明格式示例:  <pre>// Function Declarations variableName typeInfo ; { variableName typeInfo ; }</pre> </li> </ul>	<p>迁移工具按如下方式设置 typeInfo:</p> <ul style="list-style-type: none"> <li>对于记录, typeInfo 是指定同一个记录名的类型定义。</li> <li>如果该项是共享数据项,则迁移工具执行下列操作: <ul style="list-style-type: none"> <li>如果您选择了迁移语法首选项将共享数据项转换为基本项定义,并且数据项部件可用,则迁移工具将把共享项转换为使用基本定义进行声明的 EGL 变量,该基本定义基于对该数据项部件指定的类型、长度和小数位数。类型、长度和小数位数信息的迁移方式与第 181 页的表 64 中描述的迁移方式相同。</li> <li>如果您未选择迁移语法首选项将共享数据项转换为基本项定义,或者数据项部件不可用,则迁移工具将把共享项转换为使用类型定义进行声明的 EGL 变量。对于迁移来说,类型定义总是与项名相同。</li> </ul> </li> <li>如果项是非共享的数据项,则根据项类型、长度和小数位数来迁移 typeInfo,并遵循第 181 页的表 64 中描述的规则。</li> </ul>
<p>函数本地存储器:</p> <ul style="list-style-type: none"> <li>函数本地存储器类型: <ul style="list-style-type: none"> <li>记录</li> <li>项</li> </ul> </li> </ul>	<p>函数本地存储器:</p> <ul style="list-style-type: none"> <li>函数本地存储器类型: <ul style="list-style-type: none"> <li>无</li> <li>无</li> </ul> </li> </ul>	

表 92. 函数 - EXECUTE I/O 选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: 无</li> <li>I/O 选项: EXECUTE</li> </ul>	没有等效的语句。	迁移工具将除去 EXECUTE I/O 选项。

表 93. 函数 - 映射和用户界面记录的 I/O 选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: mapName</li> <li>I/O 选项: DISPLAY</li> </ul> <p>注: DISPLAY 既用于显示器映射也用于打印机映射。</p>	<p>要显示文本表单,使用 <i>display</i> 语句。要打印打印表单,使用 <i>print</i> 语句。</p> <p>下面是 display 语句和 print 语句示例:</p> <pre>display mapName; print mapName;</pre> <p>注: 在 VisualAge Generator 兼容性方式下, <i>display printForm</i> 被视为 <i>print printForm</i>。</p>	<p>迁移工具将根据映射类型转换为 display 或 print 语句。请参阅第 63 页的『用于映射的 DISPLAY 语句』以了解详细信息和潜在问题。</p>

表 93. 函数 - 映射和用户界面记录的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: mapName</li> <li>I/O 选项: CONVERSE</li> </ul>	使用 <i>converse</i> 语句。 下面是一个 <i>converse</i> 语句示例: <b>converse</b> mapName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: UIRecordName</li> <li>I/O 选项: CONVERSE</li> </ul>	使用 <i>converse</i> 语句。下面是一个 <i>converse</i> 语句示例: <b>converse</b> UIRecordName;	没有特殊的注意事项。

表 94. 函数 - 记录的 I/O - 一般信息和 I/O 错误例程

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
VAGen 记录 I/O: <ul style="list-style-type: none"> <li>I/O 选项</li> <li>I/O 对象 (始终是记录)</li> <li>I/O 错误例程 (可选)</li> </ul>	EGL 记录 I/O: <ul style="list-style-type: none"> <li>I/O 语句</li> <li>记录名</li> <li>带有错误例程名的 <i>try onException end</i> 语句 (可选)</li> </ul> 如果指定了 I/O 错误例程, 则语句包含在 <i>try...end</i> 块中。下面是一个带有错误例程的记录 I/O 示例: <pre> <b>try</b>     <b>add</b> recordName ;     [<b>onException</b> error-routine ; ] <b>end</b>                     </pre>	迁移工具执行下列操作: <ul style="list-style-type: none"> <li>将 VAGen I/O 选项更改为对应的 EGL I/O 语句。</li> </ul>
错误例程对于执行记录 I/O 的函数来说是可选的。 注: 如果存在软错误或者 EZEFECC = 1, 则调用该错误例程。	错误例程对于执行记录 I/O 的函数来说是可选的。下面是不带错误例程的 I/O 示例: <pre> <b>add</b> recordName;                     </pre> 下面是带有错误例程的 I/O 示例: <pre> <b>try</b>     <b>add</b> recordName ;     <b>onException</b> error-routine ; <b>end</b>                     </pre> 注: 如果存在软错误或者如果 <i>handleHardIOErrors</i> = 1, 则调用 <i>onException</i> 语句。	迁移工具执行下列操作: <ul style="list-style-type: none"> <li>如果未指定错误例程, 则工具不包括 <i>try</i>、<i>onException</i> 或 <i>end</i> 语句。</li> <li>如果指定了错误例程, 则工具包括 <i>try</i> 和 <i>end</i> 语句。</li> <li>迁移工具根据 VAGen 错误例程名转换为 <i>onException</i> 语句。迁移工具迁移程序时, 它总是将 VAGen 主函数名迁移为主函数标签和主函数调用语句。这样, 当迁移函数的 I/O 错误例程时, <i>mainFunctionLabel</i> 总是与 <i>mainFunctionName</i> 相同。</li> </ul>
错误例程值:  EZECL0S EZEFL0 EZERTN mainFunctionName  nonmainFunctionName	onException 块语句:  <pre> <b>onException</b> <b>exit</b> <b>program</b>; <b>onException</b> <b>exit</b> <b>stack</b>;                     省略 <i>onException</i> 语句。 <b>onException</b> <b>exit</b> <b>stack</b>     mainFunctionLabel; <b>onException</b>     nonmainFunctionName();                     </pre>	对于具有函数名的错误例程的迁移, 有一些特殊的注意事项。请参阅第 65 页的『I/O 错误例程』以了解详细信息和潜在问题。

表 95. 函数 - 串行记录、带索引记录、相对记录和消息队列记录的 I/O 选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: ADD</li> </ul>	使用 <i>add</i> 语句。下面是一个示例: <b>add</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: SCAN</li> </ul>	使用 <i>get next</i> 语句。下面是一个示例: <b>get next</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: SCANBACK</li> </ul>	使用 <i>get previous</i> 语句。下面是一个示例: <b>get previous</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: CLOSE</li> </ul>	使用 <i>close</i> 语句。下面是一个示例: <b>close</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: INQUIRY</li> </ul>	使用 <i>get</i> 语句。下面是一个示例: <b>get</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: UPDATE</li> </ul>	使用 <i>get forUpdate</i> 语句。下面是一个示例: <b>get</b> recordName <b>forUpdate</b> ;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: DELETE</li> </ul>	使用 <i>delete</i> 语句。下面是一个示例: <b>delete</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: REPLACE</li> </ul>	使用 <i>replace</i> 语句。下面是一个示例: <b>replace</b> recordName;	没有特殊的注意事项。

表 96. 函数 - 不带“执行时语句构建”的缺省（未修改的）SQL 语句的 I/O 选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: ADD</li> </ul>	使用 <i>add</i> 语句。下面是一个示例: <b>add</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: SCAN</li> </ul>	使用 <i>get next</i> 语句。下面是一个示例: <b>get next</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: CLOSE</li> </ul>	使用 <i>close</i> 语句。下面是一个示例: <b>close</b> recordName;	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: INQUIRY</li> </ul> （带有以及不带单行选择）	使用 <i>get</i> 语句。如果正在进行单行选择，则还使用 <i>singleRow</i> 。下面是一个不带单行选择的示例: <b>get</b> recordName;  下面是一个带有单行选择的示例: <b>get</b> recordName <b>singleRow</b> ;	如果在 VisualAge Generator 中指定了单行选择，则迁移工具包括 EGL singleRow 选项。

表 96. 函数 - 不带“执行时语句构建”的缺省（未修改的）SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: UPDATE</li> </ul>	<p>使用 <i>get forUpdate</i> 语句。下面是一个示例:</p> <pre>get recordName forUpdate     resultSetID;</pre>	<p>当迁移 SQL UPDATE 语句时, 迁移工具始终包括 <i>resultSetID</i>。工具将 <i>resultSetID</i> 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。</p> <p>当迁移工具不能确定记录是 SQL 记录还是非 SQL 记录时, 有一些特殊的注意事项。请参阅第 70 页的『带有多个更新的 SQL I/O』以了解详细信息和潜在问题。</p>
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: DELETE</li> </ul>	<p>使用 <i>delete</i> 语句。下面是一个示例:</p> <pre>delete recordName;</pre>	<p>没有特殊的注意事项。</p>
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: REPLACE</li> </ul> <p>(带有或不带 UPDATE/SETUPD functionName)</p>	<p>使用 <i>replace</i> 语句。以下是一些示例:</p> <pre>replace recordName; replace recordName from     resultSetID;</pre>	<p>如果在 VisualAge Generator 中包括了 UPDATE/SETUPD 函数名, 则迁移工具包括 <i>resultSetID</i> 并将 <i>resultSetID</i> 设置为后面跟有客户指定的后缀的 UPDATE/SETUPD 函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。</p>
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: SETINQ</li> </ul> <p>(带有以及不带“声明带锁定游标”)</p>	<p>使用 <i>open</i> 语句。如果正在执行声明带锁定游标, 则还使用 <i>hold</i> 选项。下面是这两种语句的一些示例:</p> <pre>open resultSetID for recordName; open resultSetID hold     for recordName;</pre>	<p>迁移工具将 <i>resultSetID</i> 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。</p> <p>如果在 VisualAge Generator 中选择了声明带锁定游标, 则迁移工具包括 EGL <i>hold</i> 选项。</p>
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: SETUPD</li> </ul> <p>(带有以及不带“声明带锁定游标”)</p>	<p>使用 <i>open forUpdate</i> 语句。如果正在执行声明带锁定游标, 则还使用 <i>hold</i> 选项。下面是这两种语句的一些示例:</p> <pre>open resultSetID forUpdate     for recordName; open resultSetID hold forUpdate     for recordName;</pre>	<p>工具将 <i>resultSetID</i> 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。</p> <p>如果在 VisualAge Generator 中选择了声明带锁定游标, 则迁移工具包括 EGL <i>hold</i> 选项。</p>

表 96. 函数 - 不带“执行时语句构建”的缺省（未修改的）SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: SQLEXEC</li> </ul> <p>带有模型 SQL 语句 注:</p> <ul style="list-style-type: none"> <li>SQL 记录名包括在这种格式的 SQLEXEC 中。</li> <li>模型类型的值是:                             <ul style="list-style-type: none"> <li>- 无</li> <li>- 更新</li> <li>- 删除</li> </ul> </li> <li>如果模型类型是“无”，则 VisualAge Generator 不执行任何 I/O。生成仍处理 I/O 错误例程，但不会出错。</li> </ul>	<p>使用 <code>execute</code> 语句。下面是一个示例:</p> <pre>execute modelType for recordName;</pre> <p>注: <code>modelType</code> 是 <code>update</code> 或 <code>delete</code>。</p>	<p>迁移工具根据 VAGen 模型 SQL 语句值来设置 EGL <code>modelType</code>。</p> <p>如果 VAGen 模型 SQL 语句是“无”，则由于 VAGen I/O 语句不执行任何操作，所以迁移工具省略 I/O 语句。迁移工具根据函数的 I/O 错误例程来包括 <code>try</code>、<code>onException</code> 和 <code>end</code> 语句。</p>

表 97. 函数 - 不带“执行时语句构建”的已修改的 SQL 语句的 I/O 选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>已修改的 SQL 语句的一般信息:</p> <ul style="list-style-type: none"> <li>VisualAge Generator 在测试和生成时根据 SQL 行记录构建表子句。表子句如下所示:                             <ul style="list-style-type: none"> <li>- 用于 ADD I/O 选项的 <code>insert into sqlTableName</code></li> <li>- 用于 INQUIRY、UPDATE、SETINQ 和 SETUPD I/O 选项的 <code>from sqlTableName sqlTableLabel</code></li> <li>- 用于 REPLACE I/O 选项的 <code>update sqlTableName</code></li> </ul> </li> <li>根据函数的上次修改时间，其它 SQL 子句可能未存储在函数定义中。如果未存储 SQL 子句，则 VisualAge Generator 根据 I/O 对象的记录定义来创建缺省子句。</li> <li>允许使用 <code>!itemColumnName</code> 变量。这些变量指定 SQL 行记录中的项的名称。在测试或生成时，VisualAge Generator 会将其替换为对应的 SQL 列名。</li> <li><code>sqlClauses</code> 是使用 SQL 语法编写的。</li> </ul>	<p>已修改的 SQL 语句的一般信息:</p> <ul style="list-style-type: none"> <li>如果需要修改任何 SQL 子句，则 EGL 要求显式地指定所有子句。必须将表子句显式地包括在 SQL 语句中。表子句如下所示:                             <ul style="list-style-type: none"> <li>- 用于 add 语句的 <code>insert into sqlTableName</code>。</li> <li>- 用于 get 和 open 语句的 <code>from sqlTableName sqlTableLabel</code>。</li> <li>- 用于 replace 语句的 <code>update sqlTableName</code>。</li> </ul> </li> <li>如果指定了任何 SQL 子句，则 EGL 要求显式地指定所有子句。必需的 SQL 子句随 I/O 类型的不同而有所变化。</li> <li>EGL 要求将 SQL 列名显式地包括在 SQL 语句中。不支持 <code>!itemColumnName</code> 变量。</li> <li><code>sqlClauses</code> 是使用 SQL 语法编写的。</li> </ul>	<p>迁移工具使用 SQL 行记录中的表和表标签来为 EGL I/O 语句构建 <code>tables</code> 子句。表名和表名主变量都包括在 EGL I/O 语句的表子句中。</p> <p>如果必需的 SQL 子句未存储在函数定义中，则迁移工具按照与 VisualAge Generator 相同的方式来根据记录定义创建缺省子句。</p> <p>迁移工具将任何 <code>!itemColumnName</code> 变量转换为它们的对应 SQL 列名。</p> <p>迁移工具将 VAGen 注释 (<code>/*</code>) 转换为 SQL 注释 (<code>--</code>)。</p> <p>如果 SQL 记录及其备用指定记录（如果有的话）在迁移期间不可用，则有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』以了解详细信息和潜在问题。</p>

表 97. 函数 - 不带“执行时语句构建”的已修改的 SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>• I/O 对象: recordName</li> <li>• I/O 选项: ADD</li> </ul> <p>可以修改的子句:</p> <ul style="list-style-type: none"> <li>• 列</li> <li>• VALUES</li> </ul>	<p>使用 <i>add</i> 语句。下面是一个示例:</p> <pre>add recordName with #sql{   insert into   sqlTablename   (columnName1, columnName2)   values   (valueInfo1, valueInfo2) };</pre>	<p>迁移工具根据记录定义中的表名来创建 <i>insert into</i> 子句。有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』以了解详细信息和潜在问题。</p>
<ul style="list-style-type: none"> <li>• I/O 对象: recordName</li> <li>• I/O 选项: INQUIRY</li> </ul> <p>(带有以及不带单行选择)</p> <p>可以修改的子句:</p> <ul style="list-style-type: none"> <li>• SELECT</li> <li>• INTO</li> <li>• WHERE, GROUP BY, HAVING</li> <li>• ORDER BY</li> </ul>	<p>使用 <i>get</i> 语句。</p> <p>下面是不带单行选择的示例:</p> <pre>get recordName singleRow with #sql{   select   Name1,   Name2,   Age   from   sqlTable1 sqlLabel1,   sqlTable2 sqlLabel2   where   Name1 = :NameX   order by   Age } into nameA, nameB, myage;</pre>	<p>如果在 VisualAge Generator 中指定了单行选择, 则迁移工具包括 EGL <i>singleRow</i> 选项。</p> <p>迁移工具根据记录定义中的表名和表标签来创建 <i>from</i> 子句。有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』以了解详细信息和潜在问题。</p>
<ul style="list-style-type: none"> <li>• I/O 对象: recordName</li> <li>• I/O 选项: UPDATE</li> </ul> <p>可以修改的子句:</p> <ul style="list-style-type: none"> <li>• SELECT</li> <li>• INTO</li> <li>• WHERE</li> <li>• FOR UPDATE OF</li> </ul>	<p>使用 <i>get forUpdate</i> 语句。</p> <p>下面是一个示例:</p> <pre>get recordName forUpdate resultsetID with #sql{   select   Name1, Name2, Age   from   sqlTable1 sqlLabel1   where   Name1 = :NameX   for update of   Name2, Age } into Name1, Name2, Age;</pre>	<p>当迁移 SQL 记录的 UPDATE 语句时, 迁移工具总是包括 <i>resultSetID</i>。工具将 <i>resultSetID</i> 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。</p> <p>迁移工具根据记录定义中的表名或表标签来创建 <i>from</i> 子句。有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』以了解详细信息和潜在问题。</p>

表 97. 函数 - 不带“执行时语句构建”的已修改的 SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 选项: REPLACE</li> </ul> ( 可选 UPDATE/SETUPD functionName ) 可以修改的子句: <ul style="list-style-type: none"> <li>SET</li> </ul>	使用 <i>replace</i> 语句。 下面是 <i>replace</i> 语句的一个示例: <pre>replace recordName with #sql{   update   sqlTableName   set   columnName1 = value1,   columnName2 = value2 } from resultSetID;</pre>	如果在 VisualAge Generator 中包括了 UPDATE/SETUPD 函数名, 则迁移工具包括 <i>from resultSetID</i> 子句。迁移工具将 resultSetID 设置为后面跟有客户指定的后缀的 UPDATE/SETUPD 函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。 迁移工具根据记录定义中的表名来创建 <i>update</i> 子句。有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』和第 68 页的『SQL I/O 和缺少的必需 SQL 子句』以了解详细信息和潜在问题。
<ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 对象: SETINQ</li> </ul> ( 带有或者不带“声明带锁定游标” ) <ul style="list-style-type: none"> <li>I/O 对象: recordName</li> <li>I/O 对象: SETINQ</li> </ul> ( 带有或者不带“声明带锁定游标” ) 可以修改的子句: <ul style="list-style-type: none"> <li>SELECT</li> <li>INTO</li> <li>WHERE, GROUP BY, HAVING</li> <li>ORDER BY</li> </ul>	使用 <i>open</i> 语句。如果正在执行声明带锁定游标, 则还使用 <i>hold</i> 选项。 下面是使用 <i>hold</i> 选项的 <i>open</i> 语句的一个示例: <pre>open resultSetID hold with #sql{   select   Name1, Name2   from   sqlTable1 sqlLabel1,   sqlTable2 sqlLabel2   where   Name1 &gt; :Name2   order by   Name1 } into Name1, Name2 for recordName;</pre>	工具将 resultSetID 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。 如果在 VisualAge Generator 中选择了声明带锁定游标, 则迁移工具在 resultSetID 之后包括 EGL <i>hold</i> 选项。 迁移工具根据记录定义中的表名和表标签来创建 <i>from</i> 子句。有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』以了解详细信息和潜在问题。
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: SETUPD</li> </ul> ( 带有或者不带“声明带锁定游标” ) 可以修改的子句: <ul style="list-style-type: none"> <li>SELECT</li> <li>INTO</li> <li>WHERE</li> <li>FOR UPDATE OF</li> </ul>	使用 <i>open forUpdate</i> 语句。下面是一个使用 <i>hold</i> 选项的示例: <pre>open resultSetID hold forUpdate with #sql{   select   Column1, Column2   from   sqlTable1 sqlLabel1,   sqlTable2 sqlLabel2   where   Column1 &gt; :Item1   for update of   Column2 } into Item1, Item2 for recordName;</pre>	迁移工具将 resultSetID 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。 如果在 VisualAge Generator 中选择了声明带锁定游标, 则迁移工具包括 EGL <i>hold</i> 选项。 迁移工具根据记录定义中的表名或表标签来创建 <i>from</i> 子句。有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』以了解详细信息和潜在问题。

表 97. 函数 - 不带“执行时语句构建”的已修改的 SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: SQLEXEC</li> </ul> <p>带有模型 SQL 语句</p> <p>注:</p> <ul style="list-style-type: none"> <li>SQL 记录名在这种格式的 SQLEXEC 中是可选的。</li> <li>模型类型的值如下所示:                             <ul style="list-style-type: none"> <li>- 无</li> <li>- 更新</li> <li>- 删除</li> </ul> </li> </ul>	<p>使用 <i>execute sql</i> 语句。下面是此语句的一个示例:</p> <pre>execute modelType #sql{   UPDATE mysqltable   set Column1 = Column1 * 2   where Column2 = :Column2 }   for recordName;</pre> <p>注: 模型类型的值包括“更新”和“删除”。</p>	<p>迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>将 SQLEXEC 转换为 <i>execute</i> 语句。</li> <li>将 I/O 对象 (如果指定的话) 用作 <i>for</i> 子句中的 <i>recordName</i>。</li> </ul> <p>迁移工具包括 VAGen 模型 SQL 语句值 (如果有的话) 作为对 EGL <i>execute</i> 语句的注释。</p> <p>迁移工具将 VAGen SQLEXEC 子句迁移为 EGL SQL 子句。</p>

表 98. 函数 - 带有“执行时语句构建”的已修改的 SQL 语句的 I/O 选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>“执行时语句构建”只能与下列 I/O 选项配合使用:</p> <ul style="list-style-type: none"> <li>INQUIRY</li> <li>UPDATE</li> <li>SETINQ</li> <li>SETUPD</li> <li>SQLEXEC</li> </ul> <p>通过指定“执行时语句构建”，可以使 VisualAge Generator 在 I/O 语句每次被执行时都动态地准备 SQL 语句。</p>	<p>在 EGL 中，每当您想要动态地准备 SQL 语句时，都可以直接编码 SQL prepare 语句。在 prepare 后面，还必须编码 open、execute 或 get 语句。例如，带有“执行时语句构建”的 VAGen INQUIRY I/O 选项的 EGL 等效项如下所示:</p> <pre>prepare prepID from "sqlStatementString" for recordName; get recordName with prepID into itemList;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>prepare 语句中的 sqlStatementString 是常量和变量的以 SQL 表示法编写的并置字符串。下面是一个既使用列名又使用变量的 where 子句示例:                             <pre>[other clauses] + " where columnName = " + itemName + " AND columnName2 = " + itemName2 + .... [other clauses]</pre> </li> <li>此表其余部分中显示的示例未包括将变量分割到双引号之外。</li> </ul>	<p>迁移工具使用函数中的 SQL 子句以及记录定义中的表名和 / 或表名变量来构建 sqlStatementString。迁移工具按以下方式构建 sqlStatementString:</p> <ul style="list-style-type: none"> <li>执行所有处理，就象是未指定“执行时语句构建”一样，这些处理包括:                             <ul style="list-style-type: none"> <li>- 使用 SQL 行记录中的表名和 / 或表标签来构建 EGL I/O 语句的 tables 子句。表名和表名主变量都包括在 EGL I/O 语句的 tables 子句中。</li> <li>- 根据记录定义来创建必需的缺省子句。</li> <li>- 将任何 !itemColumnName 变量转换为它们的对应 SQL 列名。</li> <li>- 在 prepare 语句中，将 VAGen 注释 (/*) 转换为 EGL 注释 (//)。</li> </ul> </li> </ul> <p>然后，迁移工具执行其它处理来创建 sqlStatementString，包括:</p> <ul style="list-style-type: none"> <li>将常量、列名和 SQL 运算符括在双引号中。</li> <li>将变量放在双引号外部。</li> <li>使用 + 字符串并置运算符来将字符串和变量并置在一起。</li> </ul> <p>如果 SQL 记录及其备用指定记录 (如果有的话) 在迁移期间不可用，则有一些特殊的注意事项。请参阅第 66 页的『SQL I/O 语句』和第 68 页的『SQL I/O 和缺少的必需 SQL 子句』以了解详细信息和潜在问题。</p>

表 98. 函数 - 带有“运行时语句构建”的已修改的 SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: INQUIRY</li> </ul> 带有以及不带单行选择  可以修改的子句: <ul style="list-style-type: none"> <li>SELECT</li> <li>INTO</li> <li>WHERE, GROUP BY, HAVING</li> <li>ORDER BY</li> </ul>	使用 <i>prepare</i> 语句。下面是一个示例: <pre> <b>prepare</b> prepID <b>from</b>   " <b>select</b> columnName "   + ", columnName2 "   + " <b>from</b> table1 t1 "   + "[ <b>where</b> whereClause ]"   + "[<b>order by</b> orderByClause ]"   [ <b>for</b> recordName ]; <b>get</b> recordname <b>with</b> prepID   <b>into</b> itemList;           </pre>	没有特殊的注意事项。
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: UPDATE</li> </ul> 可以修改的子句: <ul style="list-style-type: none"> <li>SELECT</li> <li>INTO</li> <li>WHERE</li> <li>FOR UPDATE OF</li> </ul>	使用 <i>prepare</i> 语句。下面是一个示例: <pre> <b>prepare</b> prepID <b>from</b>   " <b>select</b> columnName "   + ", columnName2 "   + " <b>from</b> table1 t1 "   + "[ <b>where</b> whereClause ]"   + " <b>for Update of</b> columnList "   [ <b>for</b> recordName ]; <b>get</b> recordName <b>forUpdate</b>   resultSetID   <b>with</b> prepID   <b>into</b> itemList;           </pre>	当迁移 SQL 记录的 UPDATE 语句时, 迁移工具总是包括 resultSetID。工具将 resultSetID 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: SETINQ</li> </ul> 带有或者不带“声明带锁定游标”  可以修改的子句: <ul style="list-style-type: none"> <li>SELECT</li> <li>INTO</li> <li>WHERE, GROUP BY, HAVING</li> <li>ORDER BY</li> </ul>	使用 <i>prepare</i> 语句。下面是一个示例: <pre> <b>prepare</b> prepID <b>from</b>   " <b>select</b> columnName "   + ", columnName2 "   + " <b>from</b> table1 t1 "   + "[ <b>where</b> whereClause ]"   + "[<b>order by</b> orderByClause ]"   [ <b>for</b> recordName ]; <b>open</b> resultSetID <b>hold</b>   <b>with</b> prepID   <b>into</b> itemList   [ <b>for</b> recordName ];           </pre>	迁移工具将 resultSetID 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。  如果在 VisualAge Generator 中选择了声明带锁定游标, 则迁移工具在 resultSetID 之后包括 EGL hold 选项。
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: SETUPD</li> </ul> 带有或者不带“声明带锁定游标”  可以修改的子句: <ul style="list-style-type: none"> <li>SELECT</li> <li>INTO</li> <li>WHERE</li> <li>FOR UPDATE OF</li> </ul>	使用 <i>prepare</i> 语句。下面是一个示例: <pre> <b>prepare</b> prepID <b>from</b>   " <b>select</b> columnName "   + ", columnName2 "   + " <b>from</b> table1 t1 "   + "[ <b>where</b> whereClause ] "   + " <b>for update of</b> columnList "   [ <b>for</b> recordName ] ; <b>open</b> resultSetID   [ <b>hold</b> ]   <b>forUpdate</b>   <b>with</b> prepID   <b>into</b> itemList   [ <b>for</b> recordName ] ;           </pre>	工具将 resultSetID 设置为后面跟有客户指定的后缀的函数名。可以使用第 2 阶段 VAGen 迁移语法首选项来控制后缀。  如果在 VisualAge Generator 中选择了声明带锁定游标, 则迁移工具在 resultSetID 之后包括 EGL hold 选项。

表 98. 函数 - 带有“运行时语句构建”的已修改的 SQL 语句的 I/O 选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<ul style="list-style-type: none"> <li>I/O 对象: record</li> <li>I/O 选项: SQLEXEC</li> </ul> 带有模型 SQL 语句 注: <ul style="list-style-type: none"> <li>SQL 记录名在这种格式的 SQLEXEC 中是可选的。</li> <li>模型类型的值如下所示:               <ul style="list-style-type: none"> <li>- 无</li> <li>- 更新</li> <li>- 删除</li> </ul> </li> </ul>	使用 <i>prepare</i> 语句。下面是一个示例: <pre> prepare prepID from   " grant " + group_privileges + " on " + table_name + " to " + userid [ for recordName ] ; execute prepID [ for recordName ] ; // model = type           </pre>	迁移工具包括 VAGen 模型 SQL 语句值 (如果有的话) 作为对 EGL execute 语句的注释。  迁移工具将 VAGen SQLEXEC 子句转换为 EGL SQL 子句。

## 语句

『语句』一节由下列各表组成:

- 一般规则 - 数据项限定和数字文字 (第 228 页的表 99)
- 函数调用 (第 228 页的表 100)
- 赋值、MOVE 和 MOVEA (第 228 页的表 101)
- SET (第 230 页的表 102)
- RETRIEVE 和 FIND (第 232 页的表 103)
- IF、WHILE 和 TEST, 包括 EZE Aid、EZESYS 和 I/O 错误值 (第 233 页的表 104)
- CALL (第 237 页的表 105)
- DXFR (第 237 页的表 106)
- XFER (第 238 页的表 107)

表 99. 语句 - 一般规则 - 数据项限定和数字文字

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>项限定规则: 如果未对某个项进行限定, 则 VisualAge Generator 按以下顺序查找该项:</p> <ul style="list-style-type: none"> <li>函数本地存储器或参数列表中的项。</li> <li>函数本地存储器或参数列表中的函数 I/O 对象和记录。如果项名在此类别中不是唯一的, 则必须对该项名进行限定。</li> <li>程序的主工作存储器记录、被调用参数列表、“表及其它记录”列表以及所有 I/O 对象中的记录、映射和表。如果项名在此类别中不是唯一的, 则必须对该项名进行限定。</li> <li>如果在程序中找不到该项名, 并且程序允许使用隐式项, 则 VisualAge Generator 将根据该项的用法创建数据项定义。</li> </ul>	<p>项限定规则: 如果未对某个项进行限定, 则 EGL 按以下顺序查找该项:</p> <ul style="list-style-type: none"> <li>函数本地存储器或参数列表中的项。</li> <li>函数本地存储器或参数列表中的函数 I/O 语句和记录中使用的记录 and 表单。如果项名在此类别中不是唯一的, 则必须对该项名进行限定。</li> <li>程序的数据声明、使用声明和参数列表中的记录、表单和 dataTable。如果项名在此类别中不是唯一的, 则必须对该项名进行限定。</li> <li>EGL 不允许使用隐式项。必须显式地定义每一个项。</li> </ul>	<p>请参阅第 46 页的『记录中的 Level-77 项』和第 60 页的『程序中的隐式数据项』以了解详细信息和潜在问题。</p>
<p>数字文字:</p> <ul style="list-style-type: none"> <li>未引在引号中。</li> <li>根据本地语言的不同, 可以使用句点 (.) 或逗号 (,) 作为小数点。</li> </ul>	<p>数字文字:</p> <ul style="list-style-type: none"> <li>未引在引号中。</li> <li>必须使用句点作为小数点。在生成时, decimalSymbol 构建描述符选项确定在生成的 Java 或 COBOL 代码中是使用句点还是逗号来作为小数点。</li> </ul>	<p>迁移工具将用作小数点的逗号更改为句点, 但不表对表单变量字段的初始值进行更改。</p>

表 100. 语句 - 函数调用

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 语法示例:</p> <pre>functionName( [argumentList] );</pre>	<p>EGL 语法示例:</p> <pre>functionName( [argumentList] );</pre>	<p>请参阅第 239 页的『EZE 字』以了解等效的 EGL 系统库函数。</p> <p>请参阅第 219 页的表 94 以了解从 I/O 错误例程中进行的函数调用。</p>
<p>在流语句中:</p> <pre>functionName();</pre>	<p>不支持流语句。</p> <pre>goto functionName;</pre>	<p>没有特殊的注意事项。</p>

表 101. 语句 - 赋值、MOVE 和 MOVEA

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 语法示例:</p> <pre>target = functionName       ( [argumentList] );</pre>	<p>EGL 语法示例:</p> <pre>target = functionName       ( [argumentList] );</pre>	<p>请参阅第 239 页的『EZE 字』以了解等效的 EGL 系统库函数。</p>
<pre>target = numericExpression; 或者 target = numericExpression (R;</pre>	<pre>target = numericExpression ; 或者 target = mathLib.round       ( numericExpression ) ;</pre>	<p>如果指定了 (R 选项, 则迁移工具将该选项转换为 EGL round 系统函数。</p>

表 101. 语句 - 赋值、MOVE 和 MOVEA (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>target = source; 或者 MOVE source [ T0 ] target;</p> <p>注:</p> <ul style="list-style-type: none"> <li>target 可以是记录、映射、数据项或某些 EZE 数据字。</li> <li>source 可以是记录、映射、文字、数据项或某些 EZE 数据字。</li> <li>如果 target 是记录或映射, 则 source 必须也是记录或映射。将发生“对应移动”。</li> </ul>	<p>target = source ; 或者 move source to target byName ; 或者 move source to target ;</p> <p>注:</p> <ul style="list-style-type: none"> <li>对于赋值语句来说: <ul style="list-style-type: none"> <li>target 可以是记录、项或某些系统变量。如果 target 是记录, 则 source 必须也是记录; 将把 source 逐字节地移至 target。</li> <li>source 可以是记录、文字、项或某些系统变量。</li> <li>不能在赋值语句中使用表单。</li> <li>对于赋值语句来说, 从来不执行“对应移动”。</li> </ul> </li> <li>对于 move 语句来说: <ul style="list-style-type: none"> <li>target 与 source 可以相同, 这与在 VisualAge Generator 赋值或 MOVE 语句中是一样的。</li> <li>如果指定了 byName, 则 EGL 将执行“对应移动”。</li> <li>如果未指定修饰符, 则根据 source 的部件类型的不同, 则该移动将是项到项的移动或者“对应移动”。</li> </ul> </li> </ul> <p>数据转换和截断规则与 VisualAge Generator 中的数据转换和截断规则相同。</p>	<p>当迁移赋值和 move 语句时, 迁移工具考虑下列 EGL 规则:</p> <ul style="list-style-type: none"> <li>EGL 建议对项到项的移动使用赋值语句。</li> <li>对于涉及记录或表单的移动, 需要使用 move byName 语句, 这样可以保留 VAGen 的对应移动行为。</li> <li>允许使用不带修饰符的 move 并将根据 source 的部件类型将其视为项到项的移动或者“对应移动”。</li> </ul> <p>因此, 迁移工具执行下列操作:</p> <ul style="list-style-type: none"> <li>对于下列任何情况, 转换为赋值语句: <ul style="list-style-type: none"> <li>source 或 target 是 EZE 数据字 (例如: EZEAPP)。</li> <li>source 是文字。</li> <li>source 或 target 是限定的或带下标的项。</li> <li>source 或 target 是函数的参数列表或本地存储器中的项。</li> </ul> </li> <li>如果 source 或 target 是函数的 I/O 对象或者函数参数列表或本地存储器中的记录, 则转换为 move byName。</li> <li>在所有其它情况下, 转换为不带修饰符的 move。</li> </ul> <p>请参阅第 72 页的『赋值语句』以了解详细信息和潜在问题。</p>
<p>MOVEA source [T0] target; 或者 MOVEA source [T0] target FOR occurrence;</p> <p>注: source 可以是数组或标量。</p>	<p>move source to target for all ; 或者 move source to target for occurrence ;</p>	<p>迁移工具将 MOVEA 语句转换为带有 for 修饰符的 move 语句。此工具还执行下列操作:</p> <ul style="list-style-type: none"> <li>如果在 VisualAge Generator 中未指定 FOR occurrence 选项的话, 则包括 for all 选项。</li> <li>如果在 VisualAge Generator 中指定了 FOR occurrence 选项的话, 则包括 for occurrence 选项。</li> <li>如果先前未指定下标的话, 则将 target 的下标设置为 1。</li> <li>由于 source 可以是数组或标量项, 所以不将 source 的下标设置为 1。</li> </ul>

表 102. 语句 - SET

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>一般信息:</p> <ul style="list-style-type: none"> <li>可以使用逗号或空格来将单一 SET 语句上的多个选项分隔开。</li> </ul>	<p>一般信息:</p> <ul style="list-style-type: none"> <li>需要使用逗号来将单一 set 语句上的多个选项分隔开。</li> </ul>	<p>没有特殊的注意事项。</p>
<p>SET record SCAN;</p> <p>或者</p> <p>SET record EMPTY;</p> <p>注: SET record EMPTY 不影响 Level-77 项。</p>	<p><b>set record position;</b></p> <p>或者</p> <p><b>set record empty;</b></p>	<p>迁移工具不为 Level-77 记录添加语句。</p>
<p>SET sqlItem NULL;</p> <p>注: <i>sqlItem</i> 可以是 SQL 行记录中的项或者函数的 SQLITEM 参数。</p>	<p><b>set sqlItemnull ;</b></p> <p>注: <i>sqlItem</i> 可以是 SQL 行记录中的 <i>isNullable=yes</i> 项或者函数的可空参数。</p>	<p>没有特殊的注意事项。</p>
<p>SET map [ ALARM   [ CLEAR   EMPTY ] ] ;</p> <p>注:</p> <ul style="list-style-type: none"> <li>CLEAR 与 EMPTY 是互斥的。</li> <li>ALARM 以及 CLEAR 或 EMPTY 可以与 PAGE 选项配合使用。</li> </ul>	<p><b>set form [alarm   [ initial   empty ] ] ;</b></p> <ul style="list-style-type: none"> <li><i>Initial</i> 与 <i>empty</i> 是互斥的。</li> <li>PAGE 选项的替换项不能与任何其它选项配合使用。</li> </ul>	<p>如果 ALARM、CLEAR 或 EMPTY 与 PAGE 选项配合使用, 则迁移工具将 VAGen 语句分割为两个 EGL 语句。</p>
<p>SET map PAGE ;</p> <p>注: PAGE 可以与 ALARM 以及 CLEAR 或 EMPTY 配合使用。</p>	<p><b>converseLib.clearScreen();</b> // display form</p> <p>或者</p> <p><b>converseLib.pageEject();</b> // printer form</p> <p>注: PAGE 选项的替换项不能与任何其它选项配合使用。</p>	<p>迁移工具按如下方式迁移 SET map PAGE:</p> <ul style="list-style-type: none"> <li>如果 SET map PAGE 与其它任何选项配合使用, 则迁移工具将 VAGen 语句分割为两个 EGL 语句。</li> <li>如果该映射是显示器映射, 则迁移工具将语句转换为 <i>converseLib.clearScreen()</i>;</li> <li>如果该映射是打印机映射, 则此工具将语句转换为 <i>converseLib.pageEject()</i>;</li> <li>如果该映射不可用于确定映射类型, 则迁移工具将语句转换为 <i>converseLib.EZE_SETPAGE()</i>;</li> </ul> <p>请参阅第 73 页的『SET map PAGE 语句』以了解详细信息和潜在问题。</p>

表 102. 语句 - SET (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<pre>SET mapItem [ CURSOR   FULL     [ NORMAL   DEFINED ] ] ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• <i>mapItem</i> 可以是映射中的字段或者函数的 MAPITEM 参数。</li> <li>• NORMAL 与 DEFINED 是互斥的。</li> <li>• CURSOR 和 FULL 可以与 NORMAL 或 DEFINED 配合使用。</li> <li>• VisualAge Generator 允许对打印映射设置 CURSOR、FULL、NORMAL 和 DEFINED, 但这些设置对打印输出没有影响。</li> </ul>	<pre>set formField [ cursor   full     [ normal   initialAttributes ] ] ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• <i>formField</i> 可以是表单上的变量字段或者函数的字段参数。</li> <li>• <i>normal</i> 与 <i>initialAttributes</i> 是互斥的。</li> <li>• <i>cursor</i> 和 <i>full</i> 可以与 <i>normal</i> 或 <i>initialAttributes</i> 配合使用。</li> <li>• EGL 不支持对打印表单设置 <i>cursor</i>、<i>full</i>、<i>normal</i> 或 <i>initialAttributes</i>。</li> </ul>	<p>迁移工具迁移为每个选项的 EGL 等效项, 而不考虑 <i>formField</i> 是在文本表单上还是在打印表单上。请参阅第 74 页的『SET mapItem 属性』以了解详细信息和潜在问题。</p>
<pre>SET mapItem [ CURSOR   FULL     color   extendedHighlight     MODIFIED     [ BRIGHT   DARK ]     [ PROTECT   AUTOSKIP ] ] ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• <i>mapItem</i> 可以是映射中的字段或者函数的 MAPITEM 参数。</li> <li>• BRIGHT 与 DARK 是互斥的。</li> <li>• PROTECT 与 AUTOSKIP 是互斥的。</li> <li>• 可以与任何其它选项配合使用。</li> <li>• VisualAge Generator 允许对打印映射设置这些属性。但是, 只有 USCORE 的扩展突出显示选项才对打印输出有影响。</li> </ul>	<pre>set formField [ cursor   full     color   extendedHighlight     modified     [ bold   invisible ]     [ protect   skip ] ] ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• <i>formField</i> 可以是表单上的变量字段或者函数的字段参数。</li> <li>• <i>Bold</i> 与 <i>invisible</i> 是互斥的。</li> <li>• <i>Protect</i> 与 <i>skip</i> 是互斥的。</li> <li>• 可以与任何其它选项配合使用。</li> <li>• 除了下划线的扩展突出显示选项以外, EGL 不支持对打印表单设置这些属性。</li> </ul>	<p>迁移工具迁移为每个选项的 EGL 等效项, 而不考虑 <i>formField</i> 是在文本表单上还是在打印表单上。请参阅第 74 页的『SET mapItem 属性』以了解详细信息和潜在问题。有关颜色和 <i>extendedHighlight</i> 信息, 请参阅本表中下列各行。</p>
<pre>color: MONO           BLUE           GREEN           PINK           RED           TURQ           YELLOW           WHITE</pre>	<pre>color: defaultColor           blue           green           magenta           red           cyan           yellow           white</pre>	<p>没有特殊的注意事项。</p>
<pre>extendedHighlight:         NOHILITE           BLINK           RVIDEO           USCORE</pre>	<pre>extendedHighlight:         noHighLight           blink           reverse    underline</pre>	<p>没有特殊的注意事项。</p>

表 103. 语句 - RETRIEVE 和 FIND

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<pre>RETR dataItem1     table[.searchColumn]     dataItem2     [ returnColumn ] ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• 如果未指定 <i>searchColumn</i>, 则缺省值是表中的第一列。</li> <li>• 如果未指定 <i>returnColumn</i>, 则缺省值是表中的第二列。</li> </ul>	<pre>if (dataItem1 in     dataTable.searchColumn)     dataItem2 =     dataTable.returnColumn[sysVar.arrayIndex]; end</pre> <p>注: <i>searchColumn</i> 和 <i>returnColumn</i> 都是必需的。</p>	<p>迁移工具将 RETR 语句转换为一个 <i>if</i> 语句和一个赋值语句。</p> <p>如果该表在迁移期间不可用, 则有一些特殊的注意事项。请参阅第 73 页的『RETR 语句』以了解详细信息和潜在问题。</p>
<pre>FIND dataItem     table[.searchColumn]     trueStatement;</pre> <p>或者</p> <pre>FIND dataItem     table[.searchColumn]     , falseStatement ;</pre> <p>或者</p> <pre>FIND dataItem     table[.searchColumn]     trueStatement     [,] falseStatement ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• 如果未指定 <i>searchColumn</i>, 则缺省值是表中的第一列。</li> <li>• 如果在程序流中使用 FIND, 则 <i>trueStatement</i> 和 <i>falseStatement</i> 可以是主函数的名称或 EZECLOS。</li> <li>• 如果在函数中使用 FIND, 则 <i>trueStatement</i> 和 <i>falseStatement</i> 可以是任何函数的名称、EZECLOS、EZEFLO 或 EZERTN。</li> </ul>	<pre>if (dataItem in dataTable.searchColumn)     EGLtrueStatement ; end</pre> <p>或者</p> <pre>if (dataItem in dataTable.searchColumn) else     EGLfalseStatement ; end</pre> <p>或者</p> <pre>if (dataItem in dataTable.searchColumn)     EGLtrueStatement ; else     EGLfalseStatement ; end</pre> <p>注: <i>searchColumn</i> 是必需的。</p>	<p>迁移工具将 FIND 语句转换为 <i>if</i> 语句以及 <i>true</i> 和 <i>false</i> 语句的 EGL 等效项。有关 <i>trueStatement</i> 和 <i>falseStatement</i> 到对应的 EGL 语句的转换, 请参阅本表下列各行。</p>
<p>流中的 <b>true/falseStatement:</b></p> <ul style="list-style-type: none"> <li>• functionName() (仅用于主程序)</li> <li>• EZECLOS</li> </ul>	<p>对应的 EGL 替换项:</p> <ul style="list-style-type: none"> <li>• <b>goto</b> functionName;</li> <li>• <b>exit program;</b></li> </ul>	<p>没有特殊的注意事项。</p>
<p>函数中的 <b>true/falseStatement:</b></p> <ul style="list-style-type: none"> <li>• functionName (任何函数)</li> <li>• EZECLOS</li> <li>• EZEFLO</li> <li>• EZERTN</li> </ul>	<p>对应的 EGL 替换项:</p> <ul style="list-style-type: none"> <li>• functionName();</li> <li>• <b>exit program;</b></li> <li>• <b>exit stack;</b></li> <li>• <b>return;</b></li> </ul>	<p>没有特殊的注意事项。</p>

表 104. 语句 - IF、WHILE 和 TEST, 包括 EZEAIID、EZESYS 和 I/O 错误值

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<pre>IF logicalExpression ;   { statement ; } [ ELSE;   { statement ; } ] END;</pre>	<pre>if ( EGLLogicalExpression )   { EGLStatement ; } [ else   { EGLStatement ; } ] end</pre>	<p>请参阅本表下列各行以了解 VAGen 逻辑表达式与 EGL 逻辑表达式之间的关系。</p>
<pre>WHILE logicalExpression ;   { statement ; } END;</pre>	<pre>while ( EGLLogicalExpression )   { EGLStatement ; } end</pre>	<p>请参阅本表下列各行以了解 VAGen 逻辑表达式与 EGL 逻辑表达式之间的关系。</p>
<pre>TEST testCondition   trueStatement ;  TEST testCondition   , falseStatement ;  TEST testCondition   trueStatement   [, ] falseStatement ;</pre> <p>注:</p> <ul style="list-style-type: none"> <li>• TEST 语句与 IF ... IS 语句类似。有一点例外, 即 TEST mapItem nnn、+nnn 和 -nnn 没有 IF 语句等效项。</li> <li>• 如果在程序流中使用 TEST, 则 trueStatement 和 falseStatement 可以是主函数的名称或 EZECLOS。</li> <li>• 如果在函数中使用 TEST, 则 trueStatement 和 falseStatement 可以是任何函数的名称、EZECLOS、EZEFL0 或 EZERTN。</li> </ul>	<pre>if ( EGLLogicalExpression )   EGLtrueStatement ; end  if ( EGLLogicalExpression ) else   EGLfalseStatement ; end  if ( EGLLogicalExpression )   EGLtrueStatement ; else   EGLfalseStatement ; end</pre>	<p>除了 TEST mapItem nnn、+nnn 和 -nnn 以外, 迁移工具将 TEST 语句转换为等效的 if ... is 语句以及 true 和 false 语句的 EGL 等效项。</p> <p>请参阅本表下列各行以了解 VAGen 逻辑表达式与 EGL 逻辑表达式之间的关系。</p> <p>有关 trueStatement 和 falseStatement 到对应的 EGL 语句的转换, 请参阅本表下列各行。</p>
<p>IF 和 WHILE 的 VisualAge Generator 布尔运算符:</p> <ul style="list-style-type: none"> <li>• AND</li> <li>• OR</li> </ul>	<p>if 和 while 的相应 EGL 布尔运算符:</p> <ul style="list-style-type: none"> <li>• &amp;&amp;</li> <li>•   </li> </ul>	<p>没有特殊的注意事项。</p>
<p>IF 和 WHILE 的 VisualAge Generator 关系运算符:</p> <ul style="list-style-type: none"> <li>• EQ 和 =</li> <li>• NE 和 ^=</li> <li>• LE、&lt;= 和 =&lt;</li> <li>• LT 和 &lt;</li> <li>• GE、&gt;= 和 =&gt;</li> <li>• GT 和 &gt;</li> </ul>	<p>if 和 while 的相应 EGL 关系运算符:</p> <ul style="list-style-type: none"> <li>• ==</li> <li>• !=</li> <li>• &lt;=</li> <li>• &lt;</li> <li>• &gt;=</li> <li>• &gt;</li> </ul>	<p>没有特殊的注意事项。</p>

表 104. 语句 - IF、WHILE 和 TEST, 包括 EZEAIID、EZESYS 和 I/O 错误值 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
IF 和 WHILE 的 VisualAge Generator 状态运算符: <ul style="list-style-type: none"> <li>IS</li> <li>NOT</li> </ul>	if 和 while 的相应 EGL 状态运算符: <ul style="list-style-type: none"> <li>is</li> <li>not</li> </ul>	迁移工具总是将 VAGen TEST 语句迁移为 EGL if ... is 语句。
IF 和 WHILE 的 VisualAge Generator 数组运算符: <ul style="list-style-type: none"> <li>IN</li> </ul>	if 和 while 的相应 EGL 状态运算符: <ul style="list-style-type: none"> <li>in</li> </ul>	有一些特殊的注意事项。请参阅第 75 页的『检查 IN 文字或标量』。
VisualAge Generator <i>mapItem</i> 状态条件: <ul style="list-style-type: none"> <li>BLANK 或 BLANKS</li> <li>CURSOR</li> <li>DATA</li> <li>MODIFIED</li> <li>NULL 或 NULLS</li> <li>NUMERIC</li> </ul>	对应的 EGL <i>formField</i> 状态条件: <ul style="list-style-type: none"> <li>blanks</li> <li>cursor</li> <li>data</li> <li>modified</li> <li>blanks</li> <li>numeric</li> </ul>	迁移工具转换为等效的 EGL 状态条件。  对于 <i>mapItem</i> NULL, 有一些特殊的注意事项。请参阅第 76 页的『检查 SQL 项和映射项是否为 NULL』以了解详细信息和潜在问题。
TEST 语句的特殊 <i>mapItem</i> 状态条件: <code>nnn   +nnn   -nnn</code> <b>注:</b> 这将把用户输入的数据的长度与 <code>nnn</code> 进行比较。测试是 =、> 或 <, 它们对应于 <code>nnn</code> 、 <code>+nnn</code> 或 <code>-nnn</code> 。	EGL 未提供对此状态条件的直接支持。但是, 可以通过执行下列操作实现相同的功能: <ul style="list-style-type: none"> <li>使用系统库函数 <i>converseLib.fieldInputLength</i>, 此函数返回用户输入的数据的长度。</li> <li>使用 if 语句来比较 ==、&gt; 或 &lt; (它们分别与 <code>nnn</code>、<code>+nnn</code> 或 <code>-nnn</code> 对应) 的结果长度。</li> </ul>	当迁移任何程序时, 迁移工具始终包括下面这一项的声明: <code>&lt;custPrefix&gt;EZE_ITEMLEN</code>  迁移工具对 TEST <code>nnn</code> 、 <code>+nnn</code> 或 <code>-nnn</code> 执行下列操作: <ul style="list-style-type: none"> <li>在 TEST 语句正前面添加一个附加的语句以使用系统库函数 <i>converseLib.fieldInputLength(item)</i> 来设置  <code>&lt;custPrefix&gt;EZE_ITEMLEN</code>                      。</li> <li>将 TEST 语句更改为 if 语句并将  <code>&lt;custPrefix&gt;EZE_ITEMLEN</code>                      与 == <code>nnn</code>、&gt; <code>nnn</code> 或 &lt; <code>nnn</code> 进行比较。</li> </ul>
VisualAge Generator 映射状态条件: <ul style="list-style-type: none"> <li>MODIFIED</li> </ul>	对应的 EGL 表单状态条件: <ul style="list-style-type: none"> <li>modified</li> </ul>	没有特殊的注意事项。
VisualAge Generator EZEAIID 状态条件: <ul style="list-style-type: none"> <li>ENTER</li> <li>BYPASS</li> <li>PA<math>n</math>, 其中 <math>n = 1, 2, 3</math></li> <li>PF<math>n</math>, 其中 <math>n</math> 是 1 到 24</li> <li>PA</li> <li>PF</li> </ul>	对应的 EGL <i>converseVar.eventKey</i> 状态条件: <ul style="list-style-type: none"> <li>enter</li> <li>bypass</li> <li>pa<math>n</math>, 其中 <math>n = 1, 2, 3</math></li> <li>pf<math>n</math>, 其中 <math>n</math> 是 1 到 24</li> <li>pakey</li> <li>pfkey</li> </ul>	没有特殊的注意事项。

表 104. 语句 - IF、WHILE 和 TEST, 包括 EZEAIID、EZESYS 和 I/O 错误值 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
VisualAge Generator <i>sqlItem</i> 状态条件: <ul style="list-style-type: none"> <li>• BLANK 或 BLANKS</li> <li>• NULL</li> <li>• NUMERIC</li> <li>• TRUNC</li> </ul>	对应的 EGL <i>sqlItem</i> 状态条件: <ul style="list-style-type: none"> <li>• blanks</li> <li>• null</li> <li>• numeric</li> <li>• trunc</li> </ul>	迁移工具转换为等效的 EGL 状态条件。  对于 <i>sqlItem</i> NULL, 有一些特殊的注意事项。请参阅第 76 页的『检查 SQL 项和映射项是否为 NULL』以了解详细信息和潜在问题。
VisualAge Generator 记录状态条件: <ul style="list-style-type: none"> <li>• DED</li> <li>• DUP</li> <li>• EOF</li> <li>• ERR</li> <li>• FMT</li> <li>• FNA</li> <li>• FNF</li> <li>• FUL</li> <li>• HRD</li> <li>• LOK</li> <li>• NRF</li> <li>• UNQ</li> </ul> 注: <ul style="list-style-type: none"> <li>• SQL 记录和非 SQL 记录都支持 DUP。</li> <li>• 对于 SQL 记录, DUP 和 UNQ 是相等的, 并且它们总是硬错误。</li> <li>• 对于非 SQL 记录, DUP 和 UNQ 不相等, 两者都是软错误。</li> <li>• 仅在 OS/400 上才支持 LOK, 并且 LOK 是软错误。</li> </ul>	对应的 EGL 记录状态条件: <ul style="list-style-type: none"> <li>• deadLock</li> <li>• duplicate 或 unique</li> <li>• endOfFile</li> <li>• ioError</li> <li>• invalidFormat</li> <li>• fileNotAvailable</li> <li>• fileNotFound</li> <li>• full</li> <li>• hardIOError</li> <li>• deadLock</li> <li>• noRecordFound</li> <li>• unique</li> </ul> 注: <ul style="list-style-type: none"> <li>• 只有非 SQL 记录才支持 <i>duplicate</i>, 并且它是软错误。</li> <li>• 对于 SQL 记录和非 SQL 记录, <i>unique</i> 是硬错误。</li> <li>• 将把 LOK 转换为 <i>deadLock</i>, 后者总是硬错误。</li> </ul>	迁移工具转换为等效的 EGL 状态条件。  对于基于记录类型的 DUP 迁移, 有一些特殊的注意事项。请参阅第 77 页的『I/O 错误值 UNQ 和 DUP』以了解详细信息和潜在问题。  对于迁移 LOK, 也有一些特殊的注意事项。请参阅第 78 页的『I/O 错误值 LOK』以了解详细信息和潜在问题。
用户界面记录状态条件: <ul style="list-style-type: none"> <li>• MODIFIED</li> </ul>	对应的 EGL <i>uiRecord</i> 条件: <ul style="list-style-type: none"> <li>• modified</li> </ul>	此发行版的 EGL 不支持 Web 事务或用户界面记录。但是, 预计对映射和用户界面记录进行的用户界面记录状态 MODIFIED 测试是相同的。迁移工具转换一个测试 MODIFIED 的逻辑表达式, 而不考虑所测试的部件是映射还是用户界面记录。这样可以最大程度地保留逻辑。
VisualAge Generator <i>dataItem</i> 状态条件: <ul style="list-style-type: none"> <li>• BLANK 或 BLANKS</li> <li>• NUMERIC</li> </ul>	对应的 EGL <i>dataItem</i> 状态条件: <ul style="list-style-type: none"> <li>• blanks</li> <li>• numeric</li> </ul>	没有特殊的注意事项。

表 104. 语句 - IF、WHILE 和 TEST, 包括 EZEAIID、EZESYS 和 I/O 错误值 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
VisualAge Generator EZESYS 状态条件: <ul style="list-style-type: none"> <li>• AIX</li> <li>• AIXCICS</li> <li>• HP</li> <li>• IMSBMP</li> <li>• IMSVS</li> <li>• MVSBATCH</li> <li>• MVSCICS</li> <li>• NTCICS</li> <li>• OS2</li> <li>• OS2CICS</li> <li>• OS2GUI</li> <li>• OS400</li> <li>• SCO</li> <li>• SOLACICS</li> <li>• SOLARIS</li> <li>• TSO</li> <li>• VMCMS</li> <li>• VMBATCH</li> <li>• VSEBATCH</li> <li>• VSECICS</li> <li>• WINGUI</li> <li>• WINNT</li> <li>• ITF</li> </ul>	对应的 <code>sysVar.systemType</code> 状态条件: <ul style="list-style-type: none"> <li>• aix</li> <li>• AIXCICS</li> <li>• hpux</li> <li>• imsbmp</li> <li>• imsvs</li> <li>• zosbatch</li> <li>• zoscics</li> <li>• NTCICS</li> <li>• OS2</li> <li>• OS2CICS</li> <li>• OS2GUI</li> <li>• iseriesc</li> <li>• SCO</li> <li>• SOLACICS</li> <li>• solaris</li> <li>• TSO</li> <li>• VMCMS</li> <li>• VMBATCH</li> <li>• vsebatch</li> <li>• vsecics</li> <li>• WINGUI</li> <li>• win</li> <li>• debug</li> </ul>	迁移工具转换为等效的 EGL 状态条件。  对于 EZESYS 状态检查, 有一些特殊的注意事项。请参阅第 80 页的『EZESYS』以了解详细信息和潜在问题。 注: 此发行版并不支持所有的 VAGen 运行时环境。但是, 迁移工具总是转换为相等的值, 即使该值在 EGL 中无效。如果旧的 VAGen 值当前不受支持, 在“问题”视图中将有一个错误。
流中的 <b>true/falseStatement</b> : <ul style="list-style-type: none"> <li>• functionName() (仅用于主程序)</li> <li>• EZECLAS</li> </ul>	对应的 EGL 替换项: <ul style="list-style-type: none"> <li>• <b>goto</b> functionName ;</li> <li>• <b>exit program</b>;</li> </ul>	没有特殊的注意事项。
函数中的 <b>true/falseStatement</b> : <ul style="list-style-type: none"> <li>• functionName (任何函数)</li> <li>• EZECLAS</li> <li>• EZEFLD</li> <li>• EZERTN</li> </ul>	对应的 EGL 替换项: <ul style="list-style-type: none"> <li>• functionName();</li> <li>• <b>exit program</b>;</li> <li>• <b>exit stack</b>;</li> <li>• <b>return</b>;</li> </ul>	没有特殊的注意事项。

表 105. 语句 - CALL

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>CALL <i>programName</i> <i>argument</i>                      [ { [,] <i>argument</i> } ]                      [ (<i>options</i>) ] ;</p> <p>或者</p> <p>CALL <i>serviceRoutine</i> <i>argument</i>                      [ { [,] <i>argument</i> } ]                      [ (<i>options</i>) ] ;</p> <p>注：用来分隔自变量的逗号是可选的。</p>	<p><b>call</b> <i>programName</i> <i>argument</i>                      [ { , <i>argument</i> } ]                      [ <i>options</i> ] ;</p> <p>注：</p> <ul style="list-style-type: none"> <li>用来分隔自变量的逗号是必需的。</li> <li><i>programName</i> 不能是保留字。如果程序是非 EGL 程序，则使用链接表条目来指定实名。</li> </ul>	<p>有关这些选项到对应的 EGL 语句或选项的转换，请参阅本表下列各行。</p> <p>有关为它们迁移 CALL 语句的信息，请参阅第 248 页的『服务例程』。</p>
REPLY 选项	<p>如果在 VisualAge Generator 中指定了 REPLY 选项，则对应的 EGL 语句如下：</p> <p><b>try</b>                      call <i>programName</i> <i>argument</i>                      [ { , <i>argument</i> } ]                      [ <i>otherOptions</i> ] ;</p> <p><b>end</b></p>	如果指定了 REPLY 选项，则迁移工具包括 <i>try...end</i> 块。
<p>otherOptions:</p> <ul style="list-style-type: none"> <li>NOMAPS</li> <li>NONCSP</li> </ul>	<p>对应的 EGL otherOptions:</p> <ul style="list-style-type: none"> <li>noRefresh</li> <li>externallyDefined</li> </ul>	没有特殊的注意事项。

表 106. 语句 - DXFR

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>DXFR <i>target</i>                      [ <i>recordName</i> ]                      [ (NONCSP) ] ;</p> <p>其中, <i>target</i> 是</p> <p><i>programName</i></p> <p>或者</p> <p>EZEAPP</p> <p>注：</p> <ul style="list-style-type: none"> <li>可以传递任何记录。</li> <li>如果传递了工作存储器记录，则不包括任何 Level-77 项。</li> </ul>	<p><b>transfer to program</b> <i>target</i>                      [ <b>passing</b> <i>recordName</i> ]                      [ <b>externallyDefined</b> ] ;</p> <p>其中, <i>target</i> 是</p> <p><i>programName</i></p> <p>或者</p> <p><i>sysVar.transferName</i></p> <p>注：可以传递任何记录。</p>	没有特殊的注意事项。

表 107. 语句 - XFER

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><b>变体 1 - 迁移为 Transfer (不带映射或用户界面记录)</b></p> <pre>XFER target   [ recordName ]   [ (NONCSP) ;</pre> <p>其中, target 是</p> <p>transactionName</p> <p>或者</p> <p>sysVar.transferName</p> <p>EZEAPP</p> <p><b>注:</b></p> <ul style="list-style-type: none"> <li>• 这种格式的 XFER 不包括映射或用户界面记录。</li> <li>• 可以传递任何记录。如果传递了工作存储器记录, 则不包括任何 Level-77 项。</li> <li>• transactionName 是非事务运行时环境中的 programName。</li> </ul>	<p>transfer 语句的 EGL 语法:</p> <pre>transfer to transaction target   [ passing recordName ]   [ externallyDefined ] ;</pre> <p>其中, target 是</p> <p>transactionName</p> <p>或者</p> <p>sysVar.transferName</p> <p><b>注:</b></p> <ul style="list-style-type: none"> <li>• 可以传递任何记录。</li> <li>• transactionName 是非事务运行时环境中的程序名。</li> </ul>	<p>如果语句中没有逗号, 则迁移工具将 XFER 转换为 EGL transfer to transaction 语句。</p>
<p><b>变体 2 - 迁移为 Show (带有映射的 XFER)</b></p> <pre>XFER target   [ recordName ]   , map   [ (NONCSP) ;</pre> <p>其中, target 是</p> <p>transactionName</p> <p>或者</p> <p>sysVar.transferName</p> <p>EZEAPP</p> <p><b>注:</b></p> <ul style="list-style-type: none"> <li>• 可以传递任何记录。如果传递了工作存储器记录, 则不包括任何 Level-77 项。</li> <li>• transactionName 是非事务目标环境中的 programName。</li> </ul>	<p>show 语句的 EGL 语法:</p> <pre>show formName   returning to target   [ passing recordName ]   [ externallyDefined ] ;</pre> <p>其中, target 是</p> <p>transactionName</p> <p>或者</p> <p>sysVar.transferName</p> <p><b>注:</b></p> <ul style="list-style-type: none"> <li>• 可以传递任何记录。</li> <li>• transactionName 是非事务目标环境中的 programName。</li> </ul>	<p>迁移工具按照以下方式区分变体 2 (带有映射的 XFER) 和变体 3 (带有用户界面记录的 XFER):</p> <ul style="list-style-type: none"> <li>• 如果下列任何一项成立, 则使用变体 2 (带有映射的 XFER):             <ul style="list-style-type: none"> <li>- 指定了 (NONCSP)。</li> <li>- 第二个自变量是映射。</li> </ul> </li> <li>• 如果下列任何一项成立, 则使用变体 3 (带有用户界面记录的 XFER):             <ul style="list-style-type: none"> <li>- 目标是 ' '。</li> <li>- 第二个自变量的名称的长度超过 8 个字符。</li> <li>- 第二个自变量是用户界面记录。</li> </ul> </li> <li>• 如果迁移工具无法确定第二个自变量是映射还是用户界面记录, 则此工具使用变体 3。</li> </ul> <p>如果迁移工具无法区分映射和用户界面记录, 则有一些特殊的注意事项。请参阅第 79 页的『XFER』以了解详细信息和潜在问题。</p>

表 107. 语句 - XFER (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>变体 3 - 迁移为 <b>Show</b> (带有用户界面记录的 <b>XFER</b>)</p> <p>XFER target [ recordName ]                   , UIRecord</p> <p>其中, target 是</p> <p>transactionName</p> <p>或者</p> <p>EZEAPP</p> <p>或者</p> <p>’, ’</p> <p>注:</p> <ul style="list-style-type: none"> <li>• 可以传递任何记录。如果传递了工作存储器记录, 则不包括任何 Level-77 项。</li> <li>• <i>transactionName</i> 是非事务目标环境中的 <i>programName</i>。</li> </ul>	<p><i>forward</i> 语句的 EGL 语法:</p> <pre>forward UIRecord   [ <b>returning to</b> target ]   [ <b>passing</b> recordName ] ;</pre> <p>其中, target 是</p> <p>transactionName</p> <p>或者</p> <p>sysVar.transferName</p> <p>或者</p> <p>用户界面记录</p> <p>注:</p> <ul style="list-style-type: none"> <li>• 可以传递任何记录。</li> <li>• <i>transactionName</i> 是非事务目标环境中的 <i>programName</i>。</li> </ul>	<p>此发行版的 EGL 不支持 Web 事务、用户界面记录或者替换带有用户界面记录的 XFER。但是, <i>forward</i> 语句可以用来替换带有用户界面记录的 XFER。EGL 语法可能不正确。但是, 使用这种迁移技术可以最大程度地保留逻辑。迁移工具按照以下方式区分变体 2 (带有映射的 XFER) 和变体 3 (带有用户界面记录的 XFER):</p> <ul style="list-style-type: none"> <li>• 如果下列任何一项成立, 则使用变体 2 (带有映射的 XFER):             <ul style="list-style-type: none"> <li>- 指定了 (NONCSP。</li> <li>- 第二个自变量是映射。</li> </ul> </li> <li>• 如果下列任何一项成立, 则使用变体 3 (带有用户界面记录的 XFER):             <ul style="list-style-type: none"> <li>- 目标是 ’ ’。</li> <li>- 第二个自变量的名称的长度超过 8 个字符。</li> <li>- 第二个自变量是用户界面记录。</li> </ul> </li> <li>• 如果迁移工具无法确定第二个自变量是映射还是用户界面记录, 则此工具使用变体 3。</li> </ul> <p>如果迁移工具无法区分映射和用户界面记录, 则有一些特殊的注意事项。请参阅第 79 页的『XFER』以了解详细信息和潜在问题。</p>

## EZE 字

注: 在这些表中, 左列中的 VAGen EZE 字与右列中它们的等效 EGL 字匹配。

### 程序流 EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。

表 108. 程序流 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL
<p>EZECLOS</p>	<p>这取决于位置:</p> <ul style="list-style-type: none"> <li>• 如果用作 I/O 错误例程, 则迁移工具将 EZECLOS 转换为 try...end 块中的以下内容: onException exit program;</li> <li>• 在别处使用时, 包括用作 TEST 或 FIND 的 true 或 false 操作数时, 迁移工具将 EZECLOS 转换为以下内容: exit program;</li> </ul> <p>注: 出口程序的缺省返回码为 &lt;ezeDataPrefix&gt;.returnCode, 该返回码等同于 EZERCODE。这个缺省值提供了与 VisualAge Generator 相同的功能。</p>
<p>EZEFLO</p> <p>注: 不能在流语句中使用 EZEFLO。</p>	<p>这取决于位置:</p> <ul style="list-style-type: none"> <li>• 如果用作 I/O 错误例程, 则迁移工具将 EZEFLO 转换为 try...end 块中的以下内容: onException exit stack;</li> <li>• 在别处使用时, 包括用作 TEST 或 FIND 的 true 或 false 操作数时, 迁移工具将 EZEFLO 转换为以下内容: exit stack;</li> </ul>
<p>EZERTN 或 EZERTN(return value)</p> <p>注:</p> <ul style="list-style-type: none"> <li>• 不能在流语句中使用 EZERTN。</li> <li>• 不能将 EZERTN(return value) 用作 I/O 错误例程。</li> </ul>	<p><b>EZERTN</b> - 这取决于位置:</p> <ul style="list-style-type: none"> <li>• 如果用作 I/O 错误例程, 则迁移工具包括 try...end 块, 但省略 onException 语句。</li> <li>• 在别处使用时, 迁移工具将 EZERTN 转换为以下内容: return;</li> </ul> <p>或者</p> <p>return(returnValue);</p> <p>注: 如果 returnValue 是 EZESYS, 则请参阅 EZESYS 以了解其它注意事项。</p>

## SQL EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。

表 109. SQL EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZECONCT	VGLib.connectionService  自变量与 VAGen 中的自变量相同。但是，对于调试和 Java 生成，并非工作单元自变量的所有值都受支持。JDBC 仅支持单阶段落实。
EZESQCOD	sysVar.sqlcode
EZESQISL 注： • 对于 VisualAge Generator 4.5，支持将 EZESQISL 与 ODBC 配合使用。 • 否则，对于所有环境，在 VisualAge Generator 中或者不支持或者忽略 EZESQISL，但为了兼容而将其保留。	VGVar.sqlIsolationLevel
EZESQLCA	sysVar.sqlca 注：在 EGL 中，仅部分支持 sysVar.sqlca。对于调试以及 Java 生成，EGL 不设置在 sysVar.sqlca 中包含 VGVar.sqlerrmc 值和 VGVar.sqlwarn[7] 值的字段。
EZESQRD3	VGVar.sqlerrd[3] 注：迁移工具将此内容更改为数组引用。
EZESQRRM	VGVar.sqlerrmc 注：调试或 Java 生成不支持 sqlerrmc。
EZESQWN1	VGVar.sqlwarn[2] 注：迁移工具将此内容更改为数组引用。
EZESQWN6	VGVar.sqlwarn[7] 注：迁移工具将此内容更改为数组引用。调试或 Java 生成不支持 sqlwarn[7]。
无	VGVar.sqlState 注：这是 EGL 的新增内容，它在 VisualAge Generator 4.5 中没有等效的项。迁移工具不将任何内容转换此内容。

## 日期和时间 EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。

表 110. 日期和时间 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEDAY	VGVar.currentShortJulianDate
EZEDAYL	VGVar.currentJulianDate
EZEDAYLC	VGVar.currentFormattedJulianDate
EZEDTE	VGVar.currentShortGregorianDate
EZEDTEL	VGVar.currentGregorianDate

表 110. 日期和时间 EZE 字 (续)

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEDTELC	VGVar.currentFormattedGregorianCalendar
EZETIM	VGVar.currentFormattedTime

## 其它数据 EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。

表 111. 其它数据 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEALD	converseVar.eventKey
EZEAPP	sysVar.transferName
EZECNVCM	converseVar.commitOnConverse
EZECONVT	sysVar.callConversionTable
record.EZEDEST	record.resourceAssociation 注: 限定仍然是记录名。
EZEDESTP	converseVar.printerAssociation
EZEFEC	VGVar.handleHardIOErrors
EZELOC	sysVar.remoteSystemID
EZELTERM	sysVar.terminalID
EZEMNO	<ul style="list-style-type: none"> <li>• 如果将 EZEMNO 用作 MOVE 或赋值语句的目标, 则将发生下列情况: <ul style="list-style-type: none"> <li>- 如果 EZEMNO 是根据除 9999 以外的数字文字设置的, 则 EZEMNO 变为: converseLib.validationFailed(numericLiteral);</li> <li>- 如果 EZEMNO 是根据数字文字 9999 设置的, 则 EZEMNO 变为: converseLib.validationFailed();</li> <li>- 如果 EZEMNO 是根据项设置的, 则 EZEMNO 变为: if (itemName == 9999) converseLib.validationFailed(); else converseLib.validationFailed(itemName); end</li> </ul> </li> <li>• 如果在别处使用 EZEMNO, 则将其替换为: converseVar.validationMsgNum</li> </ul>

表 111. 其它数据 EZE 字 (续)

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
<p>EZEMSG</p> <p>注: 仅当 EZEMSG 位于映射中时, 它才作为数据项存在。如果将 EZEMSG 放在多个映射中, 则必须对其进行限定。</p>	<p>&lt;custPrefix&gt;EZEMSG</p> <p>注:</p> <ul style="list-style-type: none"> <li>在 &lt;custPrefix&gt; 与 EZEMSG 之间没有点。</li> <li>当在函数中使用 EZEMSG 时, 迁移工具对 &lt;custPrefix&gt;EZEMSG 保留 EZEMSG 在那些函数中使用的限定。例如, xxxx.EZEMSG 变为 xxxx.&lt;custPrefix&gt;EZEMSG</li> <li>当在映射中使用 EZEMSG 时, 迁移工具执行下列操作: <ul style="list-style-type: none"> <li>将字段名更改为 &lt;custPrefix&gt;EZEMSG</li> <li>将映射的 msgField 属性设置为 &lt;custPrefix&gt;EZEMSG</li> </ul> </li> </ul>
EZEORDER	VGVar.handleOverflow
EZEOVERS	sysVar.overflowIndicator
<p>EZERCODE</p> <p>注: VisualAge Generator 允许 (但不建议) 对 EZERCODE 使用负数值和大于 512 的值。</p>	<p>sysVar.returnValue</p> <p>注: EGL 不允许对 sysVar.returnValue 使用负数值或大于 512 的值。</p>
EZEREPLY	VGVar.handleSysLibraryErrors
<p>EZERT2</p> <p>注: 在 VisualAge Generator 4.5 中, 仅将 EZERT2 用作 MQ 系列访问的条件代码。</p>	VGVar.mqConditionCode
<p>EZERT8</p> <p>注: 为下列各项设置 EZERT8:</p> <ul style="list-style-type: none"> <li>CALL 语句, 如果指定了 (REPLY 选项的话)。</li> <li>EZE 系统函数调用, 如果 EZEREPLY 设置为 1 的话。</li> <li>用于串行记录、带索引记录、相对记录和消息队列记录的 I/O 语句。</li> </ul>	<p>sysVar.errorCode</p> <p>注: 为下列各项设置 sysVar.errorCode:</p> <ul style="list-style-type: none"> <li>所有 CALL 语句。</li> <li>所有 sysLib 系统函数调用。</li> <li>用于串行记录、带索引记录、相对记录和消息队列记录的 I/O 语句。</li> </ul> <p>sysVar.errorCode 值在 EGL 中的更改频率比在 VisualAge Generator 中高。</p>
EZESEGM	converseVar.segmentedMode
EZESEGTR	sysVar.transactionID

表 111. 其它数据 EZE 字 (续)

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZESYS	<p>要在 <i>if</i> 或 <i>while</i> 语句中使用 EGL 值, 使用:  <code>sysVar.systemType</code></p> <p>要获取旧的 VAGen 值以便在任何其它语句中使用, 请使用:  <code>myItem = sysLib.getVAGSysType();</code></p> <p>, 然后在该语句中使用 <i>myItem</i>。</p> <p>如果需要在已迁移的 VAGen 程序中使用旧的 VAGen 值, 则使用:  <code>&lt;custPrefix&gt;EZESYS</code></p> <p>其中, &lt;custPrefix&gt; 是您在第 2 阶段迁移期间指定的重命名前缀。根据 VAGen 迁移首选项不初始化以前的 EZESYS 值, 迁移工具将包括或省略 &lt;custPrefix&gt;EZESYS 的数据声明和用于将其初始化为以前的 VAGen 值的语句。</p> <p>请参阅第 80 页的『EZESYS』以了解详细信息和潜在问题。</p>
EZETST 注: 为 IF...IN 和 MOVEA 设置此项。EZETST 是 2 个字节的二进制。	<p><code>sysVar.arrayIndex</code>  注: arrayIndex 是 int (4 个字节的二进制)。</p>
EZEUSR	<code>sysVar.sessionID</code>
EZEUSRID	<code>sysVar.userID</code>

## 一般函数 EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。除非有所指示, 否则 VisualAge Generator 中的自变量列表与 EGL 中的自变量列表相同, 因此下表省略了自变量列表。

表 112. 一般函数 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
<code>result = EZEBYTES(itemOrRecord)</code> 注: VisualAge Generator 记录只能将项和记录用作 EZEBYTES 的自变量。但是, VisualAge Generator 允许将映射用作 EZEBYTES 的自变量。	<code>result = sysLib.bytes(itemOrRecord)</code> 注: EGL 不支持将映射用作 <code>sysLib.bytes</code> 的自变量。迁移工具在转换自变量时不考虑它是项、记录还是映射。
EZECOMIT()	<code>sysLib.commit()</code>
EZECONV(target,direction,conversionTable)	<code>sysLib.convert</code>
EZEC10(xxx, yyy, zzz)	<code>sysLib.verifyChkDigitMod10</code>
EZEC11(xxx, yyy, zzz)	<code>sysLib.verifyChkDigitMod11</code>
EZEG10(xxx, yyy, zzz)	<code>sysLib.calculateChkDigitMod10</code>

表 112. 一般函数 EZE 字 (续)

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEG11(xxx, yyy, zzz)	sysLib.calculateChkDigitMod11
EZEPURGE(queueName)	sysLib.purge
EZEROLLB()	sysLib.rollback()
EZEWAIT(variableName) 注: <i>variableName</i> 提供以百分之一秒计的时间。	sysLib.wait( <i>variableName</i> ); 注: <ul style="list-style-type: none"> <li>• <i>variableName</i> 提供以秒计的时间。</li> <li>• 迁移工具将时间转换为秒。请参阅第 82 页的『EZEWAIT』以了解详细信息和潜在问题。</li> </ul>

## 字符串 EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。VisualAge Generator 中的自变量列表与 EGL 中的自变量列表相同，因此下表省略了自变量列表。

表 113. 字符串 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZESBLKT	strLib.setBlankTerminator
EZESCCWS	strLib.concatenateWithSeparator
EZESCMPR	strLib.compareStr
EZESCNT	strLib.concatenate
EZESCOPY	strLib.copyStr
EZESFIND	strLib.findStr
EZESNULLT	strLib.setNullTerminator
EZESSET	strLib.setSubStr
EZESTLEN	strLib.strLen
EZESTOKN	strLib.getNextToken

## 数学 EZE 字

表的左列显示 VisualAge Generator 4.5 EZE 字。右列显示迁移工具将该 EZE 字转换成的 EGL 内容。除非有所指示，否则 VisualAge Generator 中的自变量列表与 EGL 中的自变量列表相同，因此下表省略了自变量列表。

表 114. 数学 EZE 字 - 一般数学函数

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEABS	mathLib.abs
EZECEIL	mathLib.ceiling
EZEEXP	mathLib.exp
EZEFLOOR	mathLib.floor
EZEFREXP	mathLib.frexp
EZELDEXP	mathLib.ldexp

表 114. 数学 EZE 字 - 一般数学函数 (续)

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZELOG	mathLib.log
EZELOG10	mathLib.log10
EZEMAX	mathLib.maximum
EZEMIN	mathLib.minimum
EZEMODF	mathLib.modf
EZENCMPR	mathLib.compareNum
EZEPOW	mathLib.pow
EZEPRSCN	mathLib.precision
EZEROUND	mathLib.round 注: mathLib.round 也用来替换带有 (R 选项的 VAGen 语句。那些语句的赋值语句迁移具有以下语法: <pre>result = mathLib.round(numericExpression);</pre>
EZESQRT	mathLib.sqrt

表 115. 数学 EZE 字 - 三角数学函数

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEACOS	mathLib.acos
EZEASIN	mathLib.asin
EZEATAN	mathLib.atan
EZEATAN2	mathLib.atan2
EZECOS	mathLib.cos
EZECOSH	mathLib.cosh
EZESIN	mathLib.sin
EZESINH	mathLib.sinh
EZETAN	mathLib.tan
EZETANH	mathLib.tanh

表 116. 数学 EZE 字 - 浮点数学函数

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEFLADD	mathLib.floatingSum
EZEFLDIV	mathLib.floatingQuotient
EZEFLMOD	mathLib.floatingMod
EZEFLMUL	mathLib.floatingProduct
EZEFLSET	mathLib.floatingAssign
EZEFLSUB	mathLib.floatingDifference

## 用户界面 EZE 字

此发行版的 EGL 不支持 Web 事务或用户界面记录。但是, VisualAge Generator EZEUIxxx 特殊函数字有一些替换项可以在新的 EGL pageHandler 中使用。迁移工具将

EZEUIxxx 特殊函数转换为它们的 EGL 等效项。这样可以最大程度地保留逻辑。如果开发任何新的 EGL pageHandler, 则可能能够使用此函数。

表 117. 用户界面 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZEUIERR	sysLib.setError
EZEUILOC	sysLib.setLocale

## 对象脚本编制 EZE 字

表 118. 对象脚本编制 EZE 字

VisualAge Generator 4.5 中的 EZE 字	EGL 定义
EZESCRPT(targetScriptName)	迁移工具将发出错误消息。无法正确地迁移此函数部件。将把原始 EZESCRPT 语句作为注释包括在 EGL 源代码中。

## DL/I EZE 字

注: 此发行版的 EGL 不支持 DL/I。但是, 如果在函数中使用了任何 EZEDL 字, 则迁移工具将按下表显示的方式迁移 EZEDL 字。迁移工具使用在将来的 EGL 发行版中将用来替换 EZEDL 状态字的内容。这样可以最大程度地保留逻辑。迁移工具还将发出错误消息。在此发行版的 EGL 中, 不能测试、生成或运行任何使用该函数的程序。

当支持 DL/I 时, 此处列示的 EGL 替换可能会有所不同。

表 119. DL/I EZE 字

VisualAge Generator 4.5 中的 EZE 字	在 EGL 的将来发行版中的预期定义
EZEDLCER	dliVar.dliCicsErrorCode
EZEDLCON	dliVar.dliCicsConditionCode
EZEDLDBD	dliVar.dliDbdName
EZEDLERR	dliVar.handleHardDliErrors
EZEDLKEY	dliVar.dliKey
EZEDLKYL	dliVar.dliKeyLength
EZEDLLEV	dliVar.dliLevel
EZEDLPCB 注: 这是缺省下标为 1 的数组。	dliVar.dliPcb 注: 如果未指定下标, 则工具将此内容转换为 dliVar.dliPcb[1]。
EZEDLPRO	dliVar.dliPcbOptions
EZEDLPSB	dliVar.dliPsbName
EZEDLRST	dliVar.dliCicsProgramRestarted
EZEDLSEG	dliVar.dliSegmentName
EZEDLSSG	dliVar.dliSegmentCount
EZEDLSTC	dliVar.dliStatusCode

表 119. DLI EZE 字 (续)

VisualAge Generator 4.5 中的 EZE 字	在 EGL 的将来发行版中的预期定义
EZEDLTRM 注: EZEDLTERM 等同于 EZECNVCM。在迁移时, 它们都将被转换为 commitOnConverse。	converseVar.commitOnConverse 注: 这个前缀用于一般数据 EZE 字, 而非专门用于 DLI。

## 服务例程

『服务例程』一节由下列各表组成:

- 服务例程 - 一般语法 (第 248 页的表 120)
- 服务例程 - VisualAge Generator 和 EGL 等效例程 (第 248 页的表 121)

表 120. 服务例程 - 一般语法

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
CALL serviceRoutine [ argumentList ] ;	syslib.EGLSystemFunction [ ( argumentList ) ] ;  注: EGL 系统函数使用的自变量列表与在 VisualAge Generator 中使用的自变量列表相同。	没有特殊的注意事项。
CALL serviceRoutine [ argumentList ] (REPLY ;	<b>try</b> sysLib.EGLSystemFunction [ ( argumentList ) ] ; <b>end;</b>  注: EGL 系统函数使用的自变量列表与在 VisualAge Generator 中使用的自变量列表相同。	如果在 VisualAge Generator 中包括了 (REPLY 选项, 则迁移工具包括 try... end 块。

表 121. 服务例程 - VisualAge Generator 和 EGL 等效例程

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
CALL AUDIT	sysLib.audit	没有特殊的注意事项。
CALL COMMIT	sysLib.commit	没有特殊的注意事项。
CALL CREATX	sysLib.startTransaction	没有特殊的注意事项。
CALL CSPTDLI	dliLib.callDli 注: 在此发行版中, 不支持 CSPTDLI; 当支持 DLI 时, EGL 替换可能会稍有不同。	迁移工具将该调用转换为 dliLib.callDli 并发出错误消息。不能运行该程序。
CALL EZCHART	<b>call EZCHART</b> [ arguments ] <b>externallyDefined ;</b>  注: 在 EGL 中没有 EZCHART 的替换项。	VAGen 迁移工具将 EZCHART 转换为对外部定义的程序的调用。  如果在 VisualAge Generator 中指定了 REPLY 选项, 则迁移工具将 call 语句嵌套在 try ... end 块中。
CALL RESET	sysLib.rollback	没有特殊的注意事项。

在此发行版的 EGL 中，不支持 PSB。

---

## 控制部件

在 VisualAge Generator 中，控制部件是使用自由格式文本编辑器输入的。在生成期间实际地使用控制部件之前，不会对那些部件进行验证。内容的大小写并不重要。在 EGL 中，控制部件以 XML 表示法存储在 .eglbl 文件中，并且，对于每种类型的控制部件都有一个特殊的编辑器。在 EGL 中，大小写至关重要。本节中的各个表将您在 VisualAge Generator 自由格式文本编辑器中输入的信息与 EGL 中使用的 XML 标记或属性值作比较。这些表仅显示了标记或属性值，而未显示实际的 XML 语法。

### 注:

- 迁移工具尽可能地为现在尚不支持但当前已计划支持的运行时环境迁移生成选项、链接表选项和资源关联选项。这将以尽可能有用的方式最大程度地保留信息。例如，尽管 IMS 当前不是受支持的运行时环境，但是大部分 IMS 生成选项都会被迁移。当您使用一般的 EGL 构建部件编辑器时，这些选项不显示。但是，如果使用文本编辑器来打开该文件，就可以看到这些选项。
- 迁移工具将那些不具有相应 EGL 替换项但在确定 EGL 所需的相关信息方面可能有用的生成选项、链接表选项和资源关联选项作为注释包括在迁移结果中。例如，目前不能迁移用于 IMS 环境的 /MFSDEV 生成选项。迁移工具将该选项作为注释包括在迁移结果中以保留信息供将来使用。当您使用一般的 EGL 构建部件编辑器时，这些注释不显示。但是，如果使用文本编辑器来打开该文件，就可以看到这些注释。
- 如果信息对于确定当前的或将来的 EGL 选项来说没有用的话，迁移工具就会除去不具有相应 EGL 替换项的生成选项。例如，*lineinfo*（这是用来帮助 IBM 支持机构调试 VAGen 生成器的选项）没有替换项。此选项对于 EGL 生成器来说是没有用的，因此，迁移工具不将其作为注释包括在迁移结果中。
- 迁移工具不会将控制部件重命名。对于与 EGL 保留字有冲突的部件名，此工具发出错误消息。

『控制部件』一节由下列各表组成:

- 一般控制部件信息（第 250 页的表 122）
- 生成选项（第 250 页的表 123）
- 生成选项 - 转换表值（第 262 页的表 124）
- :calllink 的链接表选项（第 263 页的表 125）
- :filelink 的链接表选项（第 266 页的表 126）
- :crtxlink 的链接表选项（第 267 页的表 127）
- :dxfrlink 的链接表选项（第 268 页的表 128）
- 资源关联（第 269 页的表 129）
- 链接编辑选项（第 273 页的表 130）
- 绑定控制（第 273 页的表 131）

表 122. 一般控制部件信息

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 控制部件名:</p> <ul style="list-style-type: none"> <li>名称可以包含句点 (.)。</li> <li>对于绑定部件和链接编辑部件来说, 在第一个句点之后的任何名称部分都被看作后缀。可以在 /bind 和 /linkedit 生成选项中指定后缀。</li> </ul>	<p>EGL 构建部件:</p> <ul style="list-style-type: none"> <li>句点 (.) 在构建部件名中是无效的。</li> </ul>	<p>迁移工具将句点 (.) 更改为下划线 (_)。</p>
<p>在 VAGen 控制部件标记和值中, 大小写并不重要。</p>	<p>在 EGL 控制部件标记和值中, 大小写至关重要。</p>	<p>迁移工具将控制部件标记和值转换为 EGL 所需的正确大小写。</p>

## 生成选项部件

表 123. 生成选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>VAGen 生成选项部件:</p> <ul style="list-style-type: none"> <li>包含一个或多个生成选项。</li> <li>可以使用 /options 生成选项进行链接。</li> <li>在生成时可以引用工作空间中包括的任何其它控制部件。被引用的控制部件不会被认为该生成选项部件的关联项。</li> </ul>	<p>EGL 构建描述符部件:</p> <ul style="list-style-type: none"> <li>包含一个或多个构建描述符选项。</li> <li>可以使用 nextBuildDescriptor 构建描述符选项进行链接。</li> <li>只能引用其它符合下列其中一个条件的构建部件: <ul style="list-style-type: none"> <li>构建部件包括在同一个 .eglblid 文件中。</li> <li>构建部件位于由 .eglblid 文件导入的文件中。</li> </ul> </li> </ul>	<p>如果 VAGen 控制部件全都位于同一个 VisualAge Java 包或 VisualAge Smalltalk 应用程序中, 则将把控制部件全都放在同一个 .eglblid 文件中。在这种情况下, 不需要 import 语句。</p> <p>如果 VAGen 控制部件位于不同的 VisualAge Java 包或 VisualAge Smalltalk 应用程序中, 则迁移工具不创建 import 语句。您将需要添加 import 语句。如果 EGL 无法解析对其它控制部件的引用, 在“问题”视图中将有一个错误。</p>
<p>仅当 VAGen 生成选项值对目录或文件名包含特殊字符时, 才将那些值引在引号中。</p>	<p>必须将 EGL 构建描述符选项值引在引号中。但是, 如果您使用 EGL 构建部件编辑器, 则该编辑器会将引号自动插入到 XML 源代码中。您在编辑器中看不到引号。</p>	<p>当迁移工具为 .eglblid 文件构建 XML 源代码时, 它自动包括引号。</p>
<p>许多 VAGen 生成选项可以被指定为 /xxxx 或 /noxxxx 以反映该生成选项的正数或负数。下面是一个示例:</p> <ul style="list-style-type: none"> <li>/prep 指示要在生成之后立即自动启动准备步骤。</li> <li>/noprep 指示由于您计划在以后某个时间运行准备步骤, 所以您不想让该步骤自动启动。</li> </ul>	<p>许多 EGL 构建描述符选项可以被指定为 xxxx="YES" 或 xxxx="NO" 以反映该构建描述符选项的正数或负数。下面是一个示例:</p> <ul style="list-style-type: none"> <li>prep="YES" 指示要在生成之后立即自动启动准备步骤。</li> <li>prep="NO" 指示由于您计划在以后某个时间运行准备步骤, 所以您不想让该步骤自动启动。</li> </ul>	<p>迁移工具按照以下方式处理选项:</p> <ul style="list-style-type: none"> <li>除非另有指示, 否则迁移工具将 /xxxx 转换为对应的 xxxx="YES" 选项。</li> <li>除非另有指示, 否则迁移工具将 /noxxxx 转换为对应的 xxxx="NO" 选项。</li> </ul>
<p>/ansisql</p>	<p>不受支持。</p>	<p>迁移工具包括此选项作为注释。</p>
<p>/bidicontable=xxxx</p>	<p>bidiConversionTable="xxxx"</p>	<p>没有特殊的注意事项。</p>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/bind=xxxx</code></p> <p>在 VisualAge Generator 中, xxxx 是绑定部件的后缀。程序的绑定部件名为 <code>pgmname.xxxx</code>, 其中 xxxx 是由 <code>/bind</code> 选项指定的后缀。您指定 <code>/bind=suffix</code> 的原因如下:</p> <ol style="list-style-type: none"> <li>1. 由于您将程序绑定到多个 DB2 规划中, 所以程序需要特殊的绑定, 并且</li> <li>2. VisualAge Generator 未使您能够方便地创建与程序完全同名的绑定部件。</li> </ol>	<p><code>bind="xxxx"</code></p> <p><code>bind</code> 选项的含义与在 VisualAge Generator 中的含义并不相同。在 EGL 中, xxxx 是绑定部件的全名。仅当绑定部件名与程序名不同时, 才需要指定 <code>bind</code> 选项。在大多数情况下, 程序与绑定部件将具有相同的名称, 因此不需要包括 <code>bind</code> 选项。</p> <p>仅当为多个运行时环境生成同一个程序并且每个环境都需要特殊的绑定命令时, <code>bind</code> 选项才是必需的。</p> <p><code>bind</code> 选项的另一用途是指定包含绑定命令模板的部件的名称。项目管理员或 DBA 可以定义一个绑定部件并让该部件对特定于成员参数包含可替换的 SYMPARMS。可以使用 EGL <code>bind</code> 选项来指向这个模板部件。如果为每个程序都绑定包, 这种技术就能够很好地发挥作用。</p>	<p>由于 <code>bind</code> 的值在 VisualAge Generator 和 EGL 中具有不同的含义, 所以迁移工具无法迁移此选项。迁移工具将包括 <code>/bind</code> 作为注释。</p>
<p><code>/checktype=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• none</li> <li>• low</li> <li>• all</li> </ul>	<p><code>checkType="xxxx"</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• NONE</li> <li>• LOW</li> <li>• ALL</li> </ul>	<p>没有特殊的注意事项。</p>
<p><code>/cicsdbcs</code></p>	<p>不受支持。</p>	<p>由于所有受支持的 CICS 转换程序现在都包括对 DBCS 的支持, 所以迁移工具不将此选项作为注释包括在迁移结果中。</p>
<p><code>/cicsentries=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• none</li> <li>• rdo</li> <li>• macro</li> </ul>	<p><code>cicsEntries="xxxx"</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• NONE</li> <li>• RDO</li> <li>• MACRO</li> </ul>	<p>没有特殊的注意事项。</p>
<p><code>/cobollevel=le   vs</code></p>	<p>不受支持。</p>	<p>迁移工具包括此选项作为注释。</p>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>commentlevel=n</code> 或 <code>/commentlevel=commentText</code></p> <p><i>n</i> 或 <i>commentText</i> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• 0 或 minimum</li> <li>• 1 或 info</li> <li>• 2 或 logic</li> <li>• 3 或 data</li> <li>• 4 或 statements</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• 可以指定数值或与其等同的 <i>commentText</i>。</li> <li>• 0 = 仅生成选项注释</li> <li>• 1 = 别名, 标准生成信息</li> <li>• 2 = 程序和表 prolog 以及函数描述</li> <li>• 3 = 记录 prolog 和数据项描述</li> <li>• 4 = 源代码语句和注释</li> <li>• 对于 C++ 来说, 有效值只有 0 (无) 和 1 (注释)</li> </ul>	<p><code>commentLevel="n"</code></p> <p><i>n</i> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> <li>• 1</li> <li>• 1</li> <li>• 1</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• 0 = 没有注释</li> <li>• 1 = 包括注释</li> </ul>	<p>迁移工具将 <code>/commentlevel=0</code> 或 <code>minimum</code> 迁移为 0, 并将所有其它值迁移为 1。</p>
<p><code>/configmapname="xxxx"</code></p> <p><i>xxxx</i> 是 VisualAge Smalltalk 配置映射的名称。</p>	<p>不受支持。</p>	<p>由于此选项对于确定应该作为一个单元检入源代码资源库的相关 EGL 项目组可能有用, 所以迁移工具包括此选项作为注释。</p>
<p><code>/configmapversion="xxxx"</code></p> <p><i>xxxx</i> 是 <code>/configmapname</code> 指定的 VisualAge Smalltalk 配置映射的版本名。</p>	<p>不受支持。</p>	<p>由于此选项对于确定应该作为一个单元检入源代码资源库的相关 EGL 项目组可能有用, 所以迁移工具包括此选项作为注释。</p>
<p><code>/contable=xxxx</code></p> <p><i>xxxx</i> 是转换表的名称。</p>	<p><code>clientCodeSet="yyyy"</code> <code>serverCodeSet="zzzz"</code></p> <p><i>yyyy</i> 和 <i>zzzz</i> 分别是客户机转换表和服务器转换表的名称。</p>	<p>迁移工具根据 VAGen <code>/contable</code> 生成选项来设置 <code>clientCodeSet</code> 和 <code>serverCodeSet</code> 选项。请参阅第 262 页的表 124 以了解 VAGen 值与 EGL 值之间的对应关系。如果 <code>/contable=xxxx</code> 的值未包括在第 262 页的表 124 中, 则迁移工具将 <code>clientCodeSet</code> 和 <code>serverCode</code> 都设置为 <i>xxxx</i>。</p>
<p><code>/createdds</code></p>	<p><code>genDDSFile="YES"   "NO"</code></p> <p>注: 此选项用于 ISERIESC。</p>	<p>没有特殊的注意事项。</p>
<p><code>/currency=xxx</code> (1 到 3 个字符)</p>	<p><code>currencySymbol="xxx"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/data = 24   31</code></p>	<p><code>data="24"   "31"</code></p>	<p>没有特殊的注意事项。</p>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
/dbms=xxxx  xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• db2</li> <li>• oracle</li> <li>• odbc</li> </ul>	dbms="xxxx"  xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• DB2</li> <li>• ORACLE</li> <li>• DB2</li> </ul> 注: EGL Java 生成支持 JDBC, 而不支持 ODBC。	迁移工具将 odbc 更改为 DB2 并发出警告消息。
/dbpassword=xxxx	sqlPassword="xxxx"	迁移工具将 VAGen /dbpassword 和 /sqlpassword 选项合并成 EGL sqlPassword 选项。如果一个生成选项部件既包含 /dbpassword 又包含 /sqlpassword, 则迁移工具将包括两次 sqlPassword。在“问题”视图中应该会有一个错误。
/dbuser=xxxx	sqlID="xxxx"	迁移工具将 VAGen /dbuser 和 /sqlID 选项合并成 EGL sqlID 选项。如果一个生成选项部件既包含 /dbuser 又包含 /sqlID, 则迁移工具包括 sqlID 两次。在“问题”视图中应该会有一个错误。
/debugtrace	debugTrace="YES"   "NO"	没有特殊的注意事项。
/destaccount=xxxx	不受支持。	迁移工具包括此选项作为注释。
/destdir=xxxx	destDirectory="xxxx"	没有特殊的注意事项。
/desthost=xxxx	destHost="xxxx"	没有特殊的注意事项。
/destlib=xxxx	destLibrary="xxxx" 注: 此选项用于 ISERIESC。	没有特殊的注意事项。
/destpassword=xxxx	destPassword="xxxx"	没有特殊的注意事项。
/destuid=xxxx	destUserID="xxxx"	没有特殊的注意事项。
/dxfrcancel	cancelAfterTransfer="YES"   "NO"	没有特殊的注意事项。
/dxfrctl	useXctlForTransfer="YES"   "NO"	没有特殊的注意事项。
/ejbgroup=xxxx	不受支持。	迁移工具包括此选项作为注释。
/endcommarea	endCommarea="YES"   "NO"	没有特殊的注意事项。
/errdest=xxxx	errorDestination="xxxx" 注: 此选项用于 IMS。	没有特殊的注意事项。
/fastpath	imsFastPath="YES"   "NO" 注: 此选项用于 IMS。	没有特殊的注意事项。
/fold	不受支持。	迁移工具包括此选项作为注释。
/ftptranslationcmddbs=xxxx	不受支持。EGL 只支持通过 TCP/IP 来将文件传送至主机。	迁移工具包括此选项作为注释。
/ftptranslationcmdbcs=xxxx	不受支持。EGL 只支持通过 TCP/IP 来将文件传送至主机。	迁移工具包括此选项作为注释。
/genauthortimevalues	不受支持。	迁移工具包括此选项作为注释。
/genhelpmaps	genHelpFormGroup="YES"   "NO"	没有特殊的注意事项。

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
/genmaps	genFormGroup="YES"   "NO"	没有特殊的注意事项。
/genout=xxxx	genDirectory="xxxx"	没有特殊的注意事项。但是，如果为 Java 生成，则可能需要指定 genProject 构建描述符选项和 genDirectory 选项（或替代 genDirectory 选项）。如果为 HP-UX 或 SOLARIS 生成，则需要 genProject。
/genproperties /nogenproperties	genProperties="GLOBAL" genProperties="NO"  EGL 还提供了 genProperties="PROGRAM"。	迁移工具将 /genproperties 转换为 EGL <i>genProperties="GLOBAL"</i> ，这是因为，就所生成的内容来说，这是最接近的值。
/genresourcebundle	不受支持。	迁移工具包括此选项作为注释。
/genret	genReturnImmediate="YES"   "NO"	没有特殊的注意事项。
/gentables	genDataTables="YES"   "NO"	没有特殊的注意事项。
/genuirecords	genUIRecords="YES"   "NO" 注：此选项仅用于用户界面记录。	没有特殊的注意事项。
/groupname=xxxx	不受支持。	迁移工具包括此选项作为注释。
/inedit=all /inedit=inonly	validateOnlyIfModified="NO" validateOnlyIfModified="YES"	没有特殊的注意事项。
/initaddws  在 VisualAge Generator 中，始终初始化主工作存储器记录。/initaddws 生成选项为“表及其它记录”列表中指定的其它工作存储器记录提供初始化。	initNonIOData="YES"   "NO"  在 EGL 中，始终初始化被指定为主程序 inputRecord 的记录。initNonIOData 构建描述符选项为程序数据声明中指定的其它基本记录提供初始化。另外，EGL 还使您能够为程序中的任何记录声明指定 initialized 属性。与 initNonIOData 构建描述符选项相比，initialized 属性能够提供更精细的控制。	迁移工具将主程序的主工作存储器记录迁移为 EGL <i>inputRecord</i> 属性，并且还包含不具有 initialized 属性的记录声明。迁移工具将被调用程序的主工作存储器记录迁移为具有 initialized 属性的记录声明。如果迁移工具为 Level-77 项创建了另一个记录，则迁移工具为该记录包括数据声明，并且还包含 initialized 属性。这将为工作存储器记录提供与 VisualAge Generator 中相同的初始化。所有其它基本记录都是根据 <i>initNonIOData</i> 构建描述符选项进行初始化的。
/initrecd	initIORRecords="YES"   "NO"	没有特殊的注意事项。
/javadestdir=xxxx	不受支持。	迁移工具包括此选项作为注释。
/javadesthost=xxxx	不受支持。	迁移工具包括此选项作为注释。
/javadestpassword=xxxx	不受支持。	迁移工具包括此选项作为注释。
/javadestuid=xxxx	不受支持。	迁移工具包括此选项作为注释。
/javasystem=xxxx	不受支持。	迁移工具包括此选项作为注释。
/jobcard=xxxx	不受支持。等效功能如下所示：  • zOS 和 iSeries 构建服务器处理 jobcard。这些环境会忽略 JOBCARD 符号参数。	迁移工具会将此选项转换为 JOBCARD 符号参数。

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
/jobname=xxxx	不受支持。等效功能如下所示: <ul style="list-style-type: none"> <li>对于 zOS 和 iSeries, 可在构建脚本中将 \$USERID 用作作业名。EGL 生成将 \$USERID 替换为带有数字的 destUserID 构建描述符选项, 从而提供唯一的作业名。这些环境会忽略 JOBNAME 符号参数。</li> </ul>	迁移工具会将此选项转换为 JOBNAME 符号参数。
/jspreldir="xxxx"	不受支持。不再根据 EGL 源代码生成 JSP。	迁移工具包括此选项作为注释。
/leftjust	leftAlign="YES"   "NO"	没有特殊的注意事项。
/lineinfo	不受支持。	由于此选项只是对于 IBM 支持人员调试 VAGen 生成器才有意义, 所以迁移工具不将此选项作为注释包括在迁移结果中。此选项对生成的 COBOL 不起作用。
/lines=nn	不受支持。	迁移工具包括此选项作为注释。
/linkage=xxxx  xxxx 是 VAGen 链接表部件的名称。	linkage="xxxx"  xxxx 是 EGL 链接选项部件的名称。	没有特殊的注意事项。
/linkedit=xxxx  在 VisualAge Generator 中, xxxx 是链接编辑部件的后缀。程序的链接编辑部件名为 pgmname.xxxx, 其中 xxxx 是由 /linkedit 选项指定的后缀。您指定 /linkedit=suffix 的原因如下: <ol style="list-style-type: none"> <li>1. 程序需要特殊的链接编辑, 如用于 PL/I 的静态链接编辑, 并且,</li> <li>2. VisualAge Generator 未使您能够方便地创建与程序完全同名的链接编辑部件。</li> </ol>	linkEdit="xxxx"  linkEdit 的含义与在 VisualAge Generator 中的含义不相同。在 EGL 中, xxxx 是链接编辑部件的全名。仅当链接编辑部件名与程序名不同时, 才需要指定 linkEdit 选项。在大多数情况下, 程序与链接编辑部件具有相同的名称, 因此不需要包括 linkEdit 选项。仅当为多个运行时环境生成同一个程序时, linkEdit 选项才是必需的。	由于 linkEdit 的值在 VisualAge Generator 和 EGL 中具有不同的含义, 所以, 迁移工具无法迁移此选项。迁移工具包括 /linkedit 作为注释。
/listing /listingonerror /nolisting  注: 这是一个三路开关。	不受支持。	迁移工具包括此选项作为注释。
/locvalid	不受支持。	迁移工具包括此选项作为注释。
/log=xx  或者  /nolog	imsLogID="xx"  或者  包括 /nolog 作为注释  注: 此选项用于 IMS。	迁移工具按照以下方式处理此选项: <ul style="list-style-type: none"> <li>• 将 /log=xx 转换为 imsLogID="xx"</li> <li>• 将 /nolog 转换为注释。</li> </ul>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/math=xxxxx</code></p> <p>xxxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• cobol</li> <li>• cspae</li> </ul>	<p><code>math="xxxxx"</code></p> <p>xxxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• COBOL</li> <li>• CSPAE</li> </ul>	没有特殊的注意事项。
<p><code>/mfsdev=list-of-devices</code></p> <p><i>list-of-devices</i> 提供设备信息, 这些设备信息用来将 VAGen 映射生成用于 IMS 环境的 MFS 源代码。</p>	<p>当前不受支持。</p> <p>注: 此选项用于 IMS。</p>	迁移工具将此选项作为多行注释包括在迁移结果中。
<p><code>/mfseattr</code> <code>/nomfseattr</code> <code>/mfseattrncd</code></p> <p>在 VisualAge Generator 中, 这三个选项提供一个 3 路开关, 以便为 MFS 格式的映射提供生成扩展属性支持所需的信息。</p>	<p><code>mfsExtendedAttr="YES"</code> <code>mfsExtendedAttr="NO"</code> <code>mfsExtendedAttr="NCD"</code></p> <p>注: 此选项用于 IMS。</p>	没有特殊的注意事项。
<p><code>/mfsignore</code></p>	<p><code>mfsIgnore="YES"   "NO"</code></p> <p>注: 此选项用于 IMS。</p>	没有特殊的注意事项。
<p><code>/mfstest</code></p>	<p><code>mfsUseTestLibrary="YES"   "NO"</code></p> <p>注: 此选项用于 IMS。</p>	没有特殊的注意事项。
<p><code>/msgtableprefix=xxxx</code></p> <p>在 VisualAge Generator 中, 消息表前缀是在程序中指定的。当独立地生成用户界面记录时, 必须在生成期间指定消息表前缀。</p>	<p>不受支持。</p>	迁移工具包括此选项作为注释。
<p><code>/msp=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• all</li> <li>• gsam</li> <li>• mfs</li> <li>• seq</li> </ul>	<p><code>formServicePgmType="xxxx"</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• ALL</li> <li>• GSAM</li> <li>• MFS</li> <li>• SEQ</li> </ul> <p>注: 此选项用于 IMS。</p>	没有特殊的注意事项。
<p><code>/nullfill</code></p>	<p><code>fillWithNulls="YES"   "NO"</code></p>	没有特殊的注意事项。
<p><code>/numovfl</code></p>	<p><code>checkNumericOverflow="YES"   "NO"</code></p>	没有特殊的注意事项。
<p><code>/options=xxxx</code></p> <p>xxxx 是另一个 VAGen 生成选项部件的名称。</p>	<p><code>nextBuildDescriptor="xxxx"</code></p> <p>xxxx 是另一个 EGL 构建描述符部件的名称。</p>	没有特殊的注意事项。
<p><code>/packagename=xxxx</code></p>	<p>不受支持。</p>	迁移工具包括此选项作为注释。

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
/possign=x  x 是下列其中一项: • f • c	positiveSignIndicator="x"  x 是下列其中一项: • F • C  注: 此选项用于 ISERIESC。	没有特殊的注意事项。
/prep	prep ="YES"   "NO"	没有特殊的注意事项。
/prepfiler	buildPlan="YES"   "NO"	没有特殊的注意事项。
/printdest=xxxx  xxxx 是下列其中一项: • ezeprint • termid	printDestination="xxxx"  xxxx 是下列其中一项: • PROGRAMCONTROLLED • TERMINALID	没有特殊的注意事项。
/project="xxxx" [, "version"]  xxxx 是 VisualAge for Java 项目的名称, version 是指定的项目的版本名。	不受支持。	由于此选项对于确定应该作为一个单元检入源代码资源库的相关 EGL 项目组可能有用, 所以迁移工具包括此选项作为注释。
/projectid=xxxx	projectID="xxxx"	没有特殊的注意事项。
/recovery	restoreCurrentMsgOnError="YES"   "NO" 注: 此选项用于 IMS。	没有特殊的注意事项。
/resource=xxxx  xxxx 是 VAGen 资源关联部件的名称。	resourceAssociations="xxxx"  xxxx 是 EGL 资源关联部件的名称。	没有特殊的注意事项。
/resourcebundlelocale=xxxx	不受支持。  在 EGL 中, 此信息是在 PageHandler 部件中指定的。	迁移工具包括此选项作为注释。
/resvword=xxxx	reservedWord="xxxx"	没有特殊的注意事项。
/rt=xxxx	returnTransaction="xxxx"	没有特殊的注意事项。
/runfile	genRunFile="YES"   "NO"	没有特殊的注意事项。
/sendtranslationcmddbc=xxxx	不受支持。  EGL 只支持通过 TCP/IP 来将文件传送到主机。	迁移工具包括此选项作为注释。
/session=xxxx	不受支持。 注: EGL 只支持通过 TCP/IP 来将文件传送到主机。	迁移工具包括此选项作为注释。
/setfull	setFormItemFull="YES"   "NO"	没有特殊的注意事项。
/sp	checkToTransaction="YES"   "NO"	没有特殊的注意事项。
/spa=xxxx	spaSize="nnnn" 注: 此选项用于 IMS。	没有特殊的注意事项。
/spzero	spacesZero="YES"   "NO"	没有特殊的注意事项。
/sqldb=xxxx	sqlDB="xxxx"	没有特殊的注意事项。

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
/sqlid=xxxx	sqlID="xxxx"	迁移工具将 VAGen /dbuser 和 /sqlID 选项合并成 EGL <i>sqlID</i> 选项。如果一个生成选项部件既包含 /dbuser 又包含 /sqlID, 则迁移工具包括 sqlID 两次。在“问题”视图中应该会有一个错误。
/sqlpassword=xxxx	sqlPassword="xxxx"	迁移工具将 VAGen /dbpassword 和 /sqlpassword 选项合并成 EGL <i>sqlPassword</i> 选项。如果一个生成选项部件既包含 /dbpassword 又包含 /sqlpassword, 则迁移工具将包括两次 sqlPassword。在“问题”视图中应该会有一个错误。
/sqlvalid	validateSQLStatements="YES"   "NO"	没有特殊的注意事项。
/symparm=pppppppp,'vvvv' <ul style="list-style-type: none"> <li>• pppppppp 是符号参数的名称。pppppppp 是 1 - 8 个字符。</li> <li>• vvvv 是值。值中的两个连续单引号是一个单引号。</li> </ul>	EGL 与 VisualAge Generator 支持许多相同的预定义符号参数。您也可以使用任何用户定义的符号参数, 条件是它们未与任何新的 EGL 符号参数冲突。	迁移工具按照以下方式处理符号参数: <ul style="list-style-type: none"> <li>• 迁移工具将 VAGen 定义的任何符号参数转换为对应的 EGL 符号参数。</li> <li>• 如果没有对应的 EGL 符号参数, 则 VisualAge Generator 在不更改参数名或参数值的情况下迁移为 EGL 符号参数语法。迁移工具还将发出错误消息。</li> <li>• 迁移工具在不更改参数名或参数值的情况下将用户定义的任何符号参数转换为 EGL 符号参数语法。</li> </ul>
/SYMPARM=EZALTSTR,'xxxx'	transferErrorTransaction="xxxx"	没有特殊的注意事项。
/SYMPARM=EZONEAS2,'xxxx'	oneFormItemCopybook="YES"	没有特殊的注意事项。
/syncdxfr	synchOnPgmTransfer="YES"   "NO" 注: 此选项用于 CICS 环境的 DLI。	没有特殊的注意事项。
/syncxfer	synchOnTrxTransfer="YES"   "NO"	没有特殊的注意事项。
/syscodes	sysCodes="YES"   "NO"	没有特殊的注意事项。

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/system=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• MVS BATCH</li> <li>• MVSCICS</li> <li>• AIX</li> <li>• JAVAWRAPPER</li> <li>• WINNT</li> <li>• LINUX</li> <li>• OS400</li> <li>• VSE BATCH</li> <li>• VSECICS</li> <li>• HPUX</li> <li>• SOLARIS</li> </ul> <p>还可在 VAGen 中指定下列环境 (但不能用迁移工具来转换它们): IMSBMP、IMSVS、JAVA、JAVAGUI、WINGUI、OS2GUI、OS2、OS2CICS、AIXCICS、NTCICS、SOLACICS、TSO、VMCMS 和 VMBATCH</p>	<p><code>system="xxxx"</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• ZOS BATCH</li> <li>• ZOSCICS</li> <li>• AIX</li> <li>• WIN</li> <li>• WIN</li> <li>• LINUX</li> <li>• ISERIESC</li> <li>• VSE BATCH</li> <li>• VSECICS</li> <li>• HPUX</li> <li>• SOLARIS</li> </ul>	<p>迁移工具按照以下方式处理此选项:</p> <ul style="list-style-type: none"> <li>• 如果 <code>/system=xxxx</code> 在 EGL 中具有对应的值, 则迁移工具迁移为对应的 EGL 值。</li> <li>• 如果 <code>/system=xxxx</code> 在 EGL 中没有对应的值, 则迁移工具包括 <code>/system=xxxx</code> 作为注释。</li> <li>• 对于 <code>/system=JAVAWRAPPER</code> 来说, 迁移工具还包括 EGL 构建描述符选项 <code>enableJavaWrapperGen="ONLY"</code>。此选项指定您只想为程序生成 Java 包装器。</li> <li>• 对于 COBOL 环境, 迁移工具将发出一条警告消息, 指示您需要设置 <code>destPort</code> 构建描述符选项。</li> </ul>
<p><code>/targetNLS=xxx</code></p> <p>xxx 是 3 个字符的本地语言代码。</p>	<p><code>targetNLS="xxx"</code></p> <p>xxx 是 3 个字符的本地语言代码。除 ENP (大写英语) 以外的所有值在 VisualAge Generator 和 EGL 中都是完全相同的。ENP 在 EGL 中没有对应项。</p>	<p>迁移工具使用 VAGen 值来作为 targetNLS 值。如果该值为 ENP, 在“问题”视图中将有一个错误。您可以编辑 .eglbld 文件并更改该值。您可能想使用 ENU (混合大小写英语) 来替换 ENP。</p>
<p><code>/templates=xxxx</code></p> <p>在 VisualAge Generator 中, 模板用来生成准备和运行时 JCL 以及生成 CICS 事务和程序条目。</p>	<p><code>templateDir="xxxx"</code></p> <p>在 EGL 中, 构建脚本替换了构建模板。使用的模板只有用来为 ZOS BATCH 和 ISERIESC 目标环境生成运行时 JCL 的模板。</p>	<p>没有特殊的注意事项。</p>
<p><code>/trace=xxxx,yyyy</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• none</li> <li>• sqlerr</li> <li>• sqlio</li> </ul> <p>yyyy 是可选的。如果存在 yyyy, 则将其设置为 stmt。</p> <p>无论是否带有 stmt, none, sqlerr 或 sqlio 的任何组合都有效。</p>	<p><code>/trace</code> 分割为多个构建描述符选项:</p> <ul style="list-style-type: none"> <li>• 如果包括了 sqlerr - <code>sqlErrorTrace="YES"</code></li> <li>• 如果包括了 sqlio - <code>sqlIOTrace="YES"</code></li> <li>• 如果包括了 stmt - <code>statementTrace="YES"</code></li> </ul>	<p>没有特殊的注意事项。</p>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/transfertype=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• tcpip</li> <li>• sna</li> </ul>	<p>不受支持。</p> <p>EGL 只支持通过 TCP/IP 来将文件传送到主机。</p>	<p>迁移工具包括此选项作为注释。</p>
<p><code>/transid=primaryID,restartID</code></p> <p>在 VisualAge Generator 中, <code>/transid=,restart</code> 是有效的, 并且缺省主事务是程序名的前 4 个字符。</p>	<p><code>/transid</code> 分割为多个构建描述符选项:</p> <ul style="list-style-type: none"> <li>• 如果包括了 <code>primary</code> - <code>startTransactionID="primaryID"</code></li> <li>• 如果包括了 <code>,restart</code> - <code>restartTransactionID="restartID"</code></li> </ul>	<p>没有特殊的注意事项。</p>
<p><code>/twaoff=nnnn</code></p>	<p><code>twaOffset="nnnn"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/unload</code></p> <p>在 VisualAge Generator 中, <code>/unload</code> 指示批处理生成先卸装所有包含 VAGen 部件的 VisualAge Java 项目或 VisualAge Smalltalk 配置映射, 然后再装入正在为当前生成过程请求的项目或配置映射。</p>	<p>不受支持。</p>	<p>对于此选项来说, 迁移工具不包括注释。</p>
<p><code>/validmix</code></p>	<p><code>validateMixedItems="YES"   "NO"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/vmloadlib=xxxx</code></p>	<p>不受支持。</p>	<p>迁移工具包括此选项作为注释。</p>
<p><code>/vselib=xxxx</code></p>	<p><code>vseLibrary="xxxx"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/workdb=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• aux</li> <li>• main</li> <li>• dli</li> <li>• sql</li> </ul>	<p><code>workdb="xxxx"</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• AUX</li> <li>• MAIN</li> <li>• DLI</li> <li>• SQL</li> </ul> <p>注: DLI 和 SQL 值用于 IMS, 当前在 EGL 中不受支持。</p>	<p>迁移工具按照以下方式处理此选项:</p> <ul style="list-style-type: none"> <li>• 如果 <code>/workdb=aux</code> 或 <code>/workdb=main</code>, 则迁移工具将 <code>/workdb</code> 转换为对应的 EGL 值。</li> <li>• 如果 <code>/workdb=dli</code> 或 <code>/workdb=sql</code>, 则迁移工具包括此选项作为注释。</li> </ul>
<p>未使用。</p>	<p><code>vagCompatibility="YES"</code></p>	<p>根据 VAGen 迁移首选项不设置兼容性方式, 迁移工具会对每个构建描述符部件添加或省略此选项。</p>
<p>未使用。</p>	<p><code>itemsNullable="NO"</code></p>	<p>迁移工具始终对每个构建描述符部件添加此选项。这种技术用来明确地指示必须使用缺省值 NO 以保留 VAGen 行为。</p>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>在 VisualAge Generator 中, <i>decimalSymbol</i> 是在 Java 中进行 CHA 和 NUM 值赋值或比较时使用的运行时属性。</p>	<p><code>decimalSymbol="x"</code></p> <p><i>x</i> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• 句点 (.)</li> <li>• 逗号 (,)</li> </ul> <p>在 EGL 中, 可以在生成时或运行时指定此信息。</p>	<p>迁移工具不设置 <i>decimalSymbol</i>。如果您计划生成 Java 源代码, 则可以对构建描述符部件添加此属性。如果您这样做, 则将把 <i>decimalSymbol</i> 生成到任何 EGL 属性文件中。另外, 也可以将该属性直接添加到生成的 EGL 属性文件中。</p>
<p>未使用。</p>	<p><code>destPort="xxxx"</code></p> <p>在 EGL 中, <i>destPort</i> 指定在将生成输出传送到主机系统以准备执行它们时要使用的端口。对于 COBOL 生成目标环境, <i>destPort</i> 构建描述符选项是必需的。</p>	<p>迁移工具不设置 <i>destPort</i>。缺省值将随下列目标环境的不同而变化:</p> <ul style="list-style-type: none"> <li>• 对于 z/OS 环境, <i>destPort</i> 没有缺省值。必须添加 <i>destPort</i> 构建描述符选项并且该值必须与用来启动 z/OS 构建服务器的 JCL 中的值相匹配。用于启动 z/OS 构建服务器的样本 JCL 使用端口 5555。</li> <li>• 对于 iSeries 环境, <i>destPort</i> 没有缺省值。必须添加 <i>destPort</i> 构建描述符选项。该值必须与 iSeries 构建服务器使用的值相匹配。</li> </ul>
<p>未使用。</p>	<p><code>genProject="xxxx"</code></p>	<p>如果为 Java 生成, 则可能需要指定 <i>genProject</i> 构建描述符选项和 <i>genDirectory</i> 选项 (或替代 <i>genDirectory</i> 选项)。如果为 HP-UX 或 SOLARIS 生成, 则需要 <i>genProject</i>。</p>
<p>不受支持。</p> <p>在 VisualAge Generator 中, 即使程序检查 EZESYS 的值, 所有 VAGen 源代码也必须对可能要为其生成程序的每个目标环境有效。</p>	<p><code>eliminateSystemDependentCode="YES"   "NO"</code></p> <p>在 EGL 中, 如果程序检查 <i>systemType</i> 的值, 则您可以选择省略在当前目标生成环境下永远不会运行的源代码。这可以缩小所得到的 COBOL 或 Java 源代码。</p>	<p>迁移工具不设置 <i>eliminateSystemDependentCode</i>。缺省值是 "YES"。</p>
<p>未使用。</p>	<p><code>sessionBeanID="xxxx"</code></p>	<p>迁移工具不设置 <i>sessionBeanID</i>。如果正在生成 Java 或 Java 包装器, 则请参阅 EGL 联机帮助以确定是否需要设置 <i>sessionBeanID</i> 构建描述符选项。</p>

表 123. 生成选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>在 VisualAge Generator 中, 您在一个在运行时使用的属性文件中包括 SQL JDBC 驱动程序类、JNDI 名称和连接 URL 信息。</p>	<pre>sqlJDBCdriverClass="xxxx" sqlJNDIName="xxxx" sqlValidationConnectionURL="xx"</pre> <p>在 EGL 中, 可以在生成时或在运行时指定此信息。</p>	<p>迁移工具不设置 <i>sqlJDBCdriverClass</i>、<i>sqlJNDIName</i> 和 <i>sqlValidationConnectionURL</i>。如果要在生成时指定这些值, 可以执行下列操作:</p> <ul style="list-style-type: none"> <li>指定工作空间首选项。仅当正在 Eclipse 环境中进行生成时, 这种技术才能发挥作用。</li> <li>在构建描述符部件中指定构建描述符选项。当在 Eclipse 环境中进行生成时, 以及以批处理方式进行生成时, 这种技术能够发挥作用。</li> </ul> <p>在这两种情况下, 还必须包括 <i>genProperties="YES"</i> 构建描述符选项, 以便生成属性文件。</p> <p>如果要在运行时指定值, 则可以在 J2EE 部署描述符或属性文件中修改运行时属性。</p>
不受支持。	<i>cicsj2cTimeout="nnnn"</i>	<p>迁移工具不设置 <i>cicsj2cTimeout</i> 值。如果正在生成 Java, 则请参阅 EGL 联机帮助以确定是否需要设置 <i>cicsj2cTimeout</i> 构建描述符选项。</p>

表 124. 生成选项 - 转换表值

语言	转换表 VAGen /contable 值	EBCDIC 字符集 <i>serverCodeSet</i>	EGL <i>ASCII</i> 字符集 <i>clientCodeSet</i>
阿拉伯语	ELACNARA	IBM-420	IBM-1256
简体中文	ELACNCHS	IBM-935	IBM-1381
简体中文	ELACNGBK	IBM-1388	IBM-1386
繁体中文	ELACNCHT	IBM-937	IBM-950
丹麦语	ELACNDKN	IBM-277	IBM-1252
东欧语系	ELACN870	IBM-870	IBM-1250
英国英语	ELACN285	IBM-285	IBM-1252
美国英语	ELACNENU	IBM-037	IBM-1252
芬兰语	ELACNFIN	IBM-298	IBM-1252
法语	ELACNFRA	IBM-297	IBM-1252
德语	ELACNDEU	IBM-273	IBM-1252
希腊语	ELACNGRE	IBM-875	IBM-1253
希伯来语	ELACNHEB	IBM-424	IBM-1255
意大利语	ELACNITA	IBM-280	IBM-1252
日语, 片假名	ELACNJPN	IBM-930	IBM-943
日语, 拉丁文字	ELACNJPL	IBM-939	IBM-943

表 124. 生成选项 - 转换表值 (续)

语言	转换表 VAGen /contable 值	EBCDIC 字符集 EGL serverCodeSet	ASCII 字符集 EGL clientCodeSet
韩国语	ELACNKOR	IBM-933	IBM-949
挪威语	ELACNDKN	IBM-277	IBM-1252
葡萄牙语	ELACNPTB	IBM-037	IBM-1252
俄语	ELACNCYR	IBM-1025	IBM-1251
西班牙语	ELACNESP	IBM-284	IBM-1252
瑞典语	ELACNSWE	IBM-278	IBM-1252
瑞士德语	ELACNDES	IBM-500	IBM-1252
土耳其语	ELACNTUR	IBM-1026	IBM-1254
用户定义的 (未包括在以上列表中)	XXXXXXXX	XXXXXXXX	XXXXXXXX

## 链接表部件

链接表部件包括 Calllink、Filelink、Crtxlink 和 Dxfalink。

### callLink

表 125. :calllink 的链接表选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
:calllink	callLink	没有特殊的注意事项。
linktype=xxxx xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• dynamic</li> <li>• static</li> <li>• cicslink</li> <li>• remote</li> <li>• csocall</li> <li>• sessionejb</li> </ul>	调用类型, 其中, EGL 等效选项如下所示: <ul style="list-style-type: none"> <li>• localCall</li> <li>• localCall</li> <li>• localCall</li> <li>• remoteCall</li> <li>• remoteCall</li> <li>• ejbCall</li> </ul>	如果省略了 VAGen linktype, 则迁移工具使用 localCall。迁移工具还在其它位置使用 linktype 来为 EGL CallLink 信息设置其它属性。
applname=programName programName 是正在调用的程序的名称。允许使用通配符。	pgmName="programName"	没有特殊的注意事项。
externalname=applname	alias="applname"	如果由于 VAGen 程序名是 EGL 保留字而必须将该程序重命名, 则您可以在程序定义中或在链接表中使用 alias 属性来提供该程序的原始 VAGen 名称作为所生成的程序的名称。这两种技术都使您不必修改调用 VAGen 程序的非 VAGen 程序。

表 125. :callink 的链接表选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
package=packageName	package="packageName"	如果生成 Java 并且调用程序和被调用程序位于不同的包中, 则可以将包名包括在被调用程序的链接条目中。另外, 将 CALL 语句更改为用包名显式地对程序进行限定, 或者在包含 CALL 语句的文件中为该包包括一个 import 语句。
library=libraryName  或者  dllname=libraryName  在 VisualAge Generator 中, library 和 dllname 被视为同义词。	library="libraryName"	迁移工具将 VAGen library 或 dllname 合并成 EGL library 属性。
linktype=xxxx  xxxx 是下列其中一项: • dynamic • static • cicslink	linkType="xxxx"  xxxx 是下列其中一项: • DYNAMIC • STATIC • CICSLINK	没有特殊的注意事项。
parmform=xxxx  xxxx 是下列其中一项: • oslink • commptr • commdata • cicsoslink	parmForm="xxxx"  xxxx 是下列其中一项: • OSLINK • COMMPTR • COMMDATA • CICSOSLINK	没有特殊的注意事项。
contable=xxxx  xxxx 是下列其中一项: • conversionTableName • * • EZECONVT • BINARY • NONE	conversionTable="xxxx"  xxxx 是下列其中一项: • conversionTableName • * • PROGRAMCONTROLLED • 不受支持 • 不受支持	迁移工具将使用它在创建 EGL CallLink 信息时使用的那个 <i>conversionTableName</i> 。  迁移工具将 VAGen contable=BINARY 迁移为 BINARY, 后者在 EGL 中不是受支持的值。迁移工具还将发出错误消息。在“问题”视图中将会有有一个错误。您必须通过编辑 .eglblt 文件并选择要使用的受支持值来更正错误。  如果 VAGen contable=NONE, 则迁移工具省略 <i>conversionTable</i> 属性。
location=xxxx  xxxx 是下列其中一项: • systemName • EZELOC	location="xxxx"  xxxx 是下列其中一项: • systemName • PROGRAMCONTROLLED	没有特殊的注意事项。

表 125. :calllink 的链接表选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
remotecomtype=xxxx xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• appcims</li> <li>• ca400</li> <li>• cicsclient</li> <li>• dce</li> <li>• dcsecure</li> <li>• direct</li> <li>• exci</li> <li>• ipc</li> <li>• java400</li> <li>• lu2</li> <li>• tcpip</li> </ul>	remoteComType="xxxx" xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• 不受支持</li> <li>• 不受支持</li> <li>• CICSECI</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• DIRECT</li> <li>• 不受支持</li> <li>• DISTINCT</li> <li>• JAVA400</li> <li>• 不受支持</li> <li>• TCPIP</li> </ul>	迁移工具将 cicsclient 转换为 CICSECI, 这是因为后者是最接近的对应 EGL 值。 如果 VAGen :calllink 条目尚未指定 ctgport 和 ctglocation, 则迁移工具将发出一条错误消息以提醒您指定这些值。  迁移工具“按现状”迁移列示为不受支持的值并发出消息。现在, 您必须确定要使用的通信协议, 然后用正确的信息来更新 EGL CallLink 条目。在更正 CallLink 信息之前, 在“问题”视图中将有一个错误。  如果您决定使用 CICSSSL, 则必须对 EGL CallLink 信息添加 ctgPort、ctgLocation、ctgKeyStore 和 ctgKeyStorePassword 属性。  如果您决定使用 CICSJ2C, 则必须对 EGL CallLink 信息添加 pgmName、conversionTable、remotePgmType、luwControl、remoteBind、location 和 parmForm 属性。
remoteapptype=xxxx xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• vg</li> <li>• nonvg</li> <li>• vgjava</li> <li>• itf</li> </ul>	remotePgmType="xxxx" xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• EGL</li> <li>• EXTERNALLYDEFINED</li> <li>• 无</li> <li>• 不受支持</li> </ul>	如果 VisualAge Generator remoteapptype=vgjava, 则迁移工具将迁移 :calllink 条目, 但省略 remotePgmType 属性。  如果 remoteapptype=itf, 则迁移工具将把整个 :calllink 条目转换为注释。
serverid=serverName	serverID="serverName"	没有特殊的注意事项。
luwcontrol=xxxx xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• client</li> <li>• server</li> </ul>	luwControl="xxxx" xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• CLIENT</li> <li>• SERVER</li> </ul>	没有特殊的注意事项。
remotebind=xxxx xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• generation</li> <li>• runtime</li> </ul>	remoteBind="xxxx" xxxx 是下列其中一项: <ul style="list-style-type: none"> <li>• GENERATION</li> <li>• RUNTIME</li> </ul>	没有特殊的注意事项。
providerURL=URLName	providerURL="URLName"	没有特殊的注意事项。
ctglocation='tcpipInfo'	ctgLocation="tcpipInfo"	没有特殊的注意事项。
ctgport=portID	ctgPort="portID"	没有特殊的注意事项。

表 125. :callink 的链接表选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>bitmode=<i>nn</i></p> <p><i>nn</i> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• 16</li> <li>• 32</li> </ul>	不受支持。	迁移工具包括此选项作为注释。
<p>binform=<i>xxxx</i></p> <p><i>xxxx</i> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• intel</li> <li>• host</li> </ul>	不受支持。	迁移工具包括此选项作为注释。
<p>不受支持。</p> <p>在 VisualAge Generator 中, 如果被调用程序未将任何映射发送至屏幕, 则在 CALL 语句中指定 NOMAPS 选项可以获得更好的性能。</p>	refreshScreen="YES"   "NO"	迁移工具不设置此属性。如果先前已对 VAGen CALL 语句指定了 NOMAPS, 则可以继续在 EGL CALL 语句中使用 <i>noRefresh</i> 选项 (如果使用了 <i>vagCompatibility="YES"</i> 构建描述符选项的话)。另外, 通过在被调用程序的 CallLink 条目中指定 <i>refreshScreen="NO"</i> , 可以获得相同的支持。
<p>未使用。</p> <p>VisualAge Generator 支持的通信协议都不需要此信息。</p>	ctgKeyStore ctgKeyStorePassword	迁移工具不设置此属性。如果您决定使用 <i>remoteComType="CICSSSL"</i> , 则 <i>ctgKeyStore</i> 和 <i>ctgKeyStorePassword</i> 是必需的。
<p>未使用。</p> <p>在 VisualAge Generator 中, 每当您想要为被调用批处理程序生成 Java 包装器时, 使用 <i>/system=JAVAWRAPPER</i> 生成选项。</p>	javaWrapper="YES"   "NO"	迁移工具不设置此属性。如果要在每次生成被调用程序时生成 Java 包装器, 则必须指定 <i>javaWrapper="YES"</i> 。

## fileLink

表 126. :filelink 的链接表选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
:filelink	fileLink	没有特殊的注意事项。
<p>linktype=<i>xxxx</i></p> <p><i>xxxx</i> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• local</li> <li>• remote</li> </ul> <p>在 VisualAge Generator 中, 缺省值是 local。</p>	<p>文件类型, 其中, EGL 等效选项如下所示:</p> <ul style="list-style-type: none"> <li>• localFile</li> <li>• remoteFile</li> </ul>	如果未指定 VAGen linktype, 则迁移工具转换为 <i>localFile</i> 。

表 126. :filelink 的链接表选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
filename= <i>fileName</i>  <i>fileName</i> 是 VAGen 记录定义中的文件名。允许使用通配符。	fileName=" <i>fileName</i> "	没有特殊的注意事项。
contable= <i>xxxx</i>  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• conversionTableName</li> <li>• *</li> <li>• EZECONVT</li> <li>• BINARY</li> </ul>	conversionTable=" <i>xxxx</i> "  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• conversionTableName</li> <li>• *</li> <li>• PROGRAMCONTROLLED</li> <li>• 不受支持</li> </ul>	迁移工具将使用它在创建 EGL FileLink 信息时使用的那个 <i>conversionTableName</i> 。  迁移工具将 VAGen contable=BINARY 迁移为 BINARY, 后者在 EGL 中不是受支持的值。迁移工具还将发出错误消息。在“问题”视图中将会有有一个错误。您必须通过编辑 .eglbld 文件并选择要使用的受支持值来更正错误。
location= <i>xxxx</i>  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• CICS</li> <li>• EZELOC</li> </ul>	locationSpec=" <i>xxxx</i> "  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• CICS</li> <li>• PROGRAMCONTROLLED</li> </ul>	没有特殊的注意事项。

## Crtxlink

表 127. :crtxlink 的链接表选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
:crtxlink	asynchLink	没有特殊的注意事项。
linktype= <i>xxxx</i>  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• local</li> <li>• remote</li> </ul> 注: 在 VisualAge Generator 中, 缺省值是 local。	文件类型, 其中, EGL 等效选项如下所示: <ul style="list-style-type: none"> <li>• localAsynch</li> <li>• remoteAsynch</li> </ul>	如果未指定 VAGen linktype, 则迁移工具转换为 localAsynch。
recdname= <i>recordName</i>  <i>recordName</i> 是 VAGen 记录定义名。允许使用通配符。	recordName=" <i>recordName</i> "	没有特殊的注意事项。
contable= <i>xxxx</i>  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• conversionTableName</li> <li>• *</li> <li>• EZECONVT</li> <li>• BINARY</li> </ul>	conversionTable=" <i>xxxx</i> "  <i>xxxx</i> 是下列其中一项: <ul style="list-style-type: none"> <li>• conversionTableName</li> <li>• *</li> <li>• PROGRAMCONTROLLED</li> <li>• 不受支持</li> </ul>	迁移工具将使用它在创建 EGL AsynchLink 信息时使用的那个 <i>conversionTableName</i> 。  迁移工具将 VAGen contable=BINARY 转换为 BINARY, 后者在 EGL 中不是受支持的值。迁移工具还将发出错误消息。在“问题”视图中将会有有一个错误。您必须通过编辑 .eglbld 文件并选择要使用的受支持值来更正错误。

表 127. :crtxlink 的链接表选项 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
location=xxxx xxxx 是下列其中一项: • CICS • EZELOC	locationSpec="xxxx" xxxx 是下列其中一项: • CICS • PROGRAMCONTROLLED	没有特殊的注意事项。
package=packageName	package="packageName"	没有特殊的注意事项。

## Dxfrlink

表 128. :dxfrlink 的链接表选项

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
:dxfrlink	transferToProgram	没有特殊的注意事项。
fromappl=programName programName 是正在通过 DXFR 转移至另一个程序的程序的名称。不允许使用通配符。	fromPgm="programName"	没有特殊的注意事项。
toappl=programName2 programName2 是正在转移到的程序的名称。	toPgm="programName2"	没有特殊的注意事项。
linktype=xxxx xxxx 是下列其中一项: • dynamic • static • noncsp	linkType="xxxx" xxxx 是下列其中一项: • DYNAMIC • STATIC • EXTERNALLYDEFINED	如果先前已对 VAGen DXFR 语句指定了 NONCSP, 并且, 如果将 vagCompatibility="YES" 包括在构建描述符选项中, 则可以在 EGL transfer to program 语句中继续使用 externallyDefined 选项。另外, 可以通过对正在转移到的程序的 transferToProgram 条目指定 linkType="EXTERNALLYDEFINED" 来获取相同的支持。
不受支持。	alias="applname"	如果由于 VAGen 程序名是 EGL 保留字而必须将该程序重命名, 则您可以在程序定义中或在链接表中使用 alias 属性来提供该程序的原始 VAGen 名称作为所生成的程序的名称。这两种技术都使您不必修改调用 VAGen 程序的非 VAGen 程序。

## 资源关联部件

表 129. 资源关联

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p>在 VisualAge Generator 中，资源关联部件指定为特定目标环境实现文件的方式。该文件是 VAGen 记录定义中指定的文件名。</p> <p>资源关联部件还可以指定为特定目标环境实现打印输出的方式。</p> <p>当生成程序时，每个带索引输出、串行输出、相对输出或打印输出的文件名都与该资源关联部件匹配。根据文件名和生成目标环境，第一个匹配的条目就是用于该文件的条目。</p>	<p>EGL 资源关联部件指定为特定目标环境实现文件的方式。该文件是 EGL 记录定义中指定的 <i>fileName</i> 属性。</p> <p>资源关联部件还可以指定为特定目标环境实现打印输出的方式。</p> <p>当生成程序时，每个带索引输出、串行输出、相对输出或打印输出的文件名都与该资源关联部件匹配。根据文件名和生成目标环境，第一个匹配的条目就是用于该文件的条目。</p>	<p>没有特殊的注意事项。</p>
<p>对于 VisualAge Generator 来说，在运行时也使用资源关联文件。</p>	<p>对于 EGL 来说，资源关联信息存储在 EGL 部件中。</p>	<p>迁移工具支持对仅在 VAGen 资源关联文件中才有效的其它选项进行转换。</p>

表 129. 资源关联 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/system=targetSystem</code></p> <p><code>targetSystem</code> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• AIX *</li> <li>• AIXCICS *</li> <li>• HP-UX *</li> <li>• IMSBMP</li> <li>• IMSVS</li> <li>• LINUX **</li> <li>• MVS BATCH</li> <li>• MVSCICS</li> <li>• NTCICS *</li> <li>• OS2 *</li> <li>• OS2CICS</li> <li>• OS400</li> <li>• SCO *</li> <li>• SOLACICS *</li> <li>• SOLARIS *</li> <li>• TSO</li> <li>• VMCMS</li> <li>• VMBATCH</li> <li>• VSE BATCH</li> <li>• VSECICS</li> <li>• WINNT **</li> </ul> <p>注:</p> <ul style="list-style-type: none"> <li>• * - 指示用于 C++ 生成的环境。</li> <li>• ** - 指示用于 Java 生成的环境。</li> <li>• <code>/system</code> 是可选的。</li> <li>• VisualAge Generator 支持在目标系统中将 * 用作通配符。(例如, MVS* 或 *CICS)。</li> </ul>	<p>这是 EGL 目标环境。</p> <p>对应的环境值如下所示:</p> <ul style="list-style-type: none"> <li>• aix</li> <li>• 不受支持</li> <li>• hpux</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• linux</li> <li>• zosbatch</li> <li>• zoscics</li> <li>• 值将无效</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• iseriesc</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• win</li> </ul> <p>注: 不支持通配符。</p>	<p>迁移工具按照以下方式处理 <code>/system</code> 选项:</p> <ul style="list-style-type: none"> <li>• 对于列示为不受支持的目标系统, 迁移工具在 EGL 资源关联部件中包括 VAGen 资源关联条目信息作为注释。这样有助于最大程度地保留信息。</li> <li>• 如果在 VAGen 资源关联条目中省略了 <code>/system</code> 选项, 则迁移工具使用 <i>any</i> 作为 EGL 资源关联目标环境。</li> <li>• 如果 <code>/system</code> 选项使用通配符, 则迁移工具完全按原样迁移该选项, 包括通配符 (例如, mvs* 或 *cics)。迁移工具还将发出错误消息。</li> </ul>

表 129. 资源关联 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/filetype=fileType</code></p> <p><code>fileType</code> 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• BTRIEVE</li> <li>• GSAM</li> <li>• IBMCOBOL</li> <li>• MFCOBOL</li> <li>• MMSGQ</li> <li>• MQ</li> <li>• OS2COBOL</li> <li>• SEQ</li> <li>• SEQRS</li> <li>• SMSGQ</li> <li>• SPOOL</li> <li>• TEMPAUX</li> <li>• TEMPMAIN</li> <li>• TRANSIENT</li> <li>• VSAM</li> <li>• VSAMRS</li> </ul>	<p>EGL 文件类型, 其中, 对应的值如以下列表所示:</p> <ul style="list-style-type: none"> <li>• 不受支持</li> <li>• 不受支持</li> <li>• ibmcobol</li> <li>• 不受支持</li> <li>• 不受支持</li> <li>• mq</li> <li>• 不受支持</li> <li>• seq 或 seqws</li> <li>• seqrs</li> <li>• 不受支持</li> <li>• spool</li> <li>• tempaux</li> <li>• tempmain</li> <li>• transient</li> <li>• vsam</li> <li>• vsamrs</li> </ul>	<p>迁移工具按照以下方式处理 <code>/filetype</code> 选项:</p> <ul style="list-style-type: none"> <li>• 如果 <code>/system</code> 选项指定了主机目标环境, 则迁移工具将 VAGen SEQ 文件类型转换为 EGL <code>seq</code> 文件类型。</li> <li>• 如果 <code>/system</code> 选项是工作站环境, 则迁移工具将 VAGen SEQ 文件类型转换为 EGL <code>seqws</code> 文件类型。</li> <li>• 如果 <code>/system</code> 是 MVS BATCH 并且文件类型是 GSAM, 则迁移工具将资源关联条目迁移为注释。</li> <li>• 对于所有其它不受支持的文件类型值, 如果资源关联用于受支持的 <code>/system</code>, 则迁移工具使用 VAGen 文件类型来创建 EGL 资源关联条目并发出错误消息。在“问题”视图中也将会有一个错误。在可以使用 EGL 资源关联部件之前, 必须修正此错误。</li> </ul>
<p><code>/sysname=systemName</code></p>	<p><code>systemName="systemName"</code></p>	<p>迁移工具将 <code>/system</code> 选项中使用的任何符号参数转换为对应的 EGL 替换符号参数。</p>
<p><code>/replace</code> <code>/noreplace</code></p>	<p><code>replace="YES"</code> <code>replace="NO"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/dup</code> <code>/nodup</code></p>	<p><code>duplicates="YES"</code> <code>duplicates="NO"</code></p> <p>注: 此选项用于 ISERIESC。</p>	<p>没有特殊的注意事项。</p>
<p><code>/commit</code> <code>/noccommit</code></p> <p>这些选项仅用于 OS/400 目标环境。</p>	<p><code>commit="YES"</code> <code>commit="NO"</code></p> <p>注: 此选项用于 ISERIESC。</p>	<p>没有特殊的注意事项。</p>
<p><code>/blksize=xxxx,yyy,zzz</code></p> <p>在 VisualAge Generator 中, 此选项仅用于 VSE 目标环境。</p>	<p><code>blockSize="xxxx,yyy,zzz"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/sysnum=xxxx</code></p> <p>在 VisualAge Generator 中, 此选项仅用于 VSE 目标环境。</p>	<p><code>systemNumber="xxxx"</code></p>	<p>没有特殊的注意事项。</p>
<p><code>/label</code> <code>/nolabel</code></p> <p>在 VisualAge Generator 中, 此选项仅用于 VSE 目标环境。</p>	<p><code>standardLabel="YES"</code> <code>standardLabel="NO"</code></p>	<p>没有特殊的注意事项。</p>

表 129. 资源关联 (续)

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
<p><code>/pcbno=n</code></p> <p>此选项仅对 IMSVS 或 IMSBMP 目标环境有效，或者，如果文件类型为 GSAM，则对 MVS Batch 有效。</p>	<p>不受支持。</p>	<p>迁移工具将注释掉用于不受支持的目标环境的资源关联条目，或者，如果文件类型为 GSAM，则将注释掉用于 MVS Batch 的资源关联条目。</p>
<p><code>/noff</code></p> <p>在 VisualAge Generator 中没有 <code>/FF</code> 选项。此选项只有在 VAGen 资源关联文件中才受支持。</p>	<p><code>FormFeedOnClose="NO"</code>  <code>FormFeedOnClose="YES"</code></p>	<p>迁移工具将 <code>/noff</code> 转换为 <code>FormFeedOnClose="NO"</code>。</p>
<p><code>/text</code></p> <p>在 VisualAge Generator 中没有 <code>/NOTEXT</code> 选项。此选项只有在 VAGen 资源关联文件中才受支持。</p>	<p><code>text="YES"</code>  <code>text="NO"</code></p>	<p>迁移工具将 <code>/text</code> 转换为 <code>text="YES"</code>。</p>
<p><code>/contable=xxxx</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• <code>conversionTableName</code></li> <li>• <code>EZECONVT</code></li> </ul> <p>此选项只有在 VAGen 资源关联文件中才受支持。</p>	<p><code>conversionTable="xxxx"</code></p> <p>xxxx 是下列其中一项:</p> <ul style="list-style-type: none"> <li>• <code>conversionTableName</code></li> <li>• <code>PROGRAMCONTROLLED</code></li> </ul>	<p>迁移工具将使用它在创建 EGL 资源关联条目时使用的那个 <code>conversionTableName</code>。</p>
<p><code>/keys=xxxx</code></p> <p>在 VisualAge Generator 中，此选项仅与 <code>/filetype=BTRIEVE</code> 配合使用。此选项只有在 VAGen 资源关联文件中才受支持。</p>	<p><code>KEYS="xxxx"</code></p>	<p>由于 <code>BTRIEVE</code> 是在受支持的目标环境中使用的，所以迁移工具将 <code>/keys</code> 选项迁移为 EGL <code>keys</code> 选项。</p>
<p><code>/basename=xxxx</code></p> <p>在 VisualAge Generator 中，此选项仅用于 OS/2<sup>®</sup> 目标环境。此选项只有在 VAGen 资源关联文件中才受支持。</p>	<p>不受支持。</p>	<p>迁移工具将注释掉 OS/2 目标环境的任何条目。</p>

## 链接编辑部件

表 130. 链接编辑部件

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
在 VisualAge Generator 中，链接编辑部件通常名为 <i>programName.suffix</i> ，其中，名称的第一部分与 VAGen 程序名相同，后缀是 LKG。VAGen /linkedit 生成选项指定了后缀值。	缺省情况下，在 EGL 中，链接编辑部件名必须与程序名相同。如果是这种情况，则不需要指定链接编辑构建描述符选项。  如果一个程序有多个链接编辑部件，则您必须对程序的各个链接编辑构建描述符选项部件使用不同的部件名。在这种情况下，必须在 linkedit 构建描述符选项中指定完整的链接编辑部件名。	如果后缀是 .LKG，则迁移工具在创建新的 EGL 链接编辑部件时将除去该后缀。如果后缀是除 .LKG 以外的任何内容，则迁移将 <i>.suffix</i> 更改为 <i>_suffix</i> ，这是因为句点 (.) 在 EGL 部件名中不是有效的字符。
VAGen 链接编辑部件包含一些链接编辑语句，在主机环境中，在准备过程中对程序进行链接编辑时，需要这些链接编辑语句。	在 EGL 中，链接编辑部件包含一些链接编辑语句，在主机环境中，在构建过程中对程序进行链接编辑时，需要这些链接编辑语句。	迁移工具执行下列操作： <ul style="list-style-type: none"> <li>• 将链接编辑部件中使用的任何符号参数转换为对应的 EGL 替换符号参数。</li> <li>• 使用的缩进与 VAGen 部件中使用的缩进相同。</li> </ul>

## 绑定控制部件

表 131. 绑定控制部件

VisualAge Generator 4.5	迁移工具生成的 EGL	迁移工具注意事项
在 VisualAge Generator 中，绑定控制部件通常名为 <i>programName.suffix</i> ，其中，名称的第一部分与 VAGen 程序名相同，后缀是 BND。VAGen /bind 生成选项指定了后缀值。	缺省情况下，在 EGL 中，绑定控制部件名必须与程序名相同。如果是这种情况，则不需要指定绑定控制构建描述符选项。  如果一个程序有多个绑定控制部件，则必须对程序的各个绑定控制部件使用不同的部件名。在这种情况下，必须在 bind 构建描述符选项中指定完整的绑定控制部件名。	如果后缀是 .BND，则迁移工具在创建新的 EGL 绑定控制部件时将除去该后缀。如果后缀是除 .BND 以外的任何内容，则迁移将 <i>.suffix</i> 更改为 <i>_suffix</i> ，这是因为句点 (.) 在 EGL 部件名中不是有效的字符。
VAGen 绑定控制部件包含一些 DB2 绑定命令，在 MVS 主机环境中的准备过程中，在为程序绑定 DB2 数据库资源模块 (DBRM) 时需要这些绑定命令。	在 EGL 中，绑定控制部件包含一些绑定命令，在 ZOS 主机环境中的构建过程中，在为程序绑定 DBRM 时需要这些绑定命令。	迁移工具执行下列操作： <ul style="list-style-type: none"> <li>• 在绑定控制部件的开头添加其它命令。这些命令是构建服务器所需的。</li> <li>• 将绑定控制部件中使用的任何符号参数转换为对应的 EGL 替换符号参数。</li> <li>• 使用的缩进与 VAGen 部件中使用的缩进相同。</li> </ul>

## 符号参数

下列各表显示了 VAGen 符号参数与 EGL 符号参数之间的关系。

表 132. 与部件相关的符号参数

与部件相关的符号参数	对应的 EGL 符号参数
EZECOBOLTYPE	不受支持

表 132. 与部件相关的符号参数 (续)

与部件相关的符号参数	对应的 EGL 符号参数
EZEDATA	DATA
EZEDBCS	不受支持
EZEDESTLIB	不受支持
EZEDESTNAME	不受支持
EZEDLI	不适用 - 仅用于 DL/I
EZEENTRY	不受支持
EZEENV	SYSTEM
EZEGDATE	EZEGDATE
EZEGENOUT	不受支持
EZEGMBR	EZEGMBR
EZEGTIME	EZEGTIME
EZEMBR	在链接编辑部件或绑定部件中, 是 EZEALIAS。否则, 是 EZEMBR。
EZEMBRPATH	不受支持
EZEMSG	不受支持
EZEPID	EZEPID
EZEPREPDESTACCOUNT	不受支持
EZEPREPDESTHOST	不受支持
EZEPREPDESTDIR	不受支持
EZEPREPDESTPASSWORD	不受支持
EZEPREPDESTUID	不受支持
EZEPREPFTPCMDSBCS	不受支持
EZEPREPFTPCMDBCS	不受支持
EZEPREPSEND CMDDBCS	不受支持
EZEPREPSSESSION	不受支持
EZEPREPS	不受支持
EZEPREPSQLDB	不受支持
EZEPREPWORKDB	不受支持
EZEPSB	不适用 - 仅用于 DL/I 和 IMS
EZEPTYPE	不受支持
EZESQL	EZESQL
EZETBLNAME	不受支持
EZETPROC	不受支持
EZETRAN	不受支持
EZETRANSFERTYPE	不受支持
EZETWASIZE	不受支持
EZEUSERID	不受支持
EZEVMLoadLIB	不适用 - 仅用于 VM
EZEVSELIB	不适用 - 仅用于 VSE
EZEXAPP	不受支持。

表 133. 与文件相关的符号参数

与文件相关的符号参数	对应的 EGL 符号参数
EZEBLK	EZEBLK
EZEDBD	不适用 - 仅用于 DL/I
EZEDD	EZEDD
EZEDLBL	不适用 - 仅用于 VSE
EZEDSN	EZEDSN
EZELRECL	EZELRECL
EZERECFM	EZERECFM

表 134. 用户定义的符号参数

用户定义的符号参数	对应的 EGL 符号参数
COB2LIB	COBCICS
COBLIST	不受支持
DBDLIB	不适用 - 仅用于 DL/I
DSNLOAD	DSNLOAD
DSYS	DSYS
ELA	ELA
EZALTXTR	进行特殊的迁移以迁移到正常的构建选项 - 请参阅『生成选项』一节中的 transferErrorTransaction="xxx"
EZONEAS2	进行特殊的迁移以迁移到正常的构建选项 - 请参阅『生成选项』一节中的 oneFormItemCopybook="YES"
EZUAUTH	EZUAUTH
EZUINST	EZUINST
PSBLIB	不适用 - 仅用于 DL/I
PROCLIB	不适用 - 仅用于 VSE
PWRCLASS	不适用 - 仅用于 VSE
RESLIB	不适用 - 仅用于 DL/I 和 IMS
SQLDBNAM	不适用 - 仅用于 VM 和 VSE
SQLPKGNM	不适用 - 仅用于 VM 和 VSE
SQLPROPT	不适用 - 仅用于 VM 和 VSE
SQLSTMDE	不适用 - 仅用于 VM 和 VSE
SQLSTOPT	不适用 - 仅用于 VM 和 VSE
SQLUSRPW	不适用 - 仅用于 VM 和 VSE
VMFMODE	不适用 - 仅用于 VM
VMDISKADDR	不适用 - 仅用于 VM
VUSERLIB	不适用 - 仅用于 VSE



---

## 附录 C. 迁移工具发出的消息

本节包含由迁移工具发出的消息。可以在下列各节中根据消息前缀找到消息：

- HPT.EGL.00xxx - 第 1 阶段公共消息
- HPT.EGL.01xxx - 基于 VisualAge for Java 的第 1 阶段
- HPT.EGL.02xxx - 基于 VisualAge Smalltalk 的第 1 阶段
- IWN.MIG - EGL 中的第 2 阶段和第 3 阶段

每个消息号的最后一位中的字符是一个后缀，它指示该消息的严重性：

- *i* - 参考消息，用于指示状态，或者用于指示迁移工具在迁移期间由于 VisualAge Generator 与 EGL 语言之间的差别而除去了信息。用户不需要执行操作。
- *w* - 警告消息，用于指示可能的问题。例如，迁移工具对 EGL 语法进行最佳猜测。仅当验证或生成检测到错误时，用户才需要执行操作。
- *e* - 错误消息。迁移工具无法对 EGL 语法进行合理的猜测。用户需要执行操作以提供缺少的或不完整的信息。
- *t* - 跟踪消息，用于指示比参考消息所提供的状态更详细的状态。跟踪消息包括关于何时建立落实点的详细信息。跟踪消息本身带有解释性质，未包括在本迁移指南中。

---

### VisualAge Generator 到 EGL 迁移工具发出的消息 - 第 1 阶段

第 1 阶段迁移工具是作为样本交付的。在样本工具本身中，未对消息进行翻译。但是，在《迁移指南》中，这里对样本附带交付的消息进行了翻译。

#### 第 1 阶段公共消息

下列消息是用于 VisualAge for Java 和 VisualAge Smalltalk 的 VAGen 迁移工具的公共消息。

---

**HPT.CM.215.e** 无法打开文件 *filename*。返回码是 *returnCode* (*returnCodeText*)。

**说明：** 无法打开指定的文件。*returnCode* 和 *returnCodeText* 指示了原因。*returnCode* 2 指示找不到该文件。

**用户响应：** 为迁移工具提供有效的文件。

---

**HPT.EGL.0001.w** 表名 *tableName* 是保留字。必须将其重命名。

**说明：** 迁移工具不会将表重命名。

**用户响应：** 必须更改该表的名称以及对该表的所有引用，这包括任何程序的“表及其它记录”列表、逻辑语句、数据项编辑例程以及映射编辑例程中对该表进行的引用。务必更改对 *dataTable* 名进行的任何非 VAGen 引用（包括

CICS 程序定义）。或者，您可以等待迁移完成，然后在 EGL 中重命名 *dataTable*，使用 EGL *alias* 属性来指定原始表名。

---

**HPT.EGL.0002.w** 映射组名 *mapGroupName* 是保留字。必须将其重命名。

**说明：** 迁移工具不会将该映射组重命名。

**用户响应：** 必须更改该映射组的名称、该映射组中所有映射的名称以及对该映射组的所有引用，这包括作为任何程序的映射组或帮助映射组进行的引用。务必更改对该映射组名进行的任何非 VAGen 引用（包括 CICS 程序定义）。或者，您可以等待迁移完成，然后在 EGL 中重命名 *formGroup*，并使用 EGL *alias* 属性来指定原始映射组名。

---

**HPT.EGL.0003.w** 程序名 *programName* 是保留字。必须将其重命名。

说明: 迁移工具不会将该程序重命名。

用户响应: 必须更改该程序的名称以及对该程序的所有引用, 这包括 CALL、DXFR 和 XFER 语句中进行的引用。并更改与此程序相对应的任何绑定控制部件或链接编辑部件的名称。务必更改对该程序名进行的任何非 VAGen 引用 (包括 CICS 程序定义)。或者, 您可以等待迁移完成, 然后在 EGL 中重命名该程序, 并使用 EGL *alias* 属性来指定原始程序名。

---

**HPT.EGL.0004.w** 控制部件名 *partName* 是保留字。必须将其重命名。

说明: 指定的控制部件名使用点表示法, 而点前面的名称是一个保留字。迁移工具假定点前面的名称是程序名, 并假定此控制部件与该程序紧密联系。因为迁移工具不会将该程序重命名, 所以, 它也不会将点表示法中的控制部件重命名。

用户响应: 必须更改该程序的名称以及对该程序的所有引用, 这包括 CALL、DXFR 和 XFER 语句中进行的引用。并更改与此程序相对应的任何绑定控制部件或链接编辑部件的名称。务必更改对该程序名进行的任何非 VAGen 引用 (包括 CICS 程序定义)。或者, 您可以等待迁移完成, 然后在 EGL 中重命名该程序, 并使用 EGL *alias* 属性来指定原始程序名。请参阅第 173 页的附录 A, 『保留字』, 该附录列示了 EGL 保留字。

---

**HPT.EGL.0005.w** 用户界面记录 *partName* 是保留字或者以符号 # 开头。必须将其重命名。

说明: 指定的用户界面记录与 EGL 命名约定冲突。此发行版的 EGL 不支持 Web 事务或用户界面记录。第 1 阶段迁移工具将用户界面记录包括在迁移集中。第 2 阶段和第 3 阶段迁移工具过滤掉用户界面记录, 不将它们迁移为 EGL 源代码。

用户响应: 无。不需要在 VisualAge Generator 中将该用户界面记录重命名。预计第 2 阶段和第 3 阶段迁移工具将处理用户界面记录的重命名, 就象是处理其它记录的重命名一样。

---

**HPT.EGL.0006.i** *preferenceFile* 的迁移将生成 *outputList*。

说明: *preferenceFile* 的迁移将生成 *outputList*。可能的输出包括迁移规划、报告和数据库更新。

用户响应: 无。

---

**HPT.EGL.0007.w** 根据当前过滤器, 未创建任何迁移文件。

说明: 根据当前过滤器, 未创建任何迁移文件。

用户响应: 更改过滤器首选项。

---

**HPT.EGL.0008.e** *PreferenceValue* 不是首选项 *preferenceOption* 的有效值。

说明: 该值对于该首选项来说是无效的。

用户响应: 在首选项文件中更改该首选项值。

---

**HPT.EGL.0009.e** 迁移集 *migrationSetName* 要求指定用于跨映射后缀的首选项。

说明: 指定的迁移集包含一个或多个跨多个项目或多个包的映射组。迁移工具要求您指定跨映射后缀首选项以便它可以创建该映射组所需的项目或包。

用户响应: 编辑第 1 阶段迁移首选项文件。在“映射”页上的“跨映射”部分中, 为“项目后缀”和“包后缀”字段指定值。要了解更多信息, 请参阅第 91 页的『“映射”页』(对于 Java)或第 109 页的『“映射”页』(对于 Smalltalk)。

---

**HPT.EGL.0010.w** 未请求任何迁移操作。

说明: 您还没有为第 1 阶段迁移工具选择任何输出选项。

用户响应: 选择一个或多个选项。这些选项使您能够创建迁移规划文件、创建报告或更新数据库。

---

**HPT.EGL.0011.i** 正在为迁移集 *migrationSetName* 启动数据库清理。

说明: 迁移数据库已包含指定的迁移集的信息。在准备使用新的一组首选项来运行第 1 阶段时, 迁移工具删除了该迁移集信息。

用户响应: 无。

---

**HPT.EGL.0012.i** 已经为迁移集 *migrationSetName* 完成了数据库清理。

说明: 迁移数据库已包含指定的迁移集的信息。在准备使用新的一组首选项来运行第 1 阶段时, 迁移工具删除了该迁移集信息。

用户响应: 无。

---

**HPT.EGL.0013.e** 每个重命名规则都必须具有唯一的顺序值。

说明: 两个或更多个重命名规则具有相同的顺序号。

用户响应: 编辑第 1 阶段迁移首选项文件并更改重命名规则以使每个规则都具有唯一的顺序号。

---

**HPT.EGL.0014.i** 迁移集

*migrationSetName-migrationSetVersion* 生成了 *n* 条错误消息、*n* 条警告消息和 *n* 条参考消息。

说明: *n* 是第 1 阶段迁移工具对指定迁移集发出的消息的数目。参考消息的计数包括消息 HPT.EGL.0014.i。

用户响应: 无。

---

**HPT.EGL.0015.e** 派生的 EGL 项目名 *eglProjectName* 包含无效字符: *characterList*。修改重命名规则。

说明: 通过使用您指定的重命名规则, 第 1 阶段迁移工具创建了建议的 EGL 项目名, 但该名称不符合 EGL 项目命名约定。*characterList* 显示了无效的字符。

用户响应: 编辑第 1 阶段迁移首选项文件并修改项目重命名规则, 以使它们能够生成有效的 EGL 项目名。在修改重命名规则时, 务必考虑任何指定了映射上下文两者的重命名规则的影响。

---

**HPT.EGL.0016.e** 派生的 EGL 包名 *eglPackageName* 包含无效字符: *characterList*。修改重命名规则。

说明: 通过使用您指定的重命名规则, 迁移工具创建了建议的 EGL 包名, 但该名称不符合 EGL 包命名约定。*characterList* 显示了无效的字符。

用户响应: 编辑第 1 阶段迁移首选项文件并修改包重命名规则, 以使它们能够生成有效的 EGL 包名。在修改重命名规则时, 务必考虑任何指定了映射上下文两者的重命名规则的影响。

---

**HPT.EGL.0017.e** 派生的 EGL 项目名 *eglProjectName* 不能以句点 (.) 结尾。修改重命名规则。

说明: 通过使用您指定的重命名规则, 迁移工具创建了建议的 EGL 项目名, 但该名称以句点结尾。此名称不符合 EGL 项目命名约定。

用户响应: 编辑第 1 阶段迁移首选项文件并修改项目重命名规则, 以使它们能够生成有效的 EGL 项目名。在修改重命名规则时, 务必考虑任何指定了映射上下文两者的重命名规则的影响。并且, 考虑任何指定了字符串上下文任何、后面或标记的重命名规则的影响。

---

**HPT.EGL.0018.e** 派生的 EGL 包名 *eglPackageName* 不能以数字开头或以句点 (.) 结尾。修改重命名规则。

说明: 通过使用您指定的重命名规则, 迁移工具创建了建议的 EGL 包名, 但该名称以句点结尾或以数字开头。此名称不符合 EGL 包命名约定。

用户响应: 编辑第 1 阶段迁移首选项文件并修改包重命名规则, 以使它们能够生成有效的 EGL 包名。在修改重命名规则时, 务必考虑任何指定了映射上下文两者的重命名规则的影响。

---

**HPT.EGL.0019.i** 已装入迁移功能部件 *featureName* *versionName*。

说明: 此参考消息提供当前已装入到 Java 工作空间或 Smalltalk 映像中的迁移功能部件的名称和版本。

用户响应: 无。

---

**HPT.EGL.0020.i** 未装入迁移功能部件 *featureName* *versionName*。未装入 *listOfNames*。

说明: 此参考消息提供应该添加至 Java 工作空间或装入到 Smalltalk 映像中的迁移功能部件的名称和版本。但是, 一个或多个 Java 包或 Smalltalk 应用程序不具有迁移功能部件所期望的版本。*listOfNames* 提供当前已装入但不具有期望版本的 Java 包或 Smalltalk 应用程序的列表。

用户响应: 如果您未修改第 1 阶段迁移工具, 则尝试再次添加该迁移功能部件 (对于 Java) 或者再次装入该迁移功能部件 (对于 Smalltalk)。如果您已修改了第 1 阶段迁移工具, 则此消息可以提醒您已修改了哪些 Java 包或 Smalltalk 应用程序。

---

**HPT.EGL.0021.e** 无法装入外部化的 EGL 保留字列表。验证 *fileName*。

说明: 找不到指定的 *fileName*。*fileName* 是文件的完整路径名。如果您是按照第 87 页的第 4 章, 『第 1 阶段 - 从 Java 中抽取』或第 105 页的第 5 章, 『第 1 阶段 - 从 Smalltalk 中抽取』的描述来安装第 1 阶段迁移工具的, 则该文件应该具有正确的名称和位置。

用户响应: 验证 EGL 保留字文件是否位于正确的路径中并且是否具有正确的文件名。如果不是的话, 检查第 1 阶段迁移工具的安装步骤。

---

**HPT.EGL.0022.e** 部件 *partName* 具有无效的外部源格式。

说明: 第 1 阶段迁移工具进行了 3 次尝试来从资源库中抽取外部源格式。但是, 该部件的开始标记和结束标记未构成有效的配对 (例如, :record 与 :erecord)。第 1 阶段迁移工具将继续进行处理。但是, 迁移数据库未包含指定

*partName* 的正确“外部源格式”。

**用户响应:** 尝试查看资源库中的外部源格式以了解该部件是否具有任何明显错误。如果您正在使用远程资源库,则尝试将其复制至本地驱动器并再次运行第 1 阶段迁移工具。如果您无法解决该问题,请与 IBM 支持机构联系。

---

**HPT.EGL.0023.e** 由于部件重复,所以部件 *partName* 具有无效的外部源格式。

## VisualAge for Java 上的第 1 阶段

下列消息仅出现在 VisualAge Generator 到 EGL 迁移工具的 VisualAge for Java 版本中。

---

**HPT.EGL.0101.e** 当前包名 *vagenPackageName* - 生成了 EGL 包名 *eglPackageName*, 后者以 # 开头或使用保留字 *reservedWordList*。必须将其重命名。

**说明:** EGL 保留字不可以用作 EGL 包名的点表示法中的任何字。

**用户响应:** 使用第 1 阶段重命名规则来创建不违反 EGL 命名限制的 EGL 包名。请参阅第 173 页的附录 A, 『保留字』, 该附录列示了 EGL 保留字。确保所生成的 EGL 包名未以符号 # 开头。

---

**HPT.EGL.0102.e** 迁移集 *migrationSetName* - *migrationSetVersion* 引用项目 *projectName* 的版本 *projectVersion1* 和 *projectVersion2*。未创建该迁移集。

**说明:** 迁移工具为指定的迁移集版本扩展了高级 PLP 项目。扩展的高级 PLP 项目包含同一个项目的多个版本。迁移无法继续。

**用户响应:** 如果您正在使用 PLP 项目,则修改高级 PLP 项目以及它所引用的任何低级 PLP 项目,以使 PLP 链只包括每个项目的一个版本。如果您手工创建了迁移规划文件,则请修改该迁移规划文件,以便只对迁移集指定每个项目的一个版本。

---

**HPT.EGL.0103.e** 装入数据库驱动程序时发生错误。该驱动程序为 *driverName*。请进行检查以确保 **db2java.zip** 文件位于类路径中。

**说明:** 找不到在第 1 阶段首选项文件中指定的数据库驱动程序。

**用户响应:** 修改第 1 阶段首选项文件以指向正确的驱动程序名和位置。

---

**HPT.EGL.0104.e** 连接至数据库时发生错误。该数据库为 *databaseName*。

**说明:** 一个或多个具有相同部件类型的部件具有相同的部件名。迁移工具无法确定要将哪个“外部源格式”存储在迁移数据库中。

**用户响应:** 修改迁移集以使该迁移集不包含重复部件名。

---

**说明:** 第 1 阶段迁移工具无法连接至该迁移数据库。

**用户响应:** 确保已创建了指定的数据库。并且,查看第 1 阶段首选项文件中的用户标识和密码设置以确保它们正确。

---

**HPT.EGL.0105.e** 关闭数据库连接时发生错误。

**说明:** 第 1 阶段迁移工具无法关闭与迁移数据库的连接。

**用户响应:** 第 1 阶段迁移工具在尝试关闭数据库连接之前执行了落实。您应该能够关闭 VisualAge Generator 以强制关闭该连接。

---

**HPT.EGL.0106.e** 在方法 *methodName* 中访问资源库时出错。

**说明:** 第 1 阶段迁移工具的指定方法无法访问资源库。

**用户响应:** 如果正在使用远程资源库,则验证该资源库是否可供访问并且是否不存在网络问题。然后,再次尝试迁移。

---

**HPT.EGL.0107.e** 写 XML 文件 *fileName* 时发生错误。

**说明:** 第 1 阶段迁移工具无法写指定名称的文件。

**用户响应:** 验证是否有足够的空间可供该文件使用。然后,再次尝试迁移。

---

**HPT.EGL.0108.w** 由于名称 *partName* 包含无效的空格,所以未迁移 *partType* 部件。该部件位于包 *packageName* 中,版本为 *versionName*。

**说明:** 在 VisualAge Generator 中,可以创建部件名末尾包含空格的部件。这时,通常会有两个部件,这两个部件同名,只是其中一个部件的名称末尾包含空格。由于名称略有不同,所以这两个部件不是重复部件。但是,外部源格式文件中的部件名是完全相同的,尽管这些部件的其余

源代码略有不同。第 1 阶段迁移工具会将迁移集中任何包含空格的部件名省略掉。这是因为，以空格结尾的部件名不能被任何其它 VAGen 部件引用。当不存在具有类似名称的部件时，这种技术能够确保不能被其它部件引用的部件不会被迁移。当存在具有类似名称的部件时，这种技术能够确保第 2 阶段迁移工具将正确的部件定义转换为 EGL。

**用户响应：** 无。但是，您可能想在 VisualAge Generator 中查看部件以确定是否存在名称相似的部件。请使用 VAGen 部件浏览器并在迁移集中的所有部件当中搜索 *partName\**。

---

#### HPT.EGL.0109.e 发生了意外异常：

*javaExceptionStackTrace*。

**说明：** 在运行第 2 阶段或第 3 阶段迁移工具时发生了意外错误。

**用户响应：** 查看 *javaExceptionStackTrace*。根据错误的不同，可能有您可以忽略或更正的内容。例如：

- 您可以忽略指示无法转换的字符已被替换为替代字符的消息。如果外部源格式文件包含无效的字符，就会出现此消息。在迁移数据库中，该字符将被替换为空白。第 1 阶段迁移工具将继续进行处理。您可以将该迁移数据库用于第 2 阶段和第 3 阶段。

## VisualAge Smalltalk 上的第 1 阶段

下列消息仅出现在 VisualAge Generator 到 EGL 迁移工具的 VisualAge Smalltalk 版本中。

---

#### HPT.EGL.0201.e 当前应用程序名

*vagenApplicationName* - 生成了 EGL 包名 *eglPackageName*，后者以 # 开头或使用保留字 *reservedWordList*。必须将其重命名。

**说明：** EGL 保留字不可以用作 EGL 包名的点表示法中的任何字。

**用户响应：** 使用第 1 阶段重命名规则来创建不违反 EGL 命名限制的 EGL 包名。请参阅第 173 页的附录 A，『保留字』，该附录列示了 EGL 保留字。确保所生成的 EGL 包名未以符号 # 开头。

---

#### HPT.EGL.0202.e 迁移集 *migrationSetName* 引用配置映射 *configurationMapName*，后者未在资源库中定义。

**说明：** 第 1 阶段迁移工具为指定的迁移集扩展了高级配置映射。然而，当此工具扩展高级配置映射以及必需映射和应用程序的链时，一个或多个必需的映射在库中不存在。

**用户响应：** 确定是否应该将必需的映射包括在迁移集中。如果是的话，检查是否已从库中清除了所请求的配置映射

- 如果该消息指示由于 SQL 列太短而导致无法包含 EGL 文件名，则您可以更正问题。在这种情况下，由于迁移数据库中的信息无效，所以第 1 阶段迁移工具将停止进行处理。您可以通过修改 SQL 表定义以增加列长度并接着再次运行第 1 阶段来更正问题。但是，在增加 SQL 列的长度之前，请考虑您是否愿意在迁移之后滚动这些较长的名称以及较长的名称是否可能会超出 EGL 的限制。在修改了重命名规则或 SQL 列之后，再次运行第 1 阶段迁移。

如果您无法解决该问题，请与 IBM 支持机构联系以获取帮助。

---

#### HPT.EGL.0110.e 项目 *projectName* 版本 *versionName* 在资源库中未定义。

**说明：** 第 1 阶段迁移工具为迁移集（包括 PLP 项目链）扩展了高级 PLP。在 PLP 链中引用了指定的项目和版本，但它们在资源库中不可用。

**用户响应：** 确定是否应该将项目和版本包括在迁移集中。如果是的话，检查是否已从资源库中清除了所请求的项目版本。如果是这样的话，请复原项目，然后再次迁移。如果您无法解决该问题，请与 IBM 支持机构联系以获取帮助。

版本。如果是这样的话，请恢复所请求的配置映射，然后再次进行迁移。

---

#### HPT.EGL.0203.e *ProgramContext* 遇到了数据库错误 *errorMessage*。

**说明：** 发生了数据库错误。可能的问题包括模式名无效、用户权限受限或者缺少 SQL 表。

**用户响应：** 更正迁移首选项文件。如果问题仍然存在，请与 IBM 支持机构联系以获取帮助。

---

#### HPT.EGL.0204.e 连接至数据库时发生错误。 *ErrorMessage*。

**说明：** 在进行连接时发生了数据库错误。可能的问题是用户标识或密码无效，或者数据库名称无效。

**用户响应：** 更正迁移首选项文件。如果问题仍然存在，请与 IBM 支持机构联系以获取帮助。

---

#### HPT.EGL.0205.i 迁移已根据配置映射 *configMapName* 的 *v* 个版本生成了 *n* 个迁移集。首选项文件将版本深度指定为 *d*。

**说明：** 第 1 阶段迁移首选项文件指定您想要迁移由 *d* 指

定的版本数。第 1 阶段迁移工具应该已经生成了  $d$  个迁移集 - 指定配置映射的每个版本都有一个迁移集。但是，迁移工具只创建了由  $n$  指定的个数的迁移集。 $v$  指定的数目就是迁移工具在库中找到的指定配置映射的版本数。如果  $v$  小于  $d$ ，则表示存在的配置映射版本数与您预期的不同。在这种情况下， $n$  和  $v$  应该相等，这表示已在迁移集中生成了所有配置映射版本。如果  $v$  大于  $d$ ，则表示库中的配置映射版本较多。在这种情况下， $n$  与  $d$  应该相等，这表示已符合版本深度首选项。

用户响应: 无。

---

#### HPT.EGL.0206.e 迁移集 *migrationSetName* 遇到了装入错误。

说明: 无法将该迁移集装入到映像中。发生此问题的原因可能是迁移集包含重复的部件名。

用户响应: 查看系统抄本以确定错误的原因。更正问题，然后再次运行第 1 阶段迁移。

---

#### HPT.EGL.0207.w 由于名称 *partName* 包含无效的空格，所以未迁移 *partType* 部件。该部件位于应用程序 *applicationName* 中，版本为 *versionName*。

说明: 在 VisualAge Generator 中，可以创建部件名末尾包含空格的部件。这时，通常会有两个部件，这两个部件同名，只是其中一个部件的名称末尾包含空格。由于名称

.....

---

## VisualAge Generator 到 EGL 迁移工具发出的消息 - 第 2 阶段

在为 EGL 保留字执行任何必需的重命名操作之前，消息插入的内容总是 VAGen 部件名。

---

#### IWN.MIG.0001.e 对外部源格式文件 *fileName* 进行语法分析时发生异常 - 外部源格式头无效。

说明: 迁移工具只能处理从 VisualAge Generator 4.5 导出的外部源格式文件。指定的外部源格式文件的第一行不具有正确的 VisualAge Generator 4.5 外部源格式文件头。

用户响应: 将外部源格式文件导入到 VisualAge Generator 4.5 中。这将把当前部件转换为 VisualAge Generator 4.5 格式。然后，使用 VisualAge Generator 4.5 导出部件并再次运行迁移。

---

#### IWN.MIG.0002.e 对外部源格式文件 *fileName*、*partType* 和 *partName* 进行语法分析时发生异常 - *exceptionText*。

说明: 对 VisualAge Generator 生成的“外部源格式”进行语法分析时发生问题。此问题的可能原因是:

略有不同，所以这两个部件不是重复部件。但是，外部源格式文件中的部件名是完全相同的，尽管这些部件的其余源代码略有不同。第 1 阶段迁移工具会将迁移集中任何包含空格的部件名省略掉。这是因为，以空格结尾的部件名不能被任何其它 VAGen 部件引用。当不存在具有类似名称的部件时，这种技术能够确保不能被其它部件引用的部件不会被迁移。当存在具有类似名称的部件时，这种技术能够确保第 2 阶段迁移工具将正确的部件定义转换为 EGL。

用户响应: 无。但是，您可能想在 VisualAge Generator 中查看部件以确定是否存在名称相似的部件。请使用 VAGen 部件浏览器并在迁移集中的所有部件当中搜索 *partName\**。

---

#### HPT.EGL.0208.e 数据库列

*schemaName.tableName.columnName* 包含被截断的数据。

说明: 一个或多个重命名规则导致 EGL 项目名、包名或版本名太长，在对应的 SQL 列中放不下。

用户响应: 修改重命名规则，以使它们生成较短的 EGL 项目名、包名或版本名。此外，您可以修改 DB2 表以增加 SQL 列的长度。但是，在增加 SQL 列的长度之前，请考虑您是否愿意在迁移之后滚动这些较长的名称以及较长的名称是否可能会超出 EGL 的限制。在修改了重命名规则或 SQL 列之后，再次运行第 1 阶段迁移。

- 双引号未配对，这包括: 数据项的货币字段
- 控制部件中的注释定界符未配对。
- 本地语言字符对于语言环境来说无效。例如，在没有为双字节语言环境进行设置的工作站上尝试迁移使用双字节字符（如中文）的 VAGen 源代码。

用户响应: 在 VisualAge Generator 中更正部件，并再次导出“外部源格式”。然后，运行第 2 阶段迁移工具来处理该文件。如果无法在 VisualAge Generator 中更正部件，则与 IBM 支持机构联系以获取帮助。请准备好提供文件的“外部源格式”源代码。

---

#### IWN.MIG.0003.e 对文件 *fileName*、*partType* 和 *partName* 转换为 EGL 时发生异常 - *exceptionText*。

说明: 创建 EGL 源代码期间发生问题。*exceptionText* 标识所发生的特定问题。

**用户响应:** 与 IBM 支持机构联系以获取帮助。请准备好提供文件的“外部源格式”源代码。

---

**IWN.MIG.0047.i 迁移集 *Name* - 迁移已启动。**

**说明:** 这是一条参考消息，它指示迁移工具的状态。

**用户响应:** 无。

---

**IWN.MIG.0048.i 迁移集 *Name* - 迁移已完成。**

**说明:** 这是一条参考消息，它指示迁移工具的状态。

**用户响应:** 无。

---

**IWN.MIG.0049.i EGL *projectName*、*packageName* 和 *fileName* 的 *partType partName* - 迁移已启动**

**说明:** 这是一条参考消息，它指示迁移工具的状态。*partType* 是下列其中一项：程序、映射组或表。将同时迁移指定的 *partName* 的关联项。根据迁移数据库中的信息的不同，关联项可能与 *partName* 位于同一个文件中，也可能位于不同的项目、包或文件中。当程序的迁移启动时，将迁移每个相关联的映射组，接着迁移每个相关联的表。最后，迁移其余的任何关联项（记录、共享项和函数）。

**用户响应:** 无。

---

**IWN.MIG.0050.i 程序 *programName* - 其它关联项的迁移已启动**

**说明:** 这是一条参考消息，它指示迁移工具的状态。当程序的迁移启动时，将迁移每个相关联的映射组，接着迁移每个相关联的表。最后，迁移余下的任何关联项（记录、共享项和函数）。当程序的其余关联项的迁移启动时，发出消息 IWN.MIG.0050.i。

**用户响应:** 无。

---

**IWN.MIG.0051.e 对迁移集 *planName*、*partType* 和 *partName* 进行语法分析时发生异常 - 外部源格式头无效。**

**说明:** 迁移工具只能处理从 VisualAge Generator 4.5 导出的外部源格式文件。指定部件的“外部源格式”的第一行不具有对 VisualAge Generator 4.5 外部源格式文件正确的头。如果您已修改样本第 1 阶段迁移工具，或者如果您编写了自己的第 1 阶段迁移工具以装入迁移数据库，则可能会发生这种情况。

**用户响应:** 将外部源格式文件导入到 VisualAge Generator 4.5 中。这将把当前部件转换为 VisualAge Generator 4.5 格式。然后，使用第 1 阶段迁移工具来导出迁移集。

---

**IWN.MIG.0052.e 对迁移集 *planName*、*partType* 和 *partName* 进行语法分析时发生异常 - *exceptionText*。**

**说明:** 对 VisualAge Generator 生成的“外部源格式”进行语法分析时发生问题。此问题的可能原因是：

- 双引号不匹配，这包括：
  - 数据项的货币字段
- 控制部件中的注释定界符未配对。
- 本地语言字符对于语言环境来说无效。例如，在没有为双字节语言环境进行设置的工作站上尝试迁移使用双字节字符（如中文）的 VAGen 源代码。

**用户响应:** 在 VisualAge Generator 中更正部件并再次运行第 1 阶段迁移以更正数据库。然后，再次运行第 2 阶段迁移工具以处理经过更新的部件。如果无法在 VisualAge Generator 中更正部件，则与 IBM 支持机构联系以获取帮助。请准备好提供一个包含有问题的部件的小资源库（.dat 文件）。

---

**IWN.MIG.0053.e 对迁移集 *planName*、*partType* 和 *partName* 转换为 EGL 时发生异常 - *exceptionText*。**

**说明:** 创建 EGL 源代码期间发生问题。*exceptionText* 标识所发生的特定问题。

**用户响应:** 与 IBM 支持机构联系以获取帮助。请准备好提供部件的“外部源格式”源代码。

---

**IWN.MIG.0054.e 迁移集 *migrationSetName*、*partType* 和 *partName* 的外部源格式无效。**

**说明:** 为指定部件存储的“外部源格式”无效。迁移工具继续处理指定的迁移集中的其它部件。为了进行带关联部件迁移，迁移工具认为指定的部件不可用。迁移工具故意将无效的 EGL 存储在指定部件的迁移数据库中。存储的 EGL 是 EZE\_UNKNOWN\_PARTTYPE *partName*；这将确保“问题”视图中有个错误。

**用户响应:** 在 VisualAge Generator 中查看指定的部件。尝试导出该部件的“外部源格式”，并以单文件方式迁移该部件。

---

**IWN.MIG.0055.e 迁移已停止 - 超出错误限制。**

**说明:** 对于具有无效“外部源格式”的部件，超出了错误阈值。迁移工具已停止处理。

**用户响应:** 查看出现的所有 IWN.MIG.0054.e 消息。如果您已创建了自己的工具来装入迁移数据库，则可能是该工具将“外部源格式”代码装入迁移数据库的方式有问题。请参阅第 309 页的附录 G，『迁移数据库』以了解一些可能有助于确定问题原因的查询。

---

**IWN.MIG.0060.e** 装入数据库驱动程序时发生错误。驱动程序: *driverName*

说明: 找不到指定的数据库驱动程序。

用户响应: 更正数据库驱动程序名。并且, 确认数据库驱动程序位置正确。

---

**IWN.MIG.0061.e** 连接至数据库时发生错误。数据库: *databaseName.errorText*

说明: 迁移工具无法使用指定的模式名连接至指定的数据库。 *errorText* 字段提供有关连接失败原因的其它详细信息。

用户响应: 更正数据库名称。如果正在使用 COM.ibm.db2.jdbc.app.DB2Driver 来连接至远程数据库, 则确保已在本地将该数据库编目。

---

**IWN.MIG.0063.e** 关闭数据库连接时发生错误。

说明: 迁移已成功完成, 但迁移工具无法关闭数据库连接。

用户响应: 在尝试备份数据库之前关闭 EGL 开发环境。

---

**IWN.MIG.0070.e** 用户出口方法 *renameUserExitName* [*partName*] 不存在。

说明: 找不到您对“重命名”用户出口指定的 JAR 文件或者该 JAR 文件中的包和类。

用户响应: 检查您在 VAGen 迁移首选项中对“重命名”用户出口指定的 JAR 文件位置、包名和类名。

---

**IWN.MIG.0071.e** 用户出口方法 *renameUserExitName*[*partName*] 不具有必需的方法特征符。

说明: “重命名”用户出口要求包括具有特征符 *renameUserExit(String s, Connection c)* 的方法。在您在 VAGen 迁移首选项中对“重命名”用户出口指定的 JAR 文件、包和类中不存在此方法。

用户响应: 查看类定义并确保已包括了必需的方法特征符。并确保在 VAGen 迁移首选项中对“重命名”用户出口指定了正确的 JAR 文件位置、包名和类名。

---

**IWN.MIG.0072.e** 用户出口方法 *renameUserExitName* [*partName*] 强制执行 Java 语言访问控制, 并且底层的方法不可访问。

说明: 迁移应用程序无法访问指定的用户出口类的定义。

用户响应: 验证“重命名”用户出口类是否已被定义为公用类并且是否位于指定的包中。

---

**IWN.MIG.0073.e** 用户出口方法 *renameUserExitName* [*partName*] 通过抛出异常突然终止。

说明: 方法 *renameUserExit(String s, Connection c)* 返回了空值。迁移将继续进行。迁移工具将使用原始的 VAGen 部件名。

用户响应: 使用 try ... catch 块来将“重命名用户出口”中的代码括起来。如果发生了异常, 则返回原始 VAGen 部件名以避免接收到此消息。

---

**IWN.MIG.0080.i** 找不到 VAGen 迁移首选项文件 *pref\_store.ini*; 已采用缺省值。

说明: 没有 VAGen 迁移首选项文件。迁移工具使用首选项(例如, 重命名后缀和帮助映射后缀)的缺省值。这可能是因为在迁移期间指定了新的工作空间, 所以首选项不存在。首选项文件位于:

```
workspace-directory\metadata\plugins
  \com.ibm.etools.egl.vagenmigration
  \pref_store.ini
```

用户响应: 有关迁移首选项的缺省值的信息, 请参阅第 126 页的『VAGen 迁移语法首选项』。

---

**IWN.MIG.0081.i** 文件 *fileName* - 迁移已完成。

说明: 迁移工具已完成对指定文件的处理。

用户响应: 查看日志消息以了解迁移结果。

---

**IWN.MIG.0082.e** 文件 *fileName* - 未指定必需参数。

说明: 尚未指定一个或多个必需参数。 *-importFile* 参数总是必需的。如果 *-importFile* 参数指定了外部源格式文件, 则 *-eglFile* 和 *-package* 参数也是必需的。

用户响应: 查看批处理命令文件以确定哪些参数未被指定。添加参数, 然后再次运行批处理命令文件。

---

**IWN.MIG.0083.e** 文件 *fileName* - 未对参数 *parmName* 赋值。

说明: *parmName* 是下列其中一项: *-importFile*、*-eglFile* 和 *-package*。

用户响应: 更正批处理命令文件, 然后再次运行它。

---

**IWN.MIG.0084.e** 文件 *fileName* - 参数 *parmName* 和值 *value* 无效。

说明: *parmName* 是下列其中一项: *-importFile*、*-eglFile* 和 *-package*。参数名区分大小写。

用户响应: 更正批处理命令文件, 然后再次运行它。

---

---

**IWN.MIG.0085.e** 文件 *fileName* - 在参数列表中传递了无效的参数。

**说明:** 批处理命令文件有问题。不正确地输入了一个或多个参数。有效参数只有: `-importFile`、`-eglFile`、`-package` 和 `-overwrite`。

**用户响应:** 更正批处理命令文件, 然后再次运行它。

---

**IWN.MIG.0090.e** 程序 *programName* - 是一个 Web 事务程序, 迁移不支持该程序。

**说明:** 指定的程序是一个 Web 事务程序。迁移工具不迁移该程序部件。

**用户响应:** 无。您必须等待 EGL 的将来发行版才能迁移此程序。

---

**IWN.MIG.0091.e** 记录 *recordName* - 是用户界面记录, 迁移不支持该记录。

**说明:** 指定的记录是用户界面 (UI) 记录。迁移工具不迁移该记录部件。

**用户响应:** 无。您必须等待 EGL 的将来发行版才能迁移此记录。

---

**IWN.MIG.0092.e** 函数 *functionName* - 使用了带有用户界面记录的 **XFER**, 迁移不支持该记录。

**说明:** 指定的函数包含使用用户界面记录的 **XFER** 语句。此发行版的 EGL 不支持 Web 事务或用户界面记录。迁移工具将迁移此函数, 但是, 生成的 EGL 语法可能不是带有用户界面记录的 **XFER** 语句的正确替换项。但是, 这种技术可以最大程度地保留逻辑。

**用户响应:** 查看 EGL 函数。您将无法迁移任何使用此函数的 Web 事务程序或用户界面记录。因此, 在 EGL 的将来发行版之前, 您将无法使用此函数。您可能想要将此函数移至另一个项目中, 以避免 EGL “问题” 视图包含错误。

---

**IWN.MIG.0093.w** 函数 *functionName* - 使用了一个或多个 **EZEUIxxx** 特殊功能字。

**说明:** 指定的函数包含使用特殊函数数字 **EZEUIERR** 或 **EZEUILOC** 的语句。此发行版的 EGL 不支持 Web 事务和用户界面记录。但是, 存在 **EZEUIxxx** 特殊函数数字的 EGL 替换项。迁移工具将迁移此函数, 并将 **EZEUIxxx** 字转换为等效的 EGL 系统库函数。这样可以最大程度地保留逻辑。

**用户响应:** 查看 EGL 函数。您将无法迁移任何使用此函数的 Web 事务程序或用户界面记录。但是, 如果创建任何新的 EGL `pageHandler`, 则可能能够使用此函数。

---

**IWN.MIG.0094.e** 迁移不支持 **PSB** *psbName*。

**说明:** 迁移工具当前不支持 **PSB** 部件。迁移工具不迁移 **PSB** 部件。

**用户响应:** 查看 **PSB** 以及任何使用此 **PSB** 的程序。如果 **PSB** 只包含 **TP PCB**, 则当下列所有情况都成立时, 可能能够为另一环境生成该程序:

- 该程序不执行任何 **DL/I I/O**。
  - 该程序不将 **EZEDLPSB** 或 **EZEDLPCB** 作为调用自变量进行传递。
  - 该程序不期望接收 **EZELDPSB** 或 **EZEDLPCB** 作为被调用参数。
  - 该程序不使用任何 **EZEDLxxx** 状态字 (例如, **EZEDLKEY** 或 **EZEDLSTC**)。
  - 该程序不使用资源关联部件来使文件或打印输出与 **IMS** 消息队列、**GSAM** 或 **MFS** 相关联。
  - 该程序不期望调用只在 **IMS** 或 **IMS BMP** 环境中运行的程序或者被该程序调用。
- 

**IWN.MIG.0095.e** 函数 *functionName* - 迁移不支持 **EZESCRPT**。

**说明:** 指定的函数包含使用 **EZESCRPT** 特殊函数数字的语句。**VAGen** 迁移工具当前不支持 **EZESCRPT**。迁移工具将迁移此函数, 但注释掉使用 **EZESCRPT** 的语句。

**用户响应:** 查看 EGL 函数。在此发行版中, 您将无法生成或运行使用此函数的程序。

---

**IWN.MIG.0096.e** 函数 *functionName* - 使用了一个或多个 **EZEDLxxx** 状态字, 迁移不支持该状态字。

**说明:** 指定的函数包含使用一个或多个 **EZEDLxxx** 特殊函数数字 (例如, **EZEDLKEY** 和 **EZEDLSTC**) 的语句。**VAGen** 迁移工具当前不支持 **DL/I**。迁移工具将迁移该函数, 但是, 生成的 EGL 语法可能不是这个 **EZEDLxxx** 特殊函数数字的正确替换项。

**用户响应:** 查看 EGL 函数。在此发行版中, 您将无法生成或运行使用此函数的程序。您可能想要将此函数移至另一个项目中, 以避免 EGL “问题” 视图包含错误。

---

**IWN.MIG.0097.e** 程序 *programName* - 迁移不支持被调用参数 *parmName*。

**说明:** 一个 **VAGen** 程序指定了 **EZEDLPSB** 或 **EZEDLPCB** 作为被调用参数。迁移工具当前不支持 **DL/I** 和 **IMS** 运行时环境。迁移工具将迁移此程序, 但注释掉这个被调用参数的使用。

**用户响应:** 查看程序。此时, 您将无法测试或生成它。

---

---

**IWN.MIG.0098.e** 记录 *recordName* 是 DL/I 记录，迁移不支持该记录。

**说明:** 指定的记录名是一个 DL/I 记录。迁移工具当前不支持 DL/I。迁移工具不迁移该记录。

**用户响应:** 查看任何使用此记录的程序。此时，您将无法测试或生成那些程序。

---

**IWN.MIG.0099.e** 函数 *functionName* - 迁移不支持 DL/I I/O。

**说明:** 指定的函数对 DL/I 记录执行 I/O。VAGen 迁移工具当前不支持 DL/I。迁移工具将迁移该函数，但有关 I/O 语句的信息不正确。

**用户响应:** 查看 EGL 函数。在此发行版中，您将无法生成或运行使用此函数的程序。您可能想要将此函数移至另一个项目中，以避免 EGL “问题”视图包含错误。

---

**IWN.MIG.0101.e** 数据项 *dataItemName* - 无法确定 *editRoutineName* 的编辑例程类型；已假定为函数。

**说明:** VisualAge Generator 支持将 EZEC10、EZEC11、函数或表用作数据项的映射编辑例程。对于 *dataItem*，EGL 既支持 *validatorFunction* 属性也支持 *validatorDataTable* 属性。迁移工具按以下方式转换映射编辑例程：

- EZEC10 和 EZEC11 迁移为 *validatorFunction* 属性。
- 如果 *editRoutineName* 指定的部件在迁移期间可用并且是一个函数，则 *editRoutineName* 迁移为 *validatorFunction* 属性。由于在 VisualAge Generator 中表名限长 7 个字符，所以当 *editRoutineName* 的长度超过 7 个字符时，迁移工具也将编辑例程迁移为 *validatorFunction* 属性。
- 如果 *editRoutineName* 指定的部件可用并且是一个表，则 *editRoutineName* 迁移为 *validatorDataTable* 属性。由于 VisualAge Generator 仅将编辑消息与 EZEC10、EZEC11 或表配合使用，所以，当对项指定了编辑消息时，迁移工具也将编辑例程迁移为 *validatorDataTable* 属性。
- 如果 *editRoutineName* 指定的部件在迁移期间不可用，*editRoutineName* 的长度不超过 7 个字符并且未指定编辑消息，则迁移工具假定 *editRoutineName* 是一个函数并迁移为 *validatorFunction* 属性。仅在这种情况下才发出消息 IWN.MIG.0101.e。

**用户响应:** 如果指定的编辑例程不是函数，则修改 EGL *dataItem* 定义并将 *validatorFunction* 属性更改为 *validatorDataTable* 属性。有关其它注意事项，请参阅第 43 页的『共享数据项的映射项编辑例程』中有关编辑例程的信息。

---

**IWN.MIG.0102.w** 部件 *partName* 使用共享数据项 *dataItemName* - 无法迁移为基本定义；正在使用类型定义。

**说明:** 您选择了一个首选项，即每当在记录、表、被调用参数列表、函数参数列表或函数本地存储器中使用共享数据项时，将 VAGen 共享数据项迁移为 EGL 基本定义。在 *partName* 指定的部件中使用了 *dataItemName* 指定的项。但是，该数据项定义在迁移期间不可用。迁移工具使用数据项名来作为类型定义，因此迁移的代码将有效。

**用户响应:** 如果要使用类型定义，则不需要执行任何操作。如果要使用基本定义，则将指定的部件修改为使用正确的项特征。另外，将共享数据项包括在迁移集中（或者包括在外部源格式文件中，如果正在以单文件方式进行迁移的话），并再次迁移。

---

**IWN.MIG.0103.w** 数据项 *dataItemName* - 首选项导致 *evensql=y* 被忽略。

**说明:** 指定的数据项部件是带有 *evensql=y* 的 VAGen PACK (EGL 十进制) 项。VAGen 迁移首选项指定不使用 *evensql=y*。根据该首选项，迁移工具将 PACK 项转换为更高一级的奇数精度，最大长度为 18。EGL 精度的差别如下：

- 如果对某项指定了 *evensql=n*，则 EGL 精度将总是计算为 (VAGen bytes \* 2) - 1，最大值为 18。
- 如果对某项指定了 *evensql=y*，则 EGL 精度将根据首选项来计算：
  - 如果选择了该首选项（不使用 *evensql=y*），则 EGL 精度计算为 (VAGen bytes \* 2) - 1，最大值为 18。仅在此情况下才发出消息 IWN.MIG.0103.w。
  - 如果未选择该首选项（使用 *evensql=y*），则 EGL 精度计算为 (VAGen bytes \* 2) - 2，最大值为 18。不发出消息 IWN.MIG.0103.w。

**用户响应:** 无。但是，您可能想在任何 SQL WHERE 子句或 EGL prepare 语句中查看此项的使用情况。如果此项的定义未与 SQL 表定义完全匹配，性能可能会受到影响。有关详细信息，请参阅第 155 页的『除去对 VisualAge Generator 兼容性方式的使用』中有关 EVENSQL 的信息。

---

**IWN.MIG.0201.i** 记录 *recordName* 包含 Level-77 项；正在创建名为 *level77RecordName* 的附加记录。

**说明:** VisualAge Generator 支持工作存储器记录中的 Level-77 项。EGL 不支持 Level-77 项。EGL 确实允许定义独立数据项。迁移工具将包含 Level-77 项的工作存储器记录分割为 2 个独立的 *basicRecord* - 其中一个包含非 Level-77 项，另一个包含 Level-77 项。如果工作存储器记录只包含 Level-77 项，则迁移工具只创建 Level-77 *basicRecord*。如果程序指定了包含 Level-77 项的主工作存

存储器记录, 则迁移工具将原始 `basicRecord` 及 `Level-77 basicRecord` 的声明都包括在程序定义中。

**用户响应:** 无。有关其它注意事项, 包括 `recordName` 在程序和语句的迁移期间不可用时的效果, 请参阅第 46 页的『记录中的 `Level-77` 项』中有关记录中的 `Level-77` 项的信息。

---

#### **IWN.MIG.0202.i** 记录 `recordName` 重新定义了 `redefinedRecordName`。

**说明:** `recordName` 是 `VAGen` 重新定义的记录, 它将 `redefinedRecordName` 指定为正在重新定义的记录。`recordName` 为 `redefinedRecordName` 使用的同一个物理存储器提供不同的项布局。EGL 不将重新定义信息保留在记录部件中。该信息只保存在程序中。迁移工具将注释包括在 `recordName` 中以提供原始 `VAGen redefinedRecordName` 信息。当迁移程序时, 如果 `recordName` 可用并且在 `VisualAge Generator` 中生成覆盖定义, 则迁移工具对 `recordName` 声明包括 `redefines` 属性。

**用户响应:** 无。有关其它注意事项, 包括 `recordName` 在程序的迁移期间不可用时的效果, 请参阅第 45 页的『重新定义的记录』中有关重新定义的记录的信息。

---

#### **IWN.MIG.0203.e** 记录 `recordName` - 未包含任何项。

**说明:** `VisualAge Generator` 允许保存未包含任何项的记录部件。但是, 由于该记录无效, 所以不能在任何程序中使用该记录。EGL 不允许记录部件未包含任何项。迁移工具将迁移该记录。

**用户响应:** 确定是否仍需要使用该记录。如果需要的话, 编辑该记录并添加一个或多个数据项。如果不需要的话, 删除该记录。

---

#### **IWN.MIG.0204.e** 记录 `recordName` - 没有备用指定记录 `altspecRecord` 可用; 无法确定 SQL 表名。

**说明:** 记录 `recordName` 指定了名为 `altspecRecord` 的备用指定记录, 后者提供 `recordName` 的项结构。在 `VisualAge Generator` 中, 对于 SQL 记录, 备用指定记录还提供 SQL 表名。在 EGL 中, 备用指定记录仅通过使用 `embed` 语句来提供结构。必须在每个 SQL 记录部件定义中指定表名。当迁移 `recordName` 时, 被指定为备用指定记录的记录在迁移期间不可用。迁移工具无法确定正确的表名并将 `tableNames` 属性设置为 `###TABLES_NOT_FOUND###`, `recordName` 的定义无效。

**用户响应:** 编辑 `recordName` 并从 `VAGen` 备用指定记录 `altspecRecord` 复制 `tableNames` 和 / 或 `tableNameVariables` 属性。`tableNames` 属性提供实际的 SQL 表名。`tableNameVariables` 属性提供表名主变量。如果 `recordName` 混合引用实际 SQL 表名与 SQL 表名主变量,

则 `tableNames` 和 `tableNameVariables` 属性都可用。要了解其它注意事项, 请参阅第 47 页的『备用指定记录』中的信息。

---

#### **IWN.MIG.0205.e** 记录 `recordName` - 没有备用指定记录 `altspecRecord` 可用; 无法确定 SQL 键项。

**说明:** 记录 `recordName` 指定了名为 `altspecRecord` 的备用指定记录, 后者提供 `recordName` 的项结构。当 `VisualAge Generator` 确定 SQL 记录的缺省选择条件时, `VisualAge Generator` 将备用指定记录中任何指定了 `key=yes` 的项与 `recordName` 中指定的键项 (如果有的话) 合并。键是根据各个项在记录结构中的列示顺序合并的。在 EGL 中, 备用指定记录仅通过使用 `embed` 语句来提供结构。必须在每个 SQL 记录部件定义中指定所有键项。当迁移 `recordName` 时, 被指定为备用指定记录的记录在迁移期间不可用。迁移工具无法确定正确的键项, 并将 `keyItems` 属性设置为 `###KEYS_NOT_FOUND###`, 后跟 `recordName` 中的键项 (如果有的话)。`recordName` 的定义无效。

**用户响应:** 编辑 `recordName` 并更改 `keyItems` 属性, 以将 `###KEYS_NOT_FOUND###` 替换为 `VAGen` 备用指定记录 (`altspecRecord`) 中指定了 `key=yes` 的项名的列表。确保将备用指定记录中的键项与 `recordName` 的 `VAGen` 定义中指定的键项合并, 以使 `keyItems` 属性按照项在记录结构中的出现顺序来列示它们。如果某个项在备用指定记录中被指定为 `key=yes`, 并且在 `recordName` 中被指定为键项, 则在 `recordName` 中的 `keyItems` 的合并列表中只将该项包括一次。要了解其它注意事项, 请参阅第 47 页的『备用指定记录』中的信息。

---

#### **IWN.MIG.0206.i** SQL 记录 `recordName` - 包含键项 `keyItem`, 但未指定备用指定记录。

**说明:** 即使未指定备用指定记录, `VisualAge Generator` 也允许保存指定了键项的 SQL 记录。但是, 在这种情况下, `VisualAge Generator` 在测试和生成期间将忽略该键项。仅当还存在备用指定记录时, 该键项才有意义。

**用户响应:** 无。该键项在 `VisualAge Generator` 中被忽略。在迁移期间, 迁移工具将除去该键项。

---

#### **IWN.MIG.0207.i** 记录 `recordName` - 指定仅带 `Level-77` 项的备用指定记录 `altspecRecord`; 嵌入的语句被省略。

**说明:** `altspecRecord` 指定的记录是只包含 `Level-77` 项的工作存储器记录。当 `recordName` 将工作存储器记录指定为备用指定时, `VisualAge Generator` 只使用 `altspecRecord` 中的结构 (非 `Level-77` 项)。由于在 `altspecRecord` 的结构中没有项, 所以迁移工具将省略嵌入的语句。

**用户响应:** 无。但是, 您可能想要删除 `recordName`, 因

为它是空记录。务必删除程序中的所有对 *recordName* 的引用。

---

**IWN.MIG.0208.e** 记录 *recordName* - 备用指定记录  
*altspecRecord* 不可用; 无法确定  
*!itemColumnName* 变量的 SQL 列名。

**说明:** 记录 *recordName* 指定了名为 *altspecRecord* 的备用指定记录, 后者提供 *recordName* 的项结构。当 VisualAge Generator 确定 SQL 记录的缺省选择条件时, VisualAge Generator 将任何 *!itemColumnName* 变量转换为对应的 SQL 列名。在 EGL 中, 不支持 *!itemColumnName* 变量。在每个 SQL 记录部件定义的缺省选择条件中, 必须显式地命名 SQL 列。当迁移 *recordName* 时, 被指定为备用指定记录的记录在迁移期间不可用。迁移工具无法确定与缺省选择条件中的一个或多个 *!itemColumnName* 变量相对应的正确 SQL 列名。迁移工具使用 EGL 缺省选择条件中的 *!itemColumnName*。 *recordName* 的定义无效。

**用户响应:** 编辑 *recordName* 并更改 *defaultSelectCondition* 属性, 以将 *!itemColumnName* 变量替换为 VAGen 备用指定记录 (*altspecRecord*) 中的相应 SQL 列名。有关其它注意事项, 请参阅第 47 页的『备用指定记录』中有关 *!itemColumnName* 变量的信息。

---

**IWN.MIG.0209.e** 记录 *recordName* - 备用指定记录  
*altspecRecord* 不带任何项; 嵌入的语句被省略。

**说明:** 记录 *recordName* 指定了名为 *altspecRecord* 的备用指定记录, 后者提供 *recordName* 的项结构。但是, 该备用指定记录没有任何数据项。迁移工具将省略 *recordName* 的定义中的嵌入的语句。

**用户响应:** 无。但是, 您应该查看 *recordName* 和 *altspecRecord* 以确定是否需要包括数据项或者是否可以删除这两个记录。确保删除程序中对这些记录的所有引用。

---

**IWN.MIG.0210.e** 记录 *recordName* - 无法确定  
*!itemColumnName* 变量的列名。

**说明:** 指定的记录的缺省选择条件使用一个或多个 VAGen *!itemColumnName* 变量。VAGen *!itemColumnName* 变量指定 SQL 记录定义中与实际 SQL 列名相对应的项的名称。在测试和生成时, VisualAge Generator 根据 SQL 记录确定任何 *!itemColumnName* 变量的实际 SQL 列名。EGL 不支持 *!itemColumnName* 变量。而是, EGL 要求在任何已修改的 SQL 语句中使用实际的 SQL 列名。在 VisualAge Generator 中, 当 *recordName* 指定的记录无效时, 将发出消息 IWN.MIG.0210.e。在这种情况下, 记录使用一个或多个在该记录或其备用指定记录中未定义的 *!itemColumnName* 变量。迁移工具无法替换实际的 SQL 列名。

**用户响应:** 编辑该记录并将 *!itemColumnName* 变量更改为正确的 SQL 列名。

---

**IWN.MIG.0211.w** 记录 *recordName*、数据项  
*dataItemName* - 首选项导致 *evensql=y*  
被忽略。

**说明:** 指定的记录包含指定的非共享数据项。记录定义指定此非共享数据项为带有 *evensql=y* 的 VAGen PACK (EGL 十进制) 项。VAGen 迁移首选项指定不使用 *evensql=y*。根据该首选项, 迁移工具将 PACK 项转换为更高一级的奇数精度, 最大长度为 18。EGL 精度的差别如下:

- 如果对某项指定了 *evensql=n*, 则 EGL 精度将总是计算为 (VAGen bytes \* 2) - 1, 最大值为 18。
- 如果对某项指定了 *evensql=y*, 则 EGL 精度将根据首选项来计算:
  - 如果选择了该首选项 (不使用 *evensql=y*), 则 EGL 精度计算为 (VAGen bytes \* 2) - 1, 最大值为 18。仅在此情况下才发出消息 IWN.MIG.0211.w。
  - 如果未选择该首选项 (使用 *evensql=y*), 则 EGL 精度计算为 (VAGen bytes \* 2) - 2, 最大值为 18。不发出消息 IWN.MIG.0211.w。

**用户响应:** 无。但是, 您可能想在任何 SQL WHERE 子句或 EGL prepare 语句中查看此项的使用情况。如果此项的定义未与 SQL 表定义完全匹配, 性能可能会受到影响。有关详细信息, 请参阅第 155 页的『除去对 VisualAge Generator 兼容性方式的使用』中有关 EVENSQL 的信息。

---

**IWN.MIG.0301.e** 表名 *tableName* 是保留字。必须将其重命名。

**说明:** 迁移工具不会将表重命名。

**用户响应:** 必须更改该表的名称以及对它的所有引用。这包括下列位置中的引用:

- 程序使用声明语句
- 程序和函数中的逻辑语句
- 数据项 *validatorDataTable* 属性
- 表单字段 *validatorDataTable* 属性

如果要保留原始表名来作为所生成的 *dataTable* 的名称, 则将 *alias* 属性设置为原始 *dataTable* 名。如果未指定 *alias* 属性, 则务必更改对该 *dataTable* 名的任何非 EGL 引用, 包括 CICS 程序定义。

---

**IWN.MIG.0401.e** 映射组 (*formGroup*) 名称  
*mapGroupName* 是保留字。必须将其重命名。

**说明:** 迁移工具不会将映射组 (*formGroup*) 重命名。

**用户响应:** 您必须更改该 *formGroup* 的名称以及对它的所有引用, 包括程序使用声明语句中的引用。如果要保留原始映射组名来作为所生成的 *formGroup* 的名称, 则将

*alias* 属性设置为原始映射组 (*formGroup*) 名。如果未指定 *alias* 属性, 则务必更改对该 *formGroup* 名的任何非 EGL 引用, 包括 CICS 程序定义。

---

**IWN.MIG.0402.e** 映射组 *mapGroupName* - 多个设备具有相同的深度和宽度, 但具有不同的浮动区; 设备是: *devicesList*。

**说明:** VisualAge Generator 允许 (但不建议) 对具有相同设备大小的设备类型使用不同的浮动区大小。EGL 对每个设备大小只允许一个浮动区。迁移工具为每种设备类型迁移浮动区大小。该 *formGroup* 定义无效。此消息对 VisualAge Generator 中指定了不同浮动区信息的每组相同大小的设备类型重复一次。

**用户响应:** 对于每组具有相同大小的设备, 编辑 *formGroup* 并删除除一个浮动区说明以外的所有其它浮动区说明。

---

**IWN.MIG.0403.e** FormGroup *formGroupName* - 需要对 *formGroup* 中的嵌套表单进行编辑。

**说明:** 当以单文件方式进行迁移时, 迁移工具不将表单嵌套在 *formGroup* 中。而是, 迁移工具插入 EGL *use* 语句以指示属于 *formGroup* 的表单的名称。迁移工具在每个表单的开头和末尾包括注释以指示它所属的 *formGroup*。

**用户响应:** 编辑包含该 *formGroup* 的文件, 并移动表单以使它们嵌套在该 *formGroup* 中。该 *formGroup* 中的 *use* 语句指示了应该将表单移至的位置。将表单嵌套在 *formGroup* 中之后, 除去 *use* 声明语句。

---

**IWN.MIG.0404.w** 映射组 *mapGroupName* - 使用了不再受支持的设备 *deviceName* 以及大小 *depth* 和 *width*。必须对其进行更改。

**说明:** VisualAge Generator 允许使用 EGL COBOL 生成对浮动区和文本表单不再支持的一些设备类型。迁移工具将原始深度和宽度包括在 *ScreenFloatingArea* 属性内的 *screenSize* 属性中。但是, 在 EGL COBOL 生成中不再支持这个 *screenSize*。如果为 COBOL 生成, 则发生将发出错误消息。

**用户响应:** 如果您计划为 COBOL 生成, 则在 EGL 中编辑 *formGroup* 并除去这个深度和宽度的 *ScreenFloatingArea* 属性, 或者将深度和宽度更改为受支持的大小。可能还需要修改 *formGroup* 中的 *textForm*, 以便对变量和常量进行再定位以更好地适合于新的深度和宽度。

---

**IWN.MIG.0501.e** 帮助映射组 *mapGroupName* 包含带有变量字段的映射 *mapName* - *mapName* 与程序的主映射组中相同映射名冲突。

**说明:** VisualAge Generator 允许在程序的主映射组及其帮助映射组中使用相同的映射名。EGL 不允许在程序的两个 *formGroup* 中存在任何重复的表单名。即使程序未使用具有重复名称的表单, 此项限制也适用。如果程序的帮助映射组中的映射与程序的主映射组中的映射冲突并且只包含常量字段, 则迁移工具将帮助映射组中的映射重命名。如果程序的帮助映射组中的映射包含变量字段, 则迁移工具不将该映射重命名, 即使该名称与程序的主映射组中的映射名冲突亦如此。

**用户响应:** 编辑该帮助 *formGroup* 并更改表单的名称。并且, 务必更改表单定义以及在所有使用该 *formGroup* 的程序中对此表单的所有引用。有关其它注意事项, 请参阅第 52 页的『映射名和帮助映射名』中有关映射名的信息。

---

**IWN.MIG.0502.e** 映射组 *mapGroupName*, 映射 *mapName* 和变量字段 *mapItemName* - 无法确定 *editRoutineName* 的编辑例程类型; 已假定为函数。

**说明:** VisualAge Generator 支持将 EZE10、EZE11、函数或表用作映射变量的映射编辑例程。对于表单字段, EGL 既支持 *validatorFunction* 函数也支持 *validatorDataTable* 属性。迁移工具按以下方式转换映射编辑例程:

- EZE10 和 EZE11 迁移为 *validatorFunction* 属性。
- 如果 *editRoutineName* 指定的部件在迁移期间可用并且是一个函数, 则 *editRoutineName* 迁移为 *validatorFunction* 属性。由于在 VisualAge Generator 中表名限长 7 个字符, 所以当 *editRoutineName* 的长度超过 7 个字符时, 迁移工具也将编辑例程迁移为 *validatorFunction* 属性。
- 如果 *editRoutineName* 指定的部件可用并且是一个表, 则 *editRoutineName* 迁移为 *validatorDataTable* 属性。由于 VisualAge Generator 仅将编辑消息与 EZE10、EZE11 或表配合使用, 所以, 当对表单字段指定了编辑消息时, 迁移工具也将编辑例程迁移为 *validatorDataTable* 属性。
- 如果 *editRoutineName* 指定的部件在迁移期间不可用, *editRoutineName* 的长度不超过 7 个字符并且未指定编辑消息, 则迁移工具假定 *editRoutineName* 是一个函数并迁移为 *validatorFunction* 属性。仅在这种情况下才发出消息 *msgPrefix.0502.e*。

**用户响应:** 如果指定的编辑例程不是函数, 则修改表单字段并将 *validatorFunction* 属性更改为 *validatorDataTable* 属性。要了解其它注意事项, 请参阅第 55 页的『映射变量字段和编辑例程』中有关编辑例程的信息。

---

**IWN.MIG.0503.w** 映射组 *mapGroupName*, 映射 *mapName* - 在位置 (*row,column*), 未命名的变量字段被转换为常量字段。

**说明:** VisualAge Generator 允许 (但不建议) 在映射中使用未命名变量字段。程序不能访问这些未命名的变量字段。在测试和生成时, 将把未命名的变量字段转换为常量。由于一个或多个属性是非缺省值, 所以迁移工具将这个未命名的变量字段转换为常量。

**用户响应:** 查看表单定义并确保常量字段对于此字段来说是正确的迁移。有关其它注意事项, 请参阅第 57 页的『未命名变量字段』中有关未命名的变量字段的信息。

---

**IWN.MIG.0504.w** 映射组 *mapGroupName*, 映射 *mapName* - 已从位置 (*row,column*) 中除去了未命名的变量字段。

**说明:** VisualAge Generator 允许 (但不建议) 在映射中使用未命名变量字段。程序不能访问这些未命名的变量字段。在测试和生成时, 将把未命名的变量字段转换为常量。由于这个未命名的变量字段的所有属性都对一个常量字段指定了缺省值, 所以迁移工具除去了这个未命名的变量字段。EGL 不要求为表单显式地定义具有缺省属性的常量。

**用户响应:** 查看表单定义并确保除去此字段是正确的迁移。有关其它注意事项, 请参阅第 57 页的『未命名变量字段』中有关未命名的变量字段的信息。

---

**IWN.MIG.0505.w** 映射组 *mapGroupName*, 映射 *mapName* - 使用了不再受支持的设备 *deviceName* 以及大小 *depth* 和 *width*。必须对其进行更改。

**说明:** VisualAge Generator 允许使用 EGL COBOL 生成对 *textForm* 不再支持的一些设备类型。迁移工具将原始深度和宽度包括在所迁移的 *textForm* 的 *screenSizes* 属性中。但是, 在 EGL COBOL 生成中不再支持此屏幕大小。如果为 COBOL 生成, 则发生将发出错误消息。

**用户响应:** 如果您计划为 COBOL 生成, 则在 EGL 中编辑 *textForm* 并从 *ScreenFloatingArea* 属性中除去这个深度和宽度, 或者将深度和宽度更改为受支持的大小。可能还需要修改 *textForm*, 以便对变量和常量进行再定位以更好地适合于新的深度和宽度。

---

**IWN.MIG.0506.e** 映射组 *mapGroupName*, 映射 *mapName* - 行和列处的常量未受保护; 已更改为 **protect=skip**。

**说明:** 在显示器和打印机映射中, VisualAge Generator 允许未受保护的常量。EGL 要求将常量指定为 **protect=skip** 或 **protect=no**。迁移工具将为该字段设置 **protect=skip**。

**用户响应:** 如果可以接受 **protect=skip**, 则无需进行任何操作。当 **protect=skip** 时, 最终用户可以在任何紧靠变量字段前面的末尾继续输入, 附加的字符将放在下一个未受保护的变量字段中。**Protect=no** 防止最终用户在任何紧靠变量字段前面的末尾继续输入。最终用户必须按 **Tab** 键才能跳转到下一个变量字段以继续输入。

---

**IWN.MIG.0507.w** 映射组 *mapGroupName*, 映射 *mapName* - 行=0 和列=0 处的常量已更改为行=1 和列=1。

**说明:** VisualAge Generator 允许 (但不完全支持) 常量位于映射中的位置行=0 和列=0 处。位于行=0 和列=0 位置处的字段不能指定任何属性信息。EGL 不支持位于行=0 和列=0 位置处的任何字段。位于行=0 和列=0 位置处的字段是常量, 并且第一个字节被初始化为空白。迁移工具将位置更改为行=1 和列=1 并删除常量值的第一个字节。由于此信息未记录在外部源格式文件中, 所以迁移工具不包括任何字段表示属性 (如字段的颜色或突出显示)。

**用户响应:** 测试任何使用此表单的程序以确定显示外观是否发生更改。如果是的话, 则编辑表单并设置字段表示属性以得到期望的外观。

---

**IWN.MIG.0508.e** 映射组 *mapGroupName*, 映射 *mapName* - 不能更改行=0 和列=0 处的常量。

**说明:** VisualAge Generator 允许 (但不完全支持) 常量位于映射中的位置行=0 和列=0 处。位于行=0 和列=0 位置处的字段不能指定任何属性信息。EGL 不支持位于行=0 和列=0 位置处的任何字段。位于行=0 和列=0 位置处的字段是常量, 并且第一个字节不会被初始化为空白。迁移工具不更改该字段的位置, 这是因为此项更改可能导致在表单中移动或除去该常量并更改外观。由于此信息未记录在外部源格式文件中, 所以迁移工具不包括任何字段表示属性 (如字段的颜色或突出显示)。在“问题”视图中会出现错误。

**用户响应:** 编辑表单并更改该常量字段以将该字段定位在行=1 和列=1 处。如果有必要的话, 修改常量字段以除去一个字节来补偿现在占用行=1 和列=1 的属性字节。务必测试任何使用此表单的程序以确定显示外观是否发生更改。如果是的话, 则编辑表单并设置字段表示属性以得到期望的外观。

---

**IWN.MIG.0509.e** 映射组 *mapGroupName*, 映射 *mapName* - 不能更改行=0 和列=0 处的变量。

**说明:** 此映射可能来自跨系统产品或 VisualAge Generator 的旧版本。VisualAge Generator 4.5 不支持位于行=0 和列=0 位置处的变量。位于行=0 和列=0 位置处的字段不能指定任何属性信息。EGL 不支持位于行=0 和列=0 位置处的

任何字段。位于行=0 和列=0 位置处的字段是变量字段。迁移工具不更改该字段的位置，这是因为此项更改将导致该字段被移动或导致数据的第一个字节丢失。由于此信息未记录在外部源格式文件中，所以迁移工具不包括任何表示属性（如字段的颜色或突出显示）。在“问题”视图中会出现错误。

**用户响应：** 编辑表单并更改该字段以将该字段定位在行=1 和列=1 处。由于属性字节现在占用行=1 和列=1 的位置，所以，如果有必要的话，请修改该变量字段周围的其它字段以避免丢失任何数据。务必测试任何使用此表单的程序以确定显示外观是否发生更改。如果是的话，则编辑表单并设置字段表示属性以得到期望的外观。

---

#### **IWN.MIG.0510.e 映射组 *mapGroupName*，映射 *mapName* - *mapName* 与程序名冲突。**

**说明：** 此程序使用包含与程序同名的映射的映射组或帮助映射组。VisualAge Generator 允许映射名与程序名相同。EGL 不允许表单名与程序名相同。如果映射名与程序名相同，并且该映射不带任何变量字段，则迁移工具将程序的帮助映射组中的映射重命名。但是，在下列情况下，迁移工具不会将映射重命名：

- 该映射是程序的帮助映射组中带有变量字段的映射。
- 该映射是程序的主映射组中的任何映射。

**用户响应：** 编辑该 *formGroup* 并更改表单的名称。并且，务必更改表单定义以及在所有使用该 *formGroup* 的程序中对此表单的所有引用。有关其它注意事项，请参阅第 52 页的『映射名和帮助映射名』中有关映射名的信息。

---

#### **IWN.MIG.0601.w 函数 *functionName*，I/O 对象 *recordName* - 无法确定 UPDATE 选项的记录类型；已假定为非 SQL 记录。**

**说明：** 对于 SQL，如果程序中有多个 UPDATE 或 SETUPD 语句，则 VisualAge Generator 要求 REPLACE 函数指定对应的 UPDATE 或 SETUPD 语句的名称。EGL 使用 SQL 语句的 *resultSetID* 来指定 replace 语句与它的相应 get 或 open 语句之间的关系。*recordName* 指定的记录在迁移期间不可用。迁移工具假定 UPDATE 函数是用于非 SQL 记录的，并且不包括 *resultSetID*。

**用户响应：** 如果由于程序中存在多个用于同一个记录的 get 或 open 语句而导致验证或生成标志了错误，则编辑该函数并对 get for update 语句添加 *resultSetID*。在程序中，*resultSetID* 必须是唯一的。建议的 *resultSetID* 是函数名，并且后面跟着您在迁移期间使用的结果集后缀首选项。要了解其它注意事项，请参阅第 70 页的『带有多个更新的 SQL I/O』中的信息。

---

#### **IWN.MIG.0602.w 函数 *functionName* - 无法确定 I/O 对象 *mapName* 的映射类型；已假定为显示器映射。**

**说明：** VisualAge Generator 将 DISPLAY I/O 选项既用于显示器映射又用于打印机映射。EGL 只将 display 语句用于文本表单，并将 print 语句用于打印表单。在 VisualAge Generator 兼容性方式下，display 语句也可以用于打印表单。被指定为 *mapName* 的映射在迁移期间不可用。迁移工具假定该映射是显示器映射，并迁移为 EGL display 语句。

**用户响应：** 只要继续使用 VisualAge Generator 兼容性方式，或者如果该映射是显示器映射，就不需要执行任何操作。如果该映射是打印映射，并且您不想继续使用 VisualAge Generator 兼容性方式，则必须将此函数更改为使用 print 语句。有关其它注意事项，请参阅第 63 页的『用于映射的 DISPLAY 语句』。

---

#### **IWN.MIG.0603.e 函数 *functionName*，SQL I/O 对象 *recordName* - 无法确定 SQL 表名。**

**说明：** 在测试和生成时，VisualAge Generator 根据 SQL 记录确定 SQL 表名。EGL 要求将表名包括在任何已修改的 SQL 语句中。*recordName* 指定的记录在迁移期间不可用。迁移工具将 EZE\_UNKNOWN\_SQLTABLE 用作表名以确保验证和生成将标记错误。迁移工具还将语句的表标签设置为 T1。

**用户响应：** 编辑该函数并根据记录定义指定正确的表名和表标签。表名位于 EGL 记录定义的 *tableNames* 或 / 和 *tableNameVariables* 属性中。要了解其它注意事项，请参阅第 299 页的附录 D，『“问题”视图中的消息』中有关 EZE\_UNKNOWN\_TABLE 的信息。

---

#### **IWN.MIG.0604.e 函数 *functionName*，SQL I/O 对象 *recordName* - 无法确定 *!itemColumnName* 变量的列名。**

**说明：** 已修改的 SQL 语句使用一个或多个 VAGen *!itemColumnName* 变量。VAGen *!itemColumnName* 变量指定 SQL 记录定义中与实际 SQL 列名相对应的项的名称。在测试和生成时，VisualAge Generator 根据 SQL 记录确定任何 *!itemColumnName* 变量的实际 SQL 列名。EGL 不支持 *!itemColumnName* 变量。而是，EGL 要求在任何已修改的 SQL 语句中使用实际的 SQL 列名。*recordName* 指定的记录在迁移期间不可用。迁移工具使用已修改的 SQL 语句中的 *!itemColumnNames* 来提供尽可能多的信息。

**用户响应：** 编辑该函数并根据记录定义指定 SQL 列名。对于每个 *!itemColumnName*，在 SQL 记录定义中找出对应的项。该项的列名就是在 EGL I/O 语句中需要使用的列名。要了解其它注意事项，请参阅第 70 页的『SQL I/O 和 *!itemColumnName*』中的信息。

---

**IWN.MIG.0605.w** 函数 *functionName*, SQL I/O 对象 *recordName* - **SQLEXEC** 带有 **model=none**, 并且不带 SQL 子句。

**说明:** SQLEXEC 语句指定了模型类型 none, 但未包含任何 SQL 子句。结果, VisualAge Generator 对此语句生成 no op。迁移工具生成 EGL no op 语句 (仅仅是一个分号), 并且包括 VAGen Info 注释以指示模型类型为 none。如果该 VAGen 函数指定了错误例程, 则迁移工具包括适用于该错误例程的 *try*、*onException* 和 *end* 语句。

**用户响应:** 查看该函数以确定是应该除去还是应该扩展该 I/O 语句。

---

**IWN.MIG.0606.w** 函数 *functionName* - 无法确定 **CONVERSE** 的 *ioObjectName* 的 I/O 对象类型; 已假定为映射。

**说明:** VisualAge Generator 对显示器映射和用户界面记录都使用 CONVERSE I/O 选项。EGL 将 *converse* 语句既用于文本表单又用于 *pageHandler*, 但使用不同的语法。被指定为 *ioObjectName* 的部件在迁移期间不可用。迁移工具假定该 I/O 对象是显示器映射, 并使用文本表单所需的语法来将其迁移为 EGL *converse* 语句。

**用户响应:** 如果 EGL 验证或生成标志了错误, 则将该 *converse* 语句更改为使用 *pageHandler* 所需的 EGL 语法。

---

**IWN.MIG.0607.e** 函数 *functionName*, SQL I/O 对象 *recordName* - 无法确定 SQL I/O 子句 *clauseName*。

**说明:** 在 VisualAge Generator 中, 在某些时间点, 仅将已修改的 SQL 子句与该函数一起保存。在这种情况下, VisualAge Generator 将根据被指定为 I/O 对象的记录定义来为该函数创建其余的子句。指定的 *recordName* 在迁移期间不可用。迁移工具无法创建 SQL 子句。可能列示在此消息中的 *clauseName* 包括: SELECT、INTO、INSERTCOLNAME、VALUES 和 FORUPDATEOF。迁移工具将构建一个框架子句并包括 EZE\_UNKNOWN\_SQL\_CLAUSENAME。

**用户响应:** 找到消息中指定的记录。编辑该函数以包括缺少的 SQL 子句。要确定缺少的 SQL 子句所需的内容, 请使用 VAGen SQL 语句编辑器来查看 SQL 子句。请参阅第 68 页的『SQL I/O 和缺少的必需 SQL 子句』以了解更多详细信息和潜在问题。请参阅第 299 页的附录 D, 『“问题”视图中的消息』中有关 EZE\_UNKNOWN\_SQL\_clause name 的信息。

---

**IWN.MIG.0608.e** 函数 *functionName*, SQL I/O 对象 *recordName* - 无法确定备用指定 *altspecRecordName* 的 SQL I/O 子句 *clauseName*。

**说明:** 在 VisualAge Generator 中, 在某些时间点, 仅将已修改的 SQL 子句与该函数一起保存。在这种情况下, VisualAge Generator 将根据被指定为 I/O 对象的记录定义来为该函数创建其余的子句。指定的 *recordName* 在迁移期间可用。但是, *recordName* 指定了在迁移期间不可用的备用指定记录 *altspecRecordName*。迁移工具无法创建 SQL 子句。可能列示在此消息中的 *clauseNames* 包括: SELECT、INTO、WHERE、ORDERBY、INSERTCOLNAME、VALUES、FORUPDATEOF 和 SET。迁移工具将构建一个框架子句并包括 EZE\_UNKNOWN\_SQL\_CLAUSENAME。

**用户响应:** 找到消息中指定的备用指定记录。编辑该函数以包括缺少的 SQL 子句。要确定缺少的 SQL 子句所需的内容, 请使用 VAGen SQL 语句编辑器来查看 SQL 子句。请参阅第 68 页的『SQL I/O 和缺少的必需 SQL 子句』以了解更多详细信息和潜在问题。

---

**IWN.MIG.0701.e** 函数 *functionName* - 无法确定 **SET map PAGE** 语句中使用的 *mapName* 的映射类型; 已使用 **converseLib.EZE\_SETPAGE()**;

**说明:** VisualAge Generator 使用 SET map PAGE 来指示将要清除屏幕 (对于显示器映射) 或将要弹出纸张 (对于打印机映射)。EGL 只将 *converseLib.clearScreen()* 语句用于文本表单, 并将 *converseLib.pageEject* 语句用于打印表单。被指定为 *mapName* 的映射在迁移期间不可用。迁移工具不作任何关于映射类型的假定。而是, 迁移工具使用 *converseLib.EZE\_SETPAGE()* 语句来确保验证和生成将标记错误。迁移工具包括原始映射名来作为注释。

**用户响应:** 查看该函数并确定正确的选项是 *clearScreen()* 还是 *pageEject()*。要了解其它注意事项, 请参阅第 73 页的『SET map PAGE 语句』中的信息。

---

**IWN.MIG.0702.e** 函数 *functionName* - 由于缺少表 *tableName*, 所以无法确定 **RETR** 语句的返回列名。

**说明:** 如果未在 RETR 语句中指定返回列, 则 VisualAge Generator 根据指定的表的第二列自动确定返回列名。RETR 的 EGL 替换项是一个 *if* 语句, 后跟赋值语句。必须在该赋值语句中显式地指定返回列名。 *tableName* 指定的表在迁移期间不可用。迁移工具使用 EZE\_UNKNOWN\_RETURN\_COLUMN 来确保验证和生成将标记错误。如果在程序流语句中发生此问题, 则消息中将出现程序名而不是函数名。

**用户响应:** 编辑该函数并根据表定义指定正确的返回列。表中的第二列就是在 VisualAge Generator 中使用的缺省返回列。要了解其它注意事项, 请参阅第 73 页的『RETR 语句』中的信息。

---

**IWN.MIG.0703.e** 函数 *functionName* - 由于缺少表 *tableName*, 所以无法确定 **RETR** 语句的搜索列名。

**说明:** 如果未在 RETR 语句中指定搜索列, 则 VisualAge Generator 根据指定的表的第一列自动确定搜索列名。RETR 的 EGL 替换项是一个 *if* 语句, 后跟赋值语句。必须在该 *if* 语句中显式地指定搜索列名。 *tableName* 指定的表在迁移期间不可用。迁移工具使用 EZE\_UNKNOWN\_SEARCH\_COLUMN 来确保验证和生成将标记错误。如果在程序流语句中发生此问题, 则消息中将出现程序名而不是函数名。

**用户响应:** 编辑该函数并根据表定义指定正确的搜索列。表中的第一列就是在 VisualAge Generator 中使用的缺省搜索列。要了解其它注意事项, 请参阅第 73 页的『RETR 语句』中的信息。

---

**IWN.MIG.0704.e** 函数 *functionName* - 由于缺少表 *tableName*, 所以无法确定 **FIND** 语句的搜索列名。

**说明:** 如果未在 FIND 语句中指定搜索列, 则 VisualAge Generator 根据指定的表的第一列自动确定搜索列名。FIND 的 EGL 替换项是一个 *if* 语句, 后跟函数调用语句。必须在该 *if* 语句中显式地指定搜索列名。 *tableName* 指定的表在迁移期间不可用。迁移工具使用 EZE\_UNKNOWN\_SEARCH\_COLUMN 来确保验证和生成将标记错误。如果在程序流语句中发生此问题, 则消息中将出现程序名而不是函数名。

**用户响应:** 编辑该函数并根据表定义指定正确的搜索列。表中的第一列就是在 VisualAge Generator 中使用的缺省搜索列。要了解其它注意事项, 请参阅第 72 页的『FIND 语句』中的信息。

---

**IWN.MIG.0705.e** 函数 *functionName* - 使用 **CALL CSPTDLI** 语句。

**说明:** 指定的函数包含在此发行版的 EGL 中不支持的 CALL CSPTDLI 语句。

**用户响应:** 您必须等待 EGL 的将来发行版才能测试或生成任何使用此函数的程序。您可能想要将此函数移至另一个项目中, 以避免 EGL “问题” 视图包含错误。

---

**IWN.MIG.0706.e** 函数 *functionName* - 无法确定在 **IF**、**WHILE** 或 **TEST DUP** 语句中使用的 *recordName* 的记录类型; 已使用 **EZE\_DUPLICATE**。

**说明:** VisualAge Generator 支持对非 SQL 记录和 SQL 记录检查 DUP 和 UNQ。对于 SQL 记录, DUP 和 UNQ 是相同的。对于非 SQL 记录, EGL 既支持 duplicate 也支持 unique。对于 SQL 记录, EGL 仅支持 unique。 *recordName* 指定的记录在迁移期间不可用。迁移工具将 DUP 迁移为 EZE\_DUPLICATE 以确保验证和生成将标记错误。

**注:** 迁移工具将 TEST 语句迁移为 *if* 语句。

**用户响应:** 编辑该函数并将 EZE\_DUPLICATE 更改为下列其中一项:

- *unique* (对于 SQL 记录)
- *duplicate* (对于非 SQL 记录)

要了解其它注意事项, 请参阅第 77 页的『I/O 错误值 UNQ 和 DUP』中的信息。

---

**IWN.MIG.0707.e** 函数 *functionName* - 当在 **IF**、**WHILE** 或 **TEST NULL** 语句中使用项 *itemName* 时, 无法确定它是在记录中还是在映射中; 已使用 **EZE\_NULL**。

**说明:** VisualAge Generator 支持对映射项和 SQL 项检查 NULL。检查映射项是否为 NULL 等同于检查它是否为空白。检查 SQL 项是否为 NULL 将检查空指示符变量以确定该列在数据库中是否为空。等效的 EGL 语句是检查表单字段是否为空白以及检查 SQL 字段是否为空。 *itemName* 中指定的项在迁移期间不可用。迁移工具将 NULL 迁移为 EZE\_NULL 以确保验证和生成将标记错误。

**注:** 迁移工具将 TEST 语句迁移为 *if* 语句。

**用户响应:** 编辑该函数并将 EZE\_NULL 更改为下列其中一项:

- 空白 (对于表单字段)
- *null* (对于 SQL 字段)

要了解其它注意事项, 请参阅第 76 页的『检查 SQL 项和映射项是否为 NULL』中的信息。

---

**IWN.MIG.0708.w** 函数 *functionName* - 在除 **IF**、**WHILE** 或 **TEST** 以外的语句中使用了 **EZESYS**; 将使用旧的 **VAGen** 值。

**说明:** VisualAge Generator 支持在除 IF、WHILE 和 TEST 以外的语句中使用 EZESYS。迁移工具根据语句类型来迁移 EZESYS。在 IF、WHILE 和 TEST 语句中, 迁移工具将 EZESYS 转换为 sysVar.systemType, 并且还将值转

换为新的 EGL 值。对于 IF、WHILE 或 TEST 以外的语句，迁移工具将其转换为 *custPrefixEZESYS*，其中 *custPrefix* 是您为迁移设置的重命名前缀首选项。在迁移程序时，如果未选择 VAGen 迁移首选项不初始化以前的 EZESYS 值，迁移工具包括 *custPrefixEZESYS* 的声明和用于将 *custPrefixEZESYS* 初始化为原始 VAGen 值的语句。在此语句中，将使用原始 VAGen 值。

**用户响应：** 查看该函数并确定是要使用原始 VAGen 值还是要使用新的 EGL 值。如果要使用新的 EGL 值，将 *custPrefixEZESYS* 更改为 *sysVar.systemType*。

如果想要使用原始 VAGen 值并且在迁移期间选择了 VAGen 迁移首选项不初始化以前的 EZESYS 值，必须将 *custPrefixEZESYS* 的声明和初始化语句添加至使用指定函数的任何程序。如果想要使用原始 VAGen 值并且不选择 VAGen 迁移首选项不初始化以前的 EZESYS 值，则不需要任何更改。*custPrefixEZESYS* 的声明和初始化语句已经包括在所有已迁移程序中。

---

#### **IWN.MIG.0709.e 函数 *functionName* - 无法确定 XFER 的部件 *partName* 的类型；已假定为用户界面记录。**

**说明：** VisualAge Generator 使用 XFER 语句来转移至另一个程序或事务，并且首先向最终用户显示映射或用户界面记录。此发行版的 EGL 不支持 Web 事务或用户界面记录。迁移工具无法确定一个映射是否正被使用。迁移工具假定该语句是带有用户界面记录的 XFER，这是因为它的使用频率比带有映射的 XFER 高。迁移工具使用 forward 语句来迁移该函数。这是对 EGL 语法的智能猜测，在将来的发行版中，它将替换带有用户界面记录的 XFER，这样可以最大程度地保留逻辑。

**用户响应：** 查看该函数。如果该函数打算与映射配合使用，则将 forward 语句更改为 show 语句。有关 show 语句语法的详细信息，请参阅第 79 页的『XFER』或 EGL 联机帮助。如果该函数打算与 Web 事务配合使用，则您可能想将其移至另一个项目中，以避免 EGL “问题”视图包含错误。

---

#### **IWN.MIG.0801.e 程序名 *programName* 是保留字。必须将其重命名。**

**说明：** 迁移工具不会将程序重命名。

**用户响应：** 必须更改该程序的名称以及对该程序的所有引用，包括 call、transfer 和 show 语句中的引用。并更改与此程序相对应的任何绑定控制部件或链接编辑部件的名称。如果要保留原始程序名来作为所生成的程序的名称，则可以指定 *alias* 属性。如果未指定 *alias* 属性，则务必更改对该程序名的任何非 EGL 引用，包括 CICS 程序定义。

---

#### **IWN.MIG.0802.w 程序 *programName* - 允许隐式项。迁移不会为隐式项创建定义。**

**说明：** 在 VisualAge Generator 中，程序可以指定它允许隐式数据项。如果允许隐式数据项的程序实际地使用一个项而未定义该项，则 VisualAge Generator 在测试和生成时自动创建定义。EGL 不允许隐式项。迁移工具不创建隐式定义。

**用户响应：** 在 VisualAge Generator 中验证程序以确定是否正在使用任何隐式项。如果是的话，VisualAge Generator 将在验证消息中提供隐式项的定义。在 EGL 中，编辑程序定义并添加对应的 EGL 基本字段声明。不需要创建一个记录来包含该字段。可以将基本字段声明直接添加到程序中。

---

#### **IWN.MIG.0803.w 程序 *programName* - 使用 PSB *psbName*。**

**说明：** 程序在它的程序定义中指定了 PSB 名称。在此发行版中，EGL 不支持 IMS 和 IMS BMP 环境。在此发行版中，EGL 也不支持 DL/I。迁移工具为 PSB 创建使用声明语句，但将其注释掉。该程序可能是也可能不是有效的程序，这取决于它的期望运行时环境以及它对 DL/I 的使用。

**用户响应：** 查看该程序以确定它的期望运行时环境以及它是否使用 DL/I（包括对 CSPTDLI 的任何使用）。如果该 VAGen 程序是为 IMS 和 CICS 环境生成的，并且未使用 DL/I，则 PSB 仅用来与 IMS 环境进行交互。在此发行版中，可以为 CICS 环境生成 EGL 程序。您必须等待 EGL 的将来发行版才能为 IMS 环境进行生成。如果该程序使用 DL/I，则您必须等待 EGL 的将来发行版才能为 CICS 或 IMS 环境生成程序。

---

#### **IWN.MIG.0804.w 程序 *programName* - 无法确定与 CLOSE I/O 选项配合使用的 I/O 对象 *partName* 的部件类型；已假定为记录。**

**说明：** 在 VisualAge Generator 中，在测试或生成时将自动包括 I/O 对象。CLOSE I/O 选项既可以用于记录也可以用于打印映射。在 EGL 中，必须在程序中显式地声明 I/O 语句中使用的记录。表单不是显式声明的，但必须要有 formGroup 的 use 声明。将在指定的程序中使用该 CLOSE I/O 选项，并且将指定的 *partName* 用作该 CLOSE 的 I/O 对象。但是，指定的 *partName* 在迁移期间不可用。迁移工具假定该部件是记录，并且包括数据声明。

**用户响应：** 如果迁移工具的猜测不正确，则“问题”视图中将会有有一个错误。编辑该程序并除去 printForm 声明。

---

**IWN.MIG.0805.w** 程序 *programName* - 未指定执行方式; 已假定是不分段的。

**说明:** 在 VisualAge Generator 中, 在某些时间点, 不将执行方式与程序部件保存在一起。执行方式仅适用于主事务程序。指定的 *programName* 是一个主事务程序, 但未包括外部源格式的执行方式。迁移工具假定执行方式是非分段的, 并且将 `segmented=no` 属性包括在 EGL 源代码中。

**用户响应:** 如果该程序应该以非分段方式运行, 则不需要执行任何操作。如果该程序应该以分段方式运行, 则编辑该程序并将 `segmented` 属性更改为 `segmented=yes`。

---

**IWN.MIG.0806.w** 程序 *programName*, *tableName* 的使用声明 - 首选项导致 `deleteAfterUse=yes` 被省略。

**说明:** 指定程序包含指定表的使用声明。在 CSP (跨系统产品) 和某些发行版的 VisualAge Generator 中, 将在程序释放用于表的内存时确定 `keep after use` 标志。VisualAge Generator `keep after use` 标志通常迁移至表的使用声明上的 EGL `deleteAfterUse` 属性。但是, VAGen 迁移首选项指定迁移工具不应包括 `deleteAfterUse` 属性。

**用户响应:** 无。如果 VAGen 程序是使用 VisualAge Generator 4.5 修订包 4 生成的, 则行为上不应有任何差别。有关详细信息, 请参阅第 155 页的『除去对 VisualAge Generator 兼容性方式的使用』中有关 `deleteAfterUse` 的信息。

---

**IWN.MIG.1001.e** 生成选项部件 *partName* 是保留字。必须将其重命名。

**说明:** 迁移工具不会将程序重命名。由于程序可以有一个名为 *programName.opt* 的特殊生成选项部件, 所以迁移工具也不将生成选项部件重命名。

**用户响应:** 当更改程序名时, 务必更改对应的生成选项部件的名称。

---

**IWN.MIG.1002.w** 生成选项部件 *partName* - 已将 `/dbms=odbc` 迁移为 `dbms="DB2"`。

**说明:** 指定的生成选项部件包括 VAGen 生成选项 `/dbms=odbc`。EGL 仅支持 DB2 或 Oracle。迁移工具将 `/dbms=odbc` 转换为 EGL 构建描述符部件中的 `dbms="DB2"`。EGL 通过使用 JDBC 驱动程序来提供 DB2 支持。如果有用于数据库的 JDBC 驱动程序, 则可能能够使用构建描述符选项 `dbms="DB2"` 来作为数据库类型。

**用户响应:** 务必迁移、生成并测试各种使用 ODBC 支持的 VAGen 程序, 以确保所需的所有功能都能够正确地与 JDBC 驱动程序配合工作。

---

**IWN.MIG.1003.e** 生成选项部件 *partName* - 不支持 `/system=systemType`。

**说明:** 指定的生成选项部件包括 VAGen `/system` 生成选项, 并指定了 EGL 不支持的运行时环境。迁移工具将 `/system` 生成选项转换为 EGL 构建描述符部件中的注释。

**用户响应:** 确定其它构建描述符部件是否使用此构建描述符部件。如果不使用的话, 可以将此构建描述符部件删除。另外, 您可能想保留此构建描述符部件, 以便将来某个时候当 EGL 支持此运行时环境时进行参考。

---

**IWN.MIG.1004.w** 生成选项部件 *partName* - `/system=systemType` 要求设置 `destPort`。

**说明:** 指定的生成选项部件包括 `/system` 生成选项, 并指定了 COBOL 运行时环境。EGL 构建过程要求您使用 `destPort` 构建描述符选项来指定目标端口。

**用户响应:** 修改与该生成选项部件相对应的构建描述符部件并包括 `destPort` 构建描述符选项。在指定 `destPort` 的值时考虑下列各项:

- 对于 z/OS 环境, `destPort` 没有缺省值。必须添加 `destPort` 构建描述符选项, 并且该值必须与启动 z/OS 构建服务器的 JCL 中使用的值相匹配。用于启动 z/OS 构建服务器的样本 JCL 使用端口 5555。
- 对于 iSeries 环境, `destPort` 没有缺省值。必须添加 `destPort` 构建描述符选项。该值必须与 iSeries 构建服务器使用的值相匹配。

---

**IWN.MIG.1005.e** 生成选项部件 *partName* - 当前不支持 `/system=systemType`。

**说明:** 指定的生成选项部件包括 VAGen `/system` 生成选项, 并指定了 EGL 当前不支持的运行时环境。迁移工具将 `/system` 生成选项转换为 EGL 构建描述符选项以供将来使用。但是, 由于 EGL 当前不支持该运行时环境, 所以值不会出现在构建描述符部件编辑器中。可以使用文本编辑器来查看该值。

**用户响应:** 无。您应该保留此构建描述符部件, 以便有可能在 EGL 的将来发行版中使用。

---

**IWN.MIG.1099.e** 控制部件 *partName* - 不支持 `symparm symparmName`。

**说明:** 指定的控制部件使用或设置在 EGL 中不支持的 `symparm`。迁移工具通过使用原始 VAGen `symparm` 名称来“按现状”迁移该 `symparm`。但是, 在生成期间未设置此 `symparm`。

**用户响应:** 修改该控制部件以对该 `symparm` 设置缺省值。也可以修改该控制部件以使它不再使用指定的 `symparm`。

---

**IWN.MIG.1101.e** 链接表部件 *partName* 是保留字。必须将其重命名。

说明: 迁移工具不会将控制部件重命名。

用户响应: 修改链接选项部件名, 以使其不再是保留字。当更改链接选项部件名时, 务必更改所有引用该链接选项部件的构建描述符部件。

---

**IWN.MIG.1102.e** 链接表 *partName* - 不支持 **/contable=BINARY**。必须对其进行更改。

说明: VisualAge Generator 在链接表部件中支持 **/contable=BINARY**。EGL 不支持此值。迁移工具将 **conversionTable="BINARY"** 包括在 EGL 链接表部件中。此值是无效的, 但在生成之前不会检测到这一点。

用户响应: 必须将 **conversionTable** 值更改为 EGL 支持的值。有关 EGL **conversionTable** 属性以及可用选项的详细信息, 参阅联机帮助中有关链接部件的信息。

---

**IWN.MIG.1103.e** 链接表部件 *partName* - 不支持 **/remotecomtype=CICSECI**。缺省值为 **CICSECI**。

说明: VisualAge Generator 在链接表部件中支持 **/remotecomtype=CICSECI**。EGL 不支持此值。迁移工具将 **remoteComType="CICSECI"** 包括在 EGL 资源关联部件中。此值是有效的, 但可能不是您计划使用的值。如果要使用 **CICSECI**, 则需要设置 **ctgPort** 和 **ctgLocation**。

用户响应: 如果您计划使用 **CICSECI**, 则修改链接表部件并为指定了 **CICSECI** 作为 **remoteComType** 的条目设置 **ctgPort** 和 **ctgLocation** 值。如果您未计划使用 **CICSECI**, 则参阅联机帮助中有关链接部件的信息以了解关于 EGL **remoteComType** 属性以及 EGL 中的可用选项的详细信息。

---

**IWN.MIG.1104.e** 链接表部件 *partName* - 不支持 **/remotecomtype=communicationType**。必须对其进行更改。

说明: VisualAge Generator 在链接表部件中支持 **/remotecomtype=communicationType**。EGL 不支持此通信协议。迁移工具将 **remoteComType="communicationType"** 包括在 EGL 链接表部件中。此值无效, 必须对其进行更改。

用户响应: 确定您计划使用的通信协议。然后, 编辑该部件并将 **remoteComType** 更改为 EGL 支持的值。有关 EGL **remoteComType** 属性以及 EGL 中的可用选项的详细信息, 参阅联机帮助中有关链接部件的信息。

---

**IWN.MIG.1201.e** 资源关联部件 *partName* 是保留字。必须将其重命名。

说明: 迁移工具不会将控制部件重命名。

用户响应: 修改资源关联部件名, 以使其不再是保留字。当更改资源关联部件名时, 务必更改所有引用该资源关联部件的构建描述符部件。

---

**IWN.MIG.1202.e** 资源关联部件 *partName* - 不支持 **/filetype=fileType**。必须对其进行更改。

说明: 对于某些工作站环境, VisualAge Generator 支持 **/filetype=BTRIEVE** 和 **/filetype=MFCOBOL**。EGL 不支持这些文件类型。迁移工具将 **filetype** 信息包括在 EGL 资源关联部件中。此值无效, 并且将导致“问题”视图中有一个错误。

用户响应: 必须将 **filetype** 值更改为 EGL 支持的值。有关 EGL **filetype** 属性以及可用选项的详细信息, 参阅联机帮助中有关资源关联部件的信息。

---

**IWN.MIG.1203.e** 资源关联部件 *partName* - **/system** 是 *targetSystem*, 它不受支持; 已根据 **/filetype fileType** 信息进行迁移。

说明: 资源关联部件包含一个条目, 该条目使用指定的 *targetSystem*。在 EGL 中不支持此目标系统。迁移工具根据 *fileType* 来迁移该资源关联条目。例如, 如果 *targetSystem* 是 **mvs\*** 并且 *fileType* 是 **transient**, 则迁移工具将创建一个 EGL 资源关联条目并将 EGL 系统设置为 **mvs\***。这将是无效的, 并且将导致“问题”视图中有一个错误。可以通过指定有效的 EGL 系统 (对于此示例, 是 **zoscics**) 来更正该条目。如果 *targetSystem* 是 **ims\*** 并且 *fileType* 是 **smsgq**, 则迁移工具将该资源关联条目迁移为注释, 以便最大程度地保留信息以供将来参考。

用户响应: 如果在“问题”视图中有一个错误, 则通过指定有效的目标系统来更正资源关联部件中的条目。

---

**IWN.MIG.1301.e** 链接编辑部件 *partName* 是保留字。必须将其重命名。

说明: 迁移工具不会将程序重命名。由于程序名必须与其对应的链接编辑部件匹配, 所以迁移工具也不将链接编辑部件重命名。

用户响应: 当更改程序名时, 务必更改对应的链接编辑部件的名称。

---

**IWN.MIG.1401.e** 绑定控制部件 *partName* 是保留字。必须将其重命名。

说明: 迁移工具不会将程序重命名。由于程序名必须与其对应的绑定控制部件匹配, 所以迁移工具也不将绑定控制部件重命名。

用户响应: 当更改程序名时, 务必更改对应的绑定控制部件的名称。

---

## 从VisualAge Generator 传送至 EGL 迁移工具的消息 - 第 3 阶段

第 3 阶段仅生成跟踪消息。



## 附录 D. “问题”视图中的消息

在不确定情况下，迁移工具在迁移期间并不总是能够确定所要构建的正确 EGL 语法。这种情况通常是在迁移期间没有关联部件可用时发生的。在这些情况下，迁移工具有时会故意地创建无效的 EGL 语法，以便让一个错误出现在“问题”视图中。下表列示了将会导致 EGL 验证操作出错的特定文本字符串。特定的 EGL 错误消息可能会有所变化，但左列中列示的文本字符串将出现在被标记为出错的 EGL 语句附近。每当迁移工具包括这些文本字符串时，此工具还会向迁移日志发出一条消息。

表 135. 导致 EGL 语法或验证错误的 VAGen 迁移文本

EGL 语法中的 VAGen 迁移文本	问题和解决方案
###KEYS_NOT_FOUND###	<p><b>问题:</b> 当前 SQL 记录嵌入了另一个记录的结构。在迁移期间，嵌入的语句中指定的记录不可用。在 VAGen 中对当前 SQL 记录指定的任何键项都将包括在 <i>keyItems</i> 属性中，但是缺少嵌入的记录中的键。</p> <p><b>解决方案:</b> 找到嵌入的语句中指定的 EGL 记录。将 ###KEYS_NOT_FOUND### 文本替换为嵌入式 SQL 记录中列示的键。务必按照字段在嵌入的记录的记录结构中的出现顺序来将嵌入的记录键与当前记录的键项合并。如果在嵌入的记录的 <i>keyItems</i> 属性中也指定了当前记录的键项，则仅在 EGL <i>keyItems</i> 属性中将该字段包括一次。</p>
###TABLES_NOT_FOUND###	<p><b>问题:</b> 当前 SQL 记录嵌入了另一个记录的结构。在迁移期间，嵌入的语句中指定的记录不可用。</p> <p><b>解决方案:</b> 找到在嵌入的语句中命名的 EGL 记录，并将 <i>tableNames</i> 和 <i>tableNameVariables</i> 属性复制到当前 SQL 记录中。</p>
EZE_DUPLICATE	<p><b>问题:</b> VAGen IF、WHILE 或 TEST 语句中指定的记录在迁移期间不可用。</p> <p><b>解决方案:</b> 找到 EGL <i>if</i> 或 <i>while</i> 语句中指定的 EGL 记录。将 EZE_DUPLICATE 更改为下列其中一个值:</p> <ul style="list-style-type: none"> <li>• <i>duplicate</i> (对于非 SQL 记录)</li> <li>• <i>unique</i> (对于 SQL 记录)</li> </ul>
EZE_NULL	<p><b>问题:</b> 迁移工具无法确定 VAGen IF、WHILE 或 TEST 语句中指定的项是位于 SQL 记录中还是位于映射中。</p> <p><b>解决方案:</b> 查看该程序并确定该字段是在 SQL 记录中还是在表单上。对于 SQL 字段，将 EZE_NULL 替换为 <i>null</i>，对于表单字段，替换为 <i>blanks</i>。</p>
EZE_SETPAGE();	<p><b>问题:</b> 在 VAGen SET map PAGE 语句上指定的映射在迁移期间不可见。</p> <p><b>解决方案:</b> 找到 // VAGen Info 注释中指定的表单，该注释伴随着 EZE_SETPAGE() 语句一起出现。将 EZE_SETPAGE 更改为下列其中一个值:</p> <ul style="list-style-type: none"> <li>• <i>clearScreen()</i> (对于文本表单)</li> <li>• <i>pageEject()</i> (对于打印表单)</li> </ul>

表 135. 导致 EGL 语法或验证错误的 VAGen 迁移文本 (续)

EGL 语法中的 VAGen 迁移文本	问题和解决方案
EZE_UNKNOWN_PARTTYPE	<p><b>问题:</b> 存储在迁移数据库中的“外部源格式”无效。迁移工具无法确定部件类型，并且无法将该部件转换为 EGL 语法。</p> <p><b>解决方案:</b> EZE_UNKNOWN_PARTTYPE 语句中指定的部件无效。如果仅有少数部件会发生此问题，则尝试从 VisualAge Generator 中导出“外部源格式”，并以单文件方式迁移这些部件。</p> <p>如果您已创建了自己的工具来装入迁移数据库，则可能是该工具将“外部源格式”代码装入迁移数据库的方式有问题。请参阅第 309 页的附录 G, 『迁移数据库』以了解一些可能有助于确定问题原因的查询。</p>
EZE_UNKNOWN_RETURN_COLUMN	<p><b>问题:</b> VAGen RETR 语句中指定的 VAGen 表在迁移期间不可用。</p> <p><b>解决方案:</b> 找到赋值语句中指定的 EGL dataTable，并将 EZE_UNKNOWN_RETURN_COLUMN 替换为该 dataTable 中的第二列的名称。</p>
EZE_UNKNOWN_SEARCH_COLUMN	<p><b>问题:</b> VAGen FIND 或 RETR 语句中指定的 VAGen 表在迁移期间不可用。</p> <p><b>解决方案:</b> 找到 if 语句中指定的 EGL dataTable，并将 EZE_UNKNOWN_SEARCH_COLUMN 替换为该 dataTable 中的第一列的名称。</p>
EZE_UNKNOWN_SQLTABLE	<p><b>问题:</b> 被命名为 I/O 对象的 SQL 记录在迁移期间不可用。迁移工具无法确定 EGL I/O 语句的正确 tables 子句。</p> <p><b>解决方案:</b> 找到该 I/O 语句中指定的记录，并根据该记录的 tableNames 和 / 或 tableNameVariables 属性来确定正确的 tables 子句。</p>
EZE_UNKNOWN_SQL_FORUPDATEOF	<p><b>问题:</b> VisualAge Generator 创建了 SQL UPDATE 或 SETUPD I/O 选项的缺省 for update of 子句。被命名为 I/O 对象的 SQL 记录在迁移期间不可用。因此，迁移工具无法确定 EGL I/O 语句的正确 for update of 子句。</p> <p><b>解决方案:</b> 找到该 I/O 语句中指定的记录，并根据该记录的字段列表来确定正确的 for update of 子句。在 VisualAge Generator 中，缺省 for update of 子句是记录中的列名列表，并且列名顺序就是字段在记录中的列示顺序，但省略下列内容：</p> <ul style="list-style-type: none"> <li>• 该记录的 EGL keyItems 属性中列示的任何列名。</li> <li>• 与 EGL isReadOnly=yes 属性一起指定的任何列名。</li> </ul> <p>如果该 I/O 语句中指定的记录嵌套了另一个 SQL 记录，则执行下列操作：</p> <ul style="list-style-type: none"> <li>• 使用 embed 语句中指定的记录来确定列的顺序以及 isReadOnly=yes 属性。</li> <li>• 使用该 I/O 语句中指定的记录（嵌入记录）来确定 keyItems 属性。</li> </ul> <p>如果在 EGL prepare 语句中使用了 for update of 子句，则用双引号将列名列表引起来。</p>

表 135. 导致 EGL 语法或验证错误的 VAGen 迁移文本 (续)

EGL 语法中的 VAGen 迁移文本	问题和解决方案
EZE_UNKNOWN_SQL_INSERTCOLNAME	<p><b>问题:</b> VisualAge Generator 为 SQL ADD I/O 选项创建了缺省的列列表。被命名为 I/O 对象的 SQL 记录在迁移期间不可用。因此, 迁移工具无法为 EGL <i>add</i> 语句确定正确的列名列表。</p> <p><b>解决方案:</b> 找到该 I/O 语句中指定的记录, 并根据该记录的字段列表来确定正确的列列表。在 VisualAge Generator 中, 缺省列名列表是记录中的列名列表, 并且列名顺序就是字段在记录中的列示顺序, 但省略任何被指定了 EGL <i>isReadOnly=yes</i> 属性的列名。如果该 I/O 语句中指定的记录嵌入了另一个记录, 则使用嵌入的语句中指定的记录来确定列顺序以及 <i>isReadOnly=yes</i> 属性。决不会在 EGL <i>prepare</i> 语句中使用这个列名列表。</p>
EZE_UNKNOWN_SQL_INT0	<p><b>问题:</b> VisualAge Generator 为 SQL INQUIRY、SETINQ、UPDATE 或 SETUPD I/O 选项的 <i>into</i> 子句创建了缺省的数据项列表。被命名为 I/O 对象的 SQL 记录在迁移期间不可用。因此, 迁移工具无法确定 EGL I/O 语句的正确 <i>into</i> 子句。</p> <p><b>解决方案:</b> 找到该 I/O 语句中指定的记录, 并确定 <i>into</i> 子句的正确字段列表。在 VisualAge Generator 中, 缺省字段列表是记录中的列名列表, 并且列名顺序就是字段在记录中的列示顺序。如果该 I/O 语句中指定的记录嵌入了另一个记录, 则使用 <i>embed</i> 语句中指定的记录来确定字段顺序。在 EGL <i>prepare</i> 语句中, 决不会使用 <i>into</i> 子句。</p>
EZE_UNKNOWN_SQL_SELECT	<p><b>问题:</b> VisualAge Generator 为 SQL INQUIRY、SETINQ、UPDATE 或 SETUPD I/O 选项的 <i>select</i> 子句创建了缺省列列表。被命名为 I/O 对象的 SQL 记录在迁移期间不可用。因此, 迁移工具无法确定 EGL I/O 语句的正确 <i>select</i> 子句。</p> <p><b>解决方案:</b> 找到该 I/O 语句中指定的记录, 并确定 <i>select</i> 子句的正确列名列表。在 VisualAge Generator 中, 缺省列名列表是记录中的列名列表, 并且列名顺序就是字段在记录中的列示顺序。如果该 I/O 语句中指定的记录嵌入了另一个记录, 则使用 <i>embed</i> 语句中指定的记录来确定列的顺序。如果在 EGL <i>prepare</i> 语句中使用了 <i>select</i> 子句, 则用双引号将列名列表引起来。</p>
EZE_UNKNOWN_SQL_VALUES	<p><b>问题:</b> VisualAge Generator 为 SQL ADD I/O 选项创建了缺省的数据项列表。被命名为 I/O 对象的 SQL 记录在迁移期间不可用。因此, 迁移工具无法为 EGL <i>add</i> 语句确定正确的字段名列表。</p> <p><b>解决方案:</b> 找到该 I/O 语句中指定的记录, 并根据该记录的字段列表来确定正确的字段列表。在 VisualAge Generator 中, 缺省字段名列表是记录中的字段列表, 并且字段顺序就是那些字段在记录中的列示顺序, 但省略任何被指定了 EGL <i>isReadOnly=yes</i> 属性的字段。如果该 I/O 语句中指定的记录嵌入了另一个记录, 则使用 <i>embed</i> 语句中指定的记录来确定列顺序以及 <i>isReadOnly=yes</i> 属性。决不会在 EGL <i>prepare</i> 语句中使用该 <i>values</i> 子句。</p>



---

## 附录 E. “问题”视图中的 IWN.xxx 消息

从 VisualAge Generator 迁移的 EGL 源代码中要比完全在 EGL 中开发的代码中更容易出现某些 IWN.SYN、IWN.VAL 和 IWN.XML 消息。本节列示对迁移的代码具有特殊含义的消息。

---

### IWN.VAL 消息

---

**IWN.VAL.4300.e** 无法解析名为 *partName* 的部件，或者未将其解析为下列其中一种类型：  
*partType*。

**说明：** 在很多情况下会发生此问题：

- 指定的控制部件不存在。
- 指定的控制部件与发生错误的控制部件不在同一个项目或包中。由于这些控制部件在 VisualAge Generator 中没有关联项，所以迁移工具不会为这些控制部件创建 import 语句。
- XML 解析器无法全面地处理 .eglbld 文件。在这种情况下，指定的控制部件与发生错误的控制部件可能存在于同一个文件中。请检查消息 *IWN.XML.3999.e XML 验证错误 - 已经对元素 “yyyyy” 指定了属性 “xxxxx”*。此消息指示在同一个控制部件中多次指定了 xxxxx 属性。

**用户响应：** 如果指定的控制部件不存在，则请创建该部件或者除去对它的引用。

如果指定的控制部件位于另一个项目中，则请更新当前项目的属性中的“EGL 构建路径”以包括指定的控制部件所在的项目。如果指定的控制部件位于另一个包中，则编辑当前 .eglbld 文件以便为指定的控制部件所在的包添加 import 语句。

如果出现消息 *IWN.XML.3999.e*（此消息指示已指定了属性 xxxxx），则请编辑当前 .eglbld 文件，以使 xxxxx 在控制部件中只被指定一次。当您保存 .eglbld 文件时，“问题”视图中的消息应该会被更新。由于 XML 解析器在解析 .eglbld 文件中的第一个重复属性时停止处理，所以，您可能必须先解决数个错误，之后整个文件才能被解析。当所有 *IWN.XML.3999.e* 消息都被解决之后，指定的控制部件就应该可用（如果它与引用控制部件位于同一个 .eglbld 文件中的话）。

---

**IWN.VAL.4929.e** 无法解析程序 *programName* 中的 *formGroupName* 的 use 语句。

**说明：** 程序在 VisualAge Generator 中指定了映射组，但在程序中未将任何显示或转换 I/O 选项用作被调用参数，也未将任何映射用作被调用参数。另一种可能情况是程序

指定了帮助映射组，但该程序使用的任何映射都未指定帮助映射。

**用户响应：** 更改该程序以执行下列操作之一：除去 formGroup 的 use 语句，或者为包含 formGroup 的包添加 import 语句。如果在 VisualAge Generator 中不存在 formGroup 部件，则可能需要创建一个。

---

**IWN.VAL.5100.e** xxxxx 不是 yyyyy 系统字的有效限定符。

**说明：** 一个以前在 VisualAge Generator 中有效的 EZE 字当前不受 EGL 支持。迁移工具迁移 EZE 字，并进行有关将来可能要进行哪些 EGL 替换的“最佳猜测”。这将保留程序逻辑。

**用户响应：** 编辑该函数并进行逻辑更改，以便不再使用此系统字。也可以创建新项目来保留当前不能使用的已迁移函数。将包含当前不受支持的 VAGen 值的所有函数移至这个新项目中。

---

**IWN.VAL.5101.e** *mainFunctionName*。在这个语句位置使用 xxxxx 系统字是无效的。

**说明：** 使用指定的主函数的程序又调用了其它函数。函数调用链中的其中一个函数在一个语句中使用指定的系统字。迁移工具总是对作为 VAGen EZE 字的替换项的 EGL 系统字进行限定。如果未使用 sysLib、sysVar、mathLib、strLib、VGLib、VGVar、converseLib、converseVar 或 javaLib 对 xxxxx 系统字进行限定，则最有可能的原因如下所示：

- 该 VAGen 程序允许隐式数据项，并且 xxxxx 的定义是在生成期间自动创建的。EGL 不允许隐式数据项。迁移工具也不会创建隐式数据项定义。
- 未将记录、映射或表包括在迁移集中，因此迁移工具无法在程序中包括必需的 import 语句。

**用户响应：** 检查 VAGen 程序是否允许隐式项。如果是的话，则在 VisualAge Generator 中验证该程序。在 VAGen 的“查看消息”列表中将会有一条消息，该消息提供了该隐式数据项的正确定义。将该数据项的定义添加到程序的声明部分中。如果该 VAGen 程序不允许隐式项，则在

VisualAge Generator 中创建该程序的关联项列表。使用 VAGen 的“引用”工具来从关联项列表中搜索指定的数据项。“引用”工具的结果提供了有关迁移集中可能缺少哪些记录、映射或表的线索。

---

**IWN.VAL.5168.e** 在 **Is/Not** 表达式中使用 `xxxxx` 是无效的。

**说明:** 在 VisualAge Generator 中, 指定的值是 EZESYS 的有效值。此值在 EGL 中没有对应的值。迁移工具将迁移 VisualAge Generator 值以保留程序逻辑。

**用户响应:** 编辑该函数并进行逻辑更改, 以便不再使用此值。也可以创建新项目来保留当前不能使用的已迁移函数。将包含当前不受支持的 VAGen 值的所有函数移至这个新项目中。

---

**IWN.VAL.6503.e** `xxxxx SQL I/O` 语句的子句顺序混乱。`yyyyy` 必须在 `zzzzz` 子句的前面。

**说明:** 请参阅消息 IWN.VAL.6506.e。

**用户响应:** 请参阅消息 IWN.VAL.6506.e。

---

**IWN.VAL.6506.e** `xxxxx SQL I/O` 语句只允许一个 `yyyyy` 子句。

**说明:** 在很多情况下会出现此消息:

- 在 VisualAge Generator 中, 允许将 SQL 关键字用作列名。在 EGL 中, 不允许使用某些 SQL 关键字。
- 在 VisualAge Generator 中, 某些 SQL 子句不必用括号括起来。在 EGL 中, 这些子句必须用括号括起来。例如, 子查询及其相关 FROM、WHERE、GROUP BY 和 ORDER BY 子句必须括在括号中。

**用户响应:** 如果不使用子查询, 请参阅第 177 页的『需要进行特殊处理的 SQL 保留字』以获取 SQL 关键字和 SQL 技巧的列表, 可使用它们来解决列名问题。

如果要使用子查询, 为子查询及其相关 FROM、WHERE、GROUP BY 和 ORDER BY 子句加上括号。考虑以下 SQL 语句, 例如:

```
with #sql{
  SELECT EMPNO, COUNT(*)WORKDEPT
  FROM EMPLOYEE T1
  WHERE WORKDEPT LIKE '%E%'
  GROUP BY WORKDEPT
  UNION ALL
  SELECT EMPNO
  FROM EMP_ACT
  WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112')}
}
```

更改 SQL 语句, 为第二个 SELECT 语句加上括号, 如下所示:

```
with #sql{
  SELECT EMPNO, COUNT(*)WORKDEPT
  FROM EMPLOYEE T1
  WHERE WORKDEPT LIKE '%E%'
  GROUP BY WORKDEPT
  UNION ALL
  ( SELECT EMPNO
  FROM EMP_ACT
  WHERE PROJNO IN('MA2100', 'MA2110', 'MA2112'))
}
```

---

**IWN.VAL.6507.e** `xxxxx SQL I/O` 语句不允许 `yyyyy` 子句。

**说明:** 请参阅消息 IWN.VAL.6506.e。

**用户响应:** 请参阅消息 IWN.VAL.6506.e。

---

**IWN.VAL.6620.e** `functionName` - 变量访问 `xxxxx` 是不确定的。

**说明:** 确定该问题是否是对 call 语句发生的以及 `xxxxx` 是否是未限定的数据项。如果是 call 语句的问题, 则检查该数据项是否位于与程序的 `inputRecord` 属性相关联的 Level-77 记录中。如果在 CALL 语句上使用了未限定的项, 则 VisualAge Generator 使程序的主工作存储器记录中的 Level-77 项具有优先权。然而, EGL 并未向 call 语句提供相同的优先顺序。

**用户响应:** 如果是 call 语句的问题, 则您可能能够使用 Level-77 记录名来作为此项的限定。然而, 您必须确保所有调用此函数的程序都使用同一个 Level-77 记录。

---

**IWN.VAL.6695.e** `functionName` - 这个项数据引用不允许状态 `XXXXX`。

**说明:** 如果状态为 PROTECT、SKIP、INVISIBLE、BLINK 或颜色, 则该数据项位于打印表单上。VisualAge Generator 允许对打印机表单设置这些属性。EGL 则不允许。

如果状态为 EMPTY, 则确定在迁移期间是否有表单或记录不可用。EGL 允许定义独立数据项, 并且, 当找不到类型定义时, 就假定该定义用于某个项。EGL 不支持对数据项使用 set empty。

**用户响应:** 如果该问题与对打印表单设置属性相关, 则修改函数以除去该语句。另外, 如果同一个函数既用于文本表单又用于打印表单, 则必须创建该函数的一个副本以便与打印表单配合使用。

如果该问题与 `set empty` 语句相关, 则定义或迁移缺少的记录或表单。

---

**IWN.VAL.6716.e** *functionName* - 不能将自变量 *xxxxx* 传递至 **inOut** 参数 *yyyy*。自变量必须是与 **inOut** 参数兼容的引用。

**说明:** *functionName* 是发出调用的函数的名称。传递至 **inOut** 参数的自变量必须是兼容的引用。兼容的引用意味着自变量类型与参数类型必须完全匹配。VisualAge Generator 不支持自变量与参数的类型不相同,但在某些情况下允许它们不相同。

**用户响应:** 如果被调用的函数是其中一个 EGL 系统函数,则必须修改调用函数以使用匹配项,即必需的参数类型。如果被调用的函数是用户函数,则您可以更改被调用的函数以使用诸如数字之类允许更大范围的自变量类型和长度的函数参数类型。或者,查看被调用的函数以及调用它的所有位置。还必须将被调用的函数分为多个函数,每个函数有不同的 **inOut** 参数定义。

---

**IWN.VAL.7553.e** *functionName* - *systemFunctionName* 的自变量 *n* 必须是字符串项、字符串常量或字符串文字。

---

## IWN.XML 消息

---

**IWN.XML.3997.e** XML 验证错误 - 必须为元素类型 “*xxxxx*” 声明属性 “*yyyyy*”。

**说明:** 在 VisualAge Generator 中,选项 *yyyyy* 对元素 *xxxxx* 有效。EGL 不支持此组合。即使该值无效,迁移工具也迁移该值,因此“问题”视图中将会有有一个错误以提醒您解决该问题。

**用户响应:** 查看 EGL 联机帮助以了解对 *xxxxx* 有效的选项。当决定所要使用的选项时,您可能需要使用文本编辑器打开构建描述符文件以便能够进行必要的更改。

---

**IWN.XML.3998.e** XML 验证错误 - 值为 “*xxxxx*” 的属性 “**system**” 必须具有 “**ZOSCICS WIN USS ISERIESJ ZOSBATCH AIX LINUX**” 列表中的值。

**说明:** 在 VisualAge Generator 中,值 *xxxxx* 是有效的目标环境。EGL 当前不支持此环境。迁移工具为某些目标环境迁移信息以保留该信息,以便将来有可能使用该信息。

**用户响应:** 创建一个新项目以保留当前不能使用的已迁移构建描述符。将包含当前不受支持的 VAGen 值的所有构建描述符部件移至这个新项目中。要移动该部件,请使用文本编辑器打开构建描述符文件以复制并粘贴有错误的部件。

---

**IWN.XML.3999.e** (三条备用消息中的第一条) XML 验证错误 - 已经对元素 “*yyyyy*” 指定了属性 “*xxxxx*”。

**说明:** VisualAge Generator 允许将数字 (numc) 项用作 EZE 字符串函数中的字符自变量。EGL 不支持将数字项用作系统字符串 (strLib) 函数中的字符自变量。

**用户响应:** 编辑使用字符串函数的程序:

1. 在该程序中,声明一个新的字符项,并将其长度设置为与数字项相同。
2. 在调用字符串函数的函数中,将该数字项移至该字符项,然后传递该字符项来作为系统字符串函数的自变量。

即使您正在使用分段程序,这种技术也有效。此外,如果未使用分段程序,则可以在调用系统字符串函数的函数中定义新的字符项。

---

**说明:** 在同一个控制部件中多次指定了 *xxxxx* 属性。XML 解析器将停止处理该 .eglbl 文件。因此,此错误可能掩盖了其它错误,也可能导致在“问题”视图中列示其它错误。

**用户响应:** 使用文本编辑器编辑当前 .eglbl 文件,以使 *xxxxx* 在控制部件中只被指定一次。当您保存 .eglbl 文件时,“问题”视图中的消息应该会被更新。由于 XML 解析器在解析 .eglbl 文件中的第一个重复属性时停止处理,所以,您可能必须先解决数个错误,之后整个文件才能被解析。

---

**IWN.XML.3999.e** (三条备用消息中的第二条) XML 验证错误 - 元素类型 “*xxxxx*” 的内容必须与 “(*listOfValues*)” 匹配。

**说明:** 在 VisualAge Generator 中,对资源关联指定的一个或多个值是有效值。EGL 不支持此值。即使该值无效,迁移工具也迁移该值,因此“问题”视图中将会有有一个错误以提醒您解决该问题。

**用户响应:** 查看 EGL 联机帮助以了解对 *xxxxx* 有效的选项。当决定所要使用的选项时,您可能需要使用文本编辑器打开构建描述符文件以便能够进行必要的更改。

---

**IWN.XML.3999.e** (三条备用消息中的第三条) XML 验证错误 - 值为 *yyyyy* 的属性 *xxxxx* 必须具有列表 *zzzzz* 中的值。

**说明:** 并非所有 VAGen 值在 EGL 中都有相应的替换。

**用户响应:** 有关迁移如何转换值的详细信息, 请参阅链接表和资源关联的表。有关 EGL 中的可用选项的信息, 请参阅 EGL 文档。

---

## 附录 F. 不正确的外部源格式会导致创建 EGL 时发生问题的情况

在某些情况下，在运行第 1 阶段迁移工具后生成的外部源格式会导致在运行用于生成 EGL 的迁移工具时发生问题。这里对那些非常罕见并且可能性极小的情况进行讨论。

- 数据项部件
  - 映射范围编辑（最小值和最大值）可能会导致第 2 阶段因为外部源格式无效而发生异常。如果将外部源格式导入到 VisualAge Generator 中，并且从未修改过数据项，则将发生此问题。当在第 1 阶段迁移中导出外部源格式时，映射范围编辑具有无效的格式。在 VisualAge Generator 中修改数据项并保存该项。例如，对项描述添加和除去空白。也可以在运行第 1 阶段迁移之前安装 VisualAge Generator APAR PQ75621 或 APAR PQ79914 的修订。
  - 对于 VisualAge Generator on Java，当在其中一个名称中使用了符号 # 时，具有相似部件名的数据项可能会导致不可预测的结果。例如，DATAITEM#A、DATAITEM@A 和 DATAITEM\$A 可能会导致将错误的外部源格式代码存储在 DATAITEM#A 的迁移数据库中。如果已经在数据项名称中使用了符号 #，则请查看所生成的 EGL 源代码以确保所迁移的信息正确。也可以在对 **Java** 运行第 1 阶段之前安装 VisualAge Generator APAR PQ85794 的修订。
- 记录部件
  - 对于 SQL 记录来说，缺省键项可能会导致字段包含无效（或不可打印）字符的问题。如果您在第 2 阶段接收到一条指示 “NoSuchElementException” 的消息，则在 VisualAge Generator 中修改该记录，检查缺省键项字段并删除不可打印字符。保存该记录并再次运行第 1 阶段。也可以在运行第 1 阶段迁移之前安装 VisualAge Generator APAR PQ89390 的修订。



---

## 附录 G. 迁移数据库

---

### 创建 DB2 迁移数据库

除非另有说明，否则无论是正在从 Java 迁移还是正在从 Smalltalk 迁移，下列指示信息都适用。即使正在从 Smalltalk 迁移，也必须在计划运行第 2 阶段和第 3 阶段迁移的机器上设置 JDBC 驱动程序级别。

#### 为 DB2 7.2 设置 JDBC 级别

迁移工具要求 db2java.zip 具有 JDBC 2.0 级别。DB2 7.2 附带交付了两个 db2java.zip 文件 — 一个具有 JDBC 1.1 级别，另一个具有 JDBC 2.0 级别。要为 JDBC 2.0 配置 DB2 7.2，请执行下列操作：

1. 停止所有 DB2 进程。
  - a. 浏览到控制面板，然后选择管理工具 -> 服务。
  - b. 可能必须停止 DB2 - 如果您在进行第一次尝试时未停止 DB2 进程，它就会一直保持运行。
2. 打开 DOS 命令提示符窗口并转至包含 usejdbc2.bat 文件的目录。如果安装 DB2 7.2 时使用了缺省安装目录，则该文件应该位于 \SQLLIB\java12 目录中。
3. 运行 usejdbc2.bat 文件。
4. 启动您在步骤 1 中停止的所有内容。

#### 为 DB2 8.1 或更高版本设置 JDBC 级别

如果已安装了 DB2 8.1 或更高版本，则 db2java.zip 文件已具有正确的级别。

### 在 Windows XP 上使用 DB2

迁移工具具有下列要求：

- 用来访问迁移数据库的用户标识一定不能包含任何空格。
- Windows 用户标识需要具有管理员权限而不是受限权限，这样在第 2 阶段和第 3 阶段才能在迁移工具向导中查看迁移集。

### 创建迁移数据库

要创建迁移数据库，请执行下列操作：

1. 确保 DB2 以及任何其它使用它的应用程序都已关闭。例如，关闭 VisualAge Generator 和 EGL 开发环境。
2. 打开 DB2 命令窗口。
  - 如果正在从 Java 迁移，则转至 *VisualAgeJava-installation-directory\ide\vgmigration* 目录。
  - 如果正在从 Smalltalk 迁移，则转至 *VisualAge-Smalltalk-installation-directory*。
3. 运行名为 *SetupDatabase.bat* 的文件。这将运行同一目录中名为 *createdatabase.sql* 的文件并将输出保存到同一目录中名为 *createdatabase.out* 的文件中。这将创建一个名

为 VGMIG 的 DB2 数据库、连接至该数据库并配置数据库参数。创建数据库可能要花费长达一分钟的时间。请您务必等待所有命令执行完成。

**注:**

- 出现在控制台中的第一个命令可能会产生一条错误消息。您可以忽略此消息。此消息仅仅表示尚未存在 VGMIG 数据库。
  - 如果要创建具有除 VGMIG 以外的名称的数据库，则必须将 `createdatabase.sql` 中出现的所有 VGMIG 更改为您期望的数据库名称。您还必须记得更改第 1 - 3 阶段迁移工具首选项中的 VGMIG。
  - 缺省情况下，VGMIG 数据库是未受密码保护的。如果需要对其进行密码保护，则必须将该数据库更改为受密码保护。
4. 运行名为 `SetupTables.bat` 的文件。这将运行同一目录中名为 `createtables.sql` 的文件并将输出保存到同一目录中名为 `createtables.out` 的文件中。这将在迁移数据库中创建迁移工具所需的所有表和视图。这些表是使用名为 MIGSCHEMA 的高级限定符（模式）创建的。创建数据库可能要花费长达一分钟的时间。请您务必等待所有命令执行完成。

**注:**

- 出现在控制台中的前几个命令可能会产生错误消息。您可以忽略这些消息。它们仅仅表示尚未存在表和视图。
  - 如果要创建具有除 MIGSCHEMA 以外的名称的数据库，则必须将 `createtables.sql` 中出现的所有 MIGSCHEMA 更改为您期望的模式名。您还必须记得更改第 1 - 3 阶段迁移首选项中的 MIGSCHEMA。
  - 无论何时需要彻底地清除迁移数据库，都可以从 DB2 命令窗口中重新运行 `SetupTables.bat` 文件。
5. 关闭 DB2 命令窗口。

现在，迁移数据库、模式、表和视图都已创建完毕。您现在已经准备好创建供第 1 阶段迁移工具使用的首选项文件了。如果正在从 Java 迁移，则请参阅第 88 页的『设置第 1 阶段首选项』。如果正在从 Smalltalk 迁移，则请参阅第 106 页的『设置第 1 阶段首选项』。

---

## 复位迁移数据库

如果需要复位迁移数据库（例如，由于更改了重命名规则），则使用下列其中一项技术：

- 使用用来删除并重新创建迁移数据库中的所有表的工具。

在下列情况下，使用此工具：

- 如果需要删除所有迁移计划。
- 如果已迁移了一个 Java 项目的多个版本。
- 如果已迁移了一个 Smalltalk 配置映射的多个版本。

要运行用于删除并重新创建所有表的工具，请执行下列操作：

1. 从 DB2 命令窗口中，转至 **SetupTables.bat** 所在的目录。
  - 对于 Java 来说，这是 `VisualAge-for-Java-install-directory\ide\vgmigration`。

- 对于 Smalltalk 来说，这是 *VisualAge-Smalltalk-install-directory*。
2. 运行 **SetupTables.bat**。
- 使用用于删除指定迁移集的工具。如果只需要删除几个迁移集，则可以使用此工具。要运行用于删除指定迁移集的工具，请执行下列操作：
    1. 确定需要从迁移数据库中删除的迁移集标识，如下所示：
      - a. 使用“DB2 控制中心”或 SQL 查询来查看 CONFIGPLAN 表。
      - b. 查找要删除的 CONFIGPLANNAME。
      - c. 您需要指定的迁移集标识就是对应的 CONFIGPLANID 列中的值。
    2. 从 DB2 命令窗口中，转至 **deletemigsets.bat** 所在的目录。
      - 对于 Java 来说，这是 *VisualAge-for-Java-install-directory\ide\vmigration*。
      - 对于 Smalltalk 来说，这是 *VisualAge-Smalltalk-install-directory*。
    3. 使用下列其中一种格式来运行 **deletemigsets.bat** 文件：
      - 如果您只想删除一个迁移集，则请使用以下格式：
 

```
deletemigsets n
```

其中，*n* 是要删除的迁移集标识。
      - 如果要删除若干个迁移集，则使用以下格式：
 

```
deletemigsets "n1,n2"
```

其中，*n1* 和 *n2* 是要删除的迁移集标识。

---

## 使用 DB2 来对远程数据库进行编目

要在 DB2 中对远程数据库进行编目，您需要下列信息：

- 该数据库所在的远程机器的主机名或 IP 地址
- 客户机上的端口号和协议（例如 60000/tcp）
- 节点名（用于描述远程机器的别名，例如 db2node）
- 数据库名称
- 数据库别名（可选）

要使用 DB2 来对远程数据库建立 TCP/IP 连接，请执行下列操作：

1. 打开“命令提示符”窗口（在 Windows 上）或“命令终端”（在 Linux 上）。
2. 要对节点进行编目，请输入以下命令（整个命令位于一行上）：

```
db2 catalog tcpip node nodeName remote [ hostName | ipAddress ]
server [ svcname | portNumber ]
```

您可以输入 *hostName* 或 *ipAddress*。例如，要在 IP 地址为 9.10.11.123 的节点 db2node 上使用端口号 60000 来对远程服务器进行编目，请输入以下命令：

```
db2 catalog tcpip node db2node remote 9.10.11.123 server 60000
```

3. 要对数据库进行编目，请输入以下命令（整个命令位于一行上）：

```
db2 catalog database databaseName
[ as databaseAlias ] at node nodeName
```

“as *databaseAlias*”是可选的。如果未指定 *databaseAlias*，则别名将与数据库名称相同。*nodeName* 必须就是您在步骤 2 中使用的那个 *nodeName*。

例如，要对名为 SAMPLE 的远程数据库进行编目以使其在节点 db2node 上具有别名 sam1，请输入以下命令：

```
db2 catalog database sample as sam1 at node db2node
```

4. 要测试与该数据库的连接，请输入以下命令（整个命令位于一行上）：

```
db2 connect to databaseAlias use userName using password
```

如果在步骤 3 中未指定别名 (*databaseAlias*)，则使用数据库名称。例如，对于使用密码 db2password 的用户 db2user，要连接至具有别名 sam1 的数据库 SAMPLE，请输入以下命令：

```
db2 connect to sam1 user db2user using db2password
```

如果在步骤 2 中未指定 sam1 作为数据库别名，则输入以下命令：

```
db2 connect to SAMPLE user db2user using db2password
```

5. 您应该会看到数据库连接信息。

要获取其它帮助，请访问以下 Web 站点：

<https://aurora.vcu.edu/db2help/db2i4/frame3.htm#idx>

---

## 使用 DB2 来对远程数据库进行取消编目

要在 DB2 中对远程数据库进行取消编目，您需要下列信息：

- 数据库别名，或者数据库名称（如果您对该数据库进行编目时未指定别名的话）

要使用 DB2 来对远程数据库进行取消编目，请执行下列操作：

1. 打开“命令提示符”窗口（在 Windows 上）或“命令终端”（在 Linux 上）。
2. 要对数据库进行取消编目，请输入以下命令，其中 *databaseAlias* 是数据库别名：

```
db2 uncatalog database databaseAlias
```

例如，要对数据库 SAMPLE（已对其指定别名 sam1）进行取消编目，请输入以下命令：

```
db2 uncatalog database sam1
```

如果在对该数据库进行编目时未指定数据库别名，请使用该数据库的名称。例如，如果未指定 sam1 作为数据库别名，则输入以下命令：

```
db2 uncatalog database SAMPLE
```

要获取其它帮助，请访问以下 Web 站点：

<https://aurora.vcu.edu/db2help/db2i4/frame3.htm#idx>

---

## 非常有用的查询

如果您修改了样本第 1 阶段迁移工具或者开发了自己的第 1 阶段迁移工具，则下列 SQL 查询对于验证您所作的更改来说可能非常有用。

注：

- 可以从 DB2 命令窗口中运行这些示例。
- 这些示例假定您使用缺省迁移数据库名称 (VGMIG) 和缺省模式 (MIGSCHEMA)。

- 除非另有指示，否则整个 DB2 命令必须在一行上输入。由于位置有限，本文档此后显示的命令可能跨数行。
- 这些示例要求首先连接至数据库。要连接至数据库，请运行以下命令：

```
db2 connect to VGMIG
```

为了帮助您确定第 1 阶段迁移工具是否已正确地运行，请运行以下 .bat 文件，此文件位于 *VAGen-installation\ide\vgmigration* 目录（对于 Java）或 *VAGen-installation* 目录（对于 Smalltalk）中：

```
checkStage1.bat
```

这个 .bat 文件运行数个查询：

- 数据库中的迁移规划名列表
- 数据库中的部件总数
- 其它用于检查外部源格式有效性以及部件在 EGL 文件中的放置的查询。

前两个查询的结果应该大于 0。其它查询的结果应该为 0。如果您得到的结果有所不同，则表示第 1 阶段迁移出现问题，并且您应该与 IBM 支持机构联系。

要确定是否已迁移了某个 VAGen 部件：

```
db2 select configplanname, configplanversion, vgpartname, vgparttime, is_migrated
      from migschema.vgpart where vgpartname = 'yourPartName'
```

要验证迁移数据库中所有部件的外部源格式的前几个字符：

```
db2 select vgpartname, cast(vgesfsorce as char(n)) from migschema.vgpart
```

*n* 是 1 与 256 之间的数字，它是要显示的字符数。

要确定是否迁移数据库中的任何部件未以有效的外部源格式标记开头：

```
db2 select vgpartname, cast(vgesfsorce as char(n))
      from migschema.vgpart where vgesfsorce not like ':%'
```

*n* 是 1 与 256 之间的数字，它是要显示的字符数。

如果要重新运行第 2 阶段和第 3 阶段迁移而不重新运行第 1 阶段，则复位迁移数据库中的所有部件：

```
db2 update migschema.vgpart set is_migrated = 'N', eglsource = NULL,
      eglpartname = NULL
db2 delete from migschema.translation_msgs
```

要备份迁移数据库：

```
db2 backup database vgmig to x:\mybackups\backupName
```

*x:\mybackups\backupName* 是要用来放置备份的驱动器和目录。将在 *x:\mybackups\backupName* 下面创建数个子目录。

要复原先前备份的迁移数据库：

```
db2 restore database vgmig from x:\mybackups\backupName REPLACE EXISTING
```

*x:\mybackups\backupName* 是要用来放置备份的驱动器和目录。将在 *x:\mybackups\backupName* 下面创建数个子目录。



---

## 附录 H. 迁移工具的性能

有许多因素会影响迁移工具的性能。这些因素包括:

- 第 1 阶段、第 2 阶段和第 3 阶段的一般性能。
  - 内存。
  - 处理器速度。
  - 本地还是远程 DB2 数据库。对于远程数据库来说，网络连接速度十分关键。
  - Java 项目和包的数目或者 Smalltalk 配置映射和应用程序的数目。
  - 部件的数目及其按部件类型的分布。
  - 函数部件中的行数。
- 第 1 阶段的性能
  - 在开始进行迁移之前清除 Java 工作空间或 Smalltalk 映像。
  - 本地还是远程 Java 资源库或 Smalltalk 库。对于远程资源库或库来说，网络连接速度十分关键。某些客户通过将他们的资源库复制到运行迁移的机器来提高处理速度。
  - 重命名规则的复杂性。
  - 迁移集在迁移数据库中是否已存在。如果您正在重新创建具有另一组重命名规则的迁移集，则与让第 1 阶段迁移工具清除原始迁移集相比，通过运行 `setuptables.bat` 来重新创建 SQL 表更为有效。仅当在迁移数据库中没有任何其它迁移集时，重新创建 SQL 表的方法才可行。
- 第 2 阶段和第 3 阶段的性能
  - `.ini` 文件中 `VMArgs` 选项的设置。当未设置 `VMArgs` 选项时，在第 3 阶段执行的工作空间刷新和重建无法可靠地运行。您必须设置此选项。要了解有关如何设置此选项的详细信息，请参阅第 123 页的『启动参数』。

在给定所涉及的因素数目的情况下，没有任何特定的公式可以预测迁移运行时间。但是，下列各节提供了一些有关各个迁移阶段可能需要花费的时间的前瞻性指导:

- 项目、包、部件和程序的数目
- 处理器速度
- 函数部件中的行数
- 对第 1 阶段使用干净的 Java 工作空间

另外，还有一节提供了可用来帮助您预计磁盘空间要求的一些信息。

---

### 项目、包、部件和程序的数目

下表提供了有关各个迁移阶段可能需要花费的时间的信息。这些测试是使用 Windows 2000 运行的，内存速度为 1.0 千兆赫，处理器速度为 1.1 千兆赫。测量是对 EGL 6.0.0.1 进行的，所有时间都以分钟计。

表 136. 迁移集大小对迁移时间的影响

测试用例	项目数	包数	部件数	程序数	第 1 阶段时间	第 2 阶段时间	第 3 阶段写文件时间	第 3 阶段刷新或构建时间
1	1	98	18,403	1,246	53	22	177	17
2	1	10	2,771	147	23	5	5	8
3	3	29	2,660	33	8	1	4	1
4	3	44	7003	25	62	9	66	22
5	4	92	10,601	25	16	5	41	8
6	5	35	8,238	100	23	4	29	9
7	474	570	11,280	162	46	10	58	36
8	6	118	15,250	71	91	16	110	31

以下是一些基于表 136 的一般观察结果:

- 测试用例 2 与测试用例 3 具有近似的部件数，但是，测试用例 2 的第 1 阶段运行时间和第 2 阶段运行时间分别是测试用例 3 的 3 倍和 5 倍。在第 1 阶段和第 2 阶段，测试用例 2 进行关联项分析时需要分析的程序数是测试用例 3 的 4.5 倍。因此，程序数对运行时间是有影响的。
- 测试用例 5 与测试用例 7 具有近似的部件数，但测试用例 7 的第 1 阶段运行时间是测试用例 5 的 3 倍。在第 1 阶段，测试用例 7 进行关联项分析时需要分析的程序要多得多。此外，与测试用例 5 相比，测试用例 7 的数据项相对较少。在 VAGen 中，数据项从来不会有关联项，因此，第 1 阶段迁移工具不会尝试确定数据项的关联项。测试用例 7 在第 3 阶段写文件的时间也几乎是测试用例 5 的 1.5 倍。测试用例 7 要为更多的可生成部件分析关联项。由于每个可生成部件都位于它自己的文件中，因此，要写的文件也更多。与测试用例 5 相比，放到 CommonParts.egl 和 UnusedParts.egl 文件中的部件成比例地减少。这些文件将较小，并且数目较多。
- 测试用例 1 的部件数比测试用例 7 多 50%，但测试用例 7 的工作空间刷新和重建时间是测试用例 1 的 2 倍。在构建期间，测试用例 7 所要分析的项目和包要多得多。此外，在测试用例 7 中，在构建过程中检测到 162 个项目循环，因此，这导致构建时间较长。

根据表 136 中的信息，如果您有数百甚至 1000 个程序，您可能想作为单个迁移集来迁移它们，即使那些程序位于数个 subsystem 中。这假定那些 subsystem 全都使用相同的公共项目版本，并且那些 subsystem 未包含任何重复部件名。

## 处理器速度

第 317 页的表 137 显示了将处理器速度由 1.1 千兆赫更改为 1.6 千兆赫所产生的影响。这两台机器都使用 1.0 千兆赫的内存。1.1 千兆赫的机器使用 Windows 2000; 1.6 千兆赫的机器使用 Windows XP。处理器速度具有显著的影响，对于第 1 阶段和第 2 阶段处理时间来说尤其如此。测量是对 EGL 5.1.2 进行的。

表 137. 处理器速度对迁移时间的影响

测试用例	1.1 千兆赫				1.6 千兆赫			
	第 1 阶段时间 (小时)	第 2 阶段时间 (小时)	第 3 阶段写文件时间 (小时)	第 3 阶段文件刷新或构建时间 (小时)	第 1 阶段时间 (小时)	第 2 阶段时间 (小时)	第 3 阶段写文件时间 (小时)	第 3 阶段文件刷新或构建时间 (小时)
9	1.2	1.0	0.4	0.2	0.4	0.3	0.3	0.1
10	3.0	1.6	1.8	0.9	1.6	0.6	0.9	0.9

根据表 137，您在迁移期间可能想使用具有更快处理器的机器。

## 函数部件中的行数

在某个时刻，VisualAge Generator 会发生一个问题，此问题导致将一系列空白行插入到函数中。在某些情况下，会插入多达 32000 个空白行。这些额外的空白行会对性能产生严重的影响。表 138 显示了函数中的行数对迁移时间的影响。测试用例包含 2 个项目、17 个包、919 个部件以及 87 个程序。共有 497 个函数，其中的 8 个函数包含大量的空白行（数目估计在 30000 行左右）。测试是对 EGL 5.1.2 进行的，所有时间都以分钟计。

表 138. 函数中的行数对迁移时间的影响

测试用例	在 VAGen 中除去空白行之前				在 VAGen 中除去空白行之后			
	第 1 阶段时间	第 2 阶段时间	第 3 阶段写文件时间	第 3 阶段文件刷新或构建时间	第 1 阶段时间	第 2 阶段时间	第 3 阶段写文件时间	第 3 阶段文件刷新或构建时间
11	40	25	1	3	12	11	1	3

在开始进行迁移前除去额外的空白行会对第 1 阶段和第 2 阶段处理时间造成巨大的差别。如果您知道函数包含大量的空白行，则应该在迁移之前除去那些空白行。但是，由于此问题在 VisualAge Generator 中很罕见，因此，在迁移之前搜索这些函数可能并不合乎成本效益。如果有 3 个以上的连续空白行，则迁移工具将在第 2 阶段迁移期间自动除去附加的空白行。因此，第 3 阶段的处理时间没有变化。由于除去了这些空白行，所以 EGL 的性能会有所提高。

## 对第 1 阶段使用干净的 Java 工作空间

下表显示在开始进行第 1 阶段迁移时使用干净的工作空间的影响。测试用例 12 包含 3 个项目、29 个包、2660 个部件和 33 个程序。测试用例 13 包含迁移集的 7 个版本。第一个版本包含 3 个项目、4 个包、30 个部件并且不包含任何程序。最后一个版本包含 6 个项目、11 个包、66 个部件以及 7 个程序。测试是对 EGL 5.1.2 进行的，所有时间都以分钟计。

表 139. 干净的 Java 工作空间对迁移时间的影响

测试用例	不使用干净工作空间时的第 1 阶段时间	使用干净工作空间时的第 1 阶段时间
12	17	12
13	11	1

根据表 139，如果从 VisualAge Java 迁移，则您应该考虑在开始进行迁移时使用干净的工作空间。要了解有关如何在开始进行迁移时使用干净的 Java 工作空间的详细信息，请参阅第 96 页的『提高性能』。

对于 VisualAge Smalltalk 来说，有可能会节省近似的时间量。因此，如果您正在从 VisualAge Smalltalk 迁移，则也应该考虑在开始进行迁移时使用干净的映像。要了解有关如何在开始进行迁移时使用干净的 Smalltalk 映像的详细信息，请参阅第 114 页的『提高性能』。

## 磁盘空间要求

VisualAge Generator 应用程序与对应的 EGL 程序的磁盘空间要求之间没有直接的关系。

下表提供了第 316 页的表 136 中显示的那些测试用例的磁盘空间要求。EGL 测量都是在迁移完成之后、更正“问题”视图中的任何消息之前和进行任何生成之前立即进行的。测量是对 EGL 6.0.0.1 进行的，所有大小都以兆字节 (MB) 计。

表 140. 磁盘空间要求

测试用例	VAGen .dat 文件大小	第 1 阶段 DB2 备份文件大小	EGL 项目交换文件	EGL 工作空间大小	EGL .metadata 目录大小	EGL 总工作空间大小
1	25.5	124.0	6.4	40.1	12.9	53.0
2	3.5	48.0	0.5	3.6	9.3	12.9
3	3.0	36.0	0.5	3.3	8.5	11.8
4	14.9	100.0	2.1	15.6	14.1	29.7
5	11.1	60.0	1.0	8.0	10.2	18.2
6	8.0	60.0	1.0	6.6	9.9	16.5
7	33.5	12.0	6.0	37.5	58.0	95.5
8	13.2	132.0	1.5	12.8	10.7	23.5

以下是一些基于表 140 的一般观察结果：

- DB2 备份文件大小是第 1 阶段结束时迁移数据库的备份文件的大小。第 2 阶段大小将稍大一些，这是由于将 EGL 源代码添加至数据库引起的。
- EGL 项目交换文件大小比 VAGen .dat 文件大小要小一些。
- 测试用例 2、3、4、7 和 8 都具有大小相差无几的 VAGen .dat 文件和 EGL 工作空间大小。因此，使用 1:1 的比率可提供更好的 EGL 工作空间大小预计。但是，测试用例 1 的 EGL 工作空间大小大约是 VAGen .dat 文件大小的 1.5 倍。测试用例 1 具有许多程序，但与其它测试用例相比，这些通常都是较小的文件。因此，如果预期会有许多文件，则应预留更多的空间。

- 许多测试用例都具有小于 15 MB 的 EGL .metadata 目录大小。但是，测试用例 7 具有较大的 .metadata 目录。这可能是由于测试用例 7 的项目和包数较多引起的。因此，如果具有项目和包，则应为 .metadata 目录预留更多的空间。例如，如果按 VAGen 中每个项目一个程序来组织源代码，则可能应预留相当大 .metadata 目录。



---

## 声明

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

本信息是为在美国提供的产品和服务编写的。IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing  
IBM Corporation North Castle Drive Armonk, NY 10504-1785  
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation  
SWS General Legal Counsel  
Department TL3 Building 062  
P. O. Box 12195  
Research Triangle Park, NC 27709-2195

IBM 已经作了合理的努力来确保本出版物中包含的信息的准确性。如果产品附带提供了本出版物的软拷贝版本，则您应该将软拷贝版本中包含的信息视为最新并且最准确的信息。但是，本出版物是“按现状”提供的，并且 IBM 对于这里的内容、这里列示的产品或者本出版物的完整性或准确性不作任何类型的保证。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

IBM 可以对这里描述的出版物和 / 或产品进行更改。

只要遵守适当的条款和条件，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能全面地说明这些数据 and 报表，这些示例包括个人、公司、商标和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有任何雷同，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

如果您正以软拷贝格式查看本信息，图片和彩色图例可能无法显示。

---

## 商标

以下各项是 IBM Corporation 在美国和 / 或其它国家或地区的商标：

- AIX
- CICS
- DB2
- IBM
- IMS
- iSeries
- MVS
- OS/2
- OS/400
- Rational
- VisualAge
- WebSphere

- z/OS

Intel 是 Intel Corporation 在美国和 / 或其他国家或地区的商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

Microsoft<sup>®</sup>、Windows 和 Windows NT<sup>®</sup> 是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。



# 索引

## [ B ]

帮助映射 53, 199, 201, 211  
帮助映射名 52  
帮助映射组 38, 52, 126, 277, 289, 291  
绑定控制 150  
绑定控制部件 150, 151  
    特定于程序的 152  
    用作模板 151  
包 8, 10, 11, 13, 16, 17, 19, 20, 26, 27, 28, 29, 30, 31, 32, 37, 38, 48, 60, 61, 88, 91, 92, 93, 97, 100, 109, 116, 133, 134, 135, 139, 141, 142, 143, 144, 148, 177, 196, 250, 263, 267, 280, 281, 283  
    重命名 92, 110  
    命名 109  
保留字 17, 18, 20, 26, 30, 173, 194, 199, 201, 211, 278  
    表名 48  
    程序名 59  
EGL  
    列表 173  
formGroup 名 49  
Java  
    列表 177  
SQL 126  
    列表 176  
保留字列表 30  
报告 15, 16, 17, 24, 89, 98, 116, 278  
    第 1 阶段迁移 101, 111, 112, 114, 116, 130, 138  
编辑表 22, 34, 183, 186  
编辑函数 22  
编辑例程 22, 32, 33, 34, 43, 55, 186, 208, 210, 277, 286, 289  
表 23, 27, 30, 37, 41, 42, 48, 66, 99, 193, 288  
    重命名 173, 176  
    数据库 93, 111  
    FIND 语句 73  
    RETR 语句 73  
不确定情况 18, 20, 22, 25, 26, 41, 118, 148, 179  
    表 48  
    程序 59  
    函数 63  
    记录 45  
    其它语句 71  
    数据项 41  
    映射组和映射 49  
EZE 字 80

部件 16  
    不可迁移的 31  
    大量 13  
    第 1 阶段, 第 2 阶段, 第 3 阶段 14, 19  
    放置 26, 30  
        单文件方式 20, 141, 142  
        第 1 阶段, 第 2 阶段, 第 3 阶段 20, 30  
        第 1 - 3 阶段 30  
    项目列表部件 (PLP) 14  
部件名 11, 21, 26, 28, 30, 148  
冲突 37, 63, 126, 249  
重复 23  
重复的 30  
重命名 126, 176  
解析 23, 29, 30  
无效 30, 48, 49, 59, 173, 176, 273, 296  
VisualAge Generator 282

## [ C ]

程序 22, 23, 24, 26, 30, 31, 32, 33, 34, 38, 41, 45, 52, 61, 66, 76, 99, 148, 210, 211, 212, 214, 278, 283, 285, 294, 295  
    重命名 173, 176  
    单文件迁移 19  
    行为 13, 22  
    迁移, 带 33  
    属性 26, 27  
    样本 13  
        第 1 阶段工具 15, 17  
    隐式数据项 60  
XFER 79  
重命名 16, 30, 53, 89, 92, 93, 107, 116, 282, 284  
    用户出口信息 125  
重命名规则 93, 109, 280, 281, 310  
重命名前缀 30, 126, 242, 293  
重命名页 92, 110  
错误消息  
    HPT 277  
    IWN.MIG 282  
    IWN.VAL 303  
    IWN.XML 305

## [ D ]

带有多个更新的 SQL I/O 70  
单文件方式 20, 21, 30, 33, 50, 142, 143, 196, 198, 283, 286, 289, 299  
    部件放置 141  
    迁移 141  
    设置 141  
单文件迁移  
    批处理方式 142  
    用户界面 141  
导出 20, 141, 282, 283  
导入 27, 130, 133, 137, 138, 141, 282, 283  
    第 3 阶段工具 14  
    外部源格式 20  
导入到工作空间中 130, 132, 134  
第 1 阶段 14  
    Java 87  
        首选项 31, 88  
        运行 97  
Smalltalk 105  
    首选项 31, 106  
    运行 115  
第 2 阶段 14, 123  
    首选项  
        设置 128  
    运行 132  
        批处理方式 133  
        用户界面 132  
第 3 阶段 14, 137  
    首选项 137  
    运行 137  
调试  
    运行时差别 164  
    映射 164  
    SQL 164  
EZESQLCA 164  
EZESQRRM 164  
EZESQWN6 164

## [ F ]

符号参数 250, 269, 273  
    用户定义的 275  
    与部件相关的 273  
    与文件相关的 275  
服务例程 63, 179, 248  
    一般语法 248  
    VisualAge Generator 和 EGL 等效例程 248

覆盖 PLN 115, 116, 118  
附录索引条目 173, 180, 186, 193, 196,  
198, 215, 227, 239, 248, 249, 277, 299,  
307

## [ G ]

高级 PLP 项目 14, 16, 90, 98, 99  
    创建 99  
跟踪 250  
    级别 94, 111  
    消息 277  
更新数据库 93, 116  
公共部件 91, 92, 109  
公共代码 6, 16, 22, 23, 24, 25, 26, 33,  
72, 110  
工作空间 4, 5, 8, 10, 14, 18, 19, 24, 25,  
26, 28, 48, 88, 96, 99, 100, 101, 124,  
128, 133, 134, 135, 138, 141, 143, 250,  
279, 284  
    保存 97, 114  
    重复部件 21, 23  
    复原 97, 115  
    干净的 96  
工作台  
    首选项  
        设置 123  
构建部件 29, 37, 250  
构建路径 11, 23, 25, 26, 28, 29, 30, 139,  
148  
构建描述符 149, 150  
    调试 151  
        缺省 154  
        缺省 149  
        EGL 151  
构建描述符部件 147  
    查看 149  
构建描述符选项 149, 150, 151, 154, 155  
    绑定 152  
    查看  
        一般 149  
    链接编辑 152  
    COBOL 生成  
        查看 150  
    genproject 150  
    Java 生成  
        查看 150  
关联部件 4, 22, 23, 26, 29, 37, 41, 98,  
117, 283  
    迁移, 不带 34  
    迁移, 带 33  
规划迁移 3, 4  
过滤器 15, 16, 128, 278  
    版本 90  
    版本名 90, 91, 108  
    版本深度 90, 91, 108

过滤器 (续)  
    包 92  
    配置映射 107, 108  
    项目 90, 91  
    资源库 15, 89, 90, 93, 98, 100, 107,  
117, 118

## [ H ]

函数 11, 21, 22, 23, 27, 28, 29, 30, 32,  
33, 34, 36, 37, 41, 43, 44, 46, 49, 55,  
57, 64, 65, 66, 69, 70, 71, 72, 73, 74,  
76, 77, 79, 80, 82, 127, 210, 211, 214,  
215, 216, 218, 219, 227, 228, 283, 285,  
286, 289, 291, 292, 293, 294, 303, 304  
    重命名 30, 126, 173, 176  
    处理不确定情况 63  
    公共 25, 33, 38  
    I/O 220  
    SQL 37, 126  
    SQL I/O 67, 220, 222, 225

## [ J ]

记录 23, 27, 37, 41, 44, 48, 61, 72, 79,  
186, 187, 286, 287, 288, 293, 294  
    备用指定 47, 48, 188  
    重命名 30, 173, 176  
    重新定义的 45  
    串行 220  
    带索引 220  
    赋值语句 72  
    公共 38  
    工作存储器 18, 126  
    相对 220  
    消息队列 220  
    用户界面 79  
    用户界面 (UI) 30, 218, 278  
    I/O 219  
    Level-77 项 46  
    SQL 190, 299  
交叉部件迁移 4, 14, 22, 23, 26, 33, 41  
结果  
    查看 92  
    第 1 阶段 98  
        迁移数据库 116  
    迁移 284  
    试验项目 6  
    中间的 13

## [ K ]

控制部件 17, 20, 31, 130, 249, 250, 278,  
295  
    绑定控制 19, 30, 273, 296

控制部件 (续)

    链接编辑 19, 30, 273, 296  
    链接表 263  
        Calllink 263  
        Crtxlink 267  
        Dxfmlink 268  
        Filelink 266

    链接表选项 249

    链接选项 19

    生成选项 19, 29, 30, 37, 249, 250

    资源关联 19, 30, 249, 269

库 10, 107, 108, 114, 263

    管理 3, 4, 5, 6, 11, 32

    Smalltalk 14, 17, 106, 107, 115

## [ L ]

链接编辑 150

链接选项 150

链接选项部件 147

    查看 153

## [ P ]

配置映射 8, 11, 13, 14, 15, 16, 17, 18,  
21, 24, 31, 32, 33, 60, 107, 108, 109,  
110, 111, 114, 116, 117, 118, 138, 250,  
310  
批处理方式 13, 18, 19, 123, 129, 132,  
134, 142, 144

## [ Q ]

迁移工具的性能 313

迁移功能部件 106

    添加 88

    装入 105

迁移规划 15, 16, 17, 89, 90, 92, 93, 94,  
98, 99, 101, 107, 111, 112, 115, 116,  
118, 278, 310

    多个 16

    高级配置映射 117

    手工创建 100

迁移集 14, 15, 16, 17, 18, 22, 23, 25,  
26, 31, 32, 33, 34, 37, 38, 50, 61, 89,  
90, 91, 97, 98, 99, 100, 101, 107, 108,  
109, 110, 111, 115, 117, 130, 131, 132,  
133, 134, 139, 196, 279, 283, 311

迁移数据库 15, 16, 17, 18, 19, 26, 87,  
89, 90, 93, 94, 98, 100, 105, 106, 107,  
111, 116, 134, 137, 139, 283, 299, 309,  
312, 313

    表 310

    创建 309

迁移数据库 (续)

- 复位  
表 310
- 视图 310

## [ R ]

- 日志文件 15, 17, 18, 20, 98, 116, 134, 135, 142, 143, 144
- 第 2 阶段迁移 132
- 名称 94, 111, 129, 137
- 名称首选项 114

## [ S ]

- 生成 23, 25, 29, 33, 34, 292
- 报告 92, 93, 94, 116
- 表 19
- 程序 5, 14, 19, 22, 285, 286, 289, 290, 293, 294, 295
- VisualAge Generator 100
- 生成选项 9, 149, 249, 250, 273, 295
- 转换表值 262
- VisualAge Generator 27
- 生成选项部件 9
- 使用说明 295
- 首选项 16, 20, 137, 138, 278, 281
- 必需的 EGL 123
- 编辑器 26
- 重命名 126
- 单文件方式 124
- 第 1 阶段 15, 16, 31, 118
- 基于 Java 的设置 88
- 基于 Smalltalk 的设置 106
- 设置 100, 101
- Java 15
- 第 2 阶段 18, 132
- 设置 128
- 第 3 阶段 18, 19
- 工作台
- 设置 123
- 构建描述符 26
- 建议的 124
- 派生文件名 114
- 设置 141
- 样本文件 88
- 资源库过滤器 117
- SQL 111, 126
- VAGen 迁移首选项 124
- VAGen 迁移语法首选项 30, 126, 225
- VAGen 语法迁移首选项 143, 220, 222
- 首选项文件 281
- 迁移 115
- Java 16

输出文件 21

- 数据库 7, 13, 14, 15, 16, 25, 65, 66, 93, 99, 101, 128, 129, 131, 132, 137, 250, 278, 281, 293, 295
- 数据项 9, 24, 27, 28, 30, 38, 41, 42, 45, 75, 80, 180, 227, 228, 277, 282, 283, 286, 287, 288, 299, 307
- 重命名 30, 126, 173, 176
- 赋值语句 72
- 共享的 23, 25, 30, 32, 33, 41, 42, 43, 44, 48
- 首选项 127
- 隐式 60, 71, 294
- 数组 26, 204, 228, 233, 241
- 动态 3
- 多维 3
- 映射 38
- 术语 3

## [ T ]

- 填充字符 44, 183, 208

## [ W ]

- 外部源格式 14, 17, 18, 20, 21, 26, 33, 34, 36, 50, 67, 93, 98, 116, 123, 132, 141, 142, 204, 282, 283, 284, 286, 290, 295, 299, 307, 313
- 未使用的部件 16, 91, 92, 109, 110

## [ X ]

- 系统库函数 22, 33, 44, 62, 80, 228, 233
- 显示 33, 34, 38, 51, 63, 64, 73, 196, 197, 218, 299
- 显示器 22, 24, 25, 198, 199, 202, 290, 291, 292
- 相关联的程序部件 60
- 向导
- 导入 20, 144
- 项目列表部件 11
- 项目列表部件 (PLP) 15, 99
- 项目名 15, 31, 90, 91, 92, 98, 99, 101, 109, 133, 134
- 消息 17, 42, 48, 51, 55, 59, 69, 70, 98, 106, 116, 135, 142, 144, 183, 199, 201, 202, 204
- 参考 94, 111
- 第 1 阶段公共 277
- 第 2 阶段 132, 282
- 第 3 阶段 129, 137
- 跟踪 297
- 调试 94, 111
- 警告 94, 111

消息 (续)

- 迁移工具发出的 277
- 致命 94, 111
- “问题”视图 19, 124, 128, 144, 299
- HPT 277
- IWN.MIG 282
- IWN.VAL 303
- IWN.XML 305
- VisualAge for Java 上的第 1 阶段 280
- VisualAge for Smalltalk 上的第 1 阶段 281
- 性能
- 迁移工具 313

## [ Y ]

- 一般函数 EZE 字 244
- 隐式项 37, 71, 294
- 程序中的 60
- 映射 20, 23, 27, 32, 33, 37, 41, 42, 43, 51, 53, 63, 72, 100, 148, 173, 198, 228, 291, 292, 293, 294, 299
- 变量字段 22, 33, 34, 55, 202, 204, 206, 208, 209
- 错误消息 210
- 不确定情况 49
- 不受保护常量 58
- 常量字段 202, 204, 206
- 重命名 30, 126, 176
- 打印 294
- 打印机 22, 25, 33
- 一般语法、映射类型和属性 201
- 范围编辑 307
- 赋值语句 72
- 函数和 I/O 选项 218
- 跨 109
- 数字硬件属性 56
- 未命名变量字段 57
- 显示 33
- 显示器 22, 25
- 显示器映射
- 一般语法、映射类型和属性 199
- 一般信息 198
- EZEMSG 242
- XFER 238
- XFER 语句 79
- 映射编辑 44
- 映射部件 9
- 映射名 52
- 映射属性
- 错误消息 186
- 数字编辑 185
- 一般编辑 183
- 一般信息 183

- 映射项
  - 编辑例程 43
  - 检查 NULL 76
  - 隐式 60
- 映射组 20, 22, 24, 30, 31, 41, 50, 51, 100, 195, 277, 283, 288, 289, 290, 291
  - 不确定情况 49
  - 重命名 173
  - 跨 91, 109
  - 设备名、类型和大小 197
  - 一般信息 196
  - 一般语法和浮动区 196
- 映射组部件 9
- 语法 20, 65, 292, 294
  - 表 179
  - 程序主函数示例 214
  - 服务例程一般示例 248
  - 赋值、MOVE 和 MOVEA 示例 228
  - 数据项示例 181
  - 一般表示例 194
  - 一般程序示例 211
  - 一般打印机映射示例 201
  - 一般函数示例 216
  - 一般记录示例 187
  - 一般显示器映射示例 199
  - 一般约定
    - VisualAge Generator 与 EGL 的差别 179
  - 映射组示例 196
  - 语句示例
    - 函数调用 228
- EGL 4, 14, 21, 22, 26, 33, 126, 137, 148, 179, 277, 285, 299
  - 错误 38
  - 精确的 22
  - 无效 69
  - 转换 (第 2 阶段) 123
- SET 示例 230
- VAGen 23, 38, 126, 282, 283
- XFER 示例 238
- 语句 36, 37, 80, 179, 211, 219, 227
  - 不确定情况中产生的 41
  - 打印 63
  - 赋值、MOVE 和 MOVEA 228
  - 函数调用 228
  - 链接编辑 273
  - 流 210, 214, 240
  - 使用声明 49, 212, 288
  - 显示 63
  - 一般规则
    - 数据项限定和数字文字 228
  - CALL 237, 242
  - CALL, DXFR, XFER 278
  - call, transfer, show 60, 294
  - DXFR 26, 237
  - FIND 72

- 语句 (续)
  - IF、WHILE 和 TEST 233
  - I/O 41, 127, 242, 294
  - I/O 不确定性 63
  - Level-77 项 71
  - RETRIEVE, FIND 232
  - SET 230
  - SETUPD, UPDATE 291
  - SQL 70, 215
  - XFER 26, 238
  - use 声明 49
- 源代码 3, 6, 13, 14, 17, 18, 19, 20, 21, 24, 36, 98, 117, 147, 148, 163, 247, 303
  - 查看 148
  - 从 Java 中抽取 87, 97
  - 从 Smalltalk 中抽取 115
  - 试验项目 4
- 运行工具
  - 第 1 阶段 17, 115, 307
    - Java 97
    - Smalltalk 115
  - 第 2 阶段 17, 132
    - 批处理方式 18, 133
    - 用户界面 132
  - 第 3 阶段 18, 137
    - 批处理方式 19
- 运行时差别
  - COBOL
    - 映射 165
    - CALL 165
    - DXFR 165
    - XFER 165
  - Java
    - CALL 165
    - DXFR 165
    - XFER 165

## [ Z ]

- 转换 22, 24, 36, 38, 76, 218, 292
- 资源关联 150, 250, 269, 285, 296, 305
- 资源关联部件 147
  - EGL
    - 查看 153
- 资源库 10, 11, 13, 89, 97, 101, 130
  - 源代码 3, 5, 6, 14, 18, 19, 250
  - Java 14, 16, 17, 97
- 资源库管理 11
- 资源库过滤器 108
- 资源库资源管理器 10
- 子系统 13, 23, 24, 25, 26, 28, 29, 30, 34, 37, 38, 44, 46, 48, 55, 67, 69, 70, 73, 99, 118, 148

## [ 特别字符 ]

- “表及其它记录”列表 38, 61, 212, 250
- “问题”视图 10, 22, 29, 30, 34, 37, 38, 44, 48, 49, 51, 59, 60, 66, 67, 73, 74, 75, 76, 124, 142, 148, 153, 250, 263, 266, 267, 283, 290, 294, 296, 299, 303, 305

## A

- AUDIT 166

## C

- CALL AUDIT 248
- CALL COMMIT 248
- CALL CREATX 248
- CALL CSPTDLI 248
- CALL EZCHART 248
- CALL RESET 248
- CICS 5, 6, 25, 30, 49, 60, 163, 250, 263, 266, 267, 269, 275, 277, 278, 288, 289, 294
  - 不受支持的功能部件
    - 本机环境 167
  - 服务例程
    - 不受支持, 本机环境 166
  - 函数字
    - 不受支持, 本机环境 166
  - 回滚差别 167
  - 提交差别 167
  - 资源关联
    - 不受支持, 本机环境 167
  - CALL CREATX 差别 167
  - EZE 特殊数据字差别
    - EZEAPP 167
    - EZEDEST 167
    - EZEDESTP 167
    - EZELTERM 167
    - EZERCODE 167
    - EZERT8 167
    - EZESEGTR 167
    - EZEUSR 167
    - EZEUSRID 167
    - EZECONCT 差别 167
    - XFER, DXFR 167
- COBOL 生成
  - 生成和测试 154
- containerContextDependent 属性 11, 23, 25, 26, 29, 38
- converse 80

## D

deleteAfterUse 295  
destPort 295  
display 27

## E

evensql 286, 288  
EZE 字 36, 41, 179, 239  
  程序流  
    EZECLOS 239  
    EZEFL0 239  
    EZERTN 239  
  对象脚本编制  
    EZESCRPT 247  
  浮点数学函数  
    EZEFLADD 246  
    EZEFLDIV 246  
    EZEFLMOD 246  
    EZEFLMUL 246  
    EZEFLSET 246  
    EZEFLSUB 246  
  其它数据  
    EZE Aid 242  
    EZEAPP 242  
    EZECNVCM 242  
    EZECONVT 242  
    EZEDEST 242  
    EZEDESTP 242  
    EZEFEc 242  
    EZELOC 242  
    EZE LTERM 242  
    EZEMNO 242  
    EZEMSG 242  
    EZE OVER 242  
    EZE OVERS 242  
    EZERC0DE 242  
    EZEREPLY 242  
    EZERT2 242  
    EZERT8 242  
    EZESEGM 242  
    EZESEGTR 242  
    EZESYS 242  
    EZETST 242  
    EZEUSR 242  
    EZEUSRID 242  
  日期和时间  
    EZEDAY 241  
    EZEDAYL 241  
    EZEDAYLC 241  
    EZEDTE 241  
    EZEDTEL 241  
    EZEDTELC 241  
    EZETIM 241

## EZE 字 (续)

### 三角数学函数

EZEACOS 246  
EZEASIN 246  
EZEATAN 246  
EZEATAN2 246  
EZECOS 246  
EZECOSH 246  
EZESIN 246  
EZESINH 246  
EZETAN 246  
EZETANH 246

### 数学 245

#### 一般函数

EZEBYTES 244  
EZEC10 244  
EZEC11 244  
EZECOMIT 244  
EZECONV 244  
EZEG10 244  
EZEG11 244  
EZEPURGE 244  
EZEROLLB 244  
EZEWAIT 244

#### 一般数学函数

EZEABS 245  
EZECEIL 245  
EZEEXP 245  
EZE FLOOR 245  
EZE FEXP 245  
EZE LDEXP 245  
EZELOG 245  
EZELOG10 245  
EZEMAX 245  
EZEMIN 245  
EZEMODF 245  
EZENCMPR 245  
EZEPOW 245  
EZEPRSCN 245  
EZEROUND 245  
EZESQRT 245

#### 用户界面

EZEUIERR 246  
EZEUILOC 246

#### 字符串

EZESBLKT 245  
EZESCCWS 245  
EZESCMPR 245  
EZESCNCT 245  
EZESCOPY 245  
EZESFIND 245  
EZESNULT 245  
EZESSET 245  
EZESTLEN 245  
EZESTOKN 245

## EZE 字 (续)

### DL/I

EZEDLCER 247  
EZEDLCON 247  
EZEDLDBD 247  
EZEDLERR 247  
EZEDLKEY 247  
EZEDLKYL 247  
EZEDLLEV 247  
EZEDLPCB 247  
EZEDLPRO 247  
EZEDLPSB 247  
EZEDLRST 247  
EZEDLSEG 247  
EZEDLSSG 247  
EZEDLSTC 247  
EZEDLTRM 247

### EZESYS

  不确定情况 80

### EZEWAIT

  不确定情况 82

### SQL

EZECONCT 240  
EZESQC0D 240  
EZESQISL 240  
EZESQLCA 240  
EZESQRD3 240  
EZESQRRM 240  
EZESQWN1 240  
EZESQWN6 240

EZELOC 167

EZEPURGE 166

## F

formGroup 49, 50, 148

## I

import 语句 19, 21, 23, 25, 26, 27, 28,  
  29, 30, 37, 38, 61, 139, 148, 196, 198,  
  250

isDecimalDigit 56, 206

IWN.MIG 282

IWN.VAL 303

IWN.XML 305

## J

Java 和 C++ 差别

  一般 169

  映射 169

  EZE 特殊数据字

    EZECONVT 170

    EZERC0DE 170

Java 和 C++ 差别 (续)

#### SQL

EZESQLCA 170  
EZESQRRM 170  
EZESQWN6 170

Java 生成

生成和测试 155

JDBC 级别, 设置 309

SQL I/O 选项 (续)

DELETE 220, 222, 225  
INQUIRY 220, 222, 225  
REPLACE 220, 222, 225  
SCAN 220, 222, 225  
SETINQ 220, 222, 225  
SETUPD 220, 222, 225  
SQLEXEC 220, 222, 225  
UPDATE 220, 222, 225

SQL I/O 语句 66

## M

MigPreferences.xml 87, 88, 90, 92, 97,  
105, 106

样本 94, 112

## P

PSB 31, 212, 249, 285, 294

## S

SET map PAGE 73

SQL 7, 21, 181, 187, 250, 287, 288, 292

检查项是否为 NULL 76, 293

硬错误 77

语句 41

WHERE 子句 27, 41

SQL 表 17, 67, 287

SQL 查询 311, 312

SQL 行记录 42

SQL 记录 36, 287

备用指定 47

SQL 记录定义 24

SQL 语句

未修改的

不带“执行时语句构建” 220

已修改的

不带“执行时语句构建” 222

带有“执行时语句构建” 225

SQL 子句

FOR UPDATE OF 222, 225

GROUP BY 222, 225

HAVING 222, 225

INTO 222, 225

ORDER BY 222, 225

SELECT 222, 225

WHERE 222, 225

SQL EZE 字 240

SQL I/O 291, 292

执行时语句构建 127

SQL I/O 和缺少的 SQL 子句 68

SQL I/O 和 !itemColumnName 70

SQL I/O 选项

ADD 220, 222, 225

CLOSE 220, 222, 225

## V

VAGen 迁移语法首选项 134

## W

Windows XP

使用 DB2 309

---

# 读者意见表

**Rational Software Development Platform**  
**VisualAge Generator 到 EGL 迁移指南**  
版本 6 发行版 001

---

姓名

---

地址

---

单位及部门

---

电话号码



折起并封口

请勿使用钉书机

折起并封口

在此  
贴上  
邮票

IBM 中国公司上海分公司, 汉化部  
中国上海市淮海中路 333 号瑞安广场 10 楼  
邮政编码: 200021

折起并封口

请勿使用钉书机

折起并封口





程序号: 5724-J19

中国印刷