

仿真器编程



关于本书

本书提供了必要的编程信息，以帮助您使用 HCL Z and I Emulator for Windows 仿真器高级语言应用程序接口 (EHLLAPI)、和 Z and I Emulator for Windows 会话 API (PCSAPI)。主机访问类库在主机访问类库中进行了介绍。

EHLLAPI/PCSAPI 可与 Z and I Emulator for Windows 结合使用，以便为用户和程序员提供了一种访问主机表示空间的方法，其中包含一组可从工作站会话中运行的应用程序调用的功能。

在本书中，Windows 是指 Windows® 7、Windows® 8/8.1、Windows® 10、Windows® Server 2008 和 Windows® Server 2012。当信息仅与特定操作系统相关时，将在文本中指明。

本书的读者对象

本书面向编写应用程序来使用本书中所述 API 的程序员。

其中假定程序员具备 Windows® 的工作知识。有关 Windows® 的信息，请参阅 [从何处查找更多信息 \(on page 2\)](#) 下的出版物列表。

程序员还必须熟悉如何从终端或使用终端仿真软件的工作站连接到主机系统。

本书假定您熟悉所使用的语言和编译器。有关如何编写、编译或链接编辑程序的信息，请参阅 [从何处查找更多信息 \(on page 2\)](#) 以获取要使用的特定语言的相应参考。

从何处查找更多信息

Z and I Emulator for Windows 资料库包含下列出版物：

- 安装指南
- *Quick Beginnings*
- *Emulator User's Reference*
- *Administrator's Guide and Reference*
- *Emulator Programming*
- *Client/Server Communications Programming*
- *System Management Programming*
- 主机访问类库
- *Configuration File Reference*

除了出版的书籍外，还随“Z and I Emulator for Windows”提供了超文本标记语言 (HTML) 文档：

主机访问类库

HACL Java HTML 介绍如何编写符合 ActiveX/OLE 2.0 的应用程序，以将 Z and I Emulator for Windows 用作嵌入式对象。这些文件可以从与 Z and I Emulator for Windows 产品文档一起提供的 Docs_Admin_Aids 压缩文件夹中访问，路径如下：`ZIEWin_3.0_Docs_Admin_Aids.zip\publications\en_US\doc\hacl`

相关出版物列表如下：

- *IBM 3270 Information Display System Data Stream Programmer's Reference*, GA23-0059
 - *IBM 5250 Information Display System Functions Reference Manual*, SA21-9247
-

表示法

每个部分开头的表都说明了 [EHLLAPI 函数 \(on page 28\)](#)、[PCSAPI 函数 \(on page 167\)](#) 和 [WinHLLAPI 扩展函数 \(on page 156\)](#) 中的 API 功能。它显示提供该部分所述功能的产品是否支持某项功能。“是”表示主机类型支持该功能，“否”表示不支持该功能。例如，下表指示功能可用于 3270 和 VT 会话，但不可用于 5250 会话。

3270	5250	VT
是	否	是

第 1 章. 仿真器 API 简介

IBM® Z and I Emulator for Windows 产品提供多个应用程序编程接口 (API)。每个接口都有一组特定的功能，可用于不同的用途。选择最符合应用程序功能需求的编程接口。某些应用程序可能使用多个接口来实现所需的结果。编程接口包括：

- **仿真器高级语言 API (EHLLAPI)**：此接口提供用于访问仿真器“表示空间”数据（如主机屏幕上的字符）的功能。它还提供将击键发送至主机、拦截用户输入的击键、查询主机会话的状态、上载和下载文件的功能，以及其他功能。此接口通常用于自动化操作程序应用程序，它们读取主机屏幕并输入击键，而不需要用户直接介入。请参阅 [EHLLAPI 函数 \(on page 28\)](#)。
 - **IBM® 标准 HLLAPI 支持**：这是一个标准编程接口，允许对主机仿真器会话进行编程访问。请参阅 [IBM Standard EHLLAPI](#)、[IBM Enhanced EHLLAPI](#) 和 [WinHLLAPI 编程简介 \(on page 7\)](#)。
 - **IBM® 增强 HLLAPI 支持**：这是基于 IBM® 标准 HLLAPI 接口的接口。它提供所有现有功能，但使用修改后的数据结构。请参阅 [IBM Standard EHLLAPI](#)、[IBM Enhanced EHLLAPI](#) 和 [WinHLLAPI 编程简介 \(on page 7\)](#)。
 - **Windows® 高级语言 API (WinHLLAPI)**：此接口提供了许多与 IBM® 标准 EHLLAPI 相同的功能，并添加了一些利用 Windows® 环境的扩展。请参阅 [IBM Standard EHLLAPI](#)、[IBM Enhanced EHLLAPI](#) 和 [WinHLLAPI 编程简介 \(on page 7\)](#)。
- 任何接受 \返回窗口句柄和指针的 32 位 API 可能都无法与 HCL ZIEWin 正常配合使用，因为 x86 和 x64 平台之间的指针句柄大小不同。

例如：

API 启动通信通知 (80) 中以字节编号 (9-12) 返回的“数据字符串”参数在 x64 平台上可能会被截断。

- **Z and I Emulator for Windows 会话 API (PCSAPI)**：此接口用于启动、停止和控制仿真器会话和设置。请参阅 [PCSAPI 函数 \(on page 167\)](#)。

对于 Z and I Emulator for Windows Version 3.0，添加了用于控制和检索页面和打印机设置的功能。请参阅 [页面设置功能 \(on page 175\)](#) 和 [打印机设置功能 \(on page 183\)](#)。

- **HCL Z and I Emulator for Windows 主机访问类库 (ECL)**：ECL 是一组对象，允许应用程序程序员和脚本语言编写人员轻松快速地访问主机应用程序。Z and I Emulator for Windows 支持三个不同的 ECL 层 (C++ 对象、ActiveAutomation (OLE) 和 LotusScript Extension (LSX))。有关更多详细信息，请参阅 [Host Access Class Library \(HACL\)](#)。

使用 API 头文件

应用程序在包括 API 头文件之前应包括操作系统头文件。例如：

```
#include <windows.h>      // Windows main header
#include "pcsapi.h"      // ZIEWin PCSAPI header
...
```

临界区

在程序调用仿真器 API 时，请小心使用临界区 (**EnterCriticalSection** 函数)。不要在临界区内进行仿真器 API 调用。如果应用程序的一个线程建立了临界区，而另一个线程位于仿真器 API 调用内，则调用将被暂停，直到您从临界区退出。

在处理 API 调用期间，所有信号（数字协处理器信号除外）都将延迟，直到调用完成或调用需要等待传入数据。此外，从另一个进程发出的 **TerminateProcess** 将被暂挂，直到应用程序完成可能正在处理的 API 调用。

堆栈大小

仿真器 API 在执行时使用调用程序的堆栈。操作系统、应用程序和 API 都需要用于动态变量和函数参数的堆栈空间。在 API 调用时，至少应有 8196 字节 (8K) 的堆栈空间可用。应用程序有责任确保有足够的堆栈空间可用于 API。

Windows x64 平台支持

基于 x64 的 Microsoft® Windows® Server 2008 和 Microsoft® Windows® 8/8.1/10 x64 Editio 版本已经过优化，可运行本机 64 位程序，但是不支持 32 位驱动程序或 16 位应用程序。

对于这些平台，Z and I Emulator for Windows 不会安装以下库。

- 16 位 API 支持：
 - 标准 EHLLAPI 16 位接口
 - WinHLLAPI 16 位接口
 - PCSAPI 16 位接口
-

示例程序

提供了几个示例程序，每个示例程序都说明了其中一种的使用 Z and I Emulator for Windows 蜜蜂。如果您选择安装示例程序，它们将安装在 \SAMPLES 目录中。



Note: 国际商业机器公司按原样提供这些文件，不提供任何明示或暗示的保证，包括但不限于适销性或特定用途适用性的暗示保证。

示例程序文件包括以下内容的源文件和支持文件 Z and I Emulator for Windows 蜜蜂：

- 仿真器高级语言编程接口 (EHLLAPI)
- PCSAPI 函数

以下文件安装在 \SAMPLES 目录中。

Table 1. 示例程序文件

文件名	描述
EHLAPI32.H	IBM®标准 32 位 EHLLAPI 包含文件
WHLLAPI.H	WinHLLAPI 16 位包含文件
HAPI_C.H	EHLLAPI 包含文件
PCSAPI.H	PCSAPI 包含文件
PCSCALLS库	标准接口导入库
PCSCAL32库	导入库以增强界面
EHLAPI32库	IBM®标准 32 位 EHLLAPI 接口的导入库
WHLLAPI库	WinHLLAPI 16 位接口的导入库
WHLAPI32库	WinHLLAPI 32 位接口的导入库

在 \SAMPLES 目录中创建以下子目录。

Table 2. 示例程序子目录

文件名	描述
ECL	Sample\ec\cpp 文件夹包含与 HACL CPP 示例相关的所有文件。 Sample\ec\vb 文件夹包含与 HACL VB.Net 示例相关的所有文件。
HLLSMP	演示如何使用 EHLLAPI 请求击键并登录到 VM 系统。(X86)。 它支持登录、粘贴文本和发送密钥功能。有关使用 hllsmp.exe 的上述功能的详细信息，请参阅 hllsmp\Readme.txt。(X64)。

第 2 章. IBM Standard EHLLAPI、IBM Enhanced EHLLAPI 和 WinHLLAPI 编程简介

本章提供将 IBM® 标准 EHLLAPI (16 位和 32 位)、WinHLLAPI (16 位和 32 位) 和 IBM® 增强 32 位 EHLLAPI (EHLLAPI32) 函数合并到以高级语言编写的应用程序中所需的信息。其中提供了有关调用格式、内存分配注意事项、初始化接口以及编译和链接应用程序的详细信息。还包括一个简短的 EHLLAPI 程序样本，以及用于构建此程序样本的编译/链接指令。最后，介绍了 EHLLAPI 接口 (场景) 的一组可能用途。

EHLLAPI 应用程序是使用 EHLLAPI 接口访问主机 3270/5250/VT 表示空间的任何应用程序。表示空间包括可见的仿真器字符数据、字段和属性数据、击键数据和其他信息。

EHLLAPI 概述

下面概述了 HLLAPI 编程接口。

IBM 标准 EHLLAPI

EHLLAPI 是一个标准编程接口，允许对主机仿真器会话进行编程访问。提供的功能用于读取主机屏幕数据 (如字符和属性)，发送键，以及执行与仿真器相关的其他功能。

EHLLAPI 接口是单个调用点接口。有一个可调用的 API，通过它可以请求所有 EHLLAPI 函数。在对接口的每次调用中，应用程序都会提供一个用于标识所请求函数的函数编号，一个指向数据缓冲区的指针，一个指向数据缓冲区长度的指针，以及一个指向返回码的指针 (请参阅 [EHLLAPI 调用格式 \(on page 8\)](#))。

WinHLLAPI

WinHLLAPI 基于熟悉的 EHLLAPI API。它包含所有现有功能，并添加了利用 Windows® 消息驱动型环境的扩展。除非包含 WinHLLAPI 扩展，否则 HCL Z and I Emulator for Windows EHLLAPI 接口的用户不会注意到任何功能差异。

[WinHLLAPI 扩展函数 \(on page 156\)](#) 中介绍了 WinHLLAPI 扩展函数，以及任何与 EHLLAPI 表单不符的函数。有关常用函数的信息，请参阅 [EHLLAPI 函数 \(on page 28\)](#)。

WinHLLAPI 和 IBM® 标准 EHLLAPI

WinHLLAPI 的条目符号相应地为 **WinHLLAPI**。希望切换到 WinHLLAPI 实现的 EHLLAPI 用户必须从 **hllapi** 标准条目更改。新用户应遵循 [EHLLAPI 函数 \(on page 28\)](#) 中的所有指示，并使用 **WinHLLAPI** 条目代替标准 **hllapi** 条目。

IBM 增强 EHLLAPI 和 IBM 标准 EHLLAPI

IBM 增强 EHLLAPI 基于熟悉的 EHLLAPI API。它包含所有现有功能，但利用 32 位环境并使用修改后的数据结构。希望切换到 IBM® 增强 32 位 EHLLAPI 的标准接口用户只需在第一个、第三个和第四个参数中将条目符号从 LPWORD 更改为 LPINT。新用户应使用以下各节中的过程。

语言

任何可以调用 DLL 中具有“Pascal”调用约定的入口点的编程语言都可用于执行 EHLLAPI 函数。但是，Z and I Emulator for Windows EHLLAPI 工具包仅为 C++ 语言提供头文件和函数原型。使用任何其他语言都需要清楚地了解数据结构布局和调用约定。EHLLAPI 工具包支持以下 C/C++ 编译器：

- Microsoft® Visual C/C++ V4.0 和更高版本

大多数其他 C/C++ 编译器也可以使用该工具包。

EHLLAPI C/C++ 应用程序必须包含 Z and I Emulator for Windows EHLLAPI 头文件 (HAPI_C.H)。此文件定义了数据结构的布局，并为 EHLLAPI 入口点提供了原型。



注： 16 位和 32 位应用程序的数据结构布局不同（请参阅 [标准和增强接口注意事项 \(on page 22\)](#)）。

EHLLAPI 调用格式

EHLLAPI 入口点 (**hllapi**) 始终使用以下四个参数调用：

1. EHLLAPI 函数编号 (输入)
2. 数据缓冲区 (输入/输出)
3. 缓冲区长度 (输入/输出)
4. 位置 (输入) ; 返回码 (输出)

IBM® 标准 EHLLAPI 的原型是：

```
[long hllapi (LPWORD, LPSTR, LPWORD, LPWORD);
```

IBM® 增强 EHLLAPI 的原型是：

```
[long hllapi (LPINT, LPSTR, LPINT, LPINT);
```

每个参数都是通过引用而不是通过值传递的。因此，函数调用的每个参数都必须是指向值的指针，而不是值本身。例如，以下是调用 EHLLAPI 查询会话状态功能的正确示例：

```
#include "hapi_c.h"
struct HLDQuerySessionStatus QueryData;
int    Func, Len, Rc;
long   Rc;

memset(QueryData, 0, sizeof(QueryData)); // Init buffer
QueryData.qsst_shortcode = 'A';          // Session to query
Func = HA_QUERY_SESSION_STATUS;         // Function number
Len = sizeof(QueryData);                 // Len of buffer
Rc = 0;                                   // Unused on input

hllapi(&Func, (char *)&QueryData, &Len, &Rc); // Call EHLLAPI
if (Rc != 0) {                             // Check return code
    // ...Error handling
}
```

hllapi 调用中的所有参数都是指针，并且 EHLLAPI 函数的返回码以第 4 个参数的值返回，而不是作为函数的值返回。

例如，以下项**不正确**：

```
if (hllapi(&Func, (char *)&QueryData, &Len, &Rc) != 0) { // WRONG!
    // ...Error handling
}
```

虽然 **hllapi** 函数被定义为针对 IBM® 标准和增强 EHLLAPI 返回 **long** 数据类型，而针对 WinHLLAPI 返回 **void** 数据类型，但其值未定义，不应使用。

hllapi 调用的第二到第四个参数可以将信息返回到应用程序。每个 EHLLAPI 函数的描述均描述了在这些参数中返回的信息（如有）。

数据结构

许多 EHLLAPI 函数使用格式化数据结构将信息传递到应用程序或从应用程序传递信息。每个函数的描述均显示数据结构的布局。传递到或自 EHLLAPI 函数的数据必须完全按照记录的方式存在于存储中，逐个字节对应。请注意，结构布局对于所有 IBM® 标准和 WinHLLAPI 16 位和 32 位应用程序均相同。IBM® 增强 32 位应用程序的数据结构打包为 4 字节对齐。

强烈建议使用提供的头文件和数据结构定义来确保正确的数据对齐和布局。虽然在技术上是可行的，但建议不要执行以下操作：

```
char QueryData[20]; // Not recommended
...
Func = HA_QUERY_SESSION_STATUS;
hllapi(&Func, QueryData, &Len, &Rc);
if (QueryData[13] == 'F') {
    // ...this is a 5250 session
}
```

编写此函数的建议方法是：

```
#include "hapi_c.h"
struct HLDQuerySessionStatus QueryData; // Recommended
...
Func = HA_QUERY_SESSION_STATUS;
hllapi(&Func, (char *)&QueryData, &Len, &Rc);
if (QueryData.qsst_sestype == 'F') {
    // ...this is a 5250 session
}
```

内存分配

EHLLAPI 函数不会分配或释放内存。在调用 **hllapi** 入口点之前，应用程序必须为 EHLLAPI 函数预先分配所需的缓冲区空间。缓冲区空间可以作为动态变量预先分配，例如：

```
struct HLDQuerySessionStatus QueryBuff;
```

或者，可以通过调用 C 库或操作系统功能来分配，例如：

```

struct HLDQuerySessionStatus *QueryBuff;
...
QueryBuff = malloc(sizeof(struct HLDQuerySessionStatus));

```

在任何情况下，应用程序都负责在调用 EHLLAPI 函数之前分配足够的缓冲区空间，并在不需要缓冲区时释放缓冲区。

EHLLAPI 返回码

EHLLAPI 功能会在 `hllapi` 函数调用的第 4 个参数中返回完成码或返回码（**转换位置** 或 **RowCol** (99) 功能除外）。返回码表示请求的功能是成功还是失败。

除非在每个功能的描述中另有说明，下表显示了每个返回码值的含义。某些功能对这些返回码的解释可能略有不同；有关详细信息，请参阅各个功能的描述。

表 3. EHLLAPI 返回码

返回码	说明
0	功能已成功执行，或者自上次发出调用后没有更新。
1	指定的主机表示空间标识不正确。指定的会话未连接、不存在或为逻辑打印机会话。
2	遇到参数错误，或指定的功能编号不正确。（有关详细信息，请参阅各个功能。）
4	由于目标表示空间正忙，处于 X CLOCK 状态 (X [])，或处于 X SYSTEM 状态，因此此功能的执行被禁止。
5	出于返回码 4 中所述原因之外的原因，功能的执行被禁止。
6	由于指定的参数不正确（例如，导致截断的长度错误），因此出现数据错误。
7	指定的表示空间位置无效。
8	遇到功能过程错误（例如，使用冲突功能或缺少必备功能）。
9	遇到系统错误。
10	此选项不可用于 EHLLAPI。
11	此资源不可用。
12	此会话已停止。
24	未找到字符串，或表示空间未设置格式。
25	输入队列上没有可用的击键。
26	发生主机事件。有关详细信息，请参阅 查询主机更新 (24)。
27	文件传输已被 Ctrl+Break 命令终止。
28	字段长度为 0。
31	击键队列溢出。击键丢失。
32	应用程序已连接到此会话以进行通信。
33	保留。
34	发送到主机的消息已被取消。
35	从主机发送的消息已被取消。
36	与主机的联系已丢失。
37	入站通信已被禁用。

表 3. EHLLAPI 返回码 (续)

返回码	说明
38	请求的功能尚未完成其执行。
39	另一个 DDM 会话已连接。
40	断开连接尝试成功，但存在在断开连接时尚未完成的异步请求。
41	请求的缓冲区正被另一个应用程序使用。
42	没有匹配的未完成请求。
43	API 已被另一个 EHLLAPI 应用程序 (LOCK 上) 锁定或 API 未被锁定 (UNLOCK 上)。

编译和链接

应用程序可以采用动态链接的方式进行链接。这意味着我们可以通过执行动态链接来链接入口点。在这种情况下，应用程序使用操作系统调用来加载正确的 DLL 并在运行时获取入口点地址。

下表显示了哪些 DLL 应该用于动态加载。

接口	入口点	DLL
IBM®标准 (64 位)	赫拉皮	EHLAPI32.DLL
IBM®增强型 (64 位)	赫拉皮	PCSHLL32.DLL
WinHLLAPI (64 位)	温赫拉皮	WHLAPI32.DLL

动态链接方法

使用动态链接方法，应用程序可在运行时调用操作系统，以装入 Z and I Emulator for Windows EHLLAPI 模块并在其中找到 **hllapi**。此方法在应用程序中需要更多代码，但使应用程序能够更好地控制错误情况。例如，如果找不到 Z and I Emulator for Windows EHLLAPI 模块，应用程序可以向用户显示特定的错误消息。

要使用动态链接，应用程序需要装入相应的 Z and I Emulator for Windows 模块并找到入口点。建议按序号而非名称找到入口点。可在头文件中定义此序号。以下 32 位 Windows® 代码将装入 IBM® 标准 32 位 EHLLAPI 模块，找到 **hllapi** 入口点，以及进行 EHLLAPI 函数调用。

```
#include "hapi_c.h"

HMODULE Hmod; // Handle of PCSHLL32.DLL
long (APIENTRY hllapi)(int *, char *, int *, int *); // Function pointer
int HFunc, HLen, HRc; // Function parameters
char HBuff[1]; // Function parameters

Hmod = LoadLibrary("PCSHLL32.DLL"); // Load EHLLAPI module
if (Hmod == NULL) {
    // ... Error, cannot load EHLLAPI module
}

hllapi = GetProcAddress(Hmod, MAKEINTRESOURCE(ord_hllapi));
```

```

// Get EHLLAPI entry point
if (hllapi == NULL) {
    // ... Error, cannot find EHLLAPI entry point
}

HFunc = HA_RESET_SYSTEM; // Run EHLLAPI function
HLen = 0;
HRc = 0;
(*hllapi>(&HFunc, HBuf, &HLen, &HRc);
if (HRc != 0) {
    // ... EHLLAPI access error
}

```

多线程

HCL 增强型 EHLLAPI (32 位) 和 HCL® 标准 EHLLAPI 16 位基于每个进程进行连接。所有线程都访问同一个连接的主机会话。执行连接的线程也必须执行断开连接。

HCL® 标准 EHLLAPI (32 位) 和 WinHLLAPI 基于每个线程进行连接。每个线程必须维护自己的连接。这允许多线程进程一次维持与多个已连接主机会话的连接。当使用 WinHLLAPI 程序协调不同主机之间的数据时，这就消除了对多进程方案的需要。它还将必要时连接和断开连接的负担置于单个线程上。

表示空间

许多 EHLLAPI 函数需要表示空间标识 (*PSID*) 来指示该函数将使用哪个主机仿真器会话。(这也称为短会话标识)。表示空间标识是 A 到 Z 范围内的单个字符。最多有 26 个会话。

IBM® 增强型 32 位接口表示空间标识

对于 IBM® 增强型 EHLLAPI 应用程序，会使用三个附加字节扩展会话标识。必须将这些扩展会话字节设置为零，以便将来兼容。在填入所需信息之前，将 EHLLAPI 缓冲区的内容设置为全二进制零最容易实现此目的。例如，可以使用以下命令查询会话 B 的状态：

```

#include "hapi_c.h"
int HFunc, HLen, HRc; // Function parameters
struct HLDPMWindowStatus StatusData; // Function parameters

Func = HA_PM_WINDOW_STATUS;
HLen = sizeof(StatusData);
HRc = 0;

// Set data buffer to zeros and fill in request
memset(&StatusData, 0x00, sizeof(StatusData));
StatusData.cwin_shortcode = 'B'; // Short session ID
StatusData.cwin_option = 0x02; // Query command

hllapi(&HFunc, (char *)&StatusData, &HLen, &HRc);

```

表示空间的类型

可以将仿真器会话配置为显示会话或打印机会话。EHLLAPI 应用程序无法连接到 PC400 的打印机或路由器会话。**查询会话 (10)** 功能可用于确定特定会话的类型。

表示空间的大小

仿真器显示会话可配置为从 1920 字节 (24x80 屏幕大小) 到 9920 字节 (62x160 屏幕大小) 的各种屏幕大小。某些 EHLLAPI 功能 (如**将 PS 复制到字符串 (8)**) 需要应用程序分配足够的存储空间来存放 (可能) 整个表示空间。可使用**查询会话状态 (22)** 功能获取给定会话的表示空间大小。

表示空间标识

EHLLAPI 功能一次只能与一个表示空间交互。表示空间标识 (PSID) 用于标识功能要在其中运行的特定表示空间。

对于某些功能, PSID 包含在对**连接表示空间 (1)** 功能的前面调用中。对于其他功能, PSID 包含在调用数据字符串参数中。

主机连接的表示空间

通过使用**连接表示空间 (1)** 和**断开表示空间 (2)**, 可控制与主机表示空间 (或会话) 的连接。连接的状态决定了是否可以执行某些功能。它还会影响 PSID 的定义方式。下面的文本说明了如何控制与主机表示空间的连接状态:

- 在任何给定时间, 都可能不存在任何连接主机的表示空间, 也可能存在一个且只有一个主机连接的表示空间。
 - 没有缺省的主机连接的表示空间。
 - 连接后, 有一个且只有一个主机连接的表示空间。连接的主机表示空间在连接功能的调用数据字符串参数中标识。
 - 可以在没有中断断开连接的情况下执行后续的连接调用。在这种情况下, 仍有一个且只有一个主机连接的表示空间。同样, 连接的主机表示空间在连接功能的调用数据字符串参数中标识。
 - 断开连接后, 没有主机连接的表示空间。在多次连续调用连接后或在单次调用连接后, 适用此规则。
 - 无法连接到逻辑打印机会话。
-

表示空间标识处理

PSID 用于指定您希望某个功能在其中运行的主机表示空间 (或会话)。PSID 处理方式受两个因素影响:

1. 用于指定 PSID 的方法:
 - a. 作为先前的**连接表示空间 (1)** 功能调用的调用数据字符串参数
 - b. 作为正在执行的功能的调用数据字符串中的一个字符。处理方式会因字符是否为以下项而异:
 - 字母 A 至 Z
 - 空格或 null
2. 与主机表示空间的连接的状态。

以下段落介绍了如何针对这两种因素的各种组合处理 PSID。

需要连接的功能的 PSID 处理

某些功能仅与主机连接的表示空间交互。这些功能需要将**连接表示空间** (1) 功能作为前提条件调用。这些功能的 PSID 由**连接表示空间** (1) 和**断开表示空间** (2) 功能确定，如下所示：

- 当没有主机连接的表示空间时，这些功能不会与任何表示空间交互。生成返回码 1。
 - 当有一个主机连接的表示空间时，这些功能将与在最近的**连接表示空间** (1) 功能调用的调用数据字符串参数中指定的表示空间进行交互。
-

不需要连接的功能的 PSID 处理

某些功能可以与主机表示空间交互（无论是否已连接）。这些功能允许您在调用数据字符串参数中指定 PSID。如下所示：

- **连接表示空间** (1)
- **转换位置 RowCol** (99)
- **获取键** (51)
- **拦截后状态** (52)
- **查询关闭拦截** (42)
- **查询主机更新** (24)
- **查询会话状态** (22)
- **启动关闭拦截** (41)
- **启动主机通知** (23)
- **启动击键拦截** (50)
- **停止关闭拦截** (43)
- **停止主机通知** (25)
- **停止击键拦截** (53)

除前两个功能外，所有其他功能都允许您使用以下任一项指定 PSID：

- 字母 A 至 Z
- 空格或 null

前两个功能要求使用字母来指定 PSID。

如果没有主机连接的表示空间，则适用以下规则：

- 如果使用字母（不是空格或 null）指定 PSID，则功能可以与任何主机表示空间交互。
- 如果使用空格或 null 来指定 PSID，则会生成返回码 1。功能不执行。
- 使用字母指定 PSID 不会建立主机连接的表示空间，但在连接 PS 请求上除外。

如果有一个主机连接的表示空间，则适用以下规则：

- 如果使用字母指定 PSID，则功能可以与任何主机表示空间交互。
- 如果使用空格或 null 来指定 PSID，则功能将在**连接表示空间** (1) 功能的最近调用中标识的表示空间中运行。
- 使用字母指定 PSID 不会更改主机连接的表示空间的已建立 PSID，但在连接 PS 请求上除外。

以下功能可以用于打印机会话：

- **启动主机通知** (23)
- **查询主机更新** (24)
- **停止主机通知** (25)

在进程之间共享 EHLLAPI 表示空间

如果应用程序支持共享（即，如果它们被开发为协同工作，或者它们表现出可预测的行为¹），则多个 EHLLAPI 应用程序可以共享一个表示空间。要确定哪些应用程序支持共享，可将 EHLLAPI 应用程序指定为以下类型之一：

- 管理
- 允许读取权限的独占写入
- 不允许读取权限的独占写入
- 超级写入
- 读取

通过在**设置会话参数** (9) 功能调用中为每个功能设置以下读写共享选项，可以定义共享访问权的类型：

SUPER_WRITE

此应用程序允许其他允许共享和具有写入访问权的应用程序并发连接到相同的表示空间。发起应用程序执行管理类型功能，但不会为共享表示空间的其他应用程序创建错误。

WRITE_SUPER

应用程序需要写入访问权，并且仅允许管理应用程序并发连接到其表示空间。这是缺省值。

WRITE_WRITE

应用程序需要写入访问权，并允许具有可预测行为的合作伙伴或其他应用程序共享表示空间。

WRITE_READ

应用程序需要写入访问权，并允许执行只读函数的其他应用程序共享表示空间。还允许应用程序复制表示空间，并照常执行其他只读操作。

1. 这意味着两个 EHLLAPI 程序将不会同时争夺同一个表示空间；或者，这些程序中有允许程序等至 PS 可用的逻辑；或者，应用程序从不以锁定其他应用程序的方式使用会话。

WRITE_NONE

应用程序可独占使用表示空间。不允许其他应用程序共享表示空间，包括管理应用程序。允许应用程序复制表示空间并照常执行只读操作。

READ_WRITE

应用程序仅需要读取访问权即可监控表示空间，并允许执行读取或写入功能或者同时执行这两项功能的其他应用程序共享表示空间。还允许应用程序复制表示空间，并照常执行其他只读操作。

 **注：** 进程中线程之间无法共享表示空间。

表 4. EHLLAPI 读写共享选项组合

调用应用程序	Super_Write	Write_Super	Write_Write	Write_Read	Write_None	Read_Write
Super_Write	是	是	是	否	编号	是
Write_Super (缺省值)	是	否	编号	编号	编号	编号
Write_Write	是	否	是	否	编号	是
Write_Read	编号	编号	编号	编号	编号	是
Write_None	编号	编号	编号	编号	编号	编号
Read_Write	是	否	是	是	否	是

除了指定兼容的读取和写入访问选项外，设计为协同工作但不允许其他人在同一表示空间中工作的应用程序还可以在**设置会话参数 (9)** 功能调用中选择定义关键字 KEY\$nnnnnnnn。此关键字仅允许使用相同关键字的应用程序共享表示空间。

 **注：**

1. **启动击键拦截 (50)** 功能不可共享。一次只有一个应用程序可以捕获击键。
2. **连接到表示空间 (1)** 和**启动击键拦截 (50)** 功能共享通用子系统功能。应用程序成功请求共享其中任一功能会影响其他应用程序对这两个功能的请求。例如，如果应用程序 A 成功地请求具有 Write_Read 访问权并将 KEY\$abcdefgh 用作关键字的**连接到表示空间 (1)**，则仅当两个应用程序都设置了兼容的读取和写入选项时，应用程序 B 的**连接到表示空间 (1)** 和**启动击键拦截 (50)** 请求才会成功。

表 5. 必备功能和关联的依赖功能

前提条件调用	函数	访问
分配通信缓冲区 (120)	释放通信缓冲区 (120)	不适用
连接窗口服务 (101)	更改 PS 窗口名称 (106) 更改交换机列表名称 (105) 断开窗口	写入 读取

表 5. 必备功能和关联的依赖功能 (续)

前提条件调用	函数	访问
	服务 (102) 查询窗口服务 (103) 窗口状态 (104)	查询=读取 设置=写入 写入
连接表示空间 (1)	将字段复制到字符串 (34) 复制 OIA (13) 复制表示空间 (5) 将表示空间复制到字符串 (8) 将表示空间复制到剪贴板 (35) 将字符串复制到字段 (33) 将字符串复制到表示空间 (15) 断开表示空间 (2) 查找字段长度 (32) 查找字段位置 (31) 查询光标位置 (7) 查询字段属性 (14) 将剪贴板粘贴到表示空间 (36) 释放 (12) 保留 (11) 搜索字段 (30) 搜索表示空间 (6) 发送键 (3) 设置光标 (40) 开始播放宏 (110) 等待 (4)	读取 读取 读取 读取 读取 写入 写入 写入 读取 读取 读取 读取 写入 写入 写入 读取 读取 读取 写入 写入 写入 读取 读取 读取 写入 写入 读取
连接结构化字段 (120)	断开结构化字段 (121) 获取请求完成 (125) 读取结构化字段 (126) 写入结构化字段 (127)	不适用
读取结构化字段 (126)	获取请求完成 (125)	不适用
启动关闭拦截 (41)	查询关闭拦截 (42) 停止关闭拦截 (43)	不适用
启动主机通知 (23)	查询主机更新 (24) 停止主机通知 (25)	
启动击键拦截 (50)	获取键 (51) 拦截后状态 (52) 停止击键拦截 (53) 如果要发送编辑击键, 则为发送键 (3) (在增强方式下提供编辑击键支持)	不适用

表 5. 必备功能和关联的依赖功能 (续)

前提条件调用	函数	访问
写入结构化字段 (127)	获取请求完成 (125)	不适用

锁定表示空间

即使指定了共享表示空间，应用程序也可以使用**锁定表示空间** (60) 或**锁定 Windows® 服务 API** (61) 功能，获得对表示空间的独占控制。当发起应用程序解锁表示空间时，其他应用程序使用这些功能锁定的表示空间的请求将按先进先出 (FIFO) 的顺序排队和处理。

如果锁定表示空间的应用程序未过使用**解锁选项**或**重置系统** (21) 调用来将其解锁，则当应用程序终止或会话停止时，锁定将被解除。

使用鼠标操作在表示空间上选择、复制和粘贴文本

表示空间中可以使用以下鼠标操作。

- 双击鼠标左键选择一个字。
- 单击鼠标右键复制所选的字。
- 双击鼠标右键粘贴复制的字。

ASCII 助记符

源自主机键盘的击键可能具有相应的 ASCII 值。**获取键** (51) 功能对击键的响应取决于是否定义了键，以及该键是被定义为 ASCII 值还是 ASCII 助记符。

一个会话的键盘可能无法生成另一个会话所需的某些代码。表示这些代码的 ASCII 助记符可以包含在**发送键** (3) 功能的数据字符串参数中。

发送键 (3) 功能和**获取键** (51) 功能的能力允许会话交换可能不由 ASCII 值或可用键表示的击键。一组助记符（可从键盘生成）可用。这些助记符允许您使用 ASCII 字符来表示工作站键盘的特殊功能键。

非换挡键的助记符由转义字符后跟缩写组成。对于 Shift 键本身、上 Shift、Alt 和 Ctrl 也是如此。换挡键的助记符包括 Shift 键的助记符，然后是非换挡键的助记符。因此，换挡键的助记符是转义字符、缩写、转义字符、缩写的 4 个字符序列。

缺省转义字符为 @。可以使用 **设置会话参数** (9) 功能的 `ESC=C` 选项将转义字符的值更改为任何其他字符。但是，以下文本使用缺省转义字符。

不属于 ASCII 字符集的偏移量指示符通过 2 字节 ASCII 助记符呈现给主机应用程序，如下所示：

上 Shift	@S
Alt	@A
Ctrl	@r

这些 Shift 指示符的助记符不会由应用程序单独接收。同样，它们也不会通过应用程序单独发送。Shift 指示符助记符始终伴有非 Shift 指示符字符或助记符。

使用的缩写使特殊键的助记符容易记住。最常用的键使用字母键代码。例如，Clear 键为 C；Tab 键为 T，依此类推。请注意，大写和小写字母字符是不同键的助记符缩写。

下面的文本介绍了这些功能的使用。

常规

所有定义的键均由以下任一项表示：

- 1 字节 ASCII 值，它是 256 元素 ASCII 字符集的一部分，或
- 2 字节、4 字节或 6 字节 ASCII 助记符

要表示定义为 ASCII 字符的键，将使用与该字符对应的 1 字节 ASCII 值。

要表示定义为功能的键，将使用与该功能对应的 2、4 或 6 字节 ASCII 助记符。例如，要表示退格制表键，将使用 @B。

要表示 PF1，将使用 @1。要表示“擦除输入”，将使用 @A@F。请参阅以下列表：

@B	左 Tab 键	@0	主页	@h	PF17
@C	Clear	@1	PF1/F1	@i	PF18
@D	删除	@2	PF2/F2	@j	PF19
@E	Enter	@3	PF3/F3	@k	PF20
@F	擦除 EOF	@4	PF4/F4	@l	PF21
@H	帮助 (PC400)	@5	PF5/F5	@m	PF22
@I	插入	@6	PF6/F6	@n	PF23
@J	跳转	@7	PF7/F7	@o	PF24
@L	光标左移	@8	PF8/F8	@q	End 键
@N	换行	@9	PF9/F9	@u	Page UP (PC400)
@O	空格键	@a	PF10/F10	@v	Page Down (PC400)
@P	打印	@b	PF11/F11	@x	PA1
@R	重置	@c	PF12/F12	@y	PA2
@T	右 Tab 键	@d	PF13	@z	PA3
@U	光标上移	@e	PF14	@@	@ (at) 符号
@V	光标下移	@f	PF15	@\$	备用光标
@Z	光标右移				

@A@C	测试 (PC400)	@A@e	粉红色 (PC/3270)
@A@D	字词删除	@A@f	绿色 (PC/3270)
@A@E	字段退出	@A@g	黄色 (PC/3270)
@A@F	擦除输入	@A@h	蓝色 (PC/3270)

@A@H	系统请求	@A@i	青绿色 (PC/3270)
@A@I	插入切换	@A@j	白色 (PC/3270)
@A@J	光标选择	@A@l	重置主机颜色 (PC/3270)
@A@L	快速光标左移	@A@t	打印 (个人计算机)
@A@Q	注意	@A@u	汇总 (PC400)
@A@R	设备取消	@A@v	下滚 (PC400)
@A@T	打印表示空间	@A@y	正向字词制表符
@A@U	快速光标上移	@A@z	反向字词制表符
@A@V	快速光标下移	@A@-	字段 - (PC400)
@A@Z	快速光标右移	@A@+	字段 + (PC400)
@A@9	反相显示	@A@<	记录退格键 (PC400)
@A@b	下划线 (PC/3270)	@S@E	主机上的打印表示空间 (PC400)
@A@c	重置反相显示 (PC/3270)	@S@x	Dup
@A@d	红色 (PC/3270)	@S@y	字段标记



注:

1. 第一个表中的第一个 @ 符号表示转义字符。第二个表中的第一个和第二个 @ 符号是转义字符。@ 符号是缺省转义字符。可以使用 **设置会话参数** (9) 功能的 `ESC=c` 选项更改转义字符的值。

如果将转义字符更改为 #, 则用于表示退格制表、主页和擦除输入键的文字序列将分别变为 #B、#0 和 #A#F。

此外, 用于表示 @ 符号的文字序列将变为 #@。

2. 如果发送打印屏幕的助记符 (即 @P 或 @A@T), 请将其放在调用数据字符串的末尾。
3. 如果发送用于设备取消的助记符 (即 @A@R), 则会传递该助记符而不显示错误消息; 但是, 不会停止本地复制。

获取键 (51) 功能

如果终端操作员输入定义为 ASCII 字符的键, 则主机应用程序将收到与该字符对应的 1 字节 ASCII 值。

如果操作员输入定义为功能的键, 主机应用程序将收到一个 2 字节、4 字节或 6 字节的 ASCII 助记符, 该助记符与该功能对应。例如, 如果输入 **退格制表** 键, 则会收到 @B。如果按下 **PF1** 键, 则会收到 @1。如果按下 **擦除输入**, 则会收到 @A@F。

如果操作员输入定义的 Shift 键组合, 则主机应用程序将接收到 ASCII 字符, 或与定义的字符或功能相对应的 2、4 或 6 字节 ASCII 助记符。

如果操作员输入未定义的单个键, 则 **获取键** (51) 功能将返回 20 返回码, 并且不会向主机应用程序发送任何内容。

获取键 (51) 功能使用两个 ASCII 字符将发送到主机应用程序的所有字符和助记符作为前缀。第一个 ASCII 字符是将击键发送到的主机表示空间的 PSID。其他字符是 A、S 或 M，分别表示 ASCII、特殊 Shift 或助记符。请参阅 [返回参数 \(on page 77\)](#)。

发送键 (3) 功能

要将 ASCII 字符发送到另一个会话，请在**发送键 (3)** 功能的数据字符串参数中包括该字符。

要将功能键发送到另一个会话，请在**发送键 (3)** 功能的数据字符串参数中包含该函数的 ASCII 助记符。

如果**发送键 (3)** 功能向主机会话发送无法识别的助记符，可能会导致返回码拒绝该键。

正在调试

作为调试 EHLLAPI 应用程序的辅助工具，可使用 Z and I Emulator for Windows 的跟踪工具。此工具将生成所有 EHLLAPI 调用、参数、返回值和返回码的日志。有关使用跟踪工具的更多信息，请参阅 *Administrator's Guide and Reference*。

简单的 EHLLAPI 样本程序

以下样本 Windows® 应用程序将在主机会话“A”的第一个输入字段中输入字符串“ello World!”。

```
#include <stdlib.h>
#include <stdio.h>
#include <windows.h>
#include "hapi_c.h"

int main(char **argv, int argc) {
    int HFunc, HLen, HRc;
    char HBuff[1];
    struct HLDConnectPS ConnBuff;
    // Send Key string for HOME+string+ENTER:
    char SendString[] = "@0Hello World!@E";

    HFunc = HA_RESET_SYSTEM;
    HLen = 0;
    HRc = 0;
    hllapi(&HFunc, HBuff, &HLen, &HRc);
    if (HRc != HARC_SUCCESS) {
        printf("Unable to access EHLLAPI.\n");
        return 1;
    }

    HFunc = HA_CONNECT_PS;
    HLen = sizeof(ConnBuff);
    HRc = 0;
    memset(&ConnBuff, 0x00, sizeof(ConnBuff));
    ConnBuff.steps_shortcode = 'A';
    hllapi(&HFunc, (char *)&ConnBuff, &HLen, &HRc);
    switch (HRc) {
        case HARC_SUCCESS:
```

```
case HARC_BUSY:
case HARC_LOCKED: // All these are OK
    break;
case HARC_INVALID_PS:
    printf("Host session A does not exist.\n");
    return 1;
case HARC_UNAVAILABLE:
    printf("Host session A is in use by another EHLLAPI application.\n");
    return 1;
case HARC_SYSTEM_ERROR:
    printf("System error connecting to session A.\n");
    return 1;
default:
    printf("Error connecting to session A.\n");
    return 1;
}

HFunc = HA_SENDKEY;
HLen = strlen(SendString);
HRc = 0;
hllapi(&HFunc, SendString, &HLen, &HRc);
switch (HRc) {
    case HARC_SUCCESS:
        break;
    case HARC_BUSY:
    case HARC_LOCKED:
        printf("Send failed, host session locked or busy.\n");
        break;
    default:
        printf("Send failed.\n");
        break;
}

HFunc = HA_DISCONNECT_PS;
HLen = 0;
HRc = 0;
hllapi(&HFunc, HBuf, &HLen, &HRc);

printf("EHLLAPI program ended.\n");
return 0;
}
```

可以使用以下命令构建应用程序:

```
nmake /a all
```

标准和增强接口注意事项

给定平台上的标准和增强 EHLLAPI 接口之间没有功能差异。但是，有其他重要区别：

- 增强 EHLLAPI 接口将表示空间标识 (PSID) 从 1 字节扩展到 4 字节。当前未使用其他字节，但应用程序应将其设置为二进制零，以确保与增强 EHLLAPI 的未来版本兼容。
- 传入和传出 EHLLAPI 功能的内存缓冲区中数据元素的位置（偏移量）不同。增强 EHLLAPI 中的数据元素与双字边界对齐。标准 EHLLAPI 中的数据元素不会以任何特定方式对齐。不对 EHLLAPI 应用程序进行编码，以便通过偏移量（字节）值设置或检索缓冲区中的数据。相反，应该使用 HAPLC.H 文件中提供的数据结构来设置和检索数据元素。这将确保从 16 位和 32 位程序的正确位置设置和检索数据。

通过在 EHLLAPI 数据缓冲区中预先填充二进制零，并使用 HAPLC.H 中提供的数据结构，可以对应用程序进行编译，以实现标准或增强操作，而无需更改任何源代码。例如，以下部分的代码适用于标准 EHLLAPI，但对增强 EHLLAPI 将失败：

```
#include "hapi_c.h"
...
int Func, Len, Rc;
char Buff[18];
char SessType;

Func = HA_QUERY_SESSION_STATUS; // Function
Len = 18;                        // Buffer length
Rc = 0;
Buff[0] = 'A'                    // Session to query
hllapi(&Func, Buff, &Len, &Rc); // Execute function

SessType = Buff[9];              // Get session type
...
```

如果编译为增强 EHLLAPI 应用程序，上述示例将失败，原因是：

- 应用程序未将扩展会话标识字节设置为零。
- 此功能的缓冲区长度为 20，而不是 18。
- 会话类型指示符不在数据缓冲区中的偏移量 9 处，而是在偏移量 12 处。

如果编译为标准操作或增强操作，则以下功能是为正确工作而编写的相同功能。已更改的行以 > 表示：

```
#include "hapi_c.h"
...
int Func, Len, Rc;
> struct HLDQuerySessionStatus Buff;
char SessType;

Func = HA_QUERY_SESSION_STATUS; // Function
> Len = sizeof(Buff);           // Buffer length
Rc = 0;
> memset(&Buff, 0x00, sizeof(Buff)); // Zero buffer
> Buff.qsst_shortcode = 'A';    // Session to query
hllapi(&Func, (char *)&Buff, &Len, &Rc); // Execute function

> SessType = Buff.qsst_sesstype; // Get session type
...
```

主机自动化场景

此处提供的场景示例提供了有关可使用 EHLLAPI 简化的活动的概念性信息。这些场景涉及您的 EHLLAPI 编程运算符可以在以下方面履行的职责：

- 主机系统操作，包括：
 - 搜索功能
 - 发送击键
- 分布式处理，包括：
 - 数据抽取
 - 文件传输
- 集成接口

场景 1. 搜索功能

典型的主机系统事务分为四个阶段：

1. 启动事务
2. 等待主机系统响应
3. 分析响应以查看它是否是预期的响应
4. 从响应中抽取和使用数据

编程运算符可以使用一系列 EHLLAPI 功能来模拟这些操作。在确定主机系统事务的正确起点后，编程运算符可以调用**搜索表示空间** (6) 功能，以确定显示屏上显示哪些关键字消息或提示消息。

接下来，编程运算符可以使用**发送键** (3) 功能将数据输入都主机系统会话中并输入主机系统事务。然后，编程运算符可以：

- 使用**等待** (4) 功能等待 X CLOCK、X `]` 或 X SYSTEM 条件结束（如果终端已锁定，则返回键盘已锁定状态）。
如果键盘被禁止，EHLLAPI 程序可以调用**复制 OIA** (13) 功能，以获取有关错误条件的更多信息。
- 使用**搜索表示空间** (6) 功能查找预期的关键字，以验证是否已收到正确的响应。
- 使用**将表示空间复制到字符串** (8) 功能（或多个数据访问功能中的任一个），以抽取所需的数据。

搜索表示空间 (6) 功能对于模拟终端操作员的另一项任务至关重要。某些主机系统在响应前不会在 X CLOCK、X `]` 或 X SYSTEM 方式下保持锁定状态；相反，它们会快速解锁键盘并允许运算符堆叠其他请求。在这种环境中，终端操作员依靠其他一些视觉提示来知道数据已返回（可能是屏幕标题或标签）。**搜索表示空间** (6) 功能允许 EHLLAPI 程序在等待时搜索表示空间。此外，在等待响应时，调用**暂停** (18) 功能允许其他 DOS 会话共享中央处理器资源。**暂停** (18) 功能有一个选项允许 EHLLAPI 程序等待主机系统更新事件发生。

如果在一段合理的超时时间后没有发生任何主机系统事件，EHLLAPI 程序可能会调用定制的错误消息，例如：

```
No Response From Host.  Retry?
```

在这种环境下，程序修订成为非常重要的考虑因素，因为即使显示消息发生微小的变化，也必须对编程运算符进行重新编程。

例如，如果终端操作员希望显示以下消息：

```
Enter Part Number:
```

作为提示，操作员可能能够正确响应生成以下消息的应用程序更改：

```
Enter Component Number:
```

但是，由于编程运算符正在查找文字关键字字符串，因此消息语法的细微变化（即使与大小写一样微不足道）可能会使程序执行预编程错误操作。

场景 2.发送击键

在设计将击键发送到主机系统的程序时，需要注意几个注意事项。在某些应用程序环境中，发出命令就像输入字符串并按 Enter 键一样简单。其他应用程序涉及更复杂的格式化屏幕，其中可以将数据输入到多个字段中的任何一个。在此环境中，您必须了解填写显示屏幕所需的击键。

Tab 键助记符 (@T; 请参阅 [常规 \(on page 19\)](#) 以获取完整的助记符列表) 可用于在字段之间跳过。使用**发送键 (3)** 功能向字段发送击键时，应注意字段长度和内容。如果完全填充字段且下一个属性字节为 autoskip，光标将移到下一个字段。如果随后发出了一个制表符，则会跳到另一个字段。

同样，如果击键未完全填充该字段，则可能会留下先前输入的数据。应使用擦除字段结束 (EOF) 命令清除此类残留数据。

场景 3.分布式处理

某些应用程序属于所谓的协作类别。这些应用程序提供单一最终用户界面，但它们的处理在两个或多个不同的物理位置执行。

EHLLAPI 应用程序可以通过拦截主机系统和终端用户之间的通信，来与主机系统应用程序进行交互。主机系统表示空间是用于拦截此类数据的工具。每当表示空间更新或操作员按下 AID 键时，本地应用程序都会请求收到通知。

接着，此工作站应用程序可以通过以下任一种方式与主机系统应用程序配合使用：

- 在字段或表示空间基础上，使用用于处理字段的复制功能（**将字符串复制到字段 (33)** 功能或**将字段复制到字符串 (34)** 功能），或者用于从表示空间复制和复制到表示空间的功能（例如，**将字符串复制到表示空间 (15)** 功能或**将表示空间复制到字符串 (8)** 功能）。
- 在击键基础上，使用**发送键 (3)** 功能。
- 在文件基础上，用于大数据块。您可以让应用程序使用 EHLLAPI 文件传输功能（使用**发送文件 (90)** 功能或**接收文件 (91)** 功能），来传输数据或功能（如装入模块），并在本地或远程对其进行处理。

场景 4.文件传输

在这种场景下，假设要自动执行文件传输：

- 可以首先使用前面搜索场景中讨论的过程登录主机系统会话。
 - EHLLAPI 程序可以调用文件传输功能**发送文件** (90) 和**接收文件** (91) 来传输数据，而不是使用其中一个复制功能（对于复制许多数据屏幕而言效率低下）。
 - 成功完成后：
 - 如果**发送文件** (90) 功能已完成执行，则 EHLLAPI 程序可以在注销之前使用复制功能或**发送键** (3) 功能提交批处理作业。
 - 如果**接收文件** (91) 功能完成执行，则 EHLLAPI 程序可以启动本地应用程序。
-

场景 5.自动化

一个应用程序可以为另一个应用程序提供所有击键，也可以将击键与键盘上的击键散布到目标位置。有时，要做到这一点，应用程序必须锁定针对目标应用程序或表示空间的其他击键输入源（使用**保留** (11) 功能），然后再将其解锁（使用**释放** (12) 功能）。

呈现给任何应用程序的击键来源取决于应用程序的设计。击键可能来自：

- 键盘
- 集成到源应用程序中的数据
- 通过 DOS 接口检索的辅助存储器
- Z and I Emulator for Windows 界面

在所有情况下，提供给目标应用程序的击键与普通操作员输入无法区分。

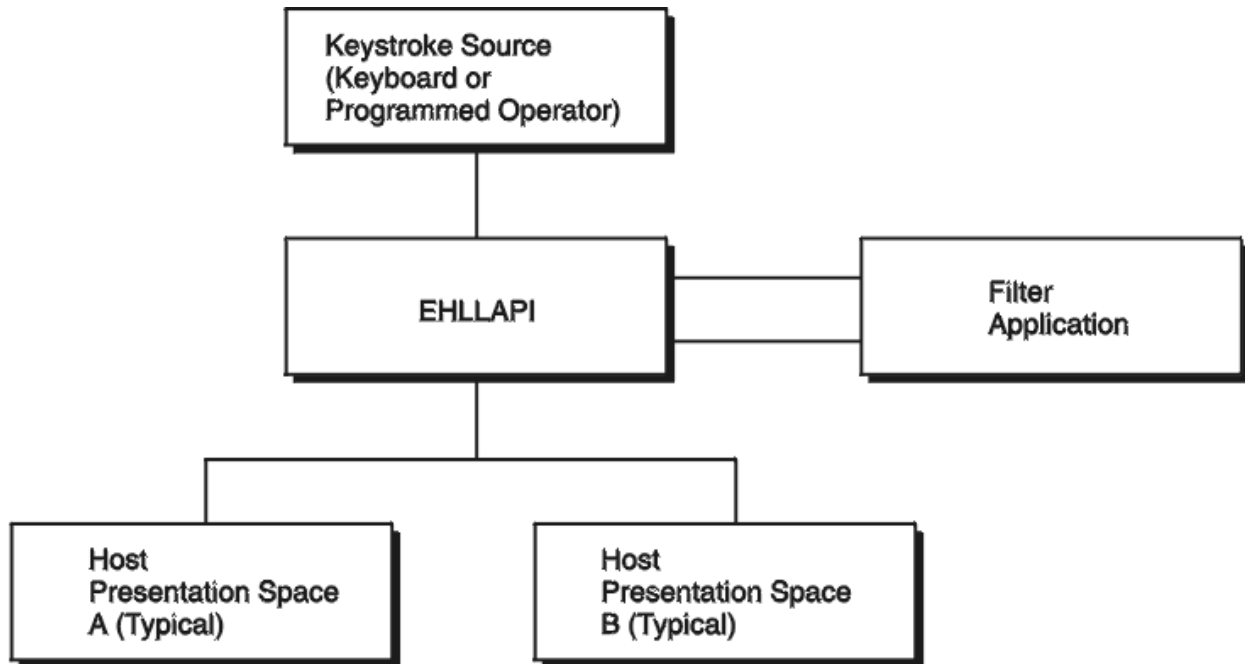
场景 6.击键过滤

用作过滤器的应用程序可以拦截来自 EHLLAPI（来自键盘或源应用程序）的击键，该击键针对另一个目标。那么，击键可以是：

- 已忽略（即已删除）
- 已重定向至另一个应用程序
- 已验证
- 已转换（例如，大写转换为小写）
- 已增强（通过键盘宏）

图 1: 击键流 (on page 27) 提供键盘增强环境中的击键流和对象的简化表示。

图 1. 击键流



场景 7. 键盘增强

此场景使用过滤来创建**增强器应用程序**。增强器应用程序是一种监控来自键盘的数据并以某种指定方式对其进行更改的应用程序。通常，这些应用程序使用称为**键盘宏**的指令，这些指令告诉他们要查找哪些击键，以及要进行哪些更改。更改可能涉及禁止击键（因此，在目标应用程序中似乎从未发送），将击键替换为另一个击键，或将单个击键替换为一系列击键。

要使用 EHLAPI 执行此操作，可以构造此场景：

1. EHLAPI 应用程序调用**连接表示空间** (1) 功能，以连接到要过滤击键的表示空间。
2. 接下来，EHLAPI 程序调用指定 L 选项的**启动击键拦截** (50) 功能。这会导致将所有击键路由到过滤应用程序。
3. 过滤应用程序现在可以定义一个循环，其中：
 - a. **获取键** (51) 功能会拦截发送到目标表示空间的所有击键。
 - b. 过滤应用程序会检查每个击键并执行键盘宏任务，例如：
 - 简化程序命令，以便将三击键或四击键命令压缩为单个击键
 - 定制命令，使其更易于记忆或与其他软件包保持一致
 - 为合同或常用信件创建**套用条款**
 - 为使用相同键执行不同功能的并发应用程序重新排列键盘
 例如，过滤应用程序可能会将组合键（如 Alt+Y）转换为命令，以便将光标移到表示空间第二行的第 35 列，并写入字符串“XYZ Tool Corporation, Dallas, Texas”。
 - c. 如果击键被拒绝，EHLAPI 程序会使用**拦截后状态** (52) 功能发出哔声。
4. EHLAPI 程序退出过滤循环后，**停止击键拦截** (53) 功能来结束过滤过程。

第 3 章. EHLLAPI 函数

本章详细介绍了每个单独的 Z and I Emulator for Windows EHLLAPI 函数，并说明了如何使用 EHLLAPI 程序采样程序。函数按名称的字母顺序排列。其中介绍了标准接口和增强接口的函数。



注： 在本章中，WinHLLAPI、IBM® 标准 32 位 HLLAPI 和 16 位 EHLLAPI 称为标准接口，而 IBM® 增强 32 位 EHLLAPI 称为增强接口。

页面布局约定

所有 EHLLAPI 函数调用都以相同的格式显示，以便可以快速检索所需的信息。格式为：

- 函数名称（函数编号）
 - 前提条件调用
 - 调用参数
 - 返回参数
 - 有关使用此功能的注释

前提条件调用

“前提条件调用” 列出在调用所讨论的函数之前必须进行的任何调用。

调用参数

“调用参数” 列出了在程序中必须定义的参数，以便调用所讨论的 EHLLAPI 函数，并解释了如何定义这些参数。如果某个参数从未被某个函数使用，则会列出不适用。如果参数可以被调用**设置会话参数** (9) 功能时定义的会话参数的某些值覆盖，则会命名此类会话参数。

返回参数

“返回参数” 列出调用讨论的 EHLLAPI 后程序必须收到的参数，并说明了如何解释这些参数。

有关使用此功能的注释

“有关使用此函数的说明” 列出了影响所讨论函数的任何会话选项。其中还提供有关使用此函数的技术信息，以及应用程序开发提示。

EHLLAPI 功能摘要

表 6: EHLLAPI 功能摘要 (on page 29) 是 EHLLAPI 功能摘要:

表 6. EHLLAPI 功能摘要

功能	3270	5250	VT
连接表示空间 (1) (on page 37)	是	是	是
断开表示空间 (2) (on page 69)	是	是	是
发送键 (3) (on page 116)	是	是	是
等待 (4) (on page 146)	是	是	是
复制表示空间 (5) (on page 53)	是	是	是
搜索表示空间 (6) (on page 113)	是	是	是
查询光标位置 (7) (on page 92)	是	是	是
将表示空间复制到字符串 (8) (on page 57)	是	是	是
设置会话参数 (9) (on page 125)	是	是	是
查询会话 (10) (on page 98)	是	是	是
保留 (11) (on page 110)	是	是	是
释放 (12) (on page 109)	是	是	是
复制 OIA (13) (on page 45)	是	是	是
查询字段属性 (14) (on page 93)	是	是	是
将字符串复制到表示空间 (15) (on page 63)	是	是	是
暂停 (18) (on page 85)	是	是	是
查询系统 (20) (on page 100)	是	是	是
重置系统 (21) (on page 111)	是	是	是
查询会话状态 (22) (on page 96)	是	是	是
启动主机通知 (23) (on page 137)	是	是	是
查询主机更新 (24) (on page 95)	是	是	是
停止主机通知 (25) (on page 144)	是	是	是
搜索字段 (30) (on page 112)	是	是	是
查找字段位置 (31) (on page 74)	是	是	是
查找字段长度 (32) (on page 72)	是	是	是
将字符串复制到字段 (33) (on page 62)	是	是	是
将字段复制到字符串 (34) (on page 41)	是	是	是
将表示空间复制到剪贴板 (35) (on page 64)	是	是	是
将剪贴板粘贴到表示空间 (36) (on page 66)	是	是	是
设置光标 (40) (on page 124)	是	是	是
启动关闭拦截 (41) (on page 133)	是	是	是
查询关闭拦截 (42) (on page 89)	是	是	是
停止关闭拦截 (43) (on page 143)	是	是	是
查询附加字段属性 DRB (on page 88)	编号	是	否

表 6. EHLLAPI 功能摘要 (续)

功能	3270	5250	VT
启动击键拦截 (50) (on page 139)	是	是	是
获取键 (51) (on page 76)	是	是	是
拦截后状态 (52) (on page 86)	是	是	是
停止击键拦截 (53) (on page 145)	是	是	是
锁定表示空间 API (60) (on page 81)	是	否	编号
锁定窗口服务 API (61) (on page 83)	是	否	编号
启动通信通知 (80) (on page 135)	是	是	是
查询通信事件 (81) (on page 91)	是	是	是
停止通信通知 (82) (on page 143)	是	是	是
发送文件 (90) (on page 161)	是	是	否
接收文件 (91) (on page 107)	是	是	否
取消文件传输 (92) (on page 31)	是	是	是
转换位置或转换 RowCol (99) (on page 39)	是	是	是
连接窗口服务 (101) (on page 38)	是	是	是
断开窗口服务 (102) (on page 69)	是	是	是
查询窗口坐标 (103) (on page 101)	是	是	是
窗口状态 (104) (on page 147)	是	是	是
更改交换机列表 LT 名称 (105) (on page 34)	是	是	是
更改 PS 窗口名称 (106) (on page 32)	是	是	是
开始播放宏 (110) (on page 142)	是	是	是
结构化字段的连接 (120) (on page 35)	是	否	编号
断开与结构化字段的连接 (121) (on page 67)	是	否	编号
查询通信缓冲区大小 (122) (on page 90)	是	否	编号
分配通信缓冲区 (123) (on page 30)	是	否	编号
释放通信缓冲区 (124) (on page 75)	是	否	编号
获取请求完成 (125) (on page 79)	是	否	编号
读取结构化字段 (126) (on page 103)	是	否	编号
写入结构化字段 (127) (on page 151)	是	否	编号

分配通信缓冲区 (123)

3270	5250	VT
是	否	编号

分配通信缓冲区功能会从操作系统获取缓冲区。必须在读取结构化字段 (126) 和写入结构化字段 (127) 功能上传递缓冲区地址。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 123	
数据字符串	请参阅下表	
Length	必须为 6	必须为 8
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1-2	1-4	32 位或 16 位缓冲区长度。(0 < size ≤ (64 KB-256 字节)=X'FF00')
3-6	5-8	分配的 32 位缓冲区地址 (返回)

返回参数

返回码	说明
0	分配通信缓冲区功能已成功。
2	指定参数时出错。
9	发生了系统错误。
11	资源不可用 (内存不可用) 。

有关使用此功能的注释

- EHLLAPI 从操作系统内存管理获取缓冲区，并将缓冲区地址放入返回参数字符串中。请求的缓冲区大小 (长度) 也会在参数字符串中传递。缓冲区大小的长度可以是 1 字节到 64 KB 减去 256 字节 (X'FF00' 字节)。
有关缓冲区大小的信息，请参阅“[查询通信缓冲区大小](#)”(122)。
- 使用此功能获得的缓冲区不能在不同进程之间共享。如果尝试执行此操作，应用程序将遇到不可预测的结果。
- EHLLAPI 应用程序必须使用[释放通信缓冲区](#) (124) 功能来释放分配的内存。
- 最多可为应用程序分配 10 个缓冲区。如果达到此限制，将返回资源不可用 (RC=11) 的返回码。
- [重置系统](#) (21) 功能会释放由此功能分配的缓冲区。

取消文件传输 (92)

3270	5250	VT
是	是	是

取消文件传输功能会导致立即返回指定会话的任何当前 EHLLAPI 启动的**发送文件**或**接收文件**。

前提条件调用

发送文件 (90) 或 **接收文件** (91)

调用参数

	增强型接口
函数编号	必须为 92
数据字符串	主机表示空间的 1 字符短名称。空白或 null 表示主机连接的表示空间的更新请求
Length	4 是隐含的
PS 位置	不适用

调用数据结构包含这些元素

字节	定义
1	1 字符表示空间短名称 (PSID)
2 - 4	保留

返回参数

返回码	定义
0	此功能已成功
1	指定的 PSID 不正确。
8	未为 PSID 调用之前的 启动通信通知 (80) 功能调用
9	遇到系统错误

有关使用此功能的注释

由于**发送文件** (90) 和**接收文件** (91) 均是分块调用，因此必须始终在不同的线程上发出此函数。

更改 PS 窗口名称 (106)

3270	5250	VT
是	是	是

更改 PS 窗口名称功能允许应用程序为表示空间窗口指定新名称，或将表示空间窗口重置为缺省名称。

前提条件调用

连接窗口服务 (101)

调用参数

	标准接口	增强型接口
函数编号	必须为 106	
数据字符串	请参阅下表	
Length	必须指定 (请参阅注释。)	必须为 68
PS 位置	不适用	



注：必须指定数据字符串长度 (对于 PC/3270，通常为 3-63；对于 PC400，通常为 4-63；对于增强接口，则为 68)。

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留
2	7	更改请求选项值，选择以下一项： <ul style="list-style-type: none"> • X'01'，表示更改表示空间窗口名称。 • X'02'，表示重置表示空间窗口名称。
3-63	6-66	从 1 (表示 PC/3270) 或 2 (表示 PC400) 到 61 字节的 ASCII 字符串，包括终止符字节。ASCII 字符串必须以 NULL 字符结束。此字符串必须至少包含一个非 NULL 字符，后跟一个 NULL 字符。
	67-68	保留

返回参数

返回码	说明
0	更改 PS 窗口名称功能已成功。
1	指定的主机表示空间短会话标识不正确，或者未连接主机表示空间。
2	指定参数时出错。
9	发生了系统错误。
12	会话已停止。

有关使用此功能的注释

字符串在找到的第一个 NULL 字符处结束。NULL 字符会覆盖指定的字符串长度。如果 NULL 字符不在指定长度的末尾，则指定长度的最后一个字节将被替换为 NULL 字符，并且数据字符串的其余部分将丢失。如果在指定长度之前找到 NULL 字符，则该字符串将在该点被截断，而数据字符串的其余部分将丢失。

如果应用程序在退出前未能重置表示空间名称，退出列表处理将重置该名称。

更改交换机列表 LT 名称 (105)

3270	5250	VT
是	是	是

更改交换机列表 LT 名称功能允许应用程序更改或重置选定逻辑终端 (LT) 的交换机列表。应用程序必须在调用时指定要插入到交换器列表中的名称。



注： 这是为了与 Communication Manager EHLAPI 兼容，其结果与**更改 PS 窗口名称 (106)** 功能相同。

前提条件调用

连接窗口服务 (101)

调用参数

	标准接口	增强型接口
函数编号	必须为 105	
数据字符串	请参阅下表	
Length	通常为 4-63	必须为 68
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强
1	1 字符表示空间短名称 (PSID)
	2-4 保留
2	7 更改请求选项；选择： <ul style="list-style-type: none"> • X'01' 表示更改交换机列表 LT 名称 • X'02' 表示重置交换机列表 LT 名称

字节		定义
3-63	6-66	2 到 61 字节的 ASCII 字符串，包括一个终止符字节。ASCII 字符串必须以 NULL 字符结束。此字符串必须至少包含一个非 NULL 字符，后跟一个 NULL 字符。
	67-68	保留

返回参数

返回码	说明
0	更改交换机列表 LT 名称功能已成功。
1	指定的主机表示空间短会话标识不正确，或者未连接主机表示空间。
2	指定参数时出错。
9	发生了系统错误。
12	会话已停止。

有关使用此功能的注释

字符串在找到的第一个 NULL 字符处结束。NULL 字符会覆盖指定的字符串长度。如果 NULL 字符不在指定长度的末尾，则指定长度的最后一个字节将被替换为 NULL 字符，并且数据字符串的其余部分将丢失。如果在指定长度之前找到 NULL 字符，则该字符串将在该点被截断，而数据字符串的其余部分将丢失。

如果应用程序在退出前未能重置交换机列表 LT 名称，退出列表处理将重置该名称。

结构化字段的连接 (120)

3270	5250	VT
是	否	编号

结构化字段的连接功能允许应用程序建立与仿真程序的连接，以便与主机应用程序交换结构化字段数据。工作站应用程序必须提供“查询应答”数据字段，并且必须在参数字符串中指向该字段。仿真器返回的目标/来源标识将被返回到应用程序。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 120	
数据字符串	请参阅下表	
Length	7 或 11	必须为 16

	标准接口	增强型接口
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留
2-5	5-8	查询应答数据缓冲区的地址
6-7	9-10	目标/来源唯一标识。(返回的 16 位字)
	11-12	保留
8-11	13-16	EHLAPI 会忽略这些位置中的数据。但是，如果迁移程序的数据位于这些位置，则不会导致错误。会接受此数据以提供与迁移应用程序的兼容性。

返回参数

返回码	说明
0	结构化字段的连接功能已成功。
1	指定的主机表示空间短会话标识无效，或未连接主机表示空间。
2	指定参数时出错。
9	发生了系统错误。
10	枚举程序不支持此功能。
32	应用程序已连接到此会话以进行通信（成功连接）。
39	一个 DDM 会话连接到此会话。

有关使用此功能的注释

1. EHLAPI 会扫描查询应答缓冲区，以获取目标/来源标识 (DOID) 自定义参数 (SDP)，以确定查询应答的 DOID 字段的内容。如果此值为 X'0000'，则仿真器将为应用程序分配 DOID，并且 EHLAPI 将使用分配的标识填写查询应答的 DOID 字段。如果应用程序在查询应答的 DOID 字段中指定的值是非零值，则仿真器会将指定的值分配为应用程序的 DOID，前提是以前未分配该标识。如果指定的 DOID 已在使用中，则返回码 2 将由 EHLAPI 返回。
2. 应用程序应在应用程序的专用内存中构建查询应答数据结构。有关 EHLAPI 支持的查询应答数据结构的详细格式和用法，请参阅 [受 EHLAPI 支持的查询应答数据结构 \(on page 197\)](#)、。
3. 只对查询应答数据执行粗略检查。仅检查结构的标识和长度是否有效。
4. 每个主机会话仅允许一个 DDM 基本类型连接。如果 DDM 连接支持目标来源标识 (DOID) 的自定义参数 (SDP)，则允许多个连接。
5. 如果收到返回码 RC=32 或 RC=39，则应用程序已连接到选定的会话，应谨慎使用该表示空间。可能会导致与文件传输和其他 EHLAPI 应用程序发生冲突。

连接表示空间 (1)

3270	5250	VT
是	是	是

连接表示空间功能会在 EHLLAPI 应用程序和主机表示空间之间建立连接。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 1	
数据字符串	主机表示空间的 1 字符短名称	
Length	1 是隐含的	必须为 4
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留

返回参数

连接表示空间功能会设置返回码，以指示尝试的状态，如果成功，则显示主机表示空间的状态。

返回码	说明
0	连接表示空间 功能已成功；主机表示空间已解锁并可供输入。
1	指定的主机表示空间标识不正确。指定的会话不存在或为逻辑打印机会话。此返回码还可能意味着未启用 EHLLAPI 的 API 设置。
4	已成功连接，但主机表示空间正忙。
5	已成功连接，但主机表示空间已锁定（输入被禁止）。
9	遇到系统错误。
11	此资源不可用。主机表示空间正由另一个系统功能在使用。

有关使用此功能的注释

1. **连接表示空间**功能受 `CONLOG/CONPHYS` 会话选项的影响。
2. EHLLAPI 应用程序不能同时连接到多个表示空间。需要**连接表示空间**功能作为前提条件的调用使用当前连接的表示空间。例如，如果应用程序按该顺序连接到表示空间 A、B 和 C，则应用程序必须再次连接到 B 或 A，才能发出功能。
3. 请求**连接表示空间**的每个线程都必须具有相应的**断开表示空间** (2)，或者其中一个线程必须发出**重置系统** (21)，这会影响到所有线程并断开所有剩余的连接。
4. 如果多个 EHLLAPI 应用程序支持共享（即，如果它们是为了协同工作而开发的，并且它们表现出可预测的行为），并且具有在**设置会话参数** (9) 功能中设置的兼容读/写访问权和关键字选项，则这些应用程序可以共享表示空间。有关更多信息，请参阅 [设置会话参数 \(9\) \(on page 125\)](#)。
5. 由于**连接表示空间**和**启动击键拦截** (50) 功能共享共同的子系统功能，因此应用程序发出的共享同一会话的其中任一功能的成功请求可能会影响其他应用程序对这两个功能的请求。例如，如果应用程序 A 成功地具有 `Write_Read` 访问权并将 `KEY$abcdefgh` 用作关键字的会话请求**连接表示空间**，则仅当两个应用程序都设置了兼容的读/写选项时，应用程序 B 才能为会话成功请求**连接表示空间**和**启动击键拦截**。
6. 无法连接到定义为逻辑打印机会话的会话。有关更多信息，请参阅 *Administrator's Guide and Reference*。

连接窗口服务 (101)

3270	5250	VT
是	是	是

连接窗口服务功能允许应用程序管理表示空间窗口。一次只能将一个 EHLLAPI 应用程序连接到窗口服务的表示空间。一个 EHLLAPI 应用程序可以同时连接到用于窗口服务的多个表示空间。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 101	
数据字符串	主机表示空间的 1 字符短会话标识	
Length	1 是隐含的	必须为 4
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强

字节		定义
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留

返回参数

返回码	说明
0	连接窗口服务 功能已成功。
1	指定的主机表示空间短会话标识错误，或者未连接会话窗口服务管理器。此返回码还可能意味着未启用 EHLLAPI 的 API 设置。
9	发生了系统错误。
10	枚举程序不支持此功能。
11	此资源不可用。主机表示空间正由另一个系统功能在使用。

有关使用此功能的注释

1. 一个 EHLLAPI 应用程序可以同时连接到多个表示空间窗口。应用程序可以在连接的表示空间窗口之间来回切换，而无需断开连接。例如，如果某个应用程序连接到表示空间窗口 A、B 和 C，则该应用程序可以同时访问 A、B 和 C，而其他应用程序则无法访问 A、B 或 C。
2. **连接窗口服务**功能足以完成此过程。但是，请求**连接窗口服务**的每个线程都必须具有相应的**断开窗口服务** (102)，或者其中一个线程必须发出**重置系统** (21)，这将影响所有线程并断开所有剩余的连接。

转换位置或转换 RowCol (99)

3270	5250	VT
是	是	是

转换位置或转换 RowCol 功能会将主机表示空间位置值转换为显示行和列坐标，或将显示行和列坐标转换为主机表示空间位置值。此功能不会更改游标位置。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 99	
数据字符串	主机表示空间短名称和 P (表示 转换位置 功能 (例如, AP 会转换会话 A 的表示空间位置) ; 或者主机表示空间短名称和 R (表示 转换 RowCol 功能) (例如, AR 会转换会话 A 的行和列坐标) 。	

	标准接口	增强型接口
Length	行（将 R 指定为数据字符串参数中的第二个字符时）。有效输入的下限为 1。有效输入的上限取决于主机表示空间的配置方式。请参阅 有关使用此功能的注释 (on page 41) 。	不适用（将 P 指定为数据字符串参数中的第二个字符时）。
PS 位置	列（将 R 指定为数据字符串参数中的第二个字符时）。有效输入的下限为 1。有效输入的上限范围为 24 到 43，具体取决于主机表示空间的配置方式。请参阅 有关使用此功能的注释 (on page 41) 。	主机表示空间位置（将 P 指定为数据字符串参数中的第二个字符时）。有效输入的下限为 1。有效输入的上限范围为 1920 到 3564，具体取决于主机表示空间的配置方式。请参阅 有关使用此功能的注释 (on page 41) 。

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留
2	7	转换选项 P 或 R
	6-8	保留

返回参数

此功能会返回一个长度和一个返回码。

长度：

对于**转换位置**功能（**P** 作为调用数据字符串中的第二个字符），将返回 1 到 43（对于 PC/3270）或 27（对于 PC400）之间的数字。此值是包含调用 PS 位置参数中包含的 PS 位置的行号。上限可以小于 43（对于 PC/3270）或 27（对于 PC400），具体取决于主机表示空间的配置方式。

对于**转换 RowCol** 功能（**R** 作为调用数据字符串中的第二个字符），值 0 表示行（调用长度参数）输入值中存在错误。

返回码：

转换位置或 RowCol 功能是第四个返回参数始终包含返回码的规则例外。对于此功能，在第四个参数中返回的值称为状态码。此状态码可以包含数据或返回码。应用程序必须提供此状态码的处理，以防止出现不可预测的结果或错误。

- 如果第四个参数的值为 0、9998 或 9999，则为返回码。
- 对于**转换位置**功能（**P** 作为调用数据字符串的第二个字符），1–132 范围内的值是包含在调用 PS 位置参数中传递的 PS 位置的列号。上限可以小于 132，具体取决于主机表示空间的配置方式。
- 对于**转换 RowCol** 功能（**R** 作为调用数据字符串的第二个字符），1–3564 范围内的值表示分别对应于调用长度和 PS 位置参数中传递的行和列值的主机表示空间位置。上限可以小于 3564，具体取决于主机表示空间的配置方式。

定义了以下状态码：

状态码	说明
0	这是不正确的 PS 位置或列。
>0	这是 PS 位置或列。
9998	指定的主机表示空间标识不正确或发生系统错误。
9999	数据字符串中的字符 2 不是 P 或 R。

有关使用此功能的注释

1. 要配置表示空间，请参阅 *Administrator's Guide and Reference*
2. 要了解表示空间中的行数和列数，请检查**查询会话状态** (22) 功能返回的数据字符串参数。请参阅 [查询会话状态 \(22\) \(on page 96\)](#)。

将字段复制到字符串 (34)

3270	5250	VT
是	是	是

将字段复制到字符串功能会将字符从主机连接的表示空间中的字段传输到字符串中。

将字段复制到字符串功能会将主机源表示空间中的字符转换为美国国家信息交换标准代码 (ASCII)。属性字节和其他字符未以 ASCII 表示，通常会转换为空格。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 34	
数据字符串	预先分配的目标数据字符串。当发出含“扩展属性字节 (EAB)”选项的 设置会话参数 (9) 功能时，数据字符串的长度必须至少是字段长度的两倍。	
Length	要复制的字节数（数据字符串的长度）。	

	标准接口	增强型接口
PS 位置	标识目标字段。这可以是目标字段中任何字节的 PS 位置。复制始终从字段的开头开始。	

返回参数

此函数会返回数据字符串、长度和返回码。

数据字符串:

包含来自主机表示空间中所标识字段的数据的字符串。返回的数据字符串中的第一个字节是主机表示空间中所标识字段的起始字节。返回的数据字符串中的字节数由以下两项中较小的项确定:

- 在调用长度参数中指定的字节数
- 主机表示空间中所标识字段中的字节数

长度:

返回的数据的长度。

返回码:

定义了以下代码:

返回码	说明
0	将字段复制到字符串功能已成功。
1	程序未连接到主机会话。
2	指定参数时出错。
6	要复制的数据和目标字段的大小不同。如果字符串长度小于复制的字段, 数据将被截断。
7	主机表示空间位置无效。
9	遇到系统错误。
24	未格式化的主机表示空间。

使用此功能的注意事项

1. 可以使用“[查找字段位置](#)”(31)和“[查找字段长度](#)”(32)函数查找字段位置和长度信息。**将字段复制到字符串**函数可用于受保护或未受保护的字段, 但仅限于字段格式的主机演示空间。
2. 当遇到下列情况之一时, 复制结束:
 - 当到达字段末尾时
 - 当超过目标字符串的长度时
3. 当使用[设置会话参数](#)(9)功能 EAB 选项时, 可以返回 EAB。EAB 与表示空间中的每个字符相关, 并在每个字符之前返回。
4. **将字段复制到字符串**函数受ATTRB / NOATTRB / NULLATTRB、EAB / NOEAB、XLATE / NOXLATE、DISPLAY / NODISPLAY、DISPLAY / NODISPLAY 会话选项。参考项目[5 on page 128](#);[13 on page 130](#)和[14 on page 131](#);[17 on page 131](#) ;和和了解更多信息。

如前所述，各种Copy（5、8和34）函数返回的属性受到Set SessionParameters（9）函数的影响。所涉及的设置会话参数的作用如下：

设置会话参数

对 COPY 功能的影响

NOEAB 和 NOEAD

不返回属性。仅文本从表示空间复制到用户缓冲区。

EAB 和 NOXLATE

属性按下表中的定义返回。

EAB 和 XLATE

返回用于显示空间显示的颜色。颜色可以重新映射；因此，当 XLATE 和 EAB 同时打开时，属性颜色不是COPY函数返回的颜色。

返回的字符属性在下表中定义。属性位位置采用IBM®格式，其中位0是字节中最左边的位。

3270个字符属性从主机返回到模拟器。下表适用于设置EAB和NOXLATE时。

位位置	含义
0-1	字符突出显示 00 = 正常 01 = 闪烁 10 = 反转视频 11 = 下划线
2-4	字符颜色（颜色重新映射可以覆盖此颜色定义。） 000 = 默认 001 = 蓝色 010 = 红色 011 = 粉色 100 = 绿色 101 = 绿松石 110 = 黄色 111 = 白色
5-6	字符属性 00 = 默认值
7	预订的

5250个字符属性从主机返回到模拟器。下表适用于设置EAB和NOXLATE时。

位位置	含义
0	倒车影像 0 = 正常图像 1 = 反转图像

位位置	含义
1	下划线 0 = 无下划线 1 = 下划线
2	眨 0 = 不闪烁 1 = 闪烁
3	列分隔符 0 = 无分隔符 1 = 分隔符
4-7	预订的

下表显示Z and I Emulator for Windows字符颜色属性。当设置 EAB 和 XLATE 时，下表适用。

位位置	含义
0-3	背景字符颜色 0000 = 黑色 0001 = 蓝色 0010=绿色 0011 = 青色 0100 = 红色 0101 = 洋红色 0110 = 棕色 (3270)、黄色 (5250) 0111=白色
4-7	前景字符颜色 0000 = 黑色 0001 = 蓝色 0010=绿色 0011 = 青色 0100 = 红色 0101 = 洋红色 0110 = 棕色 (3270)、黄色 (5250) 0111=白色 1000 = 灰色 1001 = 浅蓝色 1010=浅绿色 1011 = 浅青色 1100 = 浅红色 1101 = 浅洋红色 1110=黄色 1111 = 白色 (高强度)

对于PS/2®单色显示器，应用程序（工作站）会话中的字符显示为各种灰色阴影。这是为用户提供重新映射的颜色所必需的EHLLAPI应用程序会话，以便他们可以获得在主机应用程序演示空间中看到的内容。

5. 要使用此函数，请预先分配内存以接收返回的数据字符串参数。预分配此内存所需的语句根据编写应用程序所用的语言而有所不同。有关更多信息，请参阅[内存分配 on page 9](#)。



Note: 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真显示第 25 行。当显示来自主机的错误消息或当操作员选择 SysReq 键时，会发生这种情况。Z and I Emulator for Windows 状态栏显示第 25 行信息。通过 **EXTEND_PS** 选项，EHLLAPI 应用程序可以使用相同的接口 Communication Manager EHLLAPI 当这种情况发生时，有效的演示空间就会扩展。

复制 OIA (13)

3270	5250	VT
是	是	是

复制 OIA 功能从主机连接的表示空间返回当前的操作员信息区域 (OIA) 数据。

OIA 位于屏幕底部的分界线下，用于显示有关工作站和主机之间连接的会话状态信息。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 13	
数据字符串	预先分配的目标数据字符串	
Length	103	104
PS 位置	不适用	

返回参数

此函数会返回数据字符串和返回码。

数据字符串:

103 字节字符串表示 16 位，104 字节字符串表示 32 位。请参阅 [返回的 OIA 数据字符串的格式 \(on page 46\)](#) 以获取更多信息。

返回码:

定义了以下代码:

返回码	说明
0	返回 OIA 数据。目标表示空间已解锁。
1	程序未连接到主机会话。
2	指定字符串长度时出错。未返回 OIA 数据。
4	返回 OIA 数据。目标表示空间正忙。
5	返回 OIA 数据。目标表示空间已锁定。（输入被禁止）
9	遇到内部系统错误。未返回 OIA 数据。

有关使用此功能的注释

1. OIA 组由显示已连接会话状态的位组成。该组按表示的主机功能进行分类。（例如，第 8 组由显示会话中输入禁止的所有条件的位组成。）每个组的状态都按顺序排列，以便高位代表优先级较高的指示符。也就是说，位 7 优先于位 0。因此，如果一个组中有多个活动状态，则具有最高优先级的状态是该组中的活动状态。
2. 要使用此功能，请预先分配内存以接收返回的数据字符串参数。预先分配此内存所需的语句会因用于编写应用程序的语言而异。请参阅 [内存分配 \(on page 9\)](#) 以获取更多信息。

返回的 OIA 数据字符串的格式

OIA 数据字符串包含以下信息：


字节		定义
标准	增强	
1	1	OIA 格式字节。值为 1 (PC/3270)、9 (PC400) 或 5 (VT)。
2-81	2-81	主机代码点中的 OIA 图像。
82-103	82-103	OIA 组指示符含义。
	104	保留。

PC/3270 OIA 组指示符含义及其图像

OIA 图形组由一个 80 字节的 ASCII 字符串组成，不含主机代码点中包含 OIA 图形的属性字节。图 2: 主机表示空间字符 (on page 47) 显示十六进制代码（可在主机表示空间中找到），以及它们所表示的字符。返回的数据可以转换为 OIA 图形字符。请参阅 *Quick Beginnings* 以获取有关 OIA 指示符的信息。

要将返回的数据转换为 OIA 图形字符，请执行以下操作：

1. 将字节 2 到 81 中返回的数据打印到屏幕或打印机。
2. 使用适用于显示输出的设备的代码页图表，查找与每个字符对应的十六进制值。
3. 使用图 2: 主机表示空间字符 (on page 47)，查找与在步骤 2 中找到的每个十六进制值对应的 OIA 图形字符。

 注：组 8（字节 0）机器、通信和程序检查图像后跟与检查类型相关的三位数字。

后跟 X'20' 的短会话标识位于第 7 列。

所有组图像均由主框架交互 (MFI) 十六进制代码点表示。


 注：OIA 图像数据字符串位置减去 1 位置等于 OIA 列。

图 2. 主机表示空间字符

	0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	NUL	SP	0	&	à	ä	À	Ä	a	q	A	Q	↖	^	P	☎
x1	EM	=	1	—	è	ë	È	Ë	b	r	B	R	—		S	?
x2	FF	'	2	.	ì	ï	Ì	Ï	c	s	C	S	z	a	→	↔
x3	NL	"	3	,	ò	ö	Ò	Ö	d	t	D	T	_	°	↑	↔
x4	STP	/	4	:	ù	ü	Ù	Ü	e	u	E	U	☉	°	☎	4
x5	CR	\	5	+	ã	â	Ã	Â	f	v	F	V	☉	+	↓	—
x6			6	—	õ	ê	Õ	Ê	g	w	G	W	✕	⌋	⌋	—
x7			7	—	ÿ	î	Y	Î	h	x	H	X	■	⌋	⌋	▶
x8	>	?	8	°	à	ô	A	Ô	i	y	I	Y	←	⌋	μ	¿
x9	<	!	9		è	û	E	Û	j	z	J	Z	➡	⌋	2	☀
xA	[\$	β	^	é	á	E	Á	k	æ	K	Æ	☉	⌋	3	□
xB]	¢	§	~	ì	é	I	É	l	ø	L	Ø	☉	⌋	▶	☎
xC)	£	#	..	ò	í	O	Í	m	'a	M	À	☎	⌋	□	☎
xD	(¥	@	`	ù	ó	U	Ó	n	ç	N	Ç	B	⌋	↔	□
xE	}	Pts	%	'	Ü	Ú	Y	Ú	o	;	O	;	•	+	□	i
xF	{	☀	—	„	Ç	ñ	C	Ñ	p	*	P	*	■	×	☎	Not Supported

- 组 1 (偏移量 82) : 在线和屏幕所有权

位	意义
0-1	保留
2	SSCP-LU 会话拥有屏幕
3	LU-LU 会话拥有屏幕
4	在线且未拥有
5	子系统就绪
6-7	保留

- 组 2 (偏移量 83) : 字符选择

位	意义
0	保留
1	APL
3	字母数字
4-5	保留

- 组 3 (偏移量 84) : Shift 状态

位	意义
0	上 Shift
1	Numeric
2	CAPS
3-7	保留

- 组 4 (偏移量 85) : PSS 组 1

位	意义
0-7	保留

- 组 5 (偏移量 86) : 突出显示组 1

位	意义
0	操作员可选择
1	字段继承
2-7	保留

- 组 6 (偏移量 87) : 颜色组 1

位	意义
0	操作员可选择
1	字段继承
2-7	保留

- 组 7 (偏移量 88) : 插入

位	意义
0	插入方式

位	意义
1-7	保留

• 组 8 (偏移量 89-93) : 输入被禁止 (5 个字节)

◦ 字节 1 (偏移量 89)

位	意义
0	不可重置机器检查
1	保留
2	机器检查
3	通信检查
4	程序检查
5-7	保留

◦ 字节 2 (偏移量 90)

位	意义
0	设备正忙
1	终端等待
2	减号
3	Minus 函数
4	输入太多
5-7	保留

◦ 字节 3 (偏移量 91)

位	意义
0-2	保留
3	死键组合不正确, 键受限。
4	位置错误
5-7	保留

◦ 字节 4 (偏移量 92)

位	意义
0-1	保留
2	系统等待
3-7	保留

◦ 字节 5 (偏移量 93)

位	意义
0-7	保留

• 组 9 (偏移量 94) : PSS 组 2

位	意义
0-7	保留

• 组 10 (偏移量 95) : 突出显示组 2

位	意义
0-7	保留

- 组 11 (偏移量 96) : 颜色组 2

位	意义
0-7	保留

- 组 12 (偏移量 97) : 通信错误提醒

位	意义
0 - 6	通信错误
1-7	保留

- 组 13 (偏移量 98) : 打印机状态

位	意义
0-7	保留

- 组 14 (偏移量 99) : 图形

位	意义
0-7	保留

- 组 15 (偏移量 100) : 保留
- 组 16 (偏移量 101) : 自动键播放/录制状态

位	意义
0-7	保留

- 组 17 (偏移量 102) : 自动键退出/停止状态

位	意义
0-7	保留

- 组 18 (偏移量 103) : 扩展状态

位	意义
0-7	保留

PC400 OIA 组指示符含义及其图像

下表列出了 OIA 组的详细信息。

- 组 1 (偏移量 82) : 在线和屏幕所有权

位	意义	数据字符串的起始位置
0-2	保留	
3	可用的系统	1
6	保留	

位	意义	数据字符串的起始位置
5	子系统就绪	
6-7	保留	

- 组 2 (偏移量 83) : 字符选择

位	意义	数据字符串的起始位置
0-1	保留	
3	字母数字	
4-5	保留	

- 组 3 (偏移量 84) : Shift 状态

位	意义	数据字符串的起始位置
0	保留	
1	键盘 Shift	39
2	CAPS	
3-6	保留	

- 组 4 (偏移量 85) : PSS 组 1

位	意义	数据字符串的起始位置
0-7	保留	

- 组 5 (偏移量 86) : 突出显示组 1

位	意义	数据字符串的起始位置
0-7	保留	

- 组 6 (偏移量 87) : 颜色组 1

位	意义	数据字符串的起始位置
0-7	保留	

- 组 7 (偏移量 88) : 插入

位	意义	数据字符串的起始位置
0	插入方式	68
1-7	保留	

- 组 8 (偏移量 89-93) : 输入被禁止 (5 个字节)

- 字节 1 (偏移量 89)

位	意义	数据字符串的起始位置
0-7	保留	

- 字节 2 (偏移量 90)

位	意义	数据字符串的起始位置
0-7	保留	

- 字节 3 (偏移量 91)

位	意义	数据字符串的起始位置
0-4	保留	
5	操作员输入错误	64
6-7	保留	

- 字节 4 (偏移量 92)

位	意义	数据字符串的起始位置
0-1	保留	
2	系统等待	64
3-7	保留	

- 字节 5 (偏移量 93)

位	意义	数据字符串的起始位置
0-7	保留	

- 组 9 (偏移量 94) : PSS 组 2

位	意义	数据字符串的起始位置
0-7	保留	

- 组 10 (偏移量 95) : 突出显示组 2

位	意义	数据字符串的起始位置
0-7	保留	

- 组 11 (偏移量 96) : 颜色组 2

位	意义	数据字符串的起始位置
0-7	保留	

- 组 12 (偏移量 97) : 通信错误提醒

位	意义	数据字符串的起始位置
0	通信错误	
1-5	保留	
7	消息等待	3

- 组 13 (偏移量 98) : 打印机状态

位	意义	数据字符串的起始位置
0-7	保留	

- 组 14 (偏移量 99) : 图形

位	意义	数据字符串的起始位置
0-7	保留	

- 组 15 (偏移量 100) : 保留

- 组 16 (偏移量 101) : 自动键播放/录制状态

位	意义	数据字符串的起始位置
0-7	保留	

- 组 17 (偏移量 102) : 自动键退出/停止状态

位	意义	数据字符串的起始位置
0-7	保留	

- 组 18 (偏移量 103) : 扩展状态

位	意义	数据字符串的起始位置
0-7	保留	

VT 主机 OIA 组指示符含义及其图像

下表列出了 VT 主机 OIA 组的详细信息。

- 组 1 (偏移量 82) : 在线和屏幕所有权

位	意义
5	子系统就绪

- 组 2 (偏移量 83) : 字符选择

位	意义
0	上 Shift
2	CAPS

- 组 7 (偏移量 88) : 插入

位	意义
0	插入方式

OIA 行中的某些列针对 VT 显示的消息与针对 3270/5250 显示的消息不同。有关具体详细信息, 请参阅下表。

列	符号
1-7	VT220 7
	VT220 8
	VT100
	VT52
	VTANSI
9 - 12	LOCK
61 - 64	HOLD

复制表示空间 (5)

3270	5250	VT
是	是	是

复制表示空间功能会将主机连接的表示空间的内容复制到您在 EHLLAPI 应用程序中定义的数据字符串中。

复制表示空间功能会将主机源表示空间中的字符转换为 ASCII。未以 ASCII 表示的属性字节和其他字符通常将转换为空格。如果不希望将属性字节转换为空格，则可以使用**设置会话参数 (9)** 功能下的 ATTRB 选项覆盖此转换。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 5	
数据字符串	为目标字符串预先分配主机表示空间的大小。这可能会因主机表示空间的配置方式而异。当发出含 EAB 选项的 设置会话参数 (9) 功能时，数据字符串的长度必须至少是表示空间长度的两倍。	
Length	NA (表示主机表示空间的长度)。	
PS 位置	不适用。	

返回参数

此函数会返回数据字符串、长度和返回码。

数据字符串：

连接的主机表示空间的内容。

长度：

所复制数据的长度。

返回码：

定义了以下代码：

返回码	说明
0	主机表示空间内容已复制到应用程序。目标表示空间处于活动状态，键盘已解锁。
1	程序未连接到主机会话。
4	主机表示空间内容已复制。连接的主机表示空间正在等待主机响应。
5	主机表示空间已复制。键盘已锁定。
9	遇到系统错误。

使用此功能的注意事项

1. 当使用**设置会话参数**(9) 功能 EAB 选项时，可以返回 EAB。EAB 与表示空间中的每个字符相关，并在每个字符之前返回。
2. **复制演示空间**功能受以下会话选项影响：
 - ATTRB/NOATTRB/NULLATTRB
 - EAB/NOEAB
 - XLATE/NOXLATE
 - 空白/无空白
 - 显示/不显示
 - EXTEND_PS/NOEXTEND_PS

参考项目 [5 on page 128](#); [13 on page 130](#) , [14 on page 131](#) , [15 on page 131](#) 和 [17 on page 131](#) 和了解更多信息。

如果提供的目标数据字符串不够长，无法容纳所请求的数据，则可能会出现不可预测的结果。

如前所述，各种**Copy**（5、8 和 34）函数返回的属性受到**Set SessionParameters**（9）函数的影响。所涉及的设置会话参数的作用如下：

设置会话参数

对 COPY 功能的影响

NOEAB 和 NOEAD

不返回属性。仅文本从表示空间复制到用户缓冲区。

EAB 和 NOXLATE

属性按下表中的定义返回。

EAB 和 XLATE

返回用于显示空间显示的颜色。颜色可以重新映射；因此，当 XLATE 和 EAB 同时打开时，属性颜色不是**Copy**函数返回的颜色。

NOSO/空间SO/SO

当指定 NOSO 时，它的作用相当于 SPACESO。演示空间的大小没有改变。

返回的字符属性在下表中定义。属性位位置采用IBM®格式，其中位 0 是字节中最左边的位。

3270 个字符属性从主机返回到模拟器。下表适用于设置 EAB 和 NOXLATE 时。

位位置	含义
0-1	字符突出显示 <ul style="list-style-type: none"> 00 = 正常 01 = 闪烁 10 = 反转视频 11 = 下划线

位位置	含义
2-4	字符颜色 (颜色重新映射可以覆盖此颜色定义。) 000 = 默认 001=蓝色 010=红色 011=粉色 100 = 绿色 101 = 绿松石 110 = 黄色 111 = 白色
5-6	人物属性 00 = 默认值
7	预订的

5250 个字符属性从主机返回到模拟器。下表适用于设置 EAB 和 NOXLATE 时。

位位置	含义
0	倒车影像 0 = 正常图像 1 = 反转图像
1	下划线 0 = 无下划线 1 = 下划线
2	眨 0 = 不闪烁 1 = 闪烁
3	列分隔符 0 = 无分隔符 1 = 分隔符
4-7	预订的

下表显示 Z and I Emulator for Windows 字符颜色属性。当设置 EAB 和 XLATE 时，下表适用。

位位置	含义
0-3	背景字符颜色 0000 = 黑色 0001 = 蓝色

位位置	含义
	0010=绿色 0011 = 青色 0100 = 红色 0101 = 洋红色 0110 = 棕色 (3270)、黄色 (5250) 0111=白色
4-7	前景字符颜色 0000 = 黑色 0001 = 蓝色 0010=绿色 0011 = 青色 0100 = 红色 0101 = 洋红色 0110 = 棕色 (3270)、黄色 (5250) 0111=白色 1000 = 灰色 1001 = 浅蓝色 1010=浅绿色 1011 = 浅青色 1100 = 浅红色 1101 = 浅洋红色 1110=黄色 1111 = 白色 (高强度)

对于PS/2®单色显示器，应用程序（工作站）会话中的字符显示为各种灰色阴影。这是为用户提供重新映射的颜色所必需的EHLLAPI应用程序会话，以便他们可以获得在主机应用程序演示空间中看到的内容。

如果您只想复制主机演示空间的一部分，请使用“**将演示空间复制到字符串**” (8) 函数。

要使用此函数，请预先分配内存以接收返回的数据字符串参数。预分配此内存所需的语句根据编写应用程序所用的语言而有所不同。有关更多信息，请参阅[内存分配 on page 9](#)。



Note: 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真显示第 25 行。当显示来自主机的错误消息或当操作员选择 SysReq 键时，会发生这种情况。Z and I Emulator for Windows 在第 24 行或状态栏上显示第 25 行信息。要在状态栏上显示信息，必须配置状态栏。参考 *Quick Beginnings* 有关配置状态栏的信息。通过 **EXTEND_PS** 选项，EHLLAPI 应用程序可以使用相同的接口 Communication Manager EHLLAPI 当这种情况发生时，有效的演示空间就会扩展。

将表示空间复制到字符串 (8)

3270	5250	VT
是	是	是

将表示空间复制到字符串功能用于将全部或部分主机连接的表示空间复制到在 EHLLAPI 应用程序中定义的数据字符串中。

输入 PS 位置是进入主机表示空间的偏移量。此偏移基于布局，其中左上角（第 1 行/第 1 列）为位置 1，右下角为 3564，这是主机表示空间的最大屏幕大小。PS Position + (Length - 1) 的值不能超过主机表示空间的已配置大小。

将表示空间复制到字符串功能会将主机源表示空间中的字符转换为 ASCII。未以 ASCII 表示的属性字节和其他字符通常将转换为空格。如果不希望将属性字节转换为空格，则可以使用**设置会话参数 (9)** 功能下的 ATTRB 选项覆盖此转换。

前提条件调用

连接表示空间 (1)。

调用参数

	标准接口	增强型接口
函数编号	必须为 8	
数据字符串	为目标字符串预先分配主机表示空间的大小。当发出含 EAB 选项的 设置会话参数 (9) 功能时，数据字符串的长度必须至少是表示空间长度的两倍。	
Length	目标数据字符串的长度。	
PS 位置	在目标数据字符串中第一个字节的主机表示空间内的位置。	

返回参数

此函数会返回数据字符串和返回码。

数据字符串：

主机表示空间的内容。

返回码：

定义了以下代码：

返回码	说明
0	主机表示空间内容已复制到应用程序。目标表示空间处于活动状态，键盘已解锁。
1	程序未连接到主机会话。
2	指定字符串长度时出错，或 (长度 - 1) + PS 位置的总和大于连接的主机表示空间的大小。
4	主机表示空间内容已复制。主机表示空间正在等待主机响应。
5	主机表示空间已复制。键盘已锁定。
7	主机表示空间位置无效。

返回码	说明
9	遇到系统错误。

使用此功能的注意事项

1. 当使用**设置会话参数(9)**功能 EAB 选项时，可以返回 EAB。EAB 与表示空间中的每个字符相关，并在每个字符之后返回。
2. 将**演示空间复制到字符串**函数受以下选项影响：

- ATTRB/NOATTRB/NULLATTRB
- EAB/NOEAB
- XLATE/NOXLATE
- 空白/无空白
- 显示/不显示
- EXTEND_PS/NOEXTEND_PS

参考项目[5 on page 128](#);[13 on page 130](#)和[14 on page 131](#);[15 on page 131](#) ;[17 on page 131](#) ;和

如果提供的目标数据字符串不够大，无法容纳请求的字节数，则当到达目标数据字符串末尾时，复制成功结束 (RC=0、4 或 5) 。

如前所述，各种**Copy** (5、8 和 34) 函数返回的属性受到**Set SessionParameters** (9) 函数的影响。所涉及的设置会话参数的作用如下：

设置会话参数

对复印功能的影响

NOEAB 和 NOEAD

不返回属性。仅文本从表示空间复制到用户缓冲区。

EAB 和 NOXLATE

属性按下表中的定义返回。

EAB 和 XLATE

返回用于显示空间显示的颜色。颜色可以重新映射，因此当 XLATE 和 EAB 同时打开时，属性颜色不是**Copy**函数返回的颜色。

返回的字符属性在下表中定义。属性位位置采用 IBM 格式，其中位 0 是字节中最左边的位。

- 3270 个字符属性从主机返回到模拟器。下表适用于设置 EAB 和 NOXLATE 时。

位位置	含义
0-1	字符突出显示 00 = 正常 01 = 闪烁 10 = 反转视频 11 = 下划线
2-4	字符颜色 (颜色重新映射可以覆盖此颜色定义。)

位位置	含义
	000 = 默认 001 = 蓝色 010 = 红色 011 = 粉色 100 = 绿色 101 = 绿松石 110 = 黄色 111 = 白色
5-7	预订的

• 5250 个字符属性从主机返回到模拟器。下表适用于设置 EAB 和 NOXLATE 时。

位位置	含义
0	倒车影像 0 = 正常图像 1 = 反转图像
1	下划线 0 = 无下划线 1 = 下划线
2	眨 0 = 不闪烁 1 = 闪烁
3	列分隔符 0 = 无分隔符 1 = 分隔符
4-7	预订的

• VT 字符属性从主机返回到模拟器。下表适用于设置 EAB 和 NOXLATE 时。

位位置	含义
0-3	预订的
4	粗体 1 = 开 0 = 关闭
5	下划线 1 = 开 0 = 关闭
6	眨 1 = 开 0 = 关闭
7	撤销

位位置	含义
	0 = 开 1 = 关闭

• 下表显示Z and I Emulator for Windows字符颜色属性。当设置 EAB 和 XLATE 时，下表适用。

位位置	含义
0-3	背景字符颜色 0000 = 黑色 0001 = 蓝色 0010=绿色 0011 = 青色 0100 = 红色 0101 = 洋红色 0110 = 棕色 (3270)、黄色 (5250) 0111=白色
4-7	前景字符颜色 0000 = 黑色 0001 = 蓝色 0010=绿色 0011 = 青色 0100 = 红色 0101 = 洋红色 0110 = 棕色 (3270)、黄色 (5250) 0111=白色 1000 = 灰色 1001 = 浅蓝色 1010=浅绿色 1011 = 浅青色 1100 = 浅红色 1101 = 浅洋红色 1110=黄色 1111 = 白色 (高强度)

对于 PS/2 单色显示器，应用程序（工作站）会话中的字符显示为各种灰色阴影。这是为用户提供重新映射的颜色所必需的EHLLAPI应用程序会话，以便他们可以获得在主机应用程序演示空间中看到的内容。

- 要使用此函数，请预先分配内存以接收返回的数据字符串参数。预分配此内存所需的语句根据编写应用程序所用的语言而有所不同。有关更多信息，请参阅[内存分配 on page 9](#)。



Note: 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真显示第 25 行。当显示来自主机的错误消息或当操作员选择 SysReq 键时，会发生这种情况。Z and I Emulator for Windows在第 24 行或状态栏上显示第 25 行信息。要在状态栏上显示信息，必须配置状态栏。参



考 *Quick Beginnings* 有关配置状态栏的信息。通过 **EXTEND_PS** 选项，EHLAPI 应用程序可以使用相同的接口 **Communication Manager EHLAPI** 当这种情况发生时，有效的演示空间就会扩展。

将字符串复制到字段 (33)

3270	5250	VT
是	是	是

将字符串复制到字段 功能会将字符串传输到主机连接的表示空间中的指定字段。此功能只能在字段格式化主机表示空间中使用。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 33	
数据字符串	包含要传输到主机表示空间中目标字段的数据的字符串。	
Length	源数据字符串的长度（以字节为单位）。处于 EOT 方式时则覆盖。	
PS 位置	标识目标字段。这可以是目标字段中任何字节的 PS 位置。复制始终从字段的开头开始。	

返回参数

返回码	说明
0	将字符串复制到字段 功能已成功。
1	程序未连接到主机会话。
2	参数错误或复制长度为零。
5	目标字段受到保护或被禁止，或者向目标字段发送的数据不正确（例如字段属性）。
6	复制已完成，但数据被截断。
7	主机表示空间位置无效。
9	遇到系统错误。
24	未格式化的主机表示空间。

使用此功能的注意事项

1. **将字符串复制到字段** 功能受以下选项影响：
 - 斯特伦/街道
 - 环氧乙烷

- EAB/NOEAB
- XLATE/NOXLATE
- 普特阿布/诺普特阿布

参考项目1 on page 127和2 on page 127;13 on page 130和14 on page 131;18 on page 131 ;和和了解更多信息。

- 要传输的字符串由调用数据字符串参数指定。当遇到以下三个条件之一时，字符串结束：
 - 如果使用设置会话参数 (9) 函数选择了 EOT 模式，则在字符串中遇到文本**结束**(EOT) 分隔符时。（看[设置会话参数 \(9\) on page 125](#)）。
 - 如果不是 EOT 模式，则当达到长度中指定的数量时。
 - 当在场中遇到场结束时。



Note: 如果主机表示空间末尾的字段进行换行，则在到达表示空间末尾时会发生换行。

- 键盘助记符（请参阅[发送键\(3\) 功能](#)）无法使用“**复制字符串到字段**”功能发送。
- 要传输的数据的第一个字节始终放置在包含指定 PS 位置的字段的开头。



Note: 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真显示第 25 行。当显示来自主机的错误消息或当操作员选择 SysReq 键时，会发生这种情况。Z and I Emulator for Windows 在第 24 行或状态栏上显示第 25 行信息。要在状态栏上显示信息，必须配置状态栏。参考 *Quick Beginnings* 有关配置状态栏的信息。通过 **EXTEND_PS** 选项，EHLLAPI 应用程序可以使用相同的接口 Communication Manager EHLLAPI 当这种情况发生时，有效的演示空间就会扩展。

将字符串复制到表示空间 (15)

3270	5250	VT
是	是	是

将字符串复制到表示空间功能会将 ASCII 数据字符串直接复制到 PS 位置调用参数指定的位置处的主机表示空间中。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 15。	
数据字符串	要复制到主机表示空间的 ASCII 数据字符串。	
Length	源数据字符串的长度（以字节为单位）。处于 EOT 方式时则覆盖。	
PS 位置	在主机表示空间中要开始复制的位置，值介于 1 和主机表示空间的配置大小之间。	

返回参数

返回码	说明
0	将字符串复制到表示空间功能已成功。
1	程序未连接到主机会话。
2	参数错误或复制长度为零。
5	目标表示空间受到保护或被禁止，或者向目标表示空间发送的数据不正确（例如字段属性字节）。
6	复制已完成，但数据被截断。
7	主机表示空间位置无效。
9	遇到系统错误。

使用此功能的注意事项

1. 将字符串复制到演示空间功能受以下选项影响：

- 斯特伦/街道
- 环氧乙烷
- EAB/NOEAB
- XLATE/NOXLATE
- 普特阿布/诺普特阿布
- EXTEND_PS/NOEXTEND_PS

参考项目1 on page 127和2 on page 127;13 on page 130和14 on page 131;18 on page 131 ;和和了解更多信息。

2. 键盘助记符（请参阅发送键(3) 功能）无法使用“将字符串复制到演示空间”功能发送。
3. 如果使用设置会话参数(9) 函数选择了 EOT 模式，则当在字符串中遇到文本结束 (EOT) 分隔符时，字符串结束。（看设置会话参数 (9) on page 125）。
4. 尽管发送键(3) 功能可实现相同的目的，但此功能会以提示进行响应并更快地输入命令。由于发送键 (3) 功能模拟终端操作员从键盘输入数据，因此对于处理大量数据的应用程序来说，其处理速度很慢。此功能为主机提供了更快的输入路径。
5. 原始数据（复制的字符串）不能超过呈现空间的大小。
6. 对于某些主机应用程序，此函数调用可能会导致光标移动到意外位置。SendKey 函数可能是比此函数更好的填充字段的选择。



Note: 这只发生在VT与 ASCII 主机的会话或连接。

将表示空间复制到剪贴板 (35)

3270	5250	VT
是	是	是

将表示空间复制到剪贴板功能用于将全部或部分主机连接的表示空间复制到剪贴板。输入 PS 位置是进入主机表示空间的偏移量。此偏移基于布局，其中左上角（第 1 行/第 1 列）为位置 1，右下角为 3564，这是主机表示空间的最大屏幕大小。PS 位置 + (长度 - 1) 的值不能超过主机表示空间的已配置大小。

将表示空间复制到剪贴板会将主机源表示空间中的字符转换为 ASCII。未以 ASCII 表示的属性字节和其他字符通常将转换为空格。如果不希望将属性字节转换为空格，则可以使用**设置会话参数 (9)** 功能下的 ATTRB 选项覆盖此转换。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 35。	
数据字符串	为目标字符串预先分配主机表示空间的大小。当发出含 EAB 选项的 设置会话参数 (9) 功能时，数据字符串的长度必须至少是表示空间长度的两倍。	
Length	目标数据字符串的长度	
PS 位置	在目标数据字符串中第一个字节的主机表示空间内的位置。	

返回参数

返回码	说明
0	主机表示空间内容已复制到剪贴板。目标表示空间处于活动状态，键盘已解锁。
1	程序未连接到主机会话。
2	指定字符串长度时出错，或 (长度 - 1) + PS 位置的总和大于连接的主机表示空间的大小。
4	主机表示空间内容已复制到剪贴板。主机表示空间正在等待主机响应。
5	主机表示空间已复制到剪贴板。键盘已锁定。
7	主机表示空间位置无效。
9	遇到系统错误。

有关使用此功能的注释

- 使用**设置会话参数 (9)** 功能 EAB 选项时，可以返回 EAB。EAB 与表示空间中的每个字符相关，并在每个字符之后返回。
- 将表示空间复制到剪贴板**功能受以下选项的影响：
 - ATTRB/NOATTRB/NULLATTRB
 - EAB/NOEAB
 - XLATE/NOXLATE
 - BLANK/NOBLANK
 - DISPLAY/NODISPLAY

- PUTEAB/NOPUTEAB
- EXTEND_PS/NOEXTEND_PS

3. 数据字符串缓冲区用于处理从表示空间检索的数据并复制到剪贴板。如果提供的数据字符串缓冲区不足以容纳请求的字节数，则当到达数据字符串缓冲区的末尾时，复制将成功结束（RC=0、4 或 5）。如前所述，各种复制（5、8 和 34）功能返回的属性受**设置会话参数 (9)**功能的影响。所涉及的设置会话参数具有以下效果：

设置会话参数

对复制功能的影响

NOEAB 和 NOEAD

不返回属性。仅会将文本从表示空间复制到剪贴板。

EAB 和 NOXLATE

将按照下表中的定义返回属性。

EAB 和 XLATE

返回用于表示空间显示的颜色。可以重新映射颜色，因此属性颜色不是 XLATE 和 EAB 同时打开时 **Copy** 函数返回的颜色。

将剪贴板粘贴到表示空间 (36)

3270	5250	VT
是	是	是

将剪贴板粘贴到表示空间功能会将 ASCII 数据字符串直接粘贴到 PS 位置调用参数指定的位置处的主机表示空间中。

前提条件调用

调用参数

	标准接口	增强型接口
函数编号	必须为 36。	
数据字符串	字符串缓冲区，用于存放剪贴板中的数据，以将其粘贴到主机表示空间中。	
Length	长度，以要粘贴的字节数表示	
PS 位置	在主机表示空间中要开始复制的位置，值介于 1 和主机表示空间的配置大小之间。	

返回参数

返回码	说明
0	将剪贴板粘贴到表示空间 功能已成功。

返回码	说明
1	程序未连接到主机会话。
2	参数错误或复制长度为零
5	目标表示空间受到保护或被禁止，或者向目标表示空间发送的数据不正确（例如字段属性字节）。
6	复制已完成，但数据被截断。
7	复制已完成，但数据被截断。
9	遇到系统错误。

使用此功能的注意事项


1. 将剪贴板粘贴到演示空间功能受以下选项影响：

- 斯特伦/街道
- EAB/NOEAB
- 环氧乙烷
- XLATE/NOXLATE
- 普特阿布/诺普特阿布
- EXTEND_PS/NOEXTEND_PS

2. 如果使用**设置会话参数 (9)**函数选择了 EOT 模式，则当在字符串中遇到文本结束 (EOT) 分隔符时，字符串结束。（请参阅第 147 页上的“设置会话参数 (9)”）。

3. 原始数据（复制的字符串）不能超过呈现空间的大小。

字符串	含义	单字节字符字段	
X'000C'	(NULL)(FF) X'00' X'0C'	(SB NULL)(SB FF) X'00' X'0C'	
X'0E000C0F'	(SO)(DB FF)(SI) X'0E' X'000C' X'0F'	-S 错误	

 **Note:** SB 表示单字节字符。

注意： 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真显示第 25 行。当显示来自主机的错误消息或当操作员选择 SysReq 键时，会发生这种情况。Z and I Emulator for Windows 始终在第 24 行显示相同的信息。通过 **EXTEND_PS** 选项，EHLLAPI 应用程序可以使用与 Communication Manager EHLLAPI 相同的接口，并且在发生这种情况时扩展有效的表示空间。

断开与结构化字段的连接 (121)

3270	5250	VT
是	否	编号

断开与结构化字段的连接功能会断开仿真程序与 EHLLAPI 应用程序之间的连接。在退出系统之前，EHLLAPI 应用程序必须断开与仿真程序的连接。如果已发出以前的**结构化字段的连接**，EHLLAPI 应用程序应发出此功能请求。

重置系统 (21) 功能还将断开任何未完成的 SF 连接。

前提条件调用

结构化字段的连接 (120)

调用参数

	标准接口	增强型接口
函数编号	必须为 121	
数据字符串	请参阅下表	
Length	必须为 3	必须为 8
PS 位置	不适用	

数据字符串内容

字节	定义	
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留。
2-3	5-6	结构化字段的连接 (120) 功能返回的目标/来源唯一标识。
	7-8	保留。

返回参数

返回码	说明
0	断开与结构化字段的连接 功能已成功。
1	指定的主机表示空间短会话标识无效或未连接。
2	指定参数时出错。
9	发生了系统错误。
40	已断开连接且异步请求暂挂。

有关使用此功能的注释

- 调用**断开与结构化字段的连接**功能时，如果应用程序发出**获取请求完成** (125) 功能调用，则将返回任何未完成的异步**读取结构化字段** (126) 或**写入结构化字段** (127) 功能请求。发出断开连接调用后清除时，使用此功能的异步形式。
- 重置系统** (21) 功能还将释放所有未完成的异步请求（应用程序未使用**获取请求完成** (125) 功能检索的请求）。

断开表示空间 (2)

3270	5250	VT
是	是	是

断开表示空间功能可断开 EHLLAPI 应用程序与主机表示空间之间的连接。此外，如果使用**保留** (11) 功能保留主机表示空间，则在执行**断开表示空间**功能时将释放该空间。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 2	
数据字符串	不适用	
Length	不适用	
PS 位置	不适用	

返回参数

返回码	说明
0	断开表示空间 功能已成功。
1	您的程序当前未连接到主机表示空间。
9	遇到系统错误。

有关使用此功能的注释

1. 在调用**断开表示空间**功能后，与主机连接的显示空间交互的功能不再有效（例如，**发送键** (3)、**等待** (4)、**保留** (11) 和**释放** (12) 功能）。
2. 退出之前，EHLLAPI 应用程序应断开与主机表示空间的连接。
3. **断开表示空间**功能不会将会话参数重置为缺省值。您的 EHLLAPI 应用程序必须调用**重置系统** (21) 功能才能完成此操作。

断开窗口服务 (102)

3270	5250	VT
是	是	是

断开窗口服务功能会断开 EHLAPI 程序与指定主机表示空间窗口之间的窗口服务连接。

前提条件调用

连接窗口服务 (101)

调用参数

	标准接口	增强型接口
函数编号	必须为 102	
数据字符串	请参阅下表	
Length	1	6
PS 位置	不适用	

数据字符串内容

字节	定义
标准	增强
1	1 字符表示空间短名称 (PSID)
	2-4 保留

返回参数

返回码	说明
0	断开窗口服务 功能已成功。
1	您的程序未连接窗口服务。
9	发生了系统错误。

有关使用此功能的注释

调用**断开窗口服务**功能后，应用程序将不再管理表示空间窗口。

在退出应用程序之前，应该为所有已针对 Presentation Manager® 服务连接的表示空间请求**断开窗口服务**功能。如果应用程序退出时有未完成的窗口服务连接，子系统将取消未完成的连接。

编辑键截取

该功能部件允许您截取所有的击键以及编辑键，并将它们发送到 Windows 32 位环境中的会话。

前提条件

1. 映射定制键盘窗口中的编辑功能（例如，Ctrl+C 用于编辑复制功能）。
2. 用调用参数数据字符串值集来调用“启动击键截取”（50）EHLLAPI 功能。这些值如下所示：

字节位 置	目录
1	以下某个值： <ul style="list-style-type: none"> • 特定的主机表示空间短名称（PSID） • 空，表示主机连接的主机表示空间请求
2 至 4	保留
5	选项代码字符： <ul style="list-style-type: none"> • D，仅用于 AID 击键 • L，用于所有击键 • E，用于所有击键和编辑键 • M，用于通知的异步消息传递方式（仅 Windows）。如果指定了 M，那么代码字符 D、L 或者 E 必须在位置 13
6 至 8	保留
9 至 12	如果在位置 5 中指定了 M，那么是用于接收消息的窗口句柄。消息是 RegisterWindowMessage (PCSHLL) 的非零返回值。
13	如果在位置 5 中指定了 M，请使用下列某个值： <ul style="list-style-type: none"> • D，仅用于 AID 击键 • L，用于所有击键 • E，用于所有击键和编辑键
14 至 16	保留

3. 要获得截取的编辑键，可使用获取键 (51) EHLLAPI 功能。编辑键的数据字符串中返回的键助记符在第 5 个字节位置上将是 M（击键类型助记符）。接下来的 4 个字节将具有下列编辑键助记符之一，这些助记符基于截取的编辑键：

键助记符	截取的键
@W@C	编辑复制
@W@D	编辑清除
@W@E	编辑复制附加
@W@L	编辑复制链接
@W@N	编辑粘贴下一个
@W@V	编辑粘贴
@W@X	编辑剪切
@W@Z	编辑撤消

4. 要将编辑键发送到会话，可使用发送键 (3) EHLLAPI 功能。作为调用参数传递的数据字符串可以指定下列编辑键助记符：

键助记符	发送的键
@W@C	编辑复制
@W@D	编辑清除
@W@E	编辑复制附加
@W@L	编辑复制链接
@W@N	编辑粘贴下一个
@W@V	编辑粘贴
@W@X	编辑剪切
@W@Z	编辑撤消



注:

1. 您不必调用获取键 (51) EHLAPI 功能以使用发送键 (3) 功能。对于处理编辑键的获取键 (51) 和发送键 (3) 功能, 必须先调用启动击键拦截 (50) 并将第 5 个字节位置设为 **E**。如果第 5 个字节包含 **M**, 那么位置 13 必须包含 **E**。
2. 启动击键截取 (50)、获取键 (51) 和发送键 (3) 功能的预期返回值未更改。
3. 应该遵循现有文档中的任何先决条件以及在此说明的先决条件。

查找字段长度 (32)

3270	5250	VT
是	是	是

查找字段长度功能会返回连接的表示空间中目标字段的长度。此功能可用于查找受保护或未受保护的字段, 但仅限于字段格式化主机表示空间。


此功能会返回使用调用 PS 位置参数标识的字段中包含的字符数。这包括从目标字段开始到下一个属性字节前的字符的所有字符。

前提条件调用

连接表示空间 (1)


调用参数

	标准接口	增强型接口
函数编号	必须为 32	
数据字符串	请参阅下表	
Length	不适用	不适用
PS 位置	请参阅注释	

 **注：PS 位置：**标识主机表示空间内要开始**查找**的字段。它可以是字段中任何字节的 PS 位置，您希望在该字段中开始**查找**。

调用 2 字符数据字符串可以包含：

Code	说明
b ^D b ^D 或 T ^D b ^D	字段
P ^D b ^D	上一个字段，受保护或未受保护。
N ^D b ^D	下一个字段，受保护或未受保护
NP	下一个受保护字段
NU	下一个未受保护字段
PP	上一个受保护字段
PU	上一个未受保护字段

 **注：**The b^D 符号表示必填空格。

返回参数

此功能会返回一个长度和一个返回码。

长度：

以下长度有效：

Length	说明
= 0	返回码 = 28 时，字段长度为 0。返回码 = 24 时，主机表示空间未设置字段格式。
> 0	主机表示空间中的必填字段长度。

返回码：

定义了以下代码：

返回码	说明
0	查找字段长度 功能已成功。
1	程序未连接到主机会话。
2	遇到参数错误。
7	主机表示空间位置无效。
9	遇到系统错误。
24	未找到此类字段。
28	字段长度为 0 字节。

有关使用此功能的注释

b^Db^D 或 T^Db^D 用作调用数据字符串，如果找到的字段与从中启动**查找**的字段相同，则返回 24 的返回码。

查找字段位置(31)

3270	5250	VT
是	是	是


查找字段位置功能会返回目标字段在主机连接的表示空间中的起始位置。此功能可用于查找受保护或未受保护的字段，但仅限于字段格式化主机表示空间。

前提条件调用

连接表示空间 (1)


调用参数

	标准接口	增强型接口
函数编号	必须为 31	
数据字符串	请参阅下表	
Length	不适用	不适用
PS 位置	请参阅注释	

 **注：PS 位置：**标识主机表示空间内要开始**查找**的字段。它可以是字段中任何字节的 PS 位置，您希望在该字段中开始**查找**。

调用 2 字符数据字符串可以包含：

Code	说明
b ^d b ^d 或 T ^d b ^d	字段
P ^d	上一个字段，受保护或未受保护
N ^d	下一个字段，受保护或未受保护
NP	下一个受保护字段
NU	下一个未受保护字段
PP	上一个受保护字段
PU	上一个未受保护字段

 **注：**The b^d 符号表示必填空格。

返回参数

此功能会返回一个长度和一个返回码。

长度:

以下长度有效:

Length	说明
= 0	返回码 = 28 时, 字段长度为 0。返回码 = 24 时, 主机表示空间未设置字段格式。
> 0	请求的字段相对于主机表示空间原点的相对位置。此位置定义为属性字节后面的第一个位置。

返回码:

定义了以下代码:

返回码	说明
0	查找字段位置功能已成功。
1	程序未连接到主机会话。
2	遇到参数错误。
7	主机表示空间位置无效。
9	遇到系统错误。
24	未找到此类字段。
28	字段长度为 0 字节。

有关使用此功能的注释

bD 或 **TbD** 用作调用数据字符串, 如果找到的字段与从中启动**查找**的字段相同, 则返回 24 的返回码。

释放通信缓冲区 (124)

3270	5250	VT
是	否	编号

释放通信缓冲区功能会向管理内存返回应用程序不再需要的缓冲区。应用程序应在退出系统之前释放缓冲区。

前提条件调用

分配通信缓冲区 (123)

调用参数

	标准接口	增强型接口
函数编号	必须为 124	
数据字符串	请参阅下表	
Length	必须为 6	必须为 8

	标准接口	增强型接口
PS 位置	不适用	

数据字符串内容

字节		定义
标准	增强	
1-2	1-4	必须为 0
3-6	5-8	缓冲区的地址

返回参数

返回码	说明
0	释放通信缓冲区功能已成功。
2	指定参数时出错。
9	发生了系统错误。
41	正在使用此缓冲区。

有关使用此功能的注释

1. 如果应用程序尝试释放使用中的缓冲区，则释放请求将被拒绝，并返回 41 的返回码。
2. 对于使用**分配通信缓冲区** (123) 功能分配的所有通信缓冲区，应用程序在退出之前应请求**释放通信缓冲区** (124) 功能。
3. **重置系统** (21) 功能将释放由 **分配通信缓冲区** (123) 功能分配的缓冲区。

获取键 (51)

3270	5250	VT
是	是	是

获取键功能让 EHLLAPI 应用程序可从**启动击键拦截** (50) 功能指定的会话中检索击键，并处理、接受或拒绝该击键。通过将此功能放置在循环中，可以使用它来拦截字符串。

前提条件调用

启动击键拦截 (50)

调用参数

	标准接口	增强型接口
函数编号	必须为 51	

	标准接口	增强型接口
数据字符串	请参阅下表	
Length	10	12
PS 位置	不适用	

数据字符串内容

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 1 字符表示空间短名称 (PSID) • 表示主机连接的表示的函数调用的空格或 null
	2-4	保留
2-8	5-11	为所请求数据的符号表示保留空间的空白
	12	保留

返回参数

此函数会返回数据字符串和返回码。

数据字符串：

请参阅下表：

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 1 字符表示空间短名称 (PSID) • 表示主机连接的表示的函数调用的空格或 null
	2-4	保留
2	7	选项代码字符，以下一个字符： <ul style="list-style-type: none"> • A (表示返回的 ASCII) • M (表示击键助记符) • S (表示特殊助记符)
	6-11	这 6 个字节的预先分配缓冲区空间在内部用于击键排队和取消排队。可能的组合包括：

字节	定义
	<ul style="list-style-type: none"> • 字节 3 包含 ASCII 字符，字节 4 包含 'X'00' • 字节 3 包含转义字符 (Ⓜ 或使用功能 9 的 ESC=c 选项指定的另一个字符)，字节 4 包含功能的 1 字节缩写。(请参阅ASCII 助记符 (on page 18)) • 如果返回的 ASCII 助记符长于 2 个字节 (例如，如果 ASCII 助记符表示 Attn ⓂAⓂQ, 则字节 5 包含 Ⓜ, 而字节 6 包含 Q), 则字节 5 至 8 可能类似于字节 3 和 4。如果未使用，则将字节 5 至 8 设置为零 ('X'00')。

为了澄清，下面提供了所返回数据字符串的一些示例：



注：Ⓜ 符号是缺省转义字符。使用**设置会话参数 (9)** 功能的 ESC=c 选项，可以将转义字符的值设置为以 ASCII 表示的任何击键。如果使用此选项将转义字符更改为其他字符，则以下示例中的符号 Ⓜ 将被替换为其他字符。

16 位接口

EAt

E 是表示空间短名称。击键作为 ASCII (A) 返回，返回的键是小写字母 t。(字节 4-8 = 'X'00')。

EM@2

E 是表示空间短名称。击键作为助记符返回，返回的键是 PF2 (字节 5-8 = 'X'00')。

32 位接口

EbD_bD_bD_At

E 是表示空间短名称。击键作为 ASCII (A) 返回，返回的键是小写字母 t。(字节 7-11 = 'X'00')。

EbD_bD_bD_M@2

E 是表示空间短名称。击键作为助记符返回，返回的键是 PF2 (字节 8-11 = 'X'00')。

返回码：

以下代码有效：

返回码	说明
0	获取键功能已成功。
1	指定的表示空间不正确。
5	您在 启动击键拦截 (50) 功能下指定了仅 AID 选项，当 EHLLAPI 尝试将不正确的键写入表示空间时，非 AID 键将被此会话类型禁止。
8	未为此表示空间调用先前的 开始击键拦截 (50) 功能。
9	遇到系统错误。
20	输入了未定义的组合键。

返回码	说明
25	请求的击键在输入队列上不可用。
31	击键队列溢出，击键丢失。

有关使用此功能的注释

1. 如果**获取键**功能的返回码为 31，则：

- 增加**启动击键拦截** (50) 功能的调用长度参数的值，或
- 更频繁地执行**获取键**功能。

拦截的击键在缓冲区中占用 3 个字节。下一个拦截的击键将放置在相邻的三个字节中。当**获取键**功能检索击键（先进先出，FIFO）时，此功能占用的三个字节将可用于另一个击键。通过增加缓冲区的大小或从缓冲区中检索击键的速率，可以消除缓冲区溢出。

对于 PC/3270，消除返回码 31 的另一种方法是在恢复方式下操作 PC/3270 仿真器。

2. 您可以使用**发送键** (3) 功能将原始击键和 EHLLAPI 应用程序可能需要的任何其他击键传递到主机连接的表示空间。
3. 击键异步到达，并在 EHLLAPI 应用程序中使用**启动击键拦截** (50) 功能提供的击键队列中排队。
4. **获取键**功能的行为与读取类似。当击键可用时，会将其读取到在应用程序中提供的数据区域。
5. 对于会话的字段支持，应用程序可能只对 AID 键感兴趣，例如 Enter 键。如果是这样，则应将**启动击键拦截** (50) 功能选项代码设置为 D（意味着仅适用于 AID 键）。
6. 要使用此功能，请预先分配内存以接收返回的数据字符串参数。预先分配此内存所需的语句会因用于编写应用程序的语言而异。请参阅 [内存分配 \(on page 9\)](#) 以获取更多信息。

获取请求完成 (125)

3270	5250	VT
是	否	编号

获取请求完成功能允许应用程序确定以前向 EHLLAPI 发出的异步功能请求的状态，并在再次使用数据字符串之前获取功能参数列表。仅当用户在上一个功能调用（如**读取结构化字段** (126) 或**写入结构化字段** (127)）中指定异步 (A) 完成时，此功能才有效。

每个需要**获取请求完成**功能的异步请求都将从异步请求返回唯一的标识。应用程序必须保存此标识。此标识是**获取请求完成**功能用于标识所需请求的标识。用户拥有三个使用此功能的请求选项：

1. 应用程序可以查询或等待特定异步功能请求，方法是提供该功能的请求标识和非空会话短名称。
2. 通过提供请求标识 X'0000' 和非空会话短名称，应用程序可以查询或等待指定会话的第一个已完成的异步功能请求。

前提条件调用

连接结构化字段 (120) 和**分配通信缓冲区** (123)

并且

读取结构化字段 (126) 或写入结构化字段 (127)

调用参数

	标准接口	增强型接口
函数编号	必须为 125	
数据字符串	请参阅下表	
Length	必须为 14	必须为 24
PS 位置	不适用	

数据字符串内容

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留
2	7	N 或 W N=NOWAIT 为必填项 W=WAIT 为必填项
	6-8	保留
3-4	9-10	功能请求标识。
5-6	11-12	保留
7-10	13-16	保留
11-12	17-20	保留
13-14	21-24	保留

获取请求完成功能的行为不同，具体取决于参数字符串的第二个字符（即以下字符之一）：

否

Nowait 选项：如果提供了特定的请求标识，并且功能已完成，则会将控制权返回给应用程序，返回码为零，并且完成的数据字符串如 [返回参数 \(on page 81\)](#) 中定义。如果提供的请求标识为零，并且任何符合条件的异步功能都已完成，则会将控制权返回给应用程序，返回码为零，并且完成的数据字符串如 [返回参数 \(on page 81\)](#) 中定义。

W

Wait 选项：如果提供了特定的请求标识，但该功能尚未完成，则调用将等至该功能完成后再返回到应用程序。如果提供的请求标识为零，并且没有符合条件的异步函数已完成，则调用将等至某个功能完成后再返回到调用应用程序。返回时，返回码值将为零，数据字符串将按照 [返回参数 \(on page 81\)](#) 中的定义完成。

返回参数

字节		定义
标准	增强	
5-6	11-12	完成的异步功能的功能编号(126 或 127)。(已返回)
7-10	13-16	完成的异步功能调用的数据字符串的地址。(在请求完成之前,应用程序不得重用数据字符串)。(已返回)
11-12	17-20	完成的异步功能调用的数据字符串的长度。(已返回)
13-14	21-24	完成的异步功能调用的返回码。(已返回)

返回码	说明
0	获取请求完成功能已成功。
2	指定参数时出错。
9	遇到系统错误。
38	请求的函数未完成。
42	未找到匹配的请求。

返回码 38 和 42 之间存在一些差异:

1. 返回码 38

- 如果请求了特定的请求标识和会话,则找到了会话和标识,但请求处于待处理状态(未处于已完成状态)。
- 如果请求的是零请求标识和特定会话,则指定的会话具有待处理的请求,但这些请求未得到满足(完成)。
- 如果请求的请求标识为零且会话为空,则会找到待处理的请求,但未满足任何请求(完成)。

2. 返回码 42

- 如果请求了特定的请求标识和会话,则未找到处于待处理或已完成状态的特定请求标识。
- 如果请求的是零请求标识和特定会话,则特定会话不包含待处理或已完成的请求。
- 如果请求的是零请求标识和空白会话,则未找到待处理或已完成的请求。

有关使用此功能的注释

- 仅当用户在上一个函数调用(如**读取结构化字段**或**写入结构化字段**)上指定异步完成(A表示异步)时,此函数才有效。
- 如果返回码为0,则应用程序应检查返回的数据字符串,以获取有关完成所请求的异步函数的相关信息。

锁定表示空间 API (60)

3270	5250	VT
是	否	编号

锁定表示空间 API 功能允许应用程序获得或释放表示空间窗口的独占控制，而不是其他 Windows 32 位应用程序。锁定时，其他应用程序均无法连接到表示空间窗口。

使用锁定成功处理此功能会导致从其他 EHLLAPI 应用程序请求的 EHLLAPI 表示空间窗口功能进入队列，直到请求应用程序解锁表示空间。已正常处理来自锁定应用程序的请求。

前提条件调用

连接到表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 60	
数据字符串	请参阅下表	
Length	必须为 3	必须为 8
PS 位置	不适用	

数据字符串内容

字节	标准	增强	定义
1	1		1 字符表示空间短名称 (PSID)
	2-4		保留。
2	7		以下字符表示： <ul style="list-style-type: none"> • L 表示锁定 API。 • U 表示解锁 API。
	8		以下字符表示： <ul style="list-style-type: none"> • R 表示在表示空间已被应用程序锁定时返回。 • Q 表示在表示空间已被应用程序锁定时将锁定请求排队。
	7-8		保留。

返回参数

返回码	说明
0	锁定表示空间 API 功能已成功。
1	指定的主机表示空间短会话标识不正确或未连接。
2	指定参数时出错。
9	遇到系统错误。

返回码	说明
43	API 已被另一个 EHLLAPI 应用程序 (LOCK 上) 锁定或 API 未被锁定 (UNLOCK 上)。

有关使用此功能的注释

当锁定生效时，以下 EHLLAPI 功能将排队：

- **发送键** (3)
- **复制表示空间** (5)
- **搜索表示空间** (6)
- **将表示空间复制到字符串** (8)
- **释放** (11)
- **保留** (12)
- **查询字段属性** (14)
- **将字符串复制到表示空间** (15)
- **搜索字段** (30)
- **查找字段位置** (31)
- **查找字段长度** (32)
- **将字符串复制到字段** (33)
- **将字段复制到字符串** (34)
- **设置游标** (40)
- **发送文件** (90)
- **将表示空间复制到剪贴板** (35)
- **将剪贴板粘贴到表示空间** (36)
- **接收文件** (91)
- **连接到表示空间** (1)，其中在之前的**设置会话参数** (9) 功能调用中设置 CONPHYS 参数。

在解锁之前，这些排队的请求不会得到服务。解锁后，将按先进先出 (FIFO) 顺序处理排队的请求。未列出的 EHLLAPI 功能将像没有锁定一样运行。请求应用程序通过以下一种方法解锁表示空间窗口：

- 在仍拥有锁定的情况下断开与表示空间的连接。
- 在仍拥有锁定的情况下发出**重置系统** (21) 功能。
- 在仍拥有锁定的情况下停止应用程序。
- 停止会话。
- 使用“解锁”选项成功发布**锁定表示空间 API**。

在退出应用程序之前，应该解锁已使用**锁定表示空间 API** 功能锁定的任何表示空间窗口。如果应用程序退出时有未完成的锁定，或者发出了**重置系统** (21) 或**断开表示空间** (2) 功能，则会释放这些锁定。

建议应用程序仅在短时间内锁定表示空间，并且仅在需要专用表示空间时锁定。

锁定窗口服务 API (61)

3270	5250	VT
是	否	编号

锁定窗口服务 API 功能允许应用程序获得或释放表示空间窗口的独占控制，而不是其他 Windows 32 位应用程序。锁定时，其他应用程序均无法连接到表示空间窗口。

使用锁定成功处理此功能会导致从其他 EHLLAPI 应用程序请求的 EHLLAPI 表示空间窗口功能进入队列，直到请求应用程序解锁表示空间。已正常处理来自锁定应用程序的请求。

前提条件调用

连接窗口服务 (101)

调用参数

	标准接口	增强型接口
函数编号	必须为 61	
数据字符串	请参阅下表。	
Length	必须为 3	必须为 8
PS 位置	不适用	

数据字符串内容

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留。
2	7	以下字符表示： <ul style="list-style-type: none"> • L 表示锁定 API。 • U 表示解锁 API。
3	8	以下字符表示： <ul style="list-style-type: none"> • R 表示在表示空间已被应用程序锁定时返回。 • Q 表示在表示空间已被应用程序锁定时将锁定请求排队。
5-6	11-12	完成的异步功能的功能编号(126 或 127)。(已返回)
	7-8	保留。

返回参数

返回码	说明
0	锁定窗口服务 API 功能已成功。
1	指定了的主机表示空间短会话标识不正确或未连接。
2	指定参数时出错。
9	遇到系统错误。
38	请求的函数未完成。
43	API 已被另一个 EHLLAPI 应用程序 (LOCK 上) 锁定或 API 未被锁定 (UNLOCK 上)。

有关使用此功能的注释

当锁定生效时，以下 EHLLAPI 功能将排队：

- 窗口状态 (104)
- 更改交换机列表名称 (105)
- 更改 PS 窗口名称 (106)

在解锁之前，这些排队的请求不会得到服务。解锁后，将按先进先出 (FIFO) 顺序处理排队的请求。

请求应用程序通过以下一种方法解锁表示空间窗口：

- 使用 UNLOCK 选项成功发出**锁定窗口服务 API**。
- 在仍拥有锁定的情况下断开与表示空间的连接。
- 在仍拥有锁定的情况下发出**重置系统 (21)** 功能。
- 在仍拥有锁定的情况下停止应用程序。
- 停止会话。

在退出应用程序之前，应该解锁已使用**锁定窗口服务 API** 功能锁定的任何表示空间窗口。如果应用程序退出时有未完成的锁定，则子系统将释放这些锁定。

建议应用程序仅在短时间内锁定表示空间，并且仅在需要专用表示空间时锁定。

暂停 (18)

3270	5250	VT
是	是	是

暂停功能将等待指定的时间量。应使用它来取代计时循环，以等待事件发生。如果调用了之前的**启动主机通知 (23)** 功能并选择了 IPAUSE 选项，则主机事件可以结束**暂停**功能。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 18	
数据字符串	不适用	
Length	包含以半秒为增量的暂停持续时间	
PS 位置	不适用	

返回参数

返回码	定义
0	等待持续时间已过期。
9	遇到内部系统错误。时间结果不可预测。
26	主机会话表示空间或 OIA 已更新。使用 查询主机更新 (24) 功能获取更多信息。

有关使用此功能的注释

- 使用**设置会话参数** (9) 功能选择 FPAUSE 或 IPAUSE 选项时，会影响调用此功能时获得的暂停时间长度。请参阅 [项 6 \(on page 128\)](#) 以获取更多信息。
- 在调用长度参数中输入的值是**暂停功能**等待的最大半秒间隔数。暂停 20 秒时，必须在调用长度参数中传递十六进制值 0028（十进制数 40）。
- 如果使用 IPAUSE 选项且暂停值为零，则除非提前中断，否则该功能最多等待 2400 半秒间隔。如果使用 FPAUSE 选项且暂停值为零，则该功能将立即返回。
- 如果使用 IPAUSE 选项，一旦主机事件满足暂停条件，则应调用**查询主机更新** (24) 功能，以在下一个**暂停功能**之前清除队列。**暂停功能**将继续满足待处理事件的要求，直到**查询主机更新** (24) 功能完成。
- 暂停功能**的实际最大值为 2400。对于以下类型的任务，不能使用**暂停功能**：
 - 延迟持续时间非常长（例如数小时）。
 - 在检查系统时间时钟并继续 EHLLAPI 程序执行之前，延迟时间超过中等长度（20 分钟）。
 - 由于**暂停功能**创建的时间间隔是近似值，因此应用程序需要高分辨率计时器。
 - 将循环中的时间间隔设置为零。
- IPAUSE 设置和可中断暂停允许 EHLLAPI 应用程序确定是否更新指定的主机表示空间 (PS) 或操作员信息区域 (OIA)。将使用以下三个功能：
 - **启动主机通知** (23)
 - **查询主机更新** (24)
 - **停止主机通知** (25)

通过在调用**启动功能**时使用 IPAUSE，可以让应用程序等待主机表示空间或 OIA（或两者）收到更新。当接收完成并且应用程序可以发出**查询功能**以确定更改时，**暂停**将终止。接着，应用程序会发出**搜索表示空间** (6)，以检查是否发生了预期的更新。

拦截后状态 (52)

3270	5250	VT
是	是	是

拦截后状态功能会通知 Z and I Emulator for Windows 仿真器通过**获取键** (51) 功能获得的击键已被接受或拒绝。当应用程序拒绝击键时，**拦截后状态**功能将发出蜂鸣声。

前提条件调用

启动击键拦截 (50)

调用参数

	标准接口	增强型接口
函数编号	必须为 52	
数据字符串	请参阅下表	
Length	必须为 2	必须为 8
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 表示空间的 1 字母短名称。 • 表示主机连接的表示空间的函数调用的空白或 null。
	2-4	保留
2	7	以下字符表示： <ul style="list-style-type: none"> • A 表示接受的击键。 • R 代表被拒绝的击键。
	6-8	保留。

返回参数

返回码	说明
0	拦截后状态 功能已成功。
1	指定的表示空间不正确。
2	指定的会话选项不正确。
8	未为此表示空间标识调用先前的 开始击键拦截 (50) 功能。
9	遇到系统错误。

查询附加字段属性 (45)

3270	5250	VT
编号	是	否

查询附加字段属性功能会返回有关包含输入主机表示空间位置的 5250 字段的附加信息。此信息以所定义结构的形式在数据字符串参数中返回。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 45。	
数据字符串	8 字节长的字符串。	
Length	8 是隐含的。	
PS 位置	标识目标。这可以是目标字段中任何字节的 PS 位置。	

调用数据字符串可以包含：

字节	定义
1-8	保留

返回参数

此函数会返回数据字符串和返回码。

数据字符串：

此函数会返回以下数据字符串。

字节	定义
1-6	保留
7-8	返回两个 8 位无符号字符： <ul style="list-style-type: none"> • 如果字段为 RTL，则为 R；如果字段为 LTR，则为 L。 • 如果字段是大写，则为 U；如果字段是普通的大小写字段，则为 L。

返回码：

定义了以下返回码：

返回码	说明
0	查询附加字段属性已成功。
1	程序当前未连接到主机会话。
7	主机表示空间位置无效。
9	在此位置中未找到字段。
24	字段未格式化。

查询关闭拦截 (42)

3270	5250	VT
是	是	是

查询关闭拦截功能允许应用程序确定是否选择了关闭选项。

前提条件调用

启动关闭拦截 (41)

调用参数

	标准接口	增强型接口
函数编号	必须为 42	
数据字符串	请参阅下表。	
长度	必须为 1	必须为 4
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强
1	1 主机表示空间的 1 字符短会话标识，或者表示查询主机连接的会话的请求的空白或 null
	2-4 保留

返回参数

返回码	说明
0	未发生关闭拦截事件。
1	表示源无效。
2	指定参数时出错。
8	未为此主机表示空间调用先前的启动关闭拦截 (41) 功能。

返回码	说明
9	发生了系统错误。
12	会话已停止。
26	自上次查询关闭拦截调用以来发生了关闭拦截。

查询通信缓冲区大小 (122)

3270	5250	VT
是	否	编号

查询通信缓冲区大小功能允许应用程序确定仿真程序支持的最大和最佳缓冲区大小。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 122	
数据字符串	请参阅下表	
Length	必须为 9	必须为 20
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强
1	1 1 字符表示空间短名称 (PSID)
	2-4 保留
2-3	5-8 支持的最佳进站缓冲区大小的 16 位或 32 位字段 (返回值)
4-5	9-12 支持的最大进站缓冲区大小的 16 位或 32 位字段 (返回值)
6-7	13-16 支持的最佳出站缓冲区大小的 16 位或 32 位字段 (返回值)
8-9	17-20 支持的最大出站缓冲区大小的 16 位或 32 位字段 (返回值)

返回参数

返回码	说明
0	查询通信缓冲区大小功能已成功。
1	指定的主机表示空间短会话标识无效或未连接。
2	指定参数时出错。

返回码	说明
9	发生了系统错误。
10	仿真程序不支持此功能。

有关使用此功能的注释

1. 没有办法要求用户使用此功能。它不是必需的功能，因此可以定制应用程序，以在任何系统上运行。
2. 返回的缓冲区大小表示实际通过介质传输的记录大小。对于 DDM 连接，会删除**读取和写入结构化字段**数据缓冲区中提供的 8 字节标头，并且以包含结构化字段 AID 值的 1 字节为前缀。应用程序应将数据缓冲区中实际数据的大小（不包括 8 字节标头）与**查询通信缓冲区大小**返回的缓冲区大小减去 1 字节进行比较。对于目标/来源连接，会删除**读取和写入结构化字段**数据缓冲区中提供的 8 字节标头，然后将 9 字节作为数据的前缀。应用程序应将数据缓冲区中实际数据的大小（不包括 8 字节标头）与**查询通信缓冲区大小**返回的缓冲区大小减去 9 字节进行比较。
3. 返回的最大缓冲区大小代表工作站硬件和仿真器支持的最大字节数。仅当也将主机配置为至少接受这些最大小时，才可以使用最大缓冲区大小。
4. 返回的最佳缓冲区大小代表工作站硬件和仿真器支持的最佳字节数。某些网络配置可能会将传输限制设置为小于这些值。在这些情况下，仿真器配置概要文件中的数据传输缓冲区大小覆盖值将用于结构化字段支持。**查询通信缓冲区大小**将反映在仿真器配置概要文件中输入的任何缓冲区大小覆盖值。

查询通信事件 (81)

3270	5250	VT
是	是	是

查询通信事件功能让 EHLLAPI 程序可确定是否发生了任何通信事件。

前提条件调用

启动通信通知 (80)

调用参数

	增强型接口
函数编号	必须为 81
数据字符串	主机表示空间的 1 字符短名称，或者表示主机连接的表示空间的更新请求的空白或 null
Length	4 是隐含的
PS 位置	不适用

调用数据结构包含这些元素：

字节	定义
----	----

1	1 字符表示空间短名称 (PSID)
2 - 4	保留

返回参数

返回码	定义
0	此功能已成功
1	指定的 PSID 不正确。
8	未为 PSID 调用之前的 启动通信通知 (80) 功能调用
9	遇到系统错误
21	指示的 PSID 已连接
22	指示的 PSID 已断开连接

查询游标位置 (7)

3270	5250	VT
是	是	是

查询游标位置功能通过返回游标位置来指示游标在主机连接的表示空间中的位置。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 7	
数据字符串	不适用	
Length	不适用	
PS 位置	不适用	

返回参数

此功能会返回一个长度和一个返回码。

长度:

游标的主机表示空间位置。

返回码:

定义了以下代码:

返回码	说明
0	查询光标位置功能已成功。
1	程序当前未连接到主机会话。
9	遇到系统错误。

查询字段属性 (14)

3270	5250	VT
是	是	是

查询字段属性功能会返回包含输入主机表示空间位置的字段的属性字节。此信息将在返回的长度参数中返回。

对于 PC/3270，另请注意：

- 如果屏幕未格式化，则返回的长度参数设置为 0。
- 属性字节等于或大于十六进制 C0。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 14。	
数据字符串	不适用。	
Length	不适用。	
PS 位置	标识目标。这可以是目标字段中任何字节的 PS 位置。	

返回参数

此功能会返回一个长度和一个返回码。

长度：

属性值（如果屏幕已格式化）或 0（如果屏幕未格式化）。

返回码：

定义了以下代码：

返回码	说明
0	查询字段属性已成功。
1	程序当前未连接到主机会话。
7	主机表示空间位置无效。

返回码	说明
9	遇到系统错误。
24	未找到属性字节或主机表示空间未格式化。

有关使用此功能的注释

返回的字段属性在下表中定义。位位置采用 IBM 格式，其中位 0 是字节中最左侧的位。

• 3270 字段属性：

位位置	意义
0-1	两者 = 1, 字段属性字节
2	未受保护/受保护 0 = 未受保护的数据字段 1 = 受保护的字段
3	A/N 0 = 字母数字数据 1 = 仅数字数据
4-5	I/SPD 00 = 正常亮度, 光笔无法检测 01 = 正常亮度, 光笔可检测 10 = 高亮度, 光笔可检测 11 = 不显示, 光笔无法检测
6	保留
7	MDT 0 = 字段未被修改 1 = 字段已被修改

• 5250 字段属性：

位位置	意义
0	字段属性标志 0 = 非字段属性标志 1 = 字段属性标志
1	可见性 0 = 不显示 1 = 显示

位位置	意义
2	未受保护/受保护 0 = 未受保护的数据字段 1 = 受保护的字段
3	亮度 0 = 正常亮度 1 = 高亮度
4-6	字段类型 000 = 字母数字数据：所有字符均可用 001 = 仅字母：大写和小写、逗号、句点、连字符、空格或 Dup 键可用 010 = 数字转换：数字自动转换 011 = 仅数字数据：0-9、逗号、句点、加号、减号、空格或 Dup 键可用 101 = 仅数字数据：0-9 或 Dup 键可用 110 = 仅磁条读取设备数据 111 = 带符号的数字数据：0-9、加号、减号或 Dup 键可用
7	MDT 0 = 字段未被修改 1 = 字段已被修改

查询主机更新 (24)

3270	5250	VT
是	是	是

通过**查询主机更新**功能，编程运算符可以确定主机是否更新了主机表示空间或 OIA，因为：

- 已调用**启动主机通知** (23) 功能（仅限首次调用**查询主机更新**功能时）
- 对**查询主机更新**功能的上一次调用（对于**查询主机更新**功能的所有调用，第一次调用除外）。

前提条件调用

启动主机通知 (23)

调用参数

	标准接口	增强型接口
函数编号	必须为 24	

	标准接口	增强型接口
数据字符串	主机表示空间的 1 字符短名称，或者表示主机连接的表示空间的更新请求的空白或 null	
Length	1 是隐含的	4 是隐含的
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留

返回参数

返回码	定义
0	自上次调用以来未进行任何更新。
1	指定的主机表示空间不正确。
8	未为主机表示空间标识调用先前的 启动主机通知 (23) 功能。
9	遇到系统错误。
21	OIA 已更新。
22	表示空间已更新。
23	OIA 和主机表示空间都已更新。
44	打印已在打印机会话中完成。

有关使用此功能的注释

必须在数据字符串中指定目标表示空间，即使不需要连接到主机表示空间来检查更新也是如此。

查询会话状态 (22)

3270	5250	VT
是	是	是

查询会话状态功能用于获取特定于会话的信息。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	16 位	32 位
函数编号	必须为 22。	
数据字符串	18/20 字节的字符串，由目标表示空间的 1 字节短名称加上 17 字节的返回数据组成。位置 1 可以填充： 1. 表示 host_connected 表示空间请求的空白或 null。 2. * (星号) 表示键盘所有者表示空间请求。	
Length	必须为 18	必须为 20
PS 位置	不适用	

返回参数

此函数会返回数据字符串和返回码。

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留
2-9	5-12	会话长名称 (与概要文件名相同; 或者, 如果未设置概要文件, 则与短名称相同)
10	13	会话 类型 D 3270 显示 E 3270 打印机 F 5250 显示 G 5250 打印机 H ASCII VT
11	14	由二进制数字表示的会话特性, 包括以下会话特性位 位 0 EAB 0: 会话具有基本属性。 1: 会话具有扩展属性

字节		定义
		位 1 PSS 0: 会话不支持已编程符号 1: 会话支持已编程符号 位 2-7 保留
12-13	15-16	主机表示空间中的行数, 以二进制数字表示
14-15	17-18	主机表示空间中的列数, 以二进制数字表示
16-17	19-20	以二进制数字表示的主机代码页
18		保留

返回码:

定义了以下代码:

返回码	说明
0	查询会话状态功能已成功。
1	指定的主机表示空间不正确。
2	指定的字符串长度不正确。
9	遇到系统错误。

有关使用此功能的注释

1. 要使用此功能, 请预先分配内存以接收返回的数据字符串参数。预先分配此内存所需的语句会因用于编写应用程序的语言而异。请参阅 [内存分配 \(on page 9\)](#) 以获取更多信息。

查询会话 (10)

3270	5250	VT
是	是	是

查询会话功能会返回描述每个主机会话的 16 字节 (标准接口为 12 字节) 数据字符串。


前提条件调用

此函数没有任何前提条件调用。

调用参数

功能	描述	
	标准接口	增强型接口
函数编号	必须为 10	

功能	描述	
数据字符串	预先分配的 $16n$ 个字节长或更多的字符串 (16 位为 $12n$) (n =会话数)	
Length	$12n$ 个字节	$16n$ 个字节
PS 位置	不适用	

 **注：** 长度与会话数不匹配时，返回码为 2。

返回参数

此函数会返回数据字符串、长度和返回码。

数据字符串：

返回的数据字符串长度为 $16n$ 字节 (标准接口为 $12n$)，其中 n 是主机会话数。描述符并置到数据字符串、每个会话类型，以及主机会话的表示空间大小。

每个 16 字节 (标准接口为 12 字节) 会话描述符的格式如下所示：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留
2-9	5-12	会话长名称 (与概要文件名相同；或者，如果未设置概要文件，则与短名称相同)
10	13	连接类型 H=主机
	14	保留
11-12	15-16	主机表示空间大小 (这是一个二进制数字，不是采用显示格式)。如果会话类型为打印会话，则值为 0。

长度：

已启动的主机会话数。

返回码：

定义了以下代码：

返回码	说明
0	查询会话功能已成功。
2	指定的字符串长度不正确。
9	遇到系统错误。

有关使用此功能的注释

1. 如果应用程序接收到 RC=2 或 RC=0，则会在长度字段中返回活动会话数。应用程序可以通过此数字识别最小字符串长度。
2. **查询会话**功能受 CFGSIZE/NOCFGZISE 会话选项（请参阅项 16 (on page 131) 以获取更多信息）和 EXTEND_PS/NOEXTEND_PS 选项（请参阅项 20 (on page 132) 以获取更多信息）的影响。



注:

1. 当在**设置会话参数** (9) 中为 5250 会话设置 NOCFGZISE 时，将根据 EXTEND_PS 或 NOEXTEND_PS 的选择更改从**查询会话** (10) 返回的字节位置 11 和 12 中的表示空间大小值。
2. 在**设置会话参数** (9) 中设置 EXTEND_PS 时，从**查询会话** (10) 返回的表示空间大小将包括消息行的大小（如果存在）。
3. 如果设置了 NOEXTEND_PS，则无论是否存在消息行，该值都不会更改。在 25 行、80 列表示空间的情况下，值可以是 1920 或 2000。

查询系统 (20)

3270	5250	VT
是	是	是

EHLLAPI 应用程序可以使用**查询系统**功能，确定 Z and I Emulator for Windows 支持级别和其他系统相关值。此功能会返回包含相应系统数据的字符串。在收到返回码 9 后致电 IBM 支持中心时，这些信息中的大部分可供服务协调员使用（遇到系统错误）。

此返回字符串中的字节数在 [返回参数 \(on page 101\)](#) 中定义。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 20	
数据字符串	预先分配的 35 字节字符串	36 字节
Length	必须为 35	必须为 36
PS 位置	不适用	

返回参数

此函数会返回数据字符串和返回码。

数据字符串:

会返回 35 字节 (16 位) 或 36 字节 (32 位) 的数据字符串。这些字节的定义如下:

字节		定义
标准	增强	
1	1	EHLLAPI 版本号
2-3	2-3	EHLLAPI 级别号
4-9	4-9	保留
10-12	10-12	保留
13	13	硬件基础, U=无法确定
14	14	程序类型, 其中 P=HCL Z and I Emulator for Windows
15-16	15-16	保留
17-18	17-18	Z and I Emulator for Windows 版本/级别为 2 字节 ASCII 值
19	19 位	保留
20-23	20-23	保留
24-27	24-27	保留
28-29	28-29	保留
	30	保留
30-31	31-32	以 2 字节二进制数字表示的 NLS 类型
33-35	34-36	保留

返回码

定义了以下代码:

返回码	说明
0	查询系统功能已成功; 已返回数据字符串。
1	未装入 EHLLAPI。 (仅限 PC/3270)
2	指定的字符串长度不正确。 (仅限 PC/3270)
9	遇到系统错误。

有关使用此功能的注释

要使用此功能, 请预先分配内存以接收返回的数据字符串参数。请参阅 [内存分配 \(on page 9\)](#) 以获取更多信息。

查询窗口坐标 (103)

3270	5250	VT
是	是	是

查询窗口坐标功能请求表示空间窗口的坐标。窗口坐标以像素为单位返回。



注： (0,0) 表示窗口的左上角。

前提条件调用

连接窗口服务 (101)

调用参数

	标准接口	增强型接口
函数编号	必须为 103	
数据字符串	主机表示空间的 1 字符短会话标识	
Length	17 是隐含的	20 是隐含的
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 1 字符表示空间短名称 (PSID) • 表示当前连接表示空间的函数调用的空白或 null
	2-4	保留
2-17	5-20	保留

返回参数

此函数会返回数据字符串和返回码。

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 1 字符表示空间短会话标识 • 表示当前连接表示空间的函数调用的空白或 null
	2-4	保留

字节		定义
2-17	5-20	返回的四个 32 位无符号整数:
2-5	5-8	相对于桌面窗口的矩形窗口左 X 坐标的 XLeft 长整数 (以像素为单位)
6-9	9-12	相对于桌面窗口的矩形窗口底部 Y 坐标的 YBottom 长整数 (以像素为单位)
10-13	13-15	相对于桌面窗口的矩形窗口的右 X 坐标的 XRight 长整数 (以像素为单位)
14-17	16-20	相对于桌面窗口的矩形窗口的顶部 Y 坐标的 YTop 长整数 (以像素为单位)

返回码:

定义了以下代码:

返回码	说明
0	查看窗口坐标功能已成功。
1	您的程序当前未连接到主机会话。
9	发生了系统错误。
12	会话已停止。

读取结构化字段 (126)

3270	5250	VT
是	否	编号

读取结构化字段功能允许应用程序从主机应用程序读取结构化字段数据。如果调用指定 S (表示同步), 则在完成**读取结构化字段**之前, 应用程序不会获得控制权。如果调用指定 A (表示异步), 则应用程序会在调用后立即获得控制权。如果调用指定 M (表示异步、消息模式), 则应用程序会在调用后立即获得控制权。应用程序可以等待消息。在任何情况下 (S、A 或 M), 应用程序都提供将在其中放置来自主机的数据的缓冲区地址。

要成功地异步完成此功能, 以下陈述将适用:

参数列表中的返回码字段可能不包含所请求 I/O 的结果。如果返回码不是 0, 则请求失败。应用程序必须根据返回码采取适当的措施。

如果此请求的返回码为 0, 则应用程序必须使用此功能调用返回的请求标识, 发出**获取请求完成**功能调用, 以确定与请求标识关联的功能的完成结果。**获取请求完成**功能调用返回以下信息:

1. 功能请求标识
2. 来自异步请求的数据字符串的地址
3. 数据字符串的长度
4. 已完成功能的返回码

前提条件调用

结构化字段的连接 (120) 和分配通信缓冲区 (123)

调用参数

	标准接口	增强型接口
函数编号	必须为 126	
数据字符串	请参阅下表	
Length	8、10 或 14	20
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留。
2	7	S 或 A 或 M S = 同步。在满足读取要求之前，不会将控制权返回给应用程序。 A = 异步。将控制权立即返回给应用程序，可以等待事件对象。 M = 异步。将控制权立即返回给应用程序，可以等待消息。
	6	保留。
3-4	7-8	2 字节目标/来源标识。
5-8	9-12	要将数据读取到的缓冲区的 4 字节地址。必须使用分配通信缓冲区 (123) 功能获得缓冲区。
9-10	13-16	保留。
11-12	17-20	在位置 2 中指定 M 时，应设置接收消息的窗口的窗口句柄。消息是 RegisterWindowMessage ("PCSHLL") 的返回值 (不等于 0) 。
13-14		EHLAPI 会忽略这些位置中的数据。但是，如果迁移程序的数据位于这些位置，则不会导致错误。会接受此数据以提供与迁移应用程序的兼容性。


返回参数

此函数会返回数据字符串和返回码。

数据字符串:

如果在位置 5 (标准接口为 2) 中指定 A (异步), 并且功能已成功完成, 则返回以下数据字符串:


字节		定义
标准	增强	
9-10	13-14	2 字节功能请求标识。 获取请求完成 (125) 功能使用它来确定此功能调用的完成情况。
	15-16	保留。
	17-20	EHLLAPI 返回事件对象地址的 4 字节值。应用程序可以等待此事件对象。清除事件对象后, 应用程序必须发出 获取请求完成 (125) 功能调用 (仅限 32 位)。

 **注:** 每个成功的异步请求都将返回一个事件对象地址。不应再次使用事件对象。将为每个请求返回一个新的事件对象, 该对象仅在该请求的持续时间内有效。

数据字符串:

如果在位置 5 (16 位应用程序为 2) 中指定 “M” (异步消息方式), 并且功能已成功完成, 则返回以下数据字符串:

字节		定义
9-10	13-14	2 字节功能请求标识。 获取请求完成 (125) 功能使用它来确定此功能调用的完成情况。
	15-16	保留。
11-12	17-18	异步消息方式的任务标识。
	19-20	保留。

 **注:** 如果该功能已成功完成, 应用程序窗口将收到一条消息。消息是 RegisterWindowMessage (PCSHLL) 的返回值。wParam 参数包含功能调用返回的任务标识。lParam 参数的 HIWORD 包含返回码 0, 表示功能已成功, lParam 参数的 LOWORD 包含功能编号 126。

返回码:

定义了以下代码:

返回码	说明
0	读取结构化字段 功能已成功。
1	指定的主机表示空间短会话标识无效或未连接。
2	指定参数时出错。
9	发生了系统错误。
11	资源不可用 (内存不可用)。
35	请求被拒绝。主机的出站传输已被取消。

返回码	说明
36	请求被拒绝。与主机失去联系。
37	该功能已成功，但主机被禁用入站。

有关使用此功能的注释

1. 当从主机发出的出站传输被取消后请求第一个**读取结构化字段**或**写入结构化字段**时，将返回返回码 35。纠正措施由应用程序负责。
2. 返回码 36 要求应用程序断开与仿真程序的连接，然后重新连接，以重新建立与主机的通信。纠正措施由应用程序负责。
3. 如果主机被禁用入站，将返回返回码 37。已成功请求**读取结构化字段**功能。
4. EHLLAPI 允许每个应用程序最多可有 20 个未完成的异步请求。如果尝试的异步请求超过 20 个，将返回不可用资源的返回码 (RC=11)。

结构化字段数据包含从主机接收的应用程序结构化字段。在结构化字段数据到达应用程序之前，EHLLAPI 已删除结构化字段标头。

结构化字段数据格式如下所示：

偏移量	Length	目录
0	1 个字	X'0000'.
2	1 个字	m (消息长度：消息中数据的字节数，该数字不包括缓冲区标头前缀，该前缀包含 8 个字节)。此值由 EHLLAPI 返回。
4	1 个字	n (缓冲区大小：提供的数据缓冲区长度，其中包括 8 字节的消息头)。此值必须由应用程序设置。
6	1 个字	X'C000'.
8	8 字节	第一个 (或唯一) 结构化字段消息的长度。
10	1 个字节	结构化字段消息的第一个非长度字节。
		⋮
m+7	1 个字节	结构化字段消息中的最后一个字节。

字节 0 到 7 是缓冲区标头。前 8 个字节由仿真程序使用。缓冲区的用户部分以偏移量 8 开始。字节 8 和 9 包含第一个结构化字段 (结构化字段消息可以包含多个结构化字段) 中的字节数，包括字节 8 和 9 的 2 个字节。字节 8 到 m+7 用于从主机接收的结构化字段消息 (可能包含多个结构化字段)。

使用的应用程序必须提供完整的缓冲区，并将偏移量 0 处的字设置为零。缓冲区长度必须在偏移量 4 处的字中。偏移量 6 处的字必须是 X'C000'。仿真程序会将数据消息从偏移量 8 开始放置，并将消息的长度放在偏移量 2 处。缓冲区长度不受 EHLLAPI 干扰。

同步请求

同步请求**读取结构化字段** (数据字符串中的 S 选项) 时，只有在满足请求后，控制权才会返回到应用程序。应用程序可以假定：

- 返回码正确。
- 通信缓冲区（读取缓冲区）中的数据正确。
- 主机不再处理**读取结构化字段**请求。

异步请求

异步请求**读取结构化字段**（数据字符串中的 A 选项）时，应用程序不能假定：

- 返回码正确。
- 通信缓冲区（读取缓冲区）中的数据正确。
- 主机不再处理**读取结构化字段**请求。

异步请求时，EHLLAPI 会返回以下值：

- 数据字符串的位置 13-14（标准接口为 9-10）中的 16 位请求标识
- 数据字符串位置 17-20 中事件对象的地址

这些地址用于完成异步**读取结构化字段**调用。

要确定异步**读取结构化字段**函数调用的结果，必须完成以下步骤：

- 如果 EHLLAPI 返回码不是零，则请求失败。未发出异步请求。应用程序必须在再次尝试调用之前采取适当的措施。
- 如果返回码为零，则应用程序应使用**获取请求完成** (125) 功能或**等待单个对象**等至事件对象处于已发出信号状态。不能重用事件对象。事件对象仅在**读取结构化字段**功能调用直至完成**获取请求完成** (125) 功能调用的持续时间内有效。
- 事件对象处于已发出信号状态后，则在**获取请求完成** (125) 功能调用中，将返回的 16 位请求标识用作请求标识参数。从**获取请求完成** (125) 功能调用返回的数据字符串包含**读取结构化字段**功能调用的最终返回码。

异步请求**读取结构化字段**（数据字符串中的 M 选项）时，应用程序不能假定：

- 返回码正确。
- 通信缓冲区（读取缓冲区）中的数据正确。
- 主机不再处理**读取结构化字段**请求。

当使用 M 选项异步请求时，EHLLAPI 会返回以下值：

- 数据字符串的位置 13-14（标准接口为 9-10）中的 16 位请求标识
- 数据字符串的位置 17-18（标准接口为 11-12）中异步消息方式的任务标识。

这些地址用于完成异步**读取结构化字段**调用。

接收文件 (91)

3270	5250	VT
是	是	否

接收文件功能用于将文件从主机会话传输到工作站会话。它的使用方式与在 PC/3270 中使用 RECEIVE 命令的方式相同。**接收文件**功能可由 EHLLAPI 应用程序调用。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 91。	
数据字符串	请参阅示例。	
Length	数据字符串的长度（以字节为单位）。处于 EOT 方式时则覆盖。	

以下是单字节字符集 (SBSC) 的数据字符串示例:

3270 会话

- 要从 VM/CMS 主机系统接收文件:

```
pc_filename [id:] fn ft [fm] [(option)]
```

- 要从 MVS™/TSO 主机系统接收文件:

```
pc_filename[id:] dataset[(member)] [/password] [option]
```

- 要从 CICS® 主机系统接收文件:

```
pc_filename [id:] host_filename [(option)]
```

5250 会话

- 要从 iSeries™、eServer™ i5 或 System i5™ 主机系统接收文件:

```
pc_filename [id:] library file member [option]
```

返回参数

返回码	说明
2	参数错误或您为 EHLLAPI 缓冲区指定的长度过长（超过 255 个字节）。文件传输失败。
3	文件传输完成。
4	文件传输已完成但出现分段记录。
9	遇到系统错误。

返回码	说明
27	由于“取消”按钮或 设置会话参数 (9) 功能设置的超时，文件传输已被终止。
101	文件传输已成功（传输至/自 CICS®）。

如果收到返回码 2 或 9，则表示系统或指定数据字符串的方式存在问题。

还可以接收与主机传输程序生成的消息编号相关的其他返回码。要传输至 CICS® 主机传输程序，请从返回码中减去 100，即可获得消息的数字部分。例如，返回码 101 表示消息编号 INW0001 是由主机发出的。对于其他主机传输程序，只需使用返回码作为消息的数字部分。例如，返回 34 表示消息 TRANS34 是由主机传输程序发出的。主机传输程序的文档应提供有关特定消息含义的更多信息。

EHLLAPI 报告的操作系统错误代码大于 300。要确定错误代码，请减去 300 并参阅操作系统文档，以了解返回码。

有关使用此功能的注释

1. **设置会话参数 (9)** 功能下的四组参数与此功能相关。它们是 STRLEN/STREOT、EOT=c、QUIET/NOQUIET 和 TIMEOUT=c/TIMEOUT=0 会话选项。请参阅项 1 (on page 127) 和 2 (on page 127) 和项 7 (on page 128) 和 8 (on page 128) 以获取更多信息。
2. 如果在执行**接收文件**功能时未指定路径，则接收到的文件将存储在当前子目录中，该子目录是运行应用程序的目录。

释放 (12)

3270	5250	VT
是	是	是

发行版功能会解锁与使用**保留 (11)** 功能保留的主机表示空间关联的键盘。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 12	
数据字符串	不适用	
Length	不适用	
PS 位置	不适用	

返回参数

返回码	说明
0	释放功能已成功。
1	程序未连接到主机会话。
9	遇到系统错误。

有关使用此功能的注释

如果您未**释放**使用**保留**(11)功能保留的主机表示空间,则在调用**重置系统**(21)功能之前,您将被锁定在该会话之外,您会调用**断开表示空间**(2)功能,或者终止 EHLLAPI 应用程序。

保留 (11)

3270	5250	VT
是	是	是

保留功能会锁定与主机连接的表示空间关联的键盘,以阻止终端操作员输入。

保留的主机表示空间将保持锁定状态,直至出现以下一种情况:

- **连接**(1)功能已执行到新会话。
- **断开表示空间**(2)功能已执行。
- **释放**(12)功能已执行。
- **重置系统**(21)功能已执行。
- **开始击键拦截**(50)功能已执行。
- EHLLAPI 应用程序已终止。

前提条件调用

连接表示空间(1)

调用参数

	标准接口	增强型接口
函数编号	必须为 11	
数据字符串	不适用	
Length	不适用	
PS 位置	不适用	

返回参数

返回码	说明
0	保留功能已成功。
1	程序未连接到主机会话。
5	无法使用表示空间。
9	遇到系统错误。

有关使用此功能的注释

1. 如果 EHLLAPI 应用程序正在向主机发送一系列事务，则可能需要在应用程序处理完成之前阻止用户访问该会话。
2. 用户在键盘被此功能锁定时进行的键盘输入将在终止会话后排队并进行处理。
3. 此功能可锁定鼠标和键盘输入。应用程序必须解锁表示空间，才能启用鼠标或键盘输入。

重置系统 (21)

3270	5250	VT
是	是	是

重置系统功能会将 EHLLAPI 重新初始化至其起始状态。会话参数选项将重置为其缺省值。事件通知将被停止。保留的主机会话将被释放。主机表示空间已断开连接。击键拦截已被禁用。

您可以在初始化期间或程序终止时使用**重置系统**功能将系统重置为已知的初始状态。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 21	
数据字符串	不适用	
Length	不适用	
PS 位置	不适用	

返回参数

返回码	定义
0	重置系统 功能已成功。

返回码	定义
1	EHLLAPI 未装入。
9	遇到系统错误。

有关使用此函数的注释

对于 PC/3270，此函数可用于检查是否已装入 EHLLAPI。在启动应用程序时调用此函数并检查返回码是否为 1。

搜索字段 (30)

3270	5250	VT
是	是	是

搜索字段功能会检查连接的主机表示空间中的字段是否存在指定字符串。如果找到目标字符串，则此功能会返回从主机表示空间开始编号的字符串的十进制位置。（例如，在 24 行乘以 80 列的表示空间中，第 1 行、第 1 列的位置编号为 1，第 5 行、第 1 列的位置编号为 321。）

此功能可用于搜索受保护或未受保护的字段，但仅限于字段格式化主机表示空间。



注： 如果主机表示空间末尾的字段自动换行，则在到达表示空间末尾时将进行换行。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 30。	
数据字符串	用于搜索的目标字符串。	
Length	目标数据字符串的长度。处于 EOT 方式时则覆盖。	
PS 位置	标识目标字段。对于 SRCHALL，这可以是目标字段中任何字节的 PS 位置。对于 SRCHFROM，它是搜索 SRCHFRWD 的起点或搜索 SRCHBKWD 的终点。请参阅注释 3 (on page 113)。	

返回参数

此功能会返回一个长度和一个返回码。

长度：

定义了以下代码：

Length	说明
= 0	未找到字符串。
> 0	在指示的主机表示空间位置找到字符串。

返回码：

定义了以下代码：

返回码	说明
0	搜索字段功能已成功。
1	程序未连接到主机会话。
2	参数错误。字符串长度为零，或者指定了 EOT 方式，但在调用数据字符串中未找到 EOT 字符。
7	主机表示空间位置无效。
9	遇到系统错误。
24	未找到搜索字符串，或主机表示空间未格式化。

有关使用此功能的注释

1. **设置会话参数** (9) 功能下的四组参数与此功能相关。它们是 SRCHALL/SRCHFROM、STRLEN/STREOT、SRCHFRWD/SRCHBKWD 和 EOT=c 会话选项。请参阅项 1 (on page 127) 至 4 (on page 127) 以获取更多信息。
2. 可以使用**设置会话参数** (9) 功能来确定在字段中是向前搜索 (SRCHFRWD) 还是向后搜索 (SRCHBKWD)。
3. **搜索字段**功能通常会检查整个字段 (SRCHALL 缺省方式)。但是，可以使用功能 9 指定 SRCHFROM。在此方式下，调用 PS 位置参数的作用不仅仅是标识目标字段。它还提供搜索的起点或终点。
 - 如果 SRCHFRWD 选项有效，则对指定字符串的搜索从指定的 PS 位置开始，并继续到字段的末尾。
 - 如果 SRCHBKWD 选项有效，则对指定字符串的搜索从字段末尾开始，并向后转至指定的 PS 位置。如果未找到目标字符串，搜索将在调用 PS 位置参数中指定的 PS 位置结束。



注： 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真会显示第 25 行。当显示来自主机的错误消息或操作员选择 SysReq 键时，就会出现这种情况。Z and I Emulator for Windows 在第 24 行或状态栏上显示第 25 行信息。要在状态栏上显示信息，必须配置状态栏。请参阅 *Quick Beginnings* 以获取有关配置状态栏的信息。通过 **EXTEND_PS** 选项，EHLLAPI 应用程序可以使用与 Communication Manager EHLLAPI 相同的接口，并且在满足此条件时将扩展有效的表示空间。

搜索表示空间 (6)

3270	5250	VT
是	是	是

搜索表示空间功能让 EHLLAPI 程序可检查主机表示空间中是否存在指定字符串的实例。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 6。	
数据字符串	用于搜索的目标字符串。	
Length	目标数据字符串的长度。处于 EOT 方式时则覆盖。	
PS 位置	主机表示空间内搜索将开始 (SRCHFRWD 选项) 或结束 (SRCHBKWD 选项) 的位置。在 SRCHALL (缺省) 方式中将被覆盖。	

返回参数

此功能会返回一个长度和一个返回码。

长度:

定义了以下代码:

Length	说明
= 0	未找到字符串。
> 0	在指示的主机表示空间位置找到字符串。

返回码:

定义了以下代码:

返回码	说明
0	搜索表示空间功能已成功。
1	程序未连接到主机会话。
2	指定参数时出错。
7	主机表示空间位置无效。
9	遇到系统错误。
24	未找到搜索字符串。

有关使用此功能的注释

1. **设置会话参数** (9) 功能下的四组参数与此功能相关。它们是 SRCHALL/SRCHFROM、STRLEN/STREOT、SRCHFRWD/SRCHBKWD 和 EOT=c 会话选项。请参阅项 1 (on page 127) 至 4 (on page 127) 至 以获取更多信息。
2. 您可以使用**设置会话参数** (9) 功能指定 SRCHBKWD。当此选项生效时, 搜索操作将查找字符串的最后一个实例。

3. **搜索表示空间**功能通常会检查整个主机表示空间。但是，可以使用**设置会话参数 (9)** 功能来指定 SRCHFROM。在此方式下，调用 PS 位置参数指定搜索的起点或终点。
 - 如果 SRCHFRWD 选项有效，则对指定字符串的搜索从指定的 PS 位置开始，并继续到主机表示空间的末尾。
 - 如果 SRCHBKWD 选项有效，则对指定字符串的搜索从 PS 末尾开始，并向后转至指定的 PS 位置。如果未找到目标字符串，搜索将在调用 PS 位置参数中指定的 PS 位置结束。
4. 如果要查找可能在主机表示空间中出现多次的关键字，SRCHFROM 选项也很有用。
5. **搜索表示空间**功能在确定主机表示空间何时可用时非常有用。如果 EHLLAPI 应用程序在发送数据之前需要特定的提示或消息，则**搜索表示空间**功能允许在继续之前检查是否有提示消息。



注： 5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下，Communication Manager 5250 仿真会显示第 25 行。当显示来自主机的错误消息或操作员选择 SysReq 键时，就会出现这种情况。Z and I Emulator for Windows 在第 24 行或状态栏上显示第 25 行信息。要在状态栏上显示信息，必须配置状态栏。请参阅 *Quick Beginnings* 以获取有关配置状态栏的信息。通过 **EXTEND_PS** 选项，EHLLAPI 应用程序可以使用与 Communication Manager EHLLAPI 相同的接口，并且在满足此条件时将扩展有效的表示空间。

发送文件 (90)

3270	5250	VT
是	是	否

发送文件功能用于将文件从 EHLLAPI 正在运行的工作站会话传输到主机会话。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 90。	
数据字符串	请参阅示例。	
Length	目标数据字符串的长度。处于 EOT 方式时则覆盖。	
PS 位置	必须为 0。	

以下是 SBCS 的数据字符串示例

3270 会话

- 要将文件发送到 VM/CMS 主机系统：

```
pc_filename [id:] fn ft [fm] [(option)]
```

- 要将文件发送到 MVS/TSO 主机系统：

```
pc_filename [id:] dataset[(member)] [/password] [option]
```

- 要将文件发送到 CICS 主机系统:

```
pc_filename [id:] host_filename [(option)]
```

5250 会话

- 要将文件发送到 iSeries™、eServer™ i5 或 System i5™ 主机系统:

```
pc_filename [id:] library file member [option]
```

返回参数

返回码	说明
2	参数错误或您为 EHLLAPI 缓冲区指定的长度过长 (超过 255 个字节)。文件传输失败。
3	文件传输完成。
4	文件传输已完成但出现分段记录。
5	工作站文件名无效或未找到。文件传输已被取消。
9	遇到系统错误。
27	由于“取消”按钮或 设置会话参数 (9) 功能设置的超时, 文件传输已被终止。
101	文件传输已成功 (传输至/自 CICS)。

如果收到返回码 2 或 9, 则表示系统或指定数据字符串的方式存在问题。

还可以接收与主机传输程序生成的消息编号相关的其他返回码。要传输至 CICS 主机传输程序, 请从返回码中减去 100, 即可获得消息的数字部分。例如, 返回码 101 表示消息编号 INW0001 是由主机发出的。对于其他主机传输程序, 只需使用返回码作为消息的数字部分。例如, 返回 34 表示消息 TRANS34 是由主机传输程序发出的。主机传输程序的文档应提供有关特定消息含义的更多信息。

EHLLAPI 报告的操作系统错误代码大于 300。要确定错误代码, 请减去 300 并参阅操作系统文档, 以了解返回码。

有关使用此功能的注释

1. **设置会话参数 (9)** 功能下的四组参数与此功能相关。它们是 QUIET/NOQUIET、STRLEN/STREET、TIMEOUT=c/TIMEOUT=0 和 EOT=c 会话选项。请参阅项 1 (on page 127) 和 2 (on page 127) 以及项 7 (on page 128) 和 8 (on page 128) 以获取更多信息。

发送键 (3)

3270	5250	VT
是	是	是

发送键功能用于将击键或击键字符串发送到主机表示空间。

可以使用调用数据字符串参数定义要发送的击键字符串。击键在目标会话中显示为终端操作员输入的击键。还可以发送所有注意标识 (AID) 键，如 Enter 键等。所有受输入保护或纯数字的主机字段都必须应作相应处理。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 3。	
数据字符串	击键字符串，最多 255 个。大写和小写 ASCII 字符按文字表示。功能键和移位功能键由助记符表示。请参阅 键盘助记符 (on page 118) 。	
Length	源数据字符串的长度。处于 EOT 方式时则覆盖。	
PS 位置	不适用	

返回参数

返回码	说明
0	击键已发送；状态正常。
1	程序未连接到主机会话。
2	传递给 EHLLAPI 的参数不正确。
4	主机会话正忙；无法发送所有击键。
5	目标会话的输入被禁止或拒绝；无法发送所有击键。
9	遇到系统错误。

有关使用此功能的注释

- 设置会话参数 (9)** 功能下的参数与此功能相关。它们是 AUTORESET/NORESET、STRLEN/STREOT、EOT=c、ESC=c 和 RETRY/NORETRY 会话选项。请参阅项 [1 \(on page 127\)](#) 和 [2 \(on page 127\)](#)、[9 \(on page 129\)](#) 和 [10 \(on page 129\)](#) 以及 [19 \(on page 132\)](#) 以获取更多信息。
- 键盘被锁定或正忙时，无法将击键发送至主机会话。可以使用 **等待 (4)** 功能检查此条件。
- 如果主机正忙，输入可能会被拒绝。
- 数据字符串的长度必须由缺省长度参数显式定义，但它可以由 **设置会话参数 (9)** 功能的 EOT=c 选项隐式定义。
在显式定义长度（请参阅项 1）时，必须计算应用程序传递的长度参数的值。对于此计算，对于复合击键（如 @E）允许 2 个字节，对于复合击键（如 @A@C）允许 4 个字节。
- 要发送特殊控制键，请使用复合字符编码方案。在此编码方案中，一个击键由两到四个 ASCII 字符组成的序列表示。第一个和第三个字符始终是转义字符。第二个和第四个字符始终是键代码。
要发送序列 LOGON ABCDE，后跟 Enter 键，需要对字符串 LOGON ABCDE@E 进行编码。这些键代码的完整列表采用 [键盘助记符 \(on page 118\)](#) 表示。
这种复合编码技术允许所有必要的击键代码的 ASCII 字符串表示，而无需使用复杂的十六进制键代码。

缺省转义字符为 @。使用**设置会话参数** (9) 功能的 ESC=c 选项，可以将转义字符的值更改为任何其他字符。

- 需要更高级别性能的用户应使用**将字符串复制到字段** (33) 或 **将字符串复制到表示空间** (15) 功能，而不是使用**发送键** (3) 功能发送击键。但请记住，只有**发送键** (3) 功能可以发送特殊控制键。
- 请参阅 **设置会话参数 (9) (on page 125)** 会话选项 **10 (on page 129)** (NORESET 选项)，以提高此功能的性能。

除非需要 NORESET，否则重置助记符将作为前缀添加到击键字符串中。因此，将重置除输入禁止之外的所有可重置状态。

NORESET 选项与**重置系统** (21) 功能不同。

- 包括 AID 键在内的击键字符串通过多个路径发送到主机。每个路径都发送第一个 AID 键之前的字符串（或包括 AID 键）。EHLLAPI 会调整字符串长度和每个路径的起始位置。对于主机应用程序，AID 键进程可能会丢失任何击键。因此，不应发送包含复数 AID 键的击键列表。
- 在 @P (打印) 或 @A@T (打印表示空间) 进程中，将拒绝所有更新表示空间的请求。如果表示空间正忙或在打印请求期间出现中断请求，助记符 @A@R (设备重置 - 取消以打印表示空间) 将取消请求并重置状态。

键盘助记符

键盘助记符提供标识工作站中键盘特殊功能键的 ASCII 字符。缩写代码使特殊键的助记符容易记住。字母键代码用于最常用的键。例如，**Clear** 键为 C，**Tab** 键为 T。

表 7: 含大写字母字符的助记符 (on page 118) 使用大写字母字符显示助记符:

表 7. 含大写字母字符的助记符

助记符	意义	3270	5250	VT
@B	左 Tab 键	是	是	否
@C	Clear	是	是	否
@D	Delete	是	是	否
@E	Enter	是	是	否
@F	擦除 EOF	是	是	否
@H	帮助	编号	是	否
@I	插入	是	是	否
@J	跳转 (设置焦点)	是	是	否
@L	光标左移	是	是	是
@N	换行	是	是	是
@O	空格键	是	是	是
@P	打印	是	是	是
@R	重置	是	是	否
@T	右 Tab 键	是	是	是
@U	光标上移	是	是	是
@V	光标下移	是	是	是
@Z	光标右移	是	是	是

表 8: 含数字或小写字母的助记符 (on page 119) 使用数字或小写字母字符显示助记符。

表 8. 含数字或小写字母的助记符

助记符	意义	3270	5250	VT
@0	主页	是	是	否
@1	PF1/F1	是	是	否
@2	PF2/F2	是	是	否
@3	PF3/F3	是	是	否
@4	PF4/F4	是	是	否
@5	PF5/F5	是	是	否
@6	PF6/F6	是	是	是
@7	PF7/F7	是	是	是
@8	PF8/F8	是	是	是
@9	PF9/F9	是	是	是
@a	PF10/F10	是	是	是
@b	PF11/F11	是	是	是
@c	PF12/F12	是	是	是
@d	PF13	是	是	是
@e	PF14	是	是	是
@f	PF15	是	是	是
@g	PF16	是	是	是
@h	PF17	是	是	是
@i	PF18	是	是	是
@j	PF19	是	是	是
@k	PF20	是	是	是
@l	PF21	是	是	否
@m	PF22	是	是	否
@n	PF23	是	是	否
@o	PF24	是	是	否
@q	End 键	是	是	否
@u	Page Up 键	编号	是	否
@v	Page Down 键	编号	是	否
@x	PA1	是	是	否
@y	PA2	是	是	否
@z	PA3	是	是	否

表 9: 含 @A 和 @大写字母字符的助记符 (on page 119) 使用组合 @A 和 @大写字母 (A-Z) 键显示助记符。

表 9. 含 @A 和 @大写字母字符的助记符

助记符	意义	3270	5250	VT
@A@C	测试	编号	是	否

表 9. 含 @A 和 @大写字母字符的助记符 (续)

助记符	意义	3270	5250	VT
@A@D	字词删除	是	是	否
@A@E	字段退出	是	是	否
@A@F	擦除输入	是	是	否
@A@H	系统请求	是	是	否
@A@I	插入切换	是	是	否
@A@J	光标选择	是	是	否
@A@L	快速光标左移	是	是	否
@A@Q	注意	是	是	否
@A@R	设备取消 (取消打印 表示空间)	是	是	否
@A@T	打印表示空间	是	是	是
@A@U	快速光标上移	是	是	否
@A@V	快速光标下移	是	是	否
@A@Z	快速光标右移	是	是	否

表 10: 含 @A 和 @小写字母字符的助记符 (on page 120) 使用组合 @A 和 @数字或 @A 和 @小写字母 (a-z) 键显示助记符。

表 10. 含 @A 和 @小写字母字符的助记符

助记符	意义	3270	5250	VT
@A@9	反相显示	是	是	否
@A@b	下划线	是	否	编号
@A@c	重置反相显示	是	否	编号
@A@d	红色	是	否	编号
@A@e	粉红色	是	否	编号
@A@f	绿色	是	否	编号
@A@g	黄色	是	否	编号
@A@h	蓝色	是	否	编号
@A@i	Turquoise	是	否	编号
@A@j	白色	是	否	编号
@A@l	重置主机颜色	是	否	编号
@A@t	打印 (个人计算机)	是	是	否
@A@y	正向字词制表符	是	是	否
@A@z	反向字词制表符	是	是	否

表 11: 带 @A 和 @字母数字 (特殊) 字符的助记符 (on page 121) 使用组合 @A 和 @特殊字符显示助记符。

表 11. 带 @A 和 @ 字母数字 (特殊) 字符的助记符

助记符	意义	3270	5250	VT
@A@-	字段 -	编号	是	否
@A@+	字段 +	编号	是	否
@A@<	记录退格键	编号	是	否

表 12: 带 @S (Shift)、@W (编辑) 和 @字母字符的助记符 (on page 121) 使用组合 @S、@W 和 @小写字母显示助记符。

表 12. 带 @S (Shift)、@W (编辑) 和 @字母字符的助记符

助记符	意义	3270	5250	VT
@S@E	主机上的打印表示空间	编号	是	否
@S@x	Dup	是	是	否
@S@y	字段标记	是	是	否
@W@C	编辑复制	是	是	是
@W@D	编辑清除	是	是	是
@W@E	编辑复制附加	是	是	是
@W@L	编辑复制链接	是	是	是
@W@N	编辑粘贴下一个	是	是	是
@W@V	编辑粘贴	是	是	是
@W@X	编辑剪切	是	是	是
@W@Z	编辑撤消	是	是	是



注： @W 编辑助记符仅在增强方式下的 EHLLAPI 函数中受支持。请参阅 [EHLLAPI 功能摘要 \(on page 28\)](#) 下的“开始击键拦截”功能。

仅限 VT： 表 13: 使用 @M、@Q 和 @小写字母的助记符 (仅适用于 VT) (on page 121) 使用组合 @M 和 @数字或 @小写字母 (a-z) 显示助记符

表 13. 使用 @M、@Q 和 @小写字母的助记符 (仅适用于 VT)

助记符	意义	3270	5250	VT
@M@0	VT 数字小键盘 0	编号	编号	是
@M@1	VT 数字小键盘 1	编号	编号	是
@M@2	VT 数字小键盘 2	编号	编号	是
@M@3	VT 数字小键盘 3	编号	编号	是
@M@4	VT 数字小键盘 4	编号	编号	是
@M@5	VT 数字小键盘 5	编号	编号	是
@M@6	VT 数字小键盘 6	编号	编号	是
@M@7	VT 数字小键盘 7	编号	编号	是
@M@8	VT 数字小键盘 8	编号	编号	是

表 13. 使用 @M、@Q 和 @小写字母的助记符 (仅适用于 VT) (续)

助记符	意义	3270	5250	VT
@M@9	VT 数字小键盘 9	编号	编号	是
@M@-	VT 数字填充 -	编号	编号	是
@M@,	VT 数字填充 ,	编号	编号	是
@M@.	VT 数字填充 .	编号	编号	是
@M@e	VT 数字小键盘 Enter	编号	编号	是
@M@f	VT 编辑查找	编号	编号	是
@M@i	VT 编辑插入	编号	编号	是
@M@r	VT 编辑删除	编号	编号	是
@M@s	VT 编辑选择	编号	编号	是
@M@p	VT 编辑上一个屏幕	编号	编号	是
@M@n	VT 编辑下一个屏幕	编号	编号	是
@M@a	VT PF1	编号	编号	是
@M@b	VT PF2	编号	编号	是
@M@c	VT PF3	编号	编号	是
@M@d	VT PF4	编号	编号	是
@M@h	VT 保持屏幕	编号	编号	是
@M@(空格)	控制代码 NUL	编号	编号	是
@M@A	控制代码 SOH	编号	编号	是
@M@B	控制代码 STX	编号	编号	是
@M@C	控制代码 ETX	编号	编号	是
@M@D	控制代码 EOT	编号	编号	是
@M@E	控制代码 ENQ	编号	编号	是
@M@F	控制代码 ACK	编号	编号	是
@M@G	控制代码 BEL	编号	编号	是
@M@H	控制代码 BS	编号	编号	是
@M@I	控制代码 HT	编号	编号	是
@M@J	控制代码 LF	编号	编号	是
@M@K	控制代码 VT	编号	编号	是
@M@L	控制代码 FF	编号	编号	是
@M@M	控制代码 CR	编号	编号	是
@M@N	控制代码 SO	编号	编号	是
@M@O	控制代码 SI	编号	编号	是
@M@P	控制代码 DLE	编号	编号	是
@M@Q	控制代码 DC1	编号	编号	是
@M@R	控制代码 DC2	编号	编号	是
@M@S	控制代码 DC3	编号	编号	是
@M@T	控制代码 DC4	编号	编号	是

表 13. 使用 @M、@Q 和 @小写字母的助记符 (仅适用于 VT) (续)

助记符	意义	3270	5250	VT
@M@U	控制代码 NAK	编号	编号	是
@M@V	控制代码 SYN	编号	编号	是
@M@W	控制代码 ETB	编号	编号	是
@M@X	控制代码 CAN	编号	编号	是
@M@Y	控制代码 EM	编号	编号	是
@M@Z	控制代码 SUB	编号	编号	是
@M@u	控制代码 ESC	编号	编号	是
@M@v	控制代码 FS	编号	编号	是
@M@w	控制代码 GS	编号	编号	是
@M@x	控制代码 RS	编号	编号	是
@M@y	控制代码 US	编号	编号	是
@M@z	控制代码 DEL	编号	编号	是
@Q@A	VT 用户定义键 6	编号	编号	是
@Q@B	VT 用户定义键 7	编号	编号	是
@Q@C	VT 用户定义键 8	编号	编号	是
@Q@D	VT 用户定义键 9	编号	编号	是
@Q@E	VT 用户定义键 10	编号	编号	是
@Q@F	VT 用户定义键 11	编号	编号	是
@Q@G	VT 用户定义键 12	编号	编号	是
@Q@H	VT 用户定义键 13	编号	编号	是
@Q@I	VT 用户定义键 14	编号	编号	是
@Q@J	VT 用户定义键 15	编号	编号	是
@Q@K	VT 用户定义键 16	编号	编号	是
@Q@L	VT 用户定义键 17	编号	编号	是
@Q@M	VT 用户定义键 18	编号	编号	是
@Q@N	VT 用户定义键 19	编号	编号	是
@Q@O	VT 用户定义键 20	编号	编号	是
@Q@a	VT 退格制表	编号	编号	是
@Q@r	VT 清除页面	编号	编号	是
@Q@s	VT 编辑	编号	编号	是

下表显示了使用特殊字符的助记符。

表 14. 含特殊字符键的助记符

助记符	意义	3270	5250	VT
@@	@	是	是	是

表 14. 含特殊字符键的助记符 (续)

助记符	意义	3270	5250	VT
@\$	替代光标 (仅限 Presentation Manager® 界面)	是	是	是
@<	Backspace	是	是	是

以下字符键将按原样解释。

a-z	!	'	<	}
A-Z	\$	(>	[
0-9	%)	/]
~	&	*	:	
#	"	+	:	{

设置光标 (40)

3270	5250	VT
是	是	是

设置光标功能用于设置光标在主机表示空间内的位置。使用**设置光标**功能之前，必须将工作站应用程序连接到主机表示空间。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 40	
数据字符串	不适用	
Length	不适用	
PS 位置	所连接的主机表示空间中所需的光标位置	

返回参数

返回码	说明
0	光标已成功定位在指定位置。
1	程序未连接到主机会话。
4	会话正忙。

返回码	说明
7	指定的游标位置小于 1 或大于连接的主机表示空间的大小。
9	发生了系统错误。

设置会话参数 (9)

3270	5250	VT
是	是	是

设置会话参数功能让您可更改 EHLLAPI 中所有会话的某些缺省会话选项。装入 EHLLAPI 后，会话选项的缺省设置由 [会话选项 \(on page 127\)](#) 中显示的表中带下划线条目指示。可以通过在调用数据字符串中包括所需的选项来更改任何、部分或所有这些设置，如下所述。指定的设置将一直有效，直到：

- 由指定新值的后续 **设置会话参数 (9)** 功能更改。
- **重置系统 (21)** 功能已执行。
- EHLLAPI 应用程序已终止。

下表列出了受会话选项影响的 EHLLAPI 功能。表中未列出的功能不受任何会话选项的影响。影响每个功能的会话选项由“查看项”列中的相应条目表示。这些条目将索引到 [调用参数 \(on page 127\)](#) 后面的列表。

函数编号	功能名称	请参阅项
1	<u>连接表示空间</u>	11 (on page 130) 、 21 (on page 132) 、 22 (on page 133)
3	<u>发送键</u>	1 (on page 127) 、 2 (on page 127) 、 9 (on page 129) 、 10 (on page 129) 、 19 (on page 132)
4	<u>等待</u>	12 (on page 130)
5	<u>复制表示空间</u>	5 (on page 128) 、 13 (on page 130) 、 14 (on page 131) 、 15 (on page 131) 、 17 (on page 131) 、 20 (on page 132)
6	<u>搜索表示空间</u>	1 (on page 127) 、 2 (on page 127) 、 3 (on page 127) 、 4 (on page 127)
8	<u>将表示空间复制到字符串</u>	5 (on page 128) 、 13 (on page 130) 、 14 (on page 131) 、 15 (on page 131) 、 17 (on page 131) 、 20 (on page 132)
10	<u>查询会话</u>	16 (on page 131) 、 20 (on page 132)

函数编号	功能名称	请参阅项
15	将字符串复制到表示空间	1 (on page 127)、2 (on page 127)、13 (on page 130)、14 (on page 131)、18 (on page 131)、20 (on page 132)
18	暂停	6 (on page 128)
30	搜索字段	1 (on page 127)、2 (on page 127)、3 (on page 127)、4 (on page 127)、20 (on page 132)
33	将字符串复制到字段	1 (on page 127)、2 (on page 127)、13 (on page 130)、14 (on page 131)、18 (on page 131)、20 (on page 132)
34	将字段复制到字符串	5 (on page 128)、13 (on page 130)、14 (on page 131)、17 (on page 131)、20 (on page 132)
35	将表示空间复制到剪贴板	5 (on page 128)、13 (on page 130)、14 (on page 131)、17 (on page 131)、20 (on page 132)
36	将剪贴板粘贴到表示空间	1 (on page 127)、2 (on page 127)、13 (on page 130)、14 (on page 131)、18 (on page 131)、20 (on page 132)
51	获取键	9 (on page 129)、12 (on page 130)
90	发送文件	1 (on page 127)、2 (on page 127)、7 (on page 128)、8 (on page 128)
91	接收文件	1 (on page 127)、2 (on page 127)、7 (on page 128)、8 (on page 128)
101	连接窗口服务	21 (on page 132)、22 (on page 133)

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 9。	
数据字符串	包含要更改的会话选项的所需值的字符串。数据字符串可以包含 会话选项 (on page 127) 表中的任何值。这些值应放在数据字符串行上，用逗号或空格分隔。参数集将根据它们所影响的功能进行说明。	
Length	源数据字符串的显式长度（不允许使用 STREOT 选项）。	
PS 位置	不适用。	

会话选项

以下几个表显示了会话选项。缺省值是 underlined。

1. 下表中的值确定如何为以下功能定义数据字符串长度：**发送键** (3)、**搜索表示空间** (6)、**将字符串复制到表示空间** (15)、**搜索字段** (30)、**将字符串复制到字段** (33)、**发送文件** (90) 和 **接收文件** (91)。

价值	说明
<u>STRLEN</u>	为所有字符串传递显式长度。
<u>STREOT</u>	长度未显式编码。调用（源）数据字符串以 EOT 字符结束。

2. 下表中的语句用于指定在以下 EHLLAPI 功能的调用（源）数据字符串中用作文本结束 (EOT) 定界符的字符：**发送键** (3)、**搜索表示空间** (6)、**将字符串复制到表示空间** (15)、**搜索字段** (30)、**将字符串复制到字段** (33)、**发送文件** (90) 和 **接收文件** (91)。

价值	说明
<u>EOT=c</u>	允许您指定字符串终止符的 EOT 字符（在 STREOT 方式下）。二进制零是缺省值。请勿在等号后面留空。

要有效，c 必须以 1 字节字符串文字字符的形式输入，并且前面没有空格。仅当 STREOT 选项（见项 1）生效时，此语句指定的 EOT 字符才会用于确定调用数据字符串的长度。

3. 下表中的值影响**搜索表示空间** (6) 和**搜索字段** (30) 搜索功能。

价值	说明
<u>SRCHALL</u>	搜索表示空间 (6) 功能和 搜索字段 (30) 功能会扫描整个主机表示空间或字段。
<u>SRCHFROM</u>	搜索表示空间 (6) 功能和 搜索字段 (30) 功能会从指定的 PS 位置开始（适用于 <u>SRCHFRWD</u> ），或在指定的 PS 位置结束（适用于 <u>SRCHBKWD</u> ）。

4. 下表中的值影响**搜索表示空间** (6) 和**搜索字段** (30) 搜索功能。它们确定了搜索的方向。

价值	说明
<u>SRCHFRWD</u>	搜索表示空间 (6) 功能和 搜索字段 (30) 功能按升序方向执行。

价值	说明
SRCHBKWD	搜索表示空间 (6) 功能和 搜索字段 (30) 功能以降序方向执行。如果所请求字符串的第一个字符在为搜索指定的范围内开始，则满足搜索要求。

5. 下表中的值确定功能**复制表示空间** (5)、**将表示空间复制到字符串** (8) 和**将字段复制到字符串** (34) 的属性字节处理方式。

价值	说明
NOATTRB	将所有未知值转换为空白。
ATTRB	作为原始值传递回所有不具有 ASCII 等效值的代码。
NULLATTRB	将所有字段属性转换为 null 字符。

6. 下表中的值影响**暂停** (18) 功能。

价值	说明
FPAUSE	无论您在 暂停 (18) 功能中指定的时间长短，全时暂停都将持续。
IPAUSE	可中断暂停。执行 启动主机通知 (23) 功能后，主机事件满足暂停要求。

7. 下表中的值确定是否显示由文件传输功能**发送文件** (90) 和**接收文件** (91) 生成的消息。

价值	说明
NOQUIET	将显示 SEND 和 RECEIVE 消息。
QUIET	不显示 SEND 和 RECEIVE 消息。

8. 下表中的语句确定在执行文件传输功能**发送文件** (90) 和**接收文件** (91) 期间，Z and I Emulator for Windows EHLLAPI 将等待自动发出“取消”的时长。要有效，**c** 必须是大写字母 J-N，并且前面不能有空格。

价值	说明								
TIMEOUT=0	在 (大约) 20 秒延迟后自动发出“取消”。								
TIMEOUT=c	在指定的延迟之后自动发出“取消”。下表中的 1 字符指示符指示 Z and I Emulator for Windows 在发出“取消”本身之前应接受的 30 秒周期数。 <div style="text-align: center;"> 字符 值 (以分钟为单位) </div> <table style="margin-left: auto; margin-right: auto;"> <tr><td>1</td></tr> <tr><td>0.5</td></tr> <tr><td>2</td></tr> <tr><td>1.0</td></tr> <tr><td>5</td></tr> <tr><td>1.5</td></tr> <tr><td>6</td></tr> <tr><td>2.0</td></tr> </table>	1	0.5	2	1.0	5	1.5	6	2.0
1									
0.5									
2									
1.0									
5									
1.5									
6									
2.0									

价值	说明
	7
	2.5
	8
	3.0
	9
	3.5
	10
	4.0
	9
	4.5
	J
	5.0
	K
	5.5
	L
	6.0
	周一
	6.5
	否
	7.0

9. 下表中的语句用于定义击键助记符的转义字符。此会话选项影响功能**发送键** (3) 和**获取键** (51) 功能。c 的值必须以 1 字节文字字符串的形式输入，并且前面没有空格。

价值	说明
ESC=c	指定击键助记符的转义字符 (@ 为缺省值)。请勿在等号后面留空。空白不是有效的转义字符。

10. 下表中的值确定 EHLLAPI 是否自动在使用 **发送键** (3) 功能发送的字符串前面添加 reset。

价值	说明
AUTORESET	EHLLAPI 会尝试重置所有禁用条件，方法是在使用 发送键 (3) 功能发送的所有键字符串前面添加 reset 作为前缀。
NORESET	请勿执行 AUTORESET。

11. 下表中的值会影响**连接表示空间 (1)** 命令的运行方式。

价值	说明
CONLOG	在工作站会话和主机会话之间建立逻辑连接。在连接期间，不要跳至请求的表示空间。
CONPHYS	在工作站会话和主机会话之间建立物理连接。在连接期间，跳至请求的表示空间。


12. 下表中的值影响**等待 (4)** 功能和**获取键 (51)** 功能。对于每个值，都有两种不同的效果，每个功能一个。

价值	说明
TWAIT	对于 等待 (4) 功能，在 XCLOCK (X []) 或 XSYSTEM 超时之前最长等待一分钟。 对于 获取键 (51) 功能，在 EHLLAPI 应用程序拦截某个键（基于 开始击键拦截 (50) 功能下指定的选项的常规或 AID 键）后，请勿将控制权返回给该应用程序。
LWAIT	对于 等待 (4) 功能，等至 XCLOCK (X [])/XSYSTEM 清除。不建议使用此选项，因为在主机可用之前，控制权不会返回到应用程序。 对于 获取键 (51) 功能，在 EHLLAPI 应用程序拦截某个键（基于 开始击键拦截 (50) 功能下指定的选项的常规或 AID 键）后，请勿将控制权返回给该应用程序。
NWAIT	对于 等待 (4) 功能，检查状态并立即返回（无需等待）。 对于 获取键 (51) 功能，如果没有任何与 开始击键拦截 (50) 功能下指定的选项匹配的项在排队，则在第四个参数中返回返回码 25（击键不可用）。

 **注：** 建议使用 NWAIT。

13. 下表中的值影响**复制表示空间 (5)**、**将表示空间复制到字符串 (8)**、**将字符串复制到表示空间 (15)**、**将字符串复制到字段 (33)** 和**将字段复制到字符串 (34)**。扩展属性字节 (EAB) 包括扩展字符属性和扩展字段属性。

价值	说明
NOEAB	仅传递数据，无 EAB。
EAB	使用扩展属性字节传递表示空间数据。对于屏幕上显示的每个字符，将传递 2 个字节的数据。因此，必须预先分配一个缓冲区，该缓冲区的大小是表示空间大小的两倍；例如，对于 24 行 x 80 列表示空间，为 $2 \times 1920 = 3840$ 。 字符串的扩展属性可以报告为字段字节的属性，而不是字段中每个单独字符的属性。在这种情况下，要判断屏幕上的特定字符或字符集是否为下划线，请执行 CopyPStoString，指定字段属性字节（屏幕上显示的字段前面的字节）的位置，以获取适用于该字段中所有字符的 EAB 信息。

 **注：** 当使用 EHLLAPI 将 PS 复制到字符串时，将复制文本，并且该文本应对运算符不可见。使用 EHLLAPI 设置会话参数功能设置 NODISPLAY 选项，以确定是否存在隐藏的数据。这会导致 EHLLAPI 将不显示字段作为 null 返回。隐藏数据的另一个常见过程是将前景色和背景色设置为相同（例如黑



色)，以便显示文本，但对运算符不可见。应用程序检测此问题的唯一方法是使用 EAB 和 XLATE 会话参数，然后复制 PS。返回每个位置的前景色/背景色，并且可以确定哪些字符不可见。

14. 下表中的值影响**复制表示空间 (5)**、**将表示空间复制到字符串 (8)**、**将字符串复制到表示空间 (15)**、**将字符串复制到字段 (33)** 和**将字段复制到字符串 (34)**。

价值	说明
NOXLATE	EAB 未转换。
XLATE	EAB 已转换为 PC 彩色图形适配器 (CGA) 格式。

15. 下表中的值会影响**复制表示空间 (5)**、**将表示空间复制到字符串 (8)** 和**将表示空间复制到剪贴板 (35)** (如果指定了 NOATTRB 和 NOEAB) 。

价值	说明
BLANK	将所有未知值转换为 X'20'。
NOBLANK	将所有未知值转换为 X'00'。

缺省值为 BLANK。如果要将缺省值更改为 NOBLANK，请在位于 Z and I Emulator for Windows 用户类应用程序数据目录的 PCSWIN.INI 文件中添加以下语句：

```
[API]
NullToBlank=NO
```

16. 下表中的值会影响**查询会话 (10)** 返回的表示空间大小。

价值	说明
CFGSIZE	返回已连接表示空间的所配置大小。此选项将忽略主机对所配置大小的任何覆盖。
NOCFGSIZE	返回已连接表示空间的当前大小。

17. 下表中的值影响**复制表示空间 (5)**、**将表示空间复制到字符串 (8)**、**将字段复制到字符串 (34)** 和**将表示空间复制到剪贴板 (35)**。

价值	说明
DISPLAY	将表示空间中的不显示字段复制到目标缓冲区，方法与显示字段相同。当前应用程序正常运行。
NODISPLAY	请勿将表示空间中的不显示字段复制到目标缓冲区。将不显示字段作为 null 字符串复制到目标缓冲区。这样，应用程序就可以在表示窗口中显示复制的缓冲区，而不显示机密信息（如密码）。

18. 下表中的值影响**将字符串复制到表示空间 (15)**、**将字符串复制到字段 (33)** 和**将剪贴板粘贴到表示空间 (36)**。

价值	说明
NOPUTEAB	EAB 不包含在 将字符串复制到表示空间 或 将字符串复制到字段 的数据字符串中。
PUTEAB	EAB 与字符数据包含在 将字符串复制到表示空间 或 将字符串复制到字段 的数据字符串中。

此选项用于兼容 Communication Manager/2。对于 Communication Manager/2, EAB (或 EAD) 在**设置会话参数**中有效时, 在**将字符串复制到表示空间**或**将字符串复制到字段**中指定的数据字符串包含 EAB (或 EAD) 以及字符数据。然而, 对于之前的 Z and I Emulator for Windows, 在这些功能中指定的数据字符串只能由字符数据组成, 即使 EAB (或 EAD) 有效也是如此。但是, Z and I Emulator for Windows 允许数据字符串包含 EAB (或 EAD), 方法是设置 PUTEAB 来兼容 Communication Manager/2。

19. 下表中的值影响**发送键** (3) 功能。如果键盘被阻止或正在使用, 则不会处理击键。这些选项确定该功能是在 4 分钟超时之前尝试重新发送键, 还是功能在确定键盘被阻止或正在使用后立即返回。

价值	说明
RETRY	继续尝试发送击键, 直到发送击键或发生 4 分钟超时为止。
NORETRY	确定键盘被阻止或正在使用后立即返回。

20. 下表中的值影响**复制表示空间** (5)、**将表示空间复制到字符串** (8)、**将字符串复制到表示空间** (15)、**将字符串复制到字段** (33)、**将字段复制到字符串** (34)、**搜索字段** (30)、**查询会话**(10)、**将表示空间复制到剪贴板** (35) 和 **将剪贴板粘贴到表示空间** (36)。

价值	说明
EXTEND_PS	5250 仿真支持 24 行 x 80 列的表示空间。在某些情况下, Communication Manager 5250 仿真会显示第 25 行。当显示来自主机的错误消息或操作员选择 SysReq 键时, 就会出现这种情况。Z and I Emulator for Windows 在第 24 行显示第 25 行信息, 但是 EHLLAPI 通常会看到真正的第 24 行。通过 EXTEND_PS 选项, EHLLAPI 应用程序可以使用与 Communication Manager EHLLAPI 相同的接口, 并且在满足此条件时将扩展有效的表示空间。
NOEXTEND_PS	出现上述情况时, 不会扩展表示空间。这是缺省值。

21. 下表中的值影响**连接表示空间** (1) 和**连接窗口服务** (101) 功能。这些选项指定一个应用程序是否可以或将其共享与其使用另一个应用程序连接到的表示空间。每次调用**设置会话参数**时, 都只能指定以下一个值。

价值	说明
SUPER_WRITE	此应用程序允许其他允许共享和具有写入访问权的应用程序并发连接到相同的表示空间。发起应用程序执行管理类型功能, 但不会为共享表示空间的其他应用程序创建错误。
WRITE_SUPER	应用程序需要写入访问权, 并且仅允许管理应用程序并发连接到其表示空间。这是缺省值。
WRITE_WRITE	应用程序需要写入访问权, 并允许具有可预测行为的合作伙伴或其他应用程序共享表示空间。
WRITE_READ	应用程序需要写入访问权, 并允许执行只读函数的其他应用程序共享表示空间。还允许应用程序复制表示空间, 并照常执行其他只读操作。
WRITE_NONE	应用程序可独占使用表示空间。不允许其他应用程序共享表示空间, 包括管理应用程序。允许应用程序复制表示空间并照常执行只读操作。
READ_WRITE	应用程序仅需要读取访问权即可监控表示空间, 并允许执行读取或写入功能或者同时执行这两项功能的其他应用程序共享表示空间。还允许应用程序复制表示空间, 并照常执行其他只读操作。

22. 下表中的值允许具有表示空间共享需求的应用程序将共享限制为合作伙伴应用程序（为使用该应用程序而开发的应用程序）。

价值	说明
NOKEY	允许应用程序兼容未指定 KEY 参数的现有应用程序。
KEY\$nnnnnnnn	使用关键字限制对其支持的表示空间的共享访问。关键字的长度只能为 8 个字节。

返回参数

此功能会返回一个长度和一个返回码。

长度:

设置的有效会话参数数。

返回码:

定义了以下代码:

返回码	说明
0	已设置会话参数。
2	一个或多个参数无效。
9	遇到系统错误。

启动关闭拦截 (41)

3270	5250	VT
是	是	是

启动关闭拦截功能允许应用程序拦截用户从仿真器会话窗口中选择关闭选项时生成的关闭请求。此共揽会拦截关闭请求并将其丢弃，直到请求**停止关闭拦截** (43) 功能为止。

使用此功能后，应用程序可以使用**查询关闭拦截** (42) 功能来确定何时发生关闭请求。

前提条件调用

此函数没有任何前提条件调用。

调用参数

字节	定义
	标准接口 增强型接口

字节		定义
函数编号		必须为 41
数据字符串		请参阅下表
Length	5 或 6	必须为 12
PS 位置		不适用

数据字符串包含以下项。

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留。
4-5		EHLLAPI 会忽略这些位置中的数据。但是，如果迁移程序的数据位于这些位置，则不会导致错误。会接受此数据以提供与迁移应用程序的兼容性。
6	7	指定 M 可请求异步消息方式 (仅限 Windows)。
	6-8	保留。
2-3	9-12	在位置 5 (16 位为 6) 中指定 M，应设置接收消息的窗口的窗口句柄。消息是 RegisterWindowMessage (PCSHLL) 的返回值 (不等于 0)。

返回参数

此函数会返回数据字符串和返回码。

数据字符串:

如果未在位置 5 (标准接口为 6) 中指定异步消息方式，并且功能已成功完成，则返回以下数据字符串。

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-8	保留。
	9-12	EHLLAPI 返回事件对象地址的 4 字节值。应用程序可以等待此事件对象。(仅限 32 位)。

数据字符串:

如果在位置 5 (标准接口为 6) 中指定异步消息方式，并且功能已成功完成，则返回以下数据字符串。

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-8	保留
2-3	9-10	异步消息方式的任务标识



注： 如果用户选择关闭选项，应用程序窗口将收到一条消息。消息是 RegisterWindowMessage (PCSHLL) 的返回值。wParam 参数将包含此功能调用返回的任务标识。lParam 参数的 HIWORD 将包含返回码 26，该代码显示发生了关闭拦截，而 lParam 参数的 LOWORD 将包含功能编号 41。

返回码：

定义了以下代码：

返回码	说明
0	启动关闭拦截功能已成功。
1	指定的主机表示空间不正确。
2	发生参数错误。
9	发生了系统错误。
10	枚举程序不支持此功能。

有关使用此功能的注释

1. 返回启动请求函数时，返回的事件对象或信号量处于未发出信号状态。每次发生关闭请求时，事件对象都处于已发出信号状态。要接收多个关闭请求事件的通知，请在每次使用 **SetEvent** 或**查询关闭拦截** (42) 功能时，将事件对象置于已发出信息状态。
2. 使用此功能后，应用程序可以使用**查询关闭拦截** (42) 功能来确定何时发生关闭请求。应用程序可以等待返回的事件对象来确定事件发生的时间。
3. 这不是互斥调用。多个应用程序可以为同一个短会话标识请求此功能。
4. 如果没有应用程序拦截某个会话的关闭请求，则用户从仿真器操作对话框中选择的任何后续关闭请求都会导致为该会话请求正常停止。

启动通信通知 (80)

3270	5250	VT
是	是	是

启动通信通知功能会开始 EHLLAPI 应用程序可据以确定指定会话是否已连接到主机的流程。

使用此功能后，应用程序可以使用**查询通信事件** (81)，确定会话是已连接还是已断开连接。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	增强型接口
函数编号	必须为 80

数据字符串	预先分配 结构；请参阅下表
Length	16
PSPosition	不适用

调用数据结构包含这些元素

字节	定义
1	1 字符表示空间短名称 (PSID)
2 - 4	保留
5	以下某个值： <ul style="list-style-type: none"> • 当会话断开连接或连接到主机时，字符 C 要求发出通知。 • 字符 A 请求通知的异步方式。指定 A 时，位置 9-12 会返回事件对象的地址 (Windows)。字符 C 必须放置在位置 13。 • 字符 M 请求通知的异步消息方式。指定 M 时，必须将事件选择字符 C 放置在位置 13。
6 - 8	保留
9 - 12	在位置 5 中指定了 M 时，应设置用于接收消息的窗口的窗口句柄。消息是 RegisterWindowMessage (PCSHLL) 的返回值 (非零)。
13	如果位置 5 是 A 或 M，则应包含字符 C。
14 - 16	保留

数据字符串

如果在调用数据结构的位置 5 中指定了 A (异步方式)，并且已成功完成该函数，则返回以下数据字符串：

字节	定义
1	1 字符表示空间短名称 (PSID)
2 - 8	保留
9 - 12	EHLAPI 返回事件对象句柄的 4 字节二进制值。应用程序可以等待此事件对象。

如果在调用数据结构的位置 5 中指定了 M (异步消息方式)，并且已成功完成该函数，则返回以下数据字符串：

字节	定义
1	1 字符表示空间短名称 (PSID)
2 - 8	保留
9 - 10	异步消息方式的任务标识

当会话连接或断开连接时，应用程序窗口会收到一条消息。该消息是 RegisterWindow Message (PCSHLL) 的返回值。wParam 包含函数调用返回的任务标识。如果会话已连接到主机，则 lParam 的 HIWORD 包含 21；如果会话已断开连接，则包含 22。lParam 的 LOWORD 包含函数编号 80。

返回参数

返回码	定义
0	此功能已成功
1	指定的 PSID 不正确。
2	指定参数时出错
9	遇到系统错误

有关使用此功能的注释

1. 应用程序可以为多个主机会话发出此功能。**查询通信事件** (81) 功能可用于确定会话通信状态。
2. 如果应用程序选择异步选项，它可以使用 Windows SDK 调用 **WaitForSingleObject**，以等至会话通信状态已更改。
3. 事件对象最初处于未发出信号状态。每次发生事件时都会向其发出信号。要接收多个事件的通知，应用程序必须在每次发出信号时将事件对象置于未发出信号状态，方法是使用 Windows SDK 调用 **ResetEvent**，或使用功能 81 **查询通信事件**。
4. 将忽略同一应用程序使用相同选项多次调用此功能。
5. 这不是一个应用程序独有的。多个应用程序可以为同一会话标识请求此功能。

启动主机通知 (23)

3270	5250	VT
是	是	是

启动主机通知功能会开始 EHLLAPI 应用程序据以确定主机表示空间或 OIA 是否已更新的流程。

使用此功能后，应用程序可以使用**查询主机更新** (24) 功能来确定主机事件发生的时间。

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 23	
数据字符串	预先分配的字符串；请参阅下表	
Length	6 或 7 隐含	16
PS 位置	不适用	

调用数据字符串包含以下元素：

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 1 字符表示空间短名称 (PSID) • 空，表示主机连接的主机表示空间请求
	2-4	保留。
2	7	以下某个值： <ul style="list-style-type: none"> • 要求主机表示空间和 OIA 更新通知的字符 B。 • 仅要求 OIA 更新通知的字符 O。 • 仅要求主机表示空间更新通知的字符 P。 • 请求通知异步方式的字符 A 指定 A 时，位置 9-12 会返回事件对象的地址。事件选择字符 B、O 或 P 必须放置在位置 13。 • 请求通知的异步消息方式的字符 M。 指定 M 时，必须将事件选择字符 B、O 或 P 放置在位置 13（16 位为 7）。 <ul style="list-style-type: none"> • E 要求在打印机会话期间收到完成通知的字符 E。
	6-8	保留。
3-4	9-12	在位置 5（16 位为 2）中指定 M，应设置接收消息的窗口的窗口句柄。消息是 RegisterWindowMessage (PCSHLL) 的返回值（不等于 0）。
7	13	如果位置 5（16 位为 2）为 A 或 M，则为以下一个值： <ul style="list-style-type: none"> • 要求主机表示空间和 OIA 更新通知的字符 B • 仅要求 OIA 更新通知的字符 O • 仅要求主机表示更新通知的字符 P。
	14-16	保留。

返回参数

此函数会返回数据字符串和返回码。

数据字符串：

如果在位置 5 中指定了 A（通知的异步方式），并且已成功完成该功能，则返回以下数据字符串：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-8	保留。
	9-12	EHLAPI 返回事件对象地址的 4 字节值。应用程序可以等待此事件对象（仅限 32 位）。

数据字符串：

如果在位置 5 (标准接口为 2) 中指定 M (异步消息方式)，并且功能已成功完成，则返回以下数据字符串：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-8	保留
3-4	9-10	异步消息方式的任务标识



注： 如果更新了 OIA 或表示空间，应用程序窗口将收到一条消息。消息是 RegisterWindowMessage (PCSHLL) 的返回值。wParam 参数包含功能调用返回的任务标识。lParam 的 HIWORD 包含返回码 21 (显示 OIA 已更新)、22 (显示主机表示空间已更新) 或 23 (显示 OIA 和主机表示空间已更新)，lParam 参数的 LOWORD 包含功能编号 23。

返回码：

定义了以下代码：

返回码	定义
0	启动主机通知功能已成功。
1	指定的主机表示空间不正确。
2	指定参数时出错。
9	遇到系统错误。

有关使用此功能的注释

1. 应用程序可以为多个主机会话发出此功能。更新一个或多个主机会话 (PS、OIA 或两者) 时，**暂停 (18)** 功能会通知应用程序。**查询主机更新 (24)** 功能可用于确定是否已更新 PS、OIA 或两者。
2. 如果应用程序选择异步选项，它可以等待返回的事件对象或信号量确定主机事件发生的时间。
3. 事件对象或信号量最初处于未发出信号状态，每次发生相应事件时都会发出信号。要接收多个事件的通知，应用程序必须在每次使用 **ResetEvent** 或**查询主机更新 (24)** 功能向其发出信号时，将事件对象置于未发出信号状态。
4. 应用程序无法使用相同的选项多次请求“启动主机通知”。
5. 这不是互斥调用。多个应用程序可以为同一个短会话标识请求此功能。

启动击键拦截 (50)

3270	5250	VT
是	是	是

启动击键拦截功能允许工作站应用程序过滤终端操作员发送到会话的任何击键。调用此功能后，将拦截并保存击键，直至击键队列溢出，或直至调用**停止击键拦截** (53) 功能或**重置系统** (21) 功能。拦截的击键可能是：

- 通过**获取键** (51) 功能接收，并使用**发送键** (3) 功能发送到同一个或另一个会话
- 通过**拦截后状态** (52) 功能接受或拒绝
- 使用**发送键** (3) 功能替换为其他击键
- 用于触发其他进程

前提条件调用

此函数没有任何前提条件调用。

调用参数

	标准接口	增强型接口
函数编号	必须为 50	
数据字符串	请参阅下表	
Length	击键缓冲区大小 EHLLAPI 为此缓冲区分配至少 32 个字节。	
PS 位置	不适用	

调用数据字符串包含：

字节		定义
标准	增强	
1	1	以下某个值： <ul style="list-style-type: none"> • 特定的主机表示空间短名称 (PSID) • 空，表示主机连接的主机表示空间请求
	2-4	保留。
2	7	选项代码字符： <ul style="list-style-type: none"> • D (仅表示 AID 击键)。 • L (表示所有击键)。 • E (表示编辑键和所有击键 (仅在增强方式下可用)) • M, 用于通知的异步消息传递方式 (仅 Windows)。 指定 M 时，代码字符 D、L 或 E (增强方式) 必须放在位置 13 (16 位为 7) 。
	6-8	保留。

前提条件：键盘键必须映射到编辑功能，例如，Ctrl+C 映射到编辑复制功能。请参阅 [表 12: 带 @S \(Shift\)、@W \(编辑\) 和 @字母字符的助记符 \(on page 121\)](#) 以了解支持的编辑功能。

字节		定义
3-4	9-12	在位置 5 (16 位为 2) 中指定 M, 应设置接收消息的窗口的窗口句柄。消息是 RegisterWindowMessage (PCSHLL) 的返回值 (不等于 0)。
7	13	如果位置 5 (16 位为 2) 为 M, 则为以下一个值: <ul style="list-style-type: none"> • D (仅表示 AID 击键)。 • L (表示所有击键)。 • E (表示编辑键和所有击键)。(仅在增强方式下可用。)
	14-16	保留。

数据字符串:

如果在位置 5 (标准接口为 2) 中指定 M (异步消息方式), 并且功能已成功完成, 则返回以下数据字符串:

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-8	保留
3-4	9-10	异步消息方式的任务标识



注: 如果用户向会话发送击键, 应用程序窗口将收到一条消息。消息是 RegisterWindowMessge (PCSHLL) 的返回值。wParam 参数包含功能调用返回的任务标识。iParam 参数的 HIWORD 包含返回码 0, 表示功能已成功, iParam 参数的 LOWORD 包含功能编号 50。

返回参数

返回码	说明
0	开始击键拦截功能已成功。
1	指定的表示空间不正确。
2	指定的选项错误。
6	由于目标表示空间正忙, 此功能的执行被禁止。
9	遇到系统错误。正在使用释放。

有关使用此功能的注释

1. 如果**获取键** (51) 功能的返回码为 31, 则:

- 增加此功能的调用长度参数的值, 或者
- 更频繁地执行**获取键** (51) 功能。

拦截的击键在缓冲区中占用 3 个字节。下一个拦截的击键将放置在相邻的 3 个字节中。当**获取键** (51) 功能检索击键 (先进先出或 FIFO) 时, 此功能占用的 3 个字节将可用于另一个击键。通过增加缓冲区的大小或从缓冲区中检索击键的速率, 可以消除缓冲区溢出。

在 PC/3270 中，消除返回码 31 的另一种方法是在恢复方式下操作 PC/3270 仿真器。

2. 如果提供了选项代码 D，EHLLAPI 会将拦截的非 AID 键写入其最初预期的表示空间，并仅将 AID 键返回到应用程序。
3. 在退出 EHLLAPI 应用程序之前，调用**停止击键拦截** (53) 功能。否则，击键拦截将保持已启用状态，并产生不可预测的结果。

开始播放宏 (110)

3270	5250	VT
是	是	是

开始播放宏功能会调用宏。宏将在连接的会话中执行。



注： 此宏必须存在于 Z and I Emulator for Windows 用户类应用程序数据目录中，并且不应在宏名的函数调用中指定扩展名。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口
函数编号	必须为 110
数据字符串	请参阅下表
Length	宏名的长度，加上 3
PS 位置	不适用

字节	定义
标准	增强
1 - 2	保留
3-n	以 null 结束的宏名

返回参数

返回码	说明
0	开始播放宏 功能已成功。
1	程序未连接到主机会话。
2	指定参数时出错。
9	遇到系统错误。

停止关闭拦截 (43)

3270	5250	VT
是	是	是

停止关闭拦截功能允许应用程序关闭**启动关闭拦截** (41) 功能。应用程序发出**停止关闭拦截**功能后，后续的关闭请求将导致向逻辑终端会话发送正常停止。

前提条件调用

启动关闭拦截 (41)

调用参数

	标准接口	增强型接口
函数编号	必须为 43	
数据字符串	主机表示空间的 1 字符短会话标识	
Length	1	必须为 4
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强
1	1 字符表示空间短名称 (PSID)
	2-4 保留

返回参数

返回码	说明
0	停止关闭拦截 功能已成功。
1	指定的主机表示空间不正确。
2	指定参数时出错。
8	未发出以前的 启动关闭拦截 (41) 功能。
9	发生了系统错误。
12	会话已停止。

停止通信通知 (82)

3270	5250	VT
是	是	是

停止通信通知功能会禁用**查询通信事件** (81) 功能，以确定指定会话中是否发生了任何通信事件。

前提条件调用

启动通信通知 (80)

调用参数

	增强型接口
函数编号	必须为 82
数据字符串	主机表示空间的 1 字符短名称，或者表示主机连接的表示空间的更新请求的空白或 null
Length	4 是隐含的
PSPosition	不适用

调用数据结构包含这些元素：

字节	定义
1	1 字符表示空间短名称 (PSID)
2 - 4	保留

返回参数

返回码	定义
0	此功能已成功
1	指定的 PSID 不正确。
8	未为 PSID 调用之前的 启动通信通知 (80) 功能调用
9	遇到系统错误

停止主机通知 (25)

3270	5250	VT
是	是	是

停止主机通知功能会禁用**查询主机更新** (24) 功能以确定是否主机表示空间或 OIA 已更新。此功能还会阻止主机事件影响**暂停** (18) 功能。

前提条件调用

启动主机通知 (23)

调用参数

	标准接口	增强型接口
函数编号	必须为 121	
数据字符串	请参阅以下注释	
Length	1 是隐含的	必须为 4
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强
1	1 字符表示空间短名称 (PSID)
	2-4 保留



注： 目标表示空间标识的 1 个字符短名称，或表示主机连接的表示空间请求的空白或 null。

返回参数

返回码	定义
0	停止主机通知功能已成功。
1	指定的主机表示空间不正确。
8	未发出以前的启动主机通知 (23) 功能。
9	遇到系统错误。

停止击键拦截 (53)

3270	5250	VT
是	是	是

停止击键拦截功能会结束应用程序拦截击键的功能。

前提条件调用

启动击键拦截 (50)

调用参数

	标准接口	增强型接口
函数编号	必须为 53	

	标准接口	增强型接口
数据字符串	目标表示空间 (PSID) 的短名称	
Length	1 是隐含的	必须为 4
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留

返回参数

返回码	说明
0	停止击键拦截功能已成功。
1	指定的表示空间不正确。
8	未为此表示空间调用先前的 开始击键拦截 (50) 功能。
9	遇到系统错误。

等待 (4)

3270	5250	VT
是	是	是

等待功能会检查主机连接的表示空间的状态。如果会话正在等待主机响应 (由 XCLOCK (X II) 或 XSYSTEM 指示) , 则**等待**功能会导致 EHLLAPI 最长等待 1 分钟, 以查看条件是否已清除。

前提条件调用

连接表示空间 (1)

调用参数

	标准接口	增强型接口
函数编号	必须为 4	
数据字符串	不适用	
Length	不适用	
PS 位置	不适用	

返回参数

返回码	定义
0	键盘已解锁并可用于输入。
1	您的应用程序未连接到有效会话。
4	仍处于 XCLOCK (X []) 或 XSYSTEM 时超时。
5	键盘已锁定。
9	遇到系统错误。

有关使用此功能的注释

1. **等待**功能用于为主机请求（如**发送键** (3) 功能发出的请求）提供完成所需的时间。使用**设置会话参数** (9) 功能，可以请求 `TWAIT`、`LWAIT` 或 `NWAIT` 选项。请参阅项 [12 \(on page 130\)](#)。
2. 可以使用此功能查看主机 OIA 是否被禁止。
3. 解锁键盘的主机满足**等待**功能要求。因此，返回码 0 并不一定意味着事务已完成。要验证事务是否已完成，应使用**搜索字段** (30) 功能或**搜索表示空间** (6) 功能以及**等待**功能来查找预期的关键字提示。

窗口状态 (104)

3270	5250	VT
是	是	是

窗口状态功能允许应用程序查询或更改窗口的表示空间大小、位置或可见状态。

前提条件调用


连接窗口服务 (101)

调用参数

	标准接口	增强型接口
函数编号	必须为 104	
数据字符串	请参阅下表	
Length	16 或 20	24 或 28
PS 位置	不适用	

调用数据字符串可以包含：

字节	定义
标准	增强
1	1 字符表示空间短名称 (PSID)

字节		定义
	2-4	保留
2	7	<p>请求选项值，选择以下值之一：</p> <ul style="list-style-type: none"> • X'01' 表示设置状态 <p> 注： 将会话就地嵌入到复合 OLE 文档时，此函数的设置形式（字节 5 = X'01'）总是返回 0，但没有影响。</p> <ul style="list-style-type: none"> • X'02' 表示查询状态 • X'03' 表示查询扩展状态
	6	保留

如果请求选项值为 X'01'（设置状态）：

字节		定义
标准	增强	
3-4	7-8	<p>如果请求选项为 1（设置状态），则为包含状态设置位的 16 位或 32 位字。如果请求选项为设置状态，则以下代码是有效的返回值：</p> <p>X'0001' 更改窗口大小。（对最小化、最大化、恢复或移动无效。）</p> <p>X'0002' 移动窗口。（对最小化、最大化、大小或恢复无效。）</p> <p>X'0004' ZORDER 窗口替换。</p> <p>X'0008' 将窗口设置为可见。</p> <p>X'0010' 将窗口设置为不可见。</p> <p>X'0080' 激活窗口。（除非指定 ZORDER，否则将焦点设置为窗口并将其放在前景。在这种情况下，将使用 ZORDER 放置。）</p> <p>X'0100' 停用窗口。（除非还指定 ZORDER，否则停用窗口并将窗口设为底部窗口。在这种情况下，将使用 ZORDER 放置。）</p> <p>X'0400' 将窗口设置为最小化。（对最大化、恢复、大小或移动无效。）</p>

字节		定义
		<p>X'0800'</p> <p>将窗口设置为最大化。（对最小化、恢复、大小或移动无效。）</p> <p>X'1000'</p> <p>恢复窗口。（对最小化、最大化、大小或移动无效。）</p>
5-6	9-12	包含 X 窗口位置坐标的 16 位或 32 位字。（如果未设置移动选项，则忽略。）
7-8	13-16	包含 Y 窗口位置坐标的 16 位或 32 位字。（如果未设置移动选项，则忽略。）
9-10	17-20	包含 X 窗口大小（以设备单位表示）的 16 位或 32 位字。（如果未设置大小选项，则忽略。）
11-12	21-24	包含 Y 窗口大小（以设备单位表示）的 16 位或 32 位字。（如果未设置大小选项，则忽略。）
13-16	25-28	<p>包含相对窗口放置的窗口句柄的 16 位或 32 位字。这两个字仅用于设置选项。（如果未设置 ZORDER 选项，则忽略。）有效值如下所示：</p> <p>X'00000003' 放在所有同级窗口前面。X'00000004' 放在所有同级窗口后面。</p>

如果请求选项值为 X'02'（查询状态）：

字节		定义
标准	增强	
3-4	7-8	<p>如果请求选项为 2（查询状态），则为包含 X'0000' 的 16 位或 32 位字。如果请求选项是查询状态，则以下代码是可能的返回值。可能会有多个状态。</p> <p>X'0008'</p> <p>窗口可见。</p> <p>X'0010'</p> <p>窗口不可见。</p> <p>X'0080'</p> <p>窗口已激活。</p> <p>X'0100'</p> <p>窗口已停用。</p> <p>X'0400'</p> <p>窗口已最小化。</p>

字节		定义
		X'0800' 窗口已最大化。
5-6	9-12	包含 X 窗口位置坐标的 16 位或 32 位字。（如果未设置移动选项，则忽略。）
7-8	13-16	包含 Y 窗口位置坐标的 16 位或 32 位字。（如果未设置移动选项，则忽略。）
9-10	17-20	包含 X 窗口大小（以设备单位表示）的 16 位或 32 位字。（如果未设置大小选项，则忽略。）
11-12	21-24	包含 Y 窗口大小（以设备单位表示）的 16 位或 32 位字。（如果未设置大小选项，则忽略。）
13-16	25-28	包含相对窗口放置的窗口句柄的 16 位或 32 位字。这两个字仅用于设置选项。（如果未设置 ZORDER 选项，则忽略。）有效值如下所示： X'00000003' 放在所有同级窗口前面。X'00000004' 放在所有同级窗口后面。

如果请求选项值为 X'03'（查询扩展状态）：

字节		定义
标准	增强	
3-4	7-8	如果请求选项为 3（查询扩展状态），则为包含 X'0000' 的 16 位或 32 位字。如果请求选项是查询扩展状态，则以下代码是可能的返回值。可能会有多个状态。 X'0008' 窗口可见。 X'0010' 窗口不可见。 X'0080' 窗口已激活。 X'0100' 窗口已停用。 X'0400' 窗口已最小化。 X'0800' 窗口已最大化。
5-6	9-10	包含 X 尺寸中当前字体大小的 16 位或 32 位字。该值以屏幕像素表示。
7-8	11-12	包含 Y 尺寸中当前字体大小的 16 位或 32 位字。该值以屏幕像素表示。

字节		定义
9-12	13-16	保留。此值始终为零。
13-14	17-18	包含表示空间第一个可见字符的行号的 16 位或 32 位字。此值通常为 1，除非“固定大小字体”选项生效，并且窗口已调整大小以隐藏部分表示空间。
15-16	19-20	包含表示空间第一个可见字符的列号的 16 位或 32 位字。
17-20	21-24	包含会话的表示空间窗口句柄的 16 位或 32 位字。

返回参数

返回码	说明
0	窗口状态功能已成功。
1	表示空间无效或未连接。
2	指定的选项错误。
9	发生了系统错误。
12	会话已停止。

有关使用此功能的注释

逻辑终端 (LT) 窗口使用字符单元格。调整 LT 窗口大小时，LT 将舍入数字，以防止字符单元格被截断。请求的大小和位置可能与请求的大小和位置略有不同。按照带查询选项的设置选项来确定最终 Presentation Manager® 窗口的位置和大小。所有 x 和 y 坐标位置和大小均以像素为单位。

写入结构化字段 (127)

3270	5250	VT
是	否	编号

写入结构化字段功能允许应用程序将结构化字段数据写入主机应用程序。如果调用指定 S（表示同步），则在完成**写入结构化字段**功能之前，应用程序不会获得控制权。如果调用指定 A（表示异步），则应用程序会在调用后立即获得控制权。如果调用指定 M，则应用程序会在调用后立即获得控制权。应用程序可能会等待消息。在任何情况下（S、A 或 M），应用程序都提供将数据放置到主机的缓冲区地址。

要成功地异步完成此功能，以下陈述将适用：

参数列表中的返回码字段可能不包含所请求 I/O 的结果。如果返回码不是 0，则请求失败。应用程序必须根据返回码采取适当的措施。

如果此请求的返回码为 0，则应用程序必须使用此功能调用返回的请求标识，发出**获取请求完成**功能调用，以确定与请求标识关联的功能的完成结果。**获取请求完成**功能调用返回以下信息：

1. 功能请求标识
2. 来自异步请求的数据字符串的地址

- 3. 数据字符串的长度
- 4. 已完成功能的返回码

前提条件调用

结构化字段的连接 (120) 分配通信缓冲区 (123)

调用参数

	标准接口	增强型接口
函数编号	必须为 127	
数据字符串	请参阅下表	
Length	8、10 或 14	必须为 20
PS 位置	不适用	

调用数据字符串可以包含：

字节		定义
标准	增强	
1	1	1 字符表示空间短名称 (PSID)
	2-4	保留。
2	7	S 或 A 或 M S = 同步。在满足读取要求之前，不会将控制权返回给应用程序。 A = 异步。将控制权立即返回给应用程序，可以等待事件对象。 M = 异步。将控制权立即返回给应用程序，可以等待消息。
	6	保留。
3-4	7-8	2 字节目标/来源标识。
5-8	9-12	要从中写入数据的缓冲区的 4 字节地址。必须使用 分配通信缓冲区 (123) 功能获得缓冲区。
9-10	13-16	保留。
11-12	17-20	如果在位置 5 (16 位为 2) 中指定“M”，则应设置接收消息的窗口的窗口句柄。消息是 RegisterWindowMessage (“PCSHLL”) 的返回值 (不等于 0)。
13-14		这些位置中的数据将被 EHLLAPI 忽略。但是，如果迁移程序的数据位于这些位置，则不会导致错误。会接受此数据以提供与迁移应用程序的兼容性。


返回参数

此函数会返回数据字符串和返回码。

数据字符串:

如果在位置 5 (标准接口为 2) 中指定 A (异步), 并且功能已成功完成, 则返回以下数据字符串:


字节		定义
9-10	13-14	2 字节功能请求标识。 获取请求完成 (125) 功能使用它来确定此功能调用的完成情况。
	15-16	保留。
	17-20	EHLLAPI 返回事件对象地址的 4 字节值。应用程序可以等待此事件对象。清除事件对象后, 应用程序必须发出 获取请求完成 (125) 功能调用, 以获取 写入结构化字段 请求的结果。(仅限 32 位)。

 **注:** 每个成功的异步请求都会返回一个事件对象。不应再次使用事件对象。将为每个请求返回一个新的事件对象, 该对象仅在该请求的持续时间内有效。

数据字符串:

如果在位置 5 (标准接口为 2) 中指定 M (异步消息方式), 并且功能已成功完成, 则返回以下数据字符串:

字节		定义
9-10	13-14	2 字节功能请求标识。 获取请求完成 (125) 功能使用它来确定此功能调用的完成情况。
	15-16	保留。
11-12	17-18	异步消息方式的任务标识。
	19-20	保留。

 **注:** 如果该功能已成功完成, 应用程序窗口将收到一条消息。消息是 RegisterWindowMessage (PCSHLL) 的返回值。wParam 参数包含功能调用返回的任务标识。lParam 参数的 HIWORD 包含返回码 0, 表示功能已成功, lParam 参数的 LOWORD 包含功能编号 127。

返回码:

定义了以下代码:

返回码	说明
0	写入结构化字段 功能已成功。
1	指定的主机表示空间短会话标识无效或未连接。
2	指定参数时出错。
9	发生了系统错误。

返回码	说明
11	资源不可用（内存不可用）。
34	进站发送到主机的消息已被取消。
35	主机的出站传输已被取消。
36	请求被拒绝。与主机失去联系。
37	失败。主机已禁用进站。

有关使用此功能的注释

1. 当从主机发出的出站传输被取消后请求第一个**读取结构化字段**或**写入结构化字段**时，将返回返回码 35。纠正措施由应用程序负责。
2. 返回码 36 要求应用程序断开与仿真程序的连接，然后重新连接，以重新建立与主机的通信。纠正措施由应用程序负责。
3. 如果主机被禁用进站，将返回返回码 37。
4. EHLLAPI 允许每个应用程序最多可有 20 个未完成的异步请求。如果尝试的异步请求超过 20 个，将返回不可用资源的返回码 (RC=11)。

结构化字段数据格式如下所示：

偏移量	Length	目录
0	1 个字	X'0000'
2	1 个字	M (消息长度：消息中数据的字节数，该数字不包括缓冲区标头前缀，该前缀包含 8 个字节) 此值必须由应用程序设置。
4	1 个字	X'0000'
6	1 个字	X'0000'
8	8 字节	第一个 (或唯一) 结构化字段消息的长度。
10	1 个字节	结构化字段消息的第一个非长度字节。
		:
m+7	1 个字节	结构化字段消息中的最后一个字节。

字节 0 到 7 是缓冲区标头。前 8 个字节由仿真程序使用。缓冲区的用户部分以偏移量 8 开始。字节 8 和 9 包含第一个结构化字段 (结构化字段消息可以包含多个结构化字段) 中的字节数，包括字节 8 和 9 的 2 个字节。字节 8 到 m+7 用于发送到主机的结构化字段消息。

同步请求

同步请求**写入结构化字段** (数据字符串中的 S 选项) 时，只有在满足请求后，控制权才会返回到应用程序。应用程序可以假定：

- 返回码正确。
- 通信缓冲区 (读取缓冲区) 中的数据正确。
- 主机不再处理**写入结构化字段**请求。

异步请求

异步请求**写入结构化字段**（数据字符串中的 A 选项）时，应用程序不能假定：

- 返回码正确。
- 通信缓冲区（写入缓冲区）中的数据正确。
- 主机不再处理**写入结构化字段**请求。

异步请求时，EHLLAPI 会返回以下值：

- 数据字符串的位置 13-14（标准接口为 9-10）中的 16 位请求标识
- 数据字符串位置 17-20 中事件对象的地址。

这些地址用于完成异步**写入结构化字段**调用。

要确定异步**写入结构化字段**函数调用的结果，必须完成以下步骤：

- 如果 EHLLAPI 返回码不是零，则请求失败。未发出异步请求。应用程序必须在再次尝试调用之前采取适当的措施。
- 如果返回码为零，则应用程序应使用**获取请求完成** (125) 功能等至事件对象处于已发出信号状态。不能重用事件对象**获取请求完成** (125) 功能。事件对象仅在**写入结构化字段**功能调用直至完成**获取请求完成** (125) 功能调用的持续时间内有效。
- 事件对象处于已发出信号状态后，则在**获取请求完成** (125) 功能调用中，将返回的 16 位请求标识用作请求标识参数。从**获取请求完成** (125) 功能调用返回的数据字符串包含**写入结构化字段**功能调用的最终返回码。

异步请求

异步请求**写入结构化字段**（数据字符串中的 M 选项）时，应用程序不能假定：

- 返回码正确
- 通信缓冲区（写入缓冲区）中的数据正确
- 主机不再处理**写入结构化字段**请求

当使用 M 选项异步请求时，EHLLAPI 会返回以下值：

- 数据字符串的位置 13-14（标准接口为 9-10）中的 16 位请求标识
- 位置 17-18（标准接口为 11-12）中异步消息方式的任务标识

这些地址用于完成异步**写入结构化字段**调用。

第 4 章. WinHLLAPI 扩展函数

本章介绍使用 WinHLLAPI 编程支持时提供的扩展函数。

WinHLLAPI 函数摘要

以下 WinHLLAPI 函数可用于 3270、5250 和 VT：

- [等待 \(4\) \(on page 156\)](#)
 - [启动主机通知 \(23\) \(on page 157\)](#)
 - [启动关闭拦截 \(41\) \(on page 159\)](#)
 - [启动击键拦截 \(50\) \(on page 160\)](#)
 - [发送文件 \(90\) \(on page 161\)](#)
 - [接收文件 \(91\) \(on page 162\)](#)
-

WinHLLAPI 异步函数

以下各节介绍了 WinHLLAPI 异步函数。

WinHLLAPIAsync

此入口点用于通常需要很长时间才能完成的六个 WinHLLAPI 函数。使用 WinHLLAPIAsync，该函数将异步启动，不会干扰调用应用程序的持续进度。这些函数包括：[等待 \(04\)](#)、[启动主机通知 \(23\)](#)、[启动关闭拦截 \(41\)](#)、[启动击键拦截 \(50\)](#)、[发送文件 \(90\)](#) 和 [接收文件 \(91\)](#) 在 [WinHLLAPI 扩展函数 \(on page 156\)](#) 中介绍。

HANDLE WinHLLAPIAsync (HWIND hWnd、LPWORD lpnFunction、LPBYTE lpData、LPWORD lpnLength、LPWORD lpnRetC) *

参数列表与 WinHLLAPI 相同，只是在函数编号之前需要窗口句柄。由于函数是异步运行的，因此其完成由注册的消息来指示。需要将窗口句柄作为消息的目标。

有两个消息必须由 WinHLLAPI 应用程序通过调用 **RegisterWindowsMessage()** 来注册，其字符串为 **WinHLLAPIAsync**（适用于除 90 和 91 之外的所有功能）和 **WinHLLAPIAsyncFileTransfer**（适用于功能 90 和 91）。标准格式如下：

WPARAM

包含由原始函数调用返回的任务句柄。

LPARAM

高字包含错误代码，低字包含原始功能编号。

等待 (4)

此功能确定主机会话是否处于被禁止状态。如果由于某种原因会话处于被禁止状态，此功能将在被禁止状态过期或等待期间已过期时，向应用程序发送一条消息。等待的时间量可使用[设置会话参数 \(9\)](#) 功能进行设置。

前提条件函数

连接表示空间 (1)

WinHLLAPIAsync(*hWnd*, *lpwFunction*, *lpbyString*, *lpwLength*, *lpwReturnCode*)

调用参数

参数	说明
数据字符串	不适用
<i>Data Length</i>	不适用
PS 位置	不适用

返回码

Code	说明
WHLLOK	PS 未被禁止，可用于输入。
WHLLNOTCONNECTED	您的 WinHLLAPI 应用程序未连接到有效的主机会话。
WHLLPSBUSY	功能在仍被禁止时超时。
WHLLNHIBITED	PS 被禁止。
SHLLSYSERROR	此函数因系统错误而失败。
WHLLCANCEL	此异步函数已被取消。

注解

异步等待用于在 PS 的被禁止状态过期时通知调用应用程序。当被禁止状态过期时，此版本的**等待**会将 **WinHLLAPIAsync** 消息发布到 *hWnd* 指定的窗口。会话选项 **TWAIT**、**LWAIT** 和 **NWAIT** 会影响此功能的等待时间长度。有关这些会话选项的详细信息，请参阅 [设置会话参数 \(9\) \(on page 125\)](#)。



注：如果在会话参数中指定了 **NWAIT**，并且应用程序使用 WinHLLAPI 实现的修订版 1.1 注册，则 **WinHLLAPIAsync** 调用的工作方式与 **WinHLLAPI** 调用相同，而不发送消息。如果正在使用修订版 1.0，则**等待**将立即返回一条消息，并显示 PS 的被禁止状态。

启动主机通知 (23)

此功能让您可通知 WinHLLAPI 应用程序有关主机会话表示空间 (PS) 或操作信息区域 (OIA) 中的更改。

前提条件函数

此函数没有任何前提条件函数。

WinHLLAPIAsync (*hWnd*, *lpwFunction*, *lpbyString*, *lpwLength*, *lpwReturnCode*)

调用参数

参数	说明
数据字符串	<p>采用以下格式的 7 字节字符串：</p> <p>字节 1</p> <p>所需主机会话的短名称会话标识，或表示当前主机会话的空格或 null。</p> <p>字节 2</p> <p>通知方式。“P”表示仅更新表示空间，“O”表示仅更新 OIA，“B”表示同时更新表示空间和 OIA。调用 WinHLLAPIAsync 时，此位置可以是“A”。</p> <p>字节 3-6</p> <p>未使用。为兼容旧版应用程序而提供。</p> <p>字节 7</p> <p>如果使用 WinHLLAPIAsync 和 字节 2 中的 A，则保留或替换为以下项之一：P 表示仅更新表示空间，O 表示仅更新 OIA，B 表示同时更新表示空间和 OIA。</p>
<i>Data Length</i>	主机事件缓冲区的长度（建议 256 个）。
<i>PS 位置</i>	不适用

返回参数

参数	说明
数据字符串	与调用中的 <i>Data String</i> 相同。

返回码

Code	说明
WHLLOK	主机通知已启用。
WHLLNOTCONNECTED	指定的主机会话无效。
WHLLPARAMETERERROR	一个或多个参数无效。
WHLLSYSERROR	此函数因系统错误而失败。
WHLLCANCEL	此异步函数已被取消。

注解

启用后，将启用主机通知，直到调用**停止主机通知 (25)** 或 `WinHLLAPICancelAsyncRequest()`。此功能会启动主机通知并立即将控制权返回给 Windows HLLAPI 应用程序。这将释放您的应用程序，以在等待主机更新时执行其他任务。当发生更新时，此功能将使用已注册消息 `WinHLLAPIAsync` 通知 *hWnd* 指定的窗口。

启动关闭拦截 (41)

此功能会拦截关闭 Z and I Emulator for Windows 的用户请求。

前提条件函数

此函数没有任何前提条件函数。

WinHLLAPIAsync (*hWnd*, *lpwFunction*, *lpbyString*, *lpwLength*, *lpwReturnCode*)

调用参数

参数	说明
数据字符串	所返回信号量地址的 5 字节字符串。第一个字节是要查询的会话的会话短名称，或者对于当前会话为空格或 null。
<i>Data Length</i>	必须指定。
PS 位置	不适用

返回参数

参数	说明
数据字符串	采用以下格式的 5 字节字符串： <p>字节 1</p> <p>会话短名称，或者对于当前会话为空格或 null</p> <p>字节 2-5</p> <p>信号量地址。</p>

返回码

Code	说明
WHLLOK	此函数已成功。
WHLLNOTCONNECTED	指定的表示空间无效。
WHLLPARAMETERERROR	指定的选项无效。
WHLLSYSERROR	此函数因系统错误而失败。
WHLLCANCEL	此异步函数已被取消。

注解

启用后，主机通知将保持已启用状态，直到调用**停止关闭拦截 (43)** 或 **WinHLLAPICancelAsyncRequest ()**。最初，设置信号量。使用此功能后，用户发出的关闭请求将被丢弃，信号量将被清除。

该功能会启动关闭拦截，并立即将控制权返回给 Windows HLLAPI 应用程序。这将释放您的应用程序，以在等待关闭请求时执行其他任务。当发生关闭请求时，此功能将使用已注册消息 **WinHLLAPIAsync** 通知 *hWnd* 指定的窗口。

启动击键拦截 (50)

此函数会拦截用户发送到会话的击键。

前提条件函数

此函数没有任何前提条件函数。

WinHLLAPIAsync (*hWnd*, *lpwFunction*, *lpbyString*, *lpwLength*, *lpwReturnCode*)

调用参数

参数	说明
数据字符串	采用以下格式的 6 字节字符串： 字节 1 会话短名称，或对于当前主机会话为空格或 null。 字节 2 击键拦截代码。“D”会导致仅 AID 击键被拦截；“L”会导致所有击键被拦截。 字节 3-6 保留
<i>Data Length</i>	变量（建议使用 256）
<i>PS 位置</i>	不适用

返回码

Code	说明
WHLLOK	已启动击键拦截。
WHLLNOTCONNECTED	主机会话表示空间无效。
WHLLPARAMETERERROR	一个或多个参数无效。
WHLLPSBUSY	会话正忙。
WHLLSYSERROR	函数因系统错误而失败。
WHLLCANCEL	已取消异步函数。

注解

该功能会启动击键拦截，并立即将控制权返回给 Windows HLLAPI 应用程序。这将释放您的应用程序，以在等待击键时执行其他任务。启动后，只要用户向 PS 发送键，此功能就会向 *hWnd* 指定的窗口发送一条 **WinHLLAPIAsync** 消息。在发出通知后，可以采用正常 EHLLAPI 应用程序允许的任何方式处理拦截的击键。请注意，击键缓冲区的大小受限，因此应处理每个击键并将其从缓冲区中删除。

发送文件 (90)

此函数会将文件从 PC 传输到主机。

前提条件函数

此函数没有任何前提条件函数。

WinHLLAPIAsync (*hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode*)

调用参数

参数	说明
数据字符串	SEND 命令参数。
<i>Data Length</i>	数据字符串的长度。如果指定了会话选项 EOT, 则为 NA。
<i>PS 位置</i>	不适用

返回码

Code	说明
WHLLOK	文件传输已成功启动。
WHLLPARAMETERERROR	参数错误或数据长度为零或大于 255。
WHLLFTXCOMPLETE	文件传输完成。
WHLLFTXSEGMENTED	传输已完成, 但包含分段的记录。
WHLLSYSERROR	此函数因系统错误而失败。
WHLLTRANSABORTED	由于用户单击取消按钮或超时期间已过, 文件传输已中止。
WHLLFILENOTFOUND	找不到 PC 文件。
WHLLFTXCOMPLETECICS	文件传输已成功 (传输到 CICS) 。
WHLLACCESSDENIED	访问 PC 文件被拒绝。
WHLLMEMORY	内存不足。
WHLLINVALIDENVIRONMENT	环境无效。

注解

每个连接的主机会话都仅支持一个文件传输操作。

该功能会启动文件传输, 并立即将控制权返回给 Windows HLLAPI 应用程序。这将释放您的应用程序, 以在文件传输过程中执行其他任务。一旦启动, 此功能将定期将 **WinHLLAPIAsyncFileTransfer** 消息发布到 *hWnd* 指定的窗口。这些消息将通知 WinHLLAPI 应用程序传输的状态, 并在传输完成时发送最终消息。

wParm

是状态指示符: 高字节包含会话标识, 低字节包含状态。如果低字节为零, 则文件传输仍在进行中。如果低字节为 1, 则文件传输已完成。

IParm

如果 *wParm* 的低字节为零（进行中），则 *IParm* 是传输的字节数。如果低字节 *wParm* 为 1（已完成），则 *IParm* 是完成代码。

接收文件 (91)

此函数会将文件从 PC 传输到主机。

前提条件函数

此函数没有任何前提条件函数。

WinHLLAPIAsync (*hWnd*, *lpwFunction*, *lpbyString*, *lpwLength*, *lpwReturnCode*)

调用参数

参数	说明
数据字符串	RECEIVE 命令参数。
<i>Data Length</i>	数据字符串的长度。如果指定了会话选项 EOT，则为 NA。
PS 位置	不适用

返回码

Code	说明
WHLLOK	文件传输已成功启动。
WHLLPARAMETERERROR	参数错误或数据长度为零或大于 255。
WHLLFTXCOMPLETE	文件传输完成。
WHLLFTXSEGMENTED	传输已完成，但包含分段的记录。
WHLLSYSERROR	此函数因系统错误而失败。
WHLLTRANSABORTED	由于用户单击取消按钮或超时期间已过，文件传输已中止。
WHLLFILENOTFOUND	找不到 PC 文件。
WHLLFTXCOMPLETECICS	文件传输已成功（传输到 CICS）。
WHLLACCESSDENIED	访问 PC 文件被拒绝。
WHLLMEMORY	内存不足。
WHLLINVALIDENVIRONMENT	环境无效。

注解

每个连接的主机会话都仅支持一个文件传输操作。

该功能会启动文件传输，并立即将控制权返回给 Windows HLLAPI 应用程序。这将释放您的应用程序，以在文件传输过程中执行其他任务。一旦启动，此功能将定期将 **WinHLLAPIAsyncFileTransfer** 消息发布到 *hWnd* 指定的窗口。这些消息将通知 WinHLLAPI 应用程序传输的状态，并在传输完成时发送最终消息。

wParm

是状态指示符：高字节包含会话标识，低字节包含状态。如果低字节为零，则文件传输仍在进行中。如果低字节为 1，则文件传输已完成。

IParm

如果 *wParm* 的低字节为零（进行中），则 *IParm* 是传输的字节数。如果低字节 *wParm* 为 1（已完成），则 *IParm* 是完成代码。

WinHLLAPICancelAsyncRequest

此函数会取消调用 **WinHLLAPIAsync()** 时启动的未完成异步函数。

语法

```
int WinHLLAPICancelAsyncRequest (HANDLE hAsyncTask, WORD wFunction)
```

Parameters

hAsyncTask

在启动函数时由 **WinHLLAPIAsync()** 返回的句柄。

wFunction

要取消的异步任务的函数编号。由于修订版 1.1 需要此参数，但 1.0 中不需要此参数，因此此参数为可选。

使用此函数，以前通过调用 **WinHLLAPIAsync()** 启动的任何异步任务都可以在未完成时被取消。

退货

返回值指示指定的函数是否已实际被取消。如果函数被取消，则返回值为 **WHLLOK** (0)。如果未完成的异步函数未被取消，将返回以下一个代码。

WHLLINVALID

hAsyncTask 不是有效的任务句柄。

WHLLALREADY

由 *hAsyncTask* 指定的异步任务已完成。

初始化和终止功能

下节介绍了 WinHLLAPI 编程支持的初始化和终止功能。

WinHLLAPI 启动

此函数用于在 WinHLLAPI 实现中注册应用程序，并且应在对 WinHLLAPI 实现的任何其他调用之前调用此函数。此实现支持 WinHLLAPI 规范 V1.0 和 V1.1。WinHLLAPI 应用程序应与此函数协商版本兼容性。

语法

int WinHLLAPIStartup(WORD *wVersionRequired*, LPWHLAPIDATA *lpData*)

Parameters

wVersionRequired

这是 WinHLLAPI 应用程序需要的版本。低字节包含主要版本号，高字节包含次要版本（或修订版）号。

lpData

这是指向 WHLLAPIDATA 结构的指针，该结构将接收实现版本号和描述 WinHLLAPI 实现提供程序的字符串。WHLLAPIDATA 结构定义为：

```
#define WHLLDESCRIPTION_LEN 127
typedef struct tagWHLAPIDATA
{
    WORD wVersion;
    Char szDescription[WHLLDESCRIPTION_LEN + 1];
}WHLAPIDATA, * PWHLAPIDATA, FAR *LPWHLAPIDATA;
```

退货

返回值表示在实现中注册 WinHLLAPI 应用程序是成功还是失败。如果注册成功，则返回值为 WHLLOK（零）。否则，将是下列其中一项：

WHLLSYSNOTREADY

表示底层网络子系统不可用。

WHLLVERNOTSUPPORTED

表示此实现未提供请求的版本。此实现仅支持 V1.0 和 V1.1。

WinHLLAPI 清理

WinHLLAPI 规范建议 WinHLLAPI 应用程序使用此函数从 WinHLLAPI 实现中注销。

语法

BOOL WinHLLAPICleanup()

退货

如果注销成功，则返回 TRUE。否则返回 FALSE。

阻塞例程

以下各节介绍了 WinHLLAPI 编程支持的阻塞例程。



注：虽然 WinHLLAPI 合规性支持阻塞例程，但不建议使用它们。建议的方法是使用 WinHLLAPIAsync 函数进行异步处理。

WinHLLAPIIsBlocking

此函数会告诉调用 WinHLLAPI 应用程序线程是否正在执行阻塞调用。阻塞调用是任何需要很长时间才能执行的同步函数，直到完成后才会返回。WinHLLAPI 实现中有五个阻塞调用。阻塞调用包括：**获取键 (51)**、**等待 (4)**、**暂停 (18)**、**发送文件 (90)** 和 **接收文件 (91)**。

语法

```
BOOL WinHLLAPIIsBlocking()
```

退货

如果 WinHLLAPI 应用程序线程正在分块调用的中间，则此函数会返回 TRUE，否则返回 FALSE。

注解

由于缺省分块挂钩允许在分块调用期间处理消息，因此可以再次调用分块调用。

WinHLLAPISetBlockingHook

此函数会设置要在等待阻塞调用完成时执行的应用程序定义的过程。阻塞调用是任何需要很长时间才能执行的同步函数，直到完成后才会返回。WinHLLAPI 实现中有五个阻塞调用。阻塞调用包括：**获取键 (51)**、**等待 (4)**、**暂停 (18)**、**发送文件 (90)** 和 **接收文件 (91)**。

语法

```
FARPROC WinHLLAPISetBlockingHook(FARPROC lpfnBlockingHook)
```

Parameters

lpfnBlockingHook

这是指向新阻止过程的指针。

描述

WinHLLAPI 实现具有缺省阻塞过程，在过程仅包含一个消息处理程序。以下示例显示了此缺省机制：

```
BOOL DefaultBlockingHook
{
    MSG msg;

    if (PeekMessage (&msg, NULL, 0, 0, xFPM_NOREMOVE))
    {
        if(msg.message == WM_QUIT)
        {
```

```

        return FALSE;
    }
    PeekMessage (&msg, NULL, 0, 0, PM_REMOVE);
    TranslateMessage (&msg);
    DispatchMessage (&msg);
}
return TRUE;
}

```

阻塞挂钩是在每个线程的基础上实现的。此函数设置的阻塞挂钩将对线程保持有效，直到它被另一个 **WinHLLAPISetBlockingHook()** 调用替换，或者直到调用 **WinHLLAPIUnhookBlockingHook()** 恢复缺省值为止。

如果阻塞函数收到 **WM_QUIT** 消息，则必须返回 **FALSE**，以便 WinHLLAPI 可以将控制权返回给应用程序，以处理消息并正常终止。否则，函数应返回 **TRUE**。

退货

此函数返回指向要替换的阻止函数的指针。

WinHLLAPIUnhookBlockingHook

此函数会恢复调用线程的缺省分块挂钩。

语法

```
BOOL WinHLLAPIUnhookBlockingHook()
```

退货

如果已成功恢复缺省分块机制，则此函数会返回 **TRUE**，否则返回 **FALSE**。

WinHLLAPICancelBlockingCall

此函数会取消当前线程中正在执行的阻塞调用。阻塞调用是任何需要很长时间才能执行的同步函数，直到完成后才会返回。WinHLLAPI 实现中有五个阻塞调用。阻塞调用包括**获取键 (51)**、**等待 (4)**、**暂停 (18)**、**发送文件 (90)**和**接收文件 (91)**。如果其中一个阻塞调用被取消，被取消的函数将返回 **WHLLCANCEL**。

语法

```
int WinHLLAPICancelBlockingCall()
```

退货

返回值指示指定的函数是否已实际被取消。如果函数被取消，则返回值为 **WHLLCANCEL (0)**。如果没有未完成的阻塞函数，则将返回以下返回码：

WHLLINVALID

指示当前没有正在执行的阻塞调用。

第 5 章. PCSAPI 函数

Z and I Emulator for Windows 提供在此处定义且称为 *PCSAPI* 的 API 集。建立会话后，EHLLAPI 用于管理工作站应用程序与主机系统之间的交互，而 PCSAPI 可用于控制 Z and I Emulator for Windows 会话本身。

如何使用 PCSAPI

您可以在 C 或 C++ 中使用 PCSAPI 编写应用程序。要开发 PCSAPI 应用程序，请执行以下操作：

1. 准备源代码并添加相应的 PCSAPI 调用。
2. 在应用程序中包括头文件 PCSAPI.H。
3. 编译源代码。
4. 将生成的 .OBJ 文件与相应的对象文件或库链接。

您还必须将其与 PCSAPI 导入库、PCSCALLS.LIB（16 位）和 PCSCAL32.LIB（32 位）链接。

页面布局约定

所有 PCSAPI 函数调用都以相同的格式显示，以便可以快速检索所需的信息。格式为：

- 函数名
 - 函数类型
 - 参数类型和描述
 - 返回码
-

函数类型

“函数类型”以下列格式显示函数的类型：

TYPE **FunctionName**(*TYPE Parameter1, ...*)

参数类型和描述

“参数类型和描述”列出类型并描述要在 PCSAPI 函数调用中指定的每个参数。

返回码

“返回代码”列出调用 PCSAPI 函数后程序必须收到的代码。

pcsConnectSession

3270	5250	VT
是	是	是

pcsConnectSession 函数会启动与短会话标识指定的主机会话的通信。会话必须已启动。此调用等同于仿真器会话面板上的**通信** → **连接**菜单项。

函数类型

BOOL WINAPI pcsConnectSession(char cShortSessionID)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

返回码

返回码	意义
真	函数已成功结束。
FALSE	这意味着以下一种情况： <ul style="list-style-type: none"> • 会话尚未启动。 • 指定的会话标识不正确。 • 调用失败。

pcsDisconnectSession

3270	5250	VT
是	是	是

pcsDisconnectSession 函数会停止与通过短会话标识指定的主机会话的通信链路。这只会断开链路；不会停止会话。此调用等同于仿真器会话面板上的**通信** → **断开连接**菜单项。

函数类型

BOOL WINAPI pcsDisconnectSession(char cShortSessionID)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

返回码

返回码	意义
真	函数已成功结束。
FALSE	这意味着以下一种情况： <ul style="list-style-type: none"> • 会话尚未启动。 • 指定的会话标识不正确。 • 调用失败。

pcsQueryConnectionInfo

3270	5250	VT
是	否	编号

pcsQueryConnectionInfo 函数会返回有关指定主机会话的 Telnet 连接的信息。生成的信息将返回到应用程序提供的缓冲区中。

函数类型

BOOL WINAPI pcsQueryConnectionInfo(*char cShortSessionID*, *CONNECTIONINFO *ConnectionInfo*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

CONNECTIONINFO *ConnectionInfo

指向将返回连接信息数据的 CONNECTIONINFO 结构的指针。

返回码

返回码	意义
真	函数已成功结束。
FALSE	这意味着以下一种情况： <ul style="list-style-type: none"> • 会话尚未启动。 • 指定的会话标识不正确。 • 指定的会话不是此 API 支持的连接类型（不是 Telnet）。

ConnectionInfo

CONNECTIONINFO 结构将填充有关主机连接的信息，包括以下信息：

结构	信息
主机名	陈述当前连接的 Telnet 主机的名称。
LU 名	陈述当前分配的 LU 名。
端口号	陈述要用于连接的主机端口号。
SSL 指示符	指示安全连接 (1 = 安全; 0 = 不安全)。



注： 此 API 仅对 32 位版本的 PCSAPI 有效，并且仅适用于 Telnet 连接。

示例

```
typedef struct_CONNECTIONINFO
{ //Description of a connection @WD06A
  char hostName[63]; //telnet host name @WD06A
  char reserved[1]; //reserved @wD06A
  int portNumber; //host port number @WD06A
  char luName[17]; //LU name @WD06A
  char reserved2[3]; //reserved @WD06A
  BOOL sslIndicator; //Secure Connection @WD06A
  indicator
  char reserved3[256]; //reserved @WD06A
}CONNECTIONINFO;
```

pcsQueryEmulatorStatus

3270	5250	VT
是	是	是

pcsQueryEmulatorStatus 函数会返回通过短会话标识指定的主机会话的状态。

函数类型

ULONG WINAPI pcsQueryEmulatorStatus(char cShortSessionID)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

返回码

返回码值应以位显著的方式进行处理，即通过以下一个值或以下值中的 OR 运算值进行处理：

返回码	值	含义
PCS_SESSION_STARTED	0x00000001	指定的会话已启动。此位关闭时，指定的会话尚未启动或指定的会话标识错误。
PCS_SESSION_ONLINE	0x00000002	指定的会话处于在线状态（已连接）。此位关闭时，指定的会话处于离线状态（已断开连接）。
PCS_SESSION_API_ENABLED	0x00000004	API (EHLLAPI) 已在指定会话上启用。如果此位关闭，则已在此会话上禁用 API。

pcsQuerySessionList

3270	5250	VT
是	是	是

pcsQuerySessionList 函数会返回所有当前主机会话的列表。应用程序必须提供在 PCSAPI.H 文件中定义的 SESSINFO 结构数组，以及数组中元素的计数。此函数使用有关每个会话的信息填充结构，并返回找到的会话数。

如果数组的元素少于主机会话，则仅填充数组中提供的元素。该函数始终返回实际会话数，即使数组太小也是如此。

应用程序可以使用零个数组元素调用此函数，以确定存在的会话数。然后可以进行第二次调用，以获取会话信息。

函数类型

ULONG WINAPI pcsQuerySessionList(ULONG Count, SESSINFO *SessionList)

参数类型和描述

ULONG 计数

SessionList 数组中的元素数量。

SESSINFO *SessionList

指向在 PCSAPI.H 中定义的 SESSINFO 结构数组的指针

返回参数

返回码

Z and I Emulator for Windows 会话总数。这可能大于或小于 Count 参数。

SessionList

SESSINFO 结构的数组中填充了有关主机会话的信息。会话可以按任意顺序放在列表中。每个 SESSINFO 结构都包含以下字段（在 PCSAPI32.H 中定义）

姓名

char 和 ULONG 的联合，包含会话标识 (A-Z)。在 Z and I Emulator for Windows 的当前实现中，仅使用较低的字节 (char)，其他字节返回为零。

Status

指示会话当前状态的位标志的组合。标志 (PCS_SESSION_*) 在下表中定义。

状态值应以位显著的方式进行处理，即通过以下一个值或以下值中的 OR 运算值进行处理：

返回码	意义
PCS_SESSION_STARTED	会话正在运行。如果未设置此标志，则未定义所有其他标志。
PCS_SESSION_ONLINE	会话已建立到主机的通信链路（即会话已连接）。
PCS_SESSION_API_ENABLED	已为编程 API 启用会话。如果未设置此标志，则无法在此会话中使用 EHLLAPI 和主机访问类库 API。

示例

```

ULONG      NumSessions, i; // Session counters
SESSINFO  *SessList;      // Array of session information structures
// Find out number of sessions that exist
NumSessions = pcsQuerySessionList (0,NULL);
if (NumSessions == 0) {
    printf("There are no sessions.");
    exit;
}

// Allocate array large enough for all sessions
SessList = (SESSINFO *)malloc(NumSessions * sizeof(SESSINFO));
memset(SessList, 0x00, NumSessions * sizeof(SESSINFO));

// Now read actual session info
pcsQuerySessionList(NumSessions, SessList);

for (i=0; i<NumSessions; i++) {
    if ((SessList[i].Status & PCS_SESSION_STARTED) &&
        (SessList[i].Status & PCS_SESSION_ONLINE)) {

        printf("Session %c is started and connected.",
            SessList[i].Name.ShortName);
    }
}

exit;
    
```

pcsQueryWorkstationProfile

3270	5250	VT
是	是	是

pcsQueryWorkstationProfile 函数会返回用于调用主机会话的工作站概要文件名。要指定主机会话，必须使用短会话标识。工作站概要文件名将被复制到应用程序提供的工作缓冲区。

函数类型

BOOL WINAPI **pcsQueryWorkstationProfile**(*char cShortSessionID, PSZ lpBuffer*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

PSZ lpBuffer

用于复制以 null 结束的工作站概要文件名称的工作缓冲区。缓冲区必须足够大，以包含全限定文件名。

返回码

返回码	意义
真	函数已成功结束。
FALSE	这意味着以下一种情况： <ul style="list-style-type: none"> • 会话尚未启动。 • 指定的会话标识不正确。

pcsSetLinkTimeout

3270	5250	VT
是	是	是

pcsSetLinkTimeout 函数会设置 SSCP 拥有的 Telnet 链路的空闲超时。此函数对非 TN 连接或非 SSCP 拥有状态的连接没有影响。如果将超时值设置为零，则链路不会超时。否则，在 SSCP 拥有状态下空闲指定的分钟数后，链路将超时（断开连接）。

函数原型

ULONG WINAPI **pcsSetLinkTimeout**(*char cShortSessionID, USHORT Timeout*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

USHORT Timeout

超时值（以分钟为单位）。零值将禁用超时。

返回码

返回码	意义
PCS_SUCCESSFUL	函数已成功结束。
PCS_SYSTEM_ERROR	发生了系统错误。

pcsStartSession

3270	5250	VT
是	是	是

pcsStartSession 函数会使用指定的工作站概要文件启动主机会话。也可以指定短会话标识。

函数类型

ULONG WINAPI pcsStartSession(*PSZ lpProfile, char cShortSessionID, USHORT fuCmdShow*)

参数类型和描述

PSZ lpProfile

要加载的概要文件的路径和完整文件名。路径为可选，但必须指定完整文件名（不假定扩展名为 .ws）。

char cShortSessionID

表示空间短会话标识。空白或 NULL 表示下一个可用的会话标识。

USHORT fuCmdShow

指定窗口的显示方式。PCSAPI.H 中的以下值之一：

- PCS_HIDE
- PCS_SHOW
- PCS_MINIMIZE
- PCS_MAXIMIZE

返回码

返回码	价值	意义
PCS_SUCCESSFUL	0	函数已成功结束。
PCS_INVALID_ID	1	指定的会话标识不正确。
PCS_USED_ID	2	指定的短会话标识已被使用。
PCS_INVALID_PROFILE	3	指定工作站概要文件时出错，或者窗口参数无效。
PCS_SYSTEM_ERROR	9	发生了系统错误。

pcsStopSession

3270	5250	VT
是	是	是

pcsStopSession 函数会停止通过短会话标识指定的主机会话。

函数类型

BOOL WINAPI **pcsStopSession**(*char cShortSessionID, USHORT fuSaveProfile*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

USHORT fuSaveProfile

此参数可以是以下一个值：

fuSaveProfile	价值	意义
PCS_SAVE_AS_PROFILE	0	保存当前概要文件中指定的概要文件。
PCS_SAVE_ON_EXIT	1	退出时保存概要文件。
PCS_NOSAVE_ON_EXIT	2	退出时不保存概要文件。

返回码

返回码	意义
真	函数已成功结束。
FALSE	这意味着以下一种情况： <ul style="list-style-type: none"> • 会话尚未启动。 • 指定的会话标识不正确。

页面设置功能

本节中列出的 PCSAPI 功能让您可控制和检索 Z and I Emulator for Windows 仿真器会话**页面设置**设置。

限制

如果不满足以下限制，API 将失败。返回码指示失败的原因。

- 在参数 `cShortSessionID` 中指定的主机会话不应处于 PDT 方式。
- 调用 API 时，主机会话应不在打印。
- 不应使用 **文件** → **页面设置** 对话框。

PAGEINFO 结构中的某些成员可能有效或仅受特定会话类型支持。如果未指定限制，则该成员对以下会话类型有效或受支持：

- 3270 显示
- 3270 打印机
- 5250 显示
- ASCII VT

5250 打印机会话不受支持。

pcsGetPageSettings

3270	5250	VT
是	是	是

pcsGetPageSettings 函数会检索主机会话页面设置值（类似于 **文件** → **页面设置** 对话框设置）。仅支持对话框的 **文本** 选项卡中的设置。

函数类型

ULONG WINAPI pcsGetPageSettings(*char cShortSessionID, PAGEINFO * const pPageInfo, ULONG * const pErrorInfo*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

PAGEINFO * const pPageInfo

指向返回页面设置的 PAGEINFO 结构的指针。

nFlags

表示结构中哪些成员有效的位标志组合。这些标志可以单独使用，也可以通过将它们通过 OR 组合在一起来恢复属性页面来使用（在 PCSAPI32.H 中定义）。这些标志以及结构中相应的有效成员如下所示：

表示

结构中的有效成员

PCS_PAGE_CPI

nCPI

PCS_PAGE_LPI

nLPI

PCS_PAGE_FACE_NAME

szFaceName

PCS_PAGE_MPL

nMPL

PCS_PAGE_MPP

nMPP

nCPI

每英寸打印的字符数。

LOWORD 是实际 CPI 值。

如果在会话中配置了字体 CPI，则 HIWORD 为 1。如果未配置字体 CPI，则 HIWORD 为 0。

nLPI

每英寸打印的行数。

LOWORD 是实际 LPI 值。

如果在会话中配置了字体 LPI，则 HIWORD 为 1。如果未配置字体 LPI，则 HIWORD 为 0。

szFaceName

打印机字体的字体名称。这必须是以 null 结束的字符串。

nMPL

每页可打印的最大行数。

这也称为 MPL（最大打印行数）。支持的范围为 1 到 255。

nMPP

每行可打印的最大字符数。

这也称为 MPP（最大打印位置）。支持的范围为 1 到 255。

ULONG * const pErrorInfo

未使用。这必须由调用方设置为 NULL。

返回码

返回码	价值	意义
PCS_SUCCESSFUL	0	函数已成功结束。
PCS_INVALID_ID	1	指定的会话标识不正确。
PCS_INVALID_SESS_TYPE	2	主机会话类型不支持。

返回码	价值	意义
PCS_DIALOG_IN_USE	3	失败，因为主机会话“页面设置”或“打印机设置”对话框正在使用中。
PCS_PRINTING	4	无法获取页面设置，因为主机会话当时正在打印。
PCS_PDT_MODE	5	无法获取页面设置，因为主机会话处于 PDT 方式。
PCS_SYSTEM_ERROR	9	发生了系统错误。

示例

```

{
    ULONG Rc = 0;
    PAGEINFO *PageInfo;

    PageInfo = (PAGEINFO *) malloc(sizeof(PAGEINFO));
    memset(PageInfo, 0, sizeof(PAGEINFO));

    PageInfo->nFlags = PCS_PAGE_CPI | PCS_PAGE_LPI | PCS_PAGE_FACE_NAME |
        PCS_PAGE_MPL | PCS_PAGE_MPP;

    Rc = pcsGetPageSettings('A', PageInfo, NULL);

    if (Rc == PCS_SUCCESSFUL) {
        printf("CPI = %d,\n",
            LPI = %d,\n",
            FaceName = %s,\n",
            MPL = %d,\n",
            MPP = %d\n",
            LOWORD(PageInfo->nCPI),
            LOWORD(PageInfo->nLPI),
            PageInfo->szFaceName,
            PageInfo->nMPL,
            PageInfo->nMPP);

        if (HIWORD(PageInfo->nCPI))
            printf("FontCPI\n");
        else
            printf("No FontCPI\n");

        if (HIWORD(PageInfo->nLPI))
            printf("FontLPI\n");
        else
            printf("No FontLPI\n");
    } else
        printf("Failure. Return code = %d\n", Rc);
    free(PageInfo);
}

```

pcsRestorePageDefaults

3270	5250	VT
是	是	是

pcsRestorePageDefaults 函数会恢复在 nFlags 字段中定义的页面设置属性页面的系统缺省值。这等同于在**文件** → **页面设置**对话框的属性页面中单击**缺省值**。仅支持**文本**选项卡中的设置。

函数类型

ULONG WINAPI pcsRestorePageDefaults(char cShortSessionID, ULONG nFlags)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

ULONG nFlags

以下标志说明指定的**页面设置**对话框属性页面的名称。此标志可以按位进行 OR 运算，以恢复属性页面（在 PCSAPI32.H 中定义）。

PCS_PAGE_TEXT

此标志说明“文本”属性页面。这是当前支持的唯一属性页面。

返回码

返回码	价值	意义
PCS_SUCCESSFUL	0	函数已成功结束。
PCS_INVALID_ID	1	指定的会话标识不正确。
PCS_INVALID_SESS_TYPE	2	nFlags 参数具有一个或多个对主机会话类型无效的选项。未恢复任何设置。
PCS_DIALOG_IN_USE	3	失败，因为主机会话“页面设置”或“打印机设置”对话框正在使用中。
PCS_PRINTING	4	无法更改页面设置，因为主机会话正在打印。
PCS_PDT_MODE	5	无法更改页面设置，因为主机会话处于 PDT 方式。
PCS_SYSTEM_ERROR	9	发生了系统错误。

示例

```
{
    ULONG Rc = 0;

    Rc = pcsRestorePageDefaults('A', PCS_PAGE_TEXT);

    if (Rc != PCS_SUCCESSFUL)
```

```
printf("Failure. Return code = %d\n", Rc);
}
```

pcsSetPageSettings

3270	5250	VT
是	是	是

pcsSetPageSettings 函数用于设置主机会话页面设置。这与配置文件 → 页面设置对话框设置类似。仅支持文本选项卡中的设置。



注:

1. CPI、LPI 和 FontSize 取决于主机会话中配置的 FaceName。如果此 API 用于同时设置 CPI、LPI、FontSize 和 FaceName，则首先设置 FaceName，然后设置从属属性。
2. 如果此 API 用于在单独调用中设置 FaceName 和从属属性，请先设置 FaceName，然后设置 CPI、LPI 和 FontSize。否则，每次设置 FaceName 时，查询 CPI、LPI 和 FontSize 并确保它们具有所需的值。
3. 如果在 FaceName 之前设置 CPI、LPI 或 FontSize，则可能会在主机会话中为 CPI、LPI 或 FontSize 配置不同的值。如果当前的 CPI、LPI 或 FontSize 值对新的 FaceName 集无效，则可能会发生这种情况。

函数类型

ULONG WINAPI pcsSetPageSettings(char cShortSessionID, const PAGEINFO * const pPageInfo, ULONG * const pErrorInfo)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

const PAGEINFO * const pPageInfo

指向 PAGEINFO 结构的指针，其中提到了页面设置。

nFlags

表示结构中哪些成员有效的位标志组合。这些标志可以单独使用，也可以通过将它们通过 OR 组合在一起来恢复属性页面来使用（在 PCSAPI32.H 中定义）。这些标志以及结构中相应的有效成员如下所示：

表示

结构中的有效成员

PCS_PAGE_CPI

nCPI

PCS_PAGE_LPI

nLPI

PCS_PAGE_FACE_NAME

szFaceName

PCS_PAGE_MPL

nMPL

PCS_PAGE_MPP

nMPP

nCPI

每英寸打印的字符数。

要选择“字体 CPI”，请将 nCPI 的 HIWORD 设置为 1。nCPI 的 LOWORD 将被忽略。

要选择特定的 CPI 值，请执行以下操作：

1. 将 nCPI 的 HIWORD 设置为 0。
2. 将 nCPI 的 LOWORD 设置为实际 CPI 值。

nLPI

每英寸打印的行数。

要选择“字体 LPI”，请将 nLPI 的 HIWORD 设置为 1。nLPI 的 LOWORD 将被忽略。

要选择特定的 LPI 值，请执行以下操作：

1. 将 nLPI 的 HIWORD 设置为 0。
2. 将 nLPI 的 LOWORD 设置为实际 LPI 值。

szFaceName

打印机字体的字体名称。这必须是以 null 结束的字符串。

nMPL

每页可打印的最大行数。

这也称为 MPL（最大打印行数）。支持的范围为 1 到 255。

nMPP

每行可打印的最大字符数。

这也称为 MPP（最大打印位置）。支持的范围为 1 到 255。

ULONG * const pErrorInfo

包含 API 失败时的扩展错误信息，返回码为 PCS_FAILURE。如果不需要详细的错误信息，则调用方必须将此标志设置为 NULL。

这是位标志的组合，用于描述无法成功设置 PAGEINFO 结构的哪些成员。在 PCSAPI32.H 中定义的标志如下所示：

表示

结构中的有效成员

PCS_PAGE_CPI

只有 nCPI 无效。

PCS_PAGE_LPI

只有 nLPI 无效。

PCS_PAGE_FACE_NAME

只有 szFaceName 无效。

PCS_PAGE_MPL

只有 nMPL 无效。

PCS_PAGE_MPP

只有 nMPP 无效。

返回码

返回码	价值	意义
PCS_SUCCESSFUL	0	函数已成功结束。
PCS_INVALID_ID	1	指定的会话标识不正确。
PCS_INVALID_SESS_TYPE	2	主机会话类型不支持。
PCS_DIALOG_IN_USE	3	失败，因为主机会话“页面设置”或“打印机设置”对话框正在使用中。
PCS_PRINTING	4	无法更改页面设置，因为主机会话正在打印。
PCS_PDT_MODE	5	无法更改页面设置，因为主机会话处于 PDT 方式。
PCS_FAILURE	6	未完全应用主机会话页面设置。这可能是针对 PAGEINFO 结构中的部分或全部字段提供的数据无效。有关未应用的设置的详细信息，请检查 pErrorInfo。
PCS_SYSTEM_ERROR	9	发生了系统错误。

示例

```
{
    ULONG Rc = 0, Error = 0;
    PAGEINFO *PageInfo;

    PageInfo = (PAGEINFO *) malloc(sizeof(PAGEINFO));
    memset(PageInfo, 0, sizeof(PAGEINFO));

    PageInfo->nFlags = PCS_PAGE_CPI | PCS_PAGE_LPI |
```

```

        PCS_PAGE_FACE_NAME| PCS_PAGE_MPL |
        PCS_PAGE_MPP;
PageInfo->nCPI = MAKELONG(10, 0);
PageInfo->nLPI = MAKELONG(8, 0);
PageInfo->nMPL = 40;
PageInfo->nMPP = 60;
strcpy(PageInfo->szFaceName, "CourierPS");

Rc = pcsSetPageSettings('A', PageInfo, &Error);

if (Rc != PCS_SUCCESSFUL) {
    printf("Failure. Return code = %d\n", Rc);
    printf("Following members could not be set : ");

    if (Rc == PCS_FAILURE) {
        if (Error & PCS_PAGE_CPI) printf(" nCPI");
        if (Error & PCS_PAGE_LPI) printf(" nLPI");
        if (Error & PCS_PAGE_FACE_NAME) printf(" szFaceName");
        if (Error & PCS_PAGE_MPL) printf(" nMPL");
        if (Error & PCS_PAGE_MPP) printf(" nMPP");
        printf("\n");
    }
}
free(PageInfo);
}

```

打印机设置功能

本节中列出的 PCSAPI 功能让您可控制和检索 Z and I Emulator for Windows 仿真器会话**打印机设置**。

限制

如果不符合以下限制，API 将失败。返回码指示失败的原因。

- 调用 API 时，主机会话应不在打印。
- 不应使用**文件** → **打印机设置**对话框。

pcsGetPrinterSettings

3270	5250	VT
是	是	是

pcsGetPrinterSettings 函数会检索主机会话打印机设置（类似于**文件** → **打印机设置**对话框设置）。

函数类型

ULONG WINAPI pcsGetPrinterSettings(*char cShortSessionID, PRINTINFO * const pPrintInfo, ULONG * const pErrorInfo*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

PRINTINFO * const pPrintInfo

指向 PRINTINFO 结构的指针，在该结构中指定了打印机设置。

nFlags

必须设置为 0。这将被忽略。

nBufSize

为以下字段分配的缓冲区大小：

- lpPDFile
- lpPrtToDskAppFile
- lpPrtToDskSepFile
- lpPrinterName

如果在单个 API 调用中检索到这些成员中的多个，则调用方必须为所有缓冲区分配相同的大小，并在此成员中传递该大小。

如果将此成员设置为 0，则忽略字段。在 nSizeNeeded 中返回字段缓冲区所需的最大大小。

nSizeNeeded

此成员的值由与以下字段相关的条件决定：

- lpPDFile
- lpPrtToDskAppFile
- lpPrtToDskSepFile
- lpPrinterName

条件如下所示：

- 如果调用方分配的缓冲区的大小不足以返回上面列出的字段，则该值是所需的字节数。
- 如果调用方获得上面列出的多个字段，则该值是所需缓冲区的最大大小。
- 如果调用方将 nBufSize 设置为 0，则该成员包含上述字段的缓冲区所需的最大大小。

bPromptDialog

有效值如下：

- 如果为 TRUE，则在打印前显示“打印机设置”对话框。
- 如果为 FALSE，则在打印前不会显示“打印机设置”对话框。

bPDTMode

有效值如下：

- 如果为 TRUE，则主机会话处于 PDT 方式。
- 如果为 FALSE，则主机会话处于非 PDT 方式（GDI 方式）。

lpPDTFile

如果调用方不想获取此成员，则必须将其设置为 NULL。如果这不是 null 指针，则返回 PDT 文件。这必须指向由调用方分配的、大小为 nBufSize 的缓冲区。

返回 API 时，此成员包含以下一项：

- 会话 PDT 文件的全限定路径名。
- 如果未在会话中 PDT 文件，则为空字符串 ("")。
- 缓冲区大小不足时截断的文件名。成员 nSizeNeeded 包含所需缓冲区的大小。

nPrtMode

这是指示连接的 PrintMode 的枚举值。枚举数据类型 PRINTMODE 在 PCSAPI32.H 中定义。nPrtMode 设置必须是以下一项：

- **PrtToDskAppend** (打印到磁盘 - 附加方式)

这相当于在主机会话的**打印机设置** → **打印机** → **打印到磁盘**对话框中选择**追加**选项。

- **PrtToDskSeparate** (打印到磁盘 - 单独方式)

这相当于在主机会话的**打印机设置** → **打印机** → **打印到磁盘**对话框中选择**单独**选项。

- **WinDefaultPrinter** (Windows 缺省打印机方式)

这相当于在主机会话**打印机设置**对话框中选择**使用 Windows 缺省打印机**选项。

- **SpecificPrinter** (特定打印机方式)

这相当于在主机会话**打印机设置**对话框中选择打印机，而不选中**使用 Windows 缺省打印机**。

lpPrtToDskAppFile

如果调用方不想获取此成员，则必须将其设置为 NULL。如果这不是 null 指针，则返回**打印到磁盘 - 附加**文件。这必须指向由调用方分配的、大小为 nBufSize 的缓冲区。

返回 API 时，此成员包含以下一项：

- 会话**打印到磁盘 - 附加**文件的全限定路径名。
- 如果未为会话配置**打印到磁盘 - 附加**文件，则为空字符串 ("")。
- 缓冲区大小不足时截断的文件名。nSizeNeeded 成员包含所需缓冲区的大小。

lpPrtToDskSepFile

如果调用方不想获取此成员，则必须将其设置为 NULL。如果这不是 null 指针，则返回**打印到磁盘 - 单独**文件。这必须指向由调用方分配的大小为 nBufSize 的缓冲区。

返回 API 时，此成员包含以下一项：

- 会话**打印到磁盘 - 单独**文件的全限定路径名。
- 如果未为会话配置**打印到磁盘 - 单独**文件，则为空字符串 ("")。
- 缓冲区大小不足时截断的文件名。nSizeNeeded 成员包含所需缓冲区的大小。

lpPrinterName

如果调用方不想获取此成员，则必须将其设置为 NULL。如果这不是 null 指针，则返回打印机的名称。这必须指向由调用方分配的大小为 nBufSize 的缓冲区。

返回 API 时，此成员具有以下一项：

- 会话中配置的特定打印机的名称（如果主机会话 nPrtMode 是 SpecificPrinter）。
- 会话中配置的 Windows 缺省打印机的名称（如果主机会话 nPrtMode 为 WinDefaultPrinter）。
- 空字符串 ("")（如果主机会话 nPrtMode 为 PrtToDskAppend 或 PrtToDskSeparate）。
- 截断的打印机名称（如果缓冲区大小不足）。nSizeNeeded 具有所需的缓冲区大小。

PrinterName 必须采用以下格式：

```
<Printer name> on <Port Name>
```

例如：

- IBM InfoPrint 40 PS on Network Port
- HP LaserJet 4050 Series PCL 6 on LPT1

ULONG * const pErrorInfo

当 API 失败且返回码为 PCS_FAILURE 时，这将填充扩展错误信息。如果不需要错误的详细信息，则调用方必须将 pErrorInfo 设置为 NULL。

下节介绍了在 PCSAPI32.H 中定义的标志。

PRINTINFO 结构的 pErrorInfo 成员的标志**PCS_PRINT_PRINTMODE_ERROR**

主机会话中未配置 PrintMode。

PCS_PRINT_PDTFILE_SIZEERR

缓冲区大小不足以容纳 lpPDTFile，因此文件名被截断。nSizeNeeded 成员包含返回 PDT 文件所需的缓冲区的实际大小。

PCS_PRINT_DSKAPPPFILE_SIZEERR

缓冲区大小不足以容纳 lpPrtToDskAppFile，因此文件名被截断。nSizeNeeded 成员包含返回**打印到磁盘 - 附加文件**所需的缓冲区的实际大小。

PCS_PRINT_DSKSEPFFILE_SIZEERR

缓冲区大小不足以容纳 lpPrtToDskSepFile，因此文件名被截断。nSizeNeeded 成员包含返回**打印到磁盘 - 单独文件**所需的缓冲区的实际大小。

PCS_PRINT_PRINTERNAME_SIZEERR

缓冲区大小不足以容纳 lpPrinterName，因此打印机名被截断。nSizeNeeded 成员包含返回打印机名所需的缓冲区的实际大小。

返回码

返回码	价值	意义
PCS_SUCCESSFUL	0	函数已成功结束。
PCS_INVALID_ID	1	指定的会话标识不正确。
PCS_DIALOG_IN_USE	3	失败，因为主机会话“页面设置”或“打印机设置”对话框正在使用中。
PCS_PRINTING	4	无法更改打印机设置，因为主机会话正在打印。应用程序必须稍后重试
PCS_FAILURE	6	无法成功检索某些打印机设置。pErrorInfo 包含无法检索设置的详细错误信息。
PCS_SYSTEM_ERROR	9	发生了系统错误。

示例

```
{
    ULONG Rc = 0, Error=0, Size;
    PRINTINFO *PrintInfo;

    PrintInfo = (PRINTINFO *) malloc(sizeof(PRINTINFO));
    memset(PrintInfo, 0, sizeof(PRINTINFO));

    PrintInfo->nBufSize = 0;

    Rc = pcsGetPrinterSettings('A', PrintInfo, &Error);
    if (Rc != PCS_SUCCESSFUL)
        printf("Failure. Return code = %d\n", Rc);
    else {
        Size = PrintInfo->nSizeNeeded;
        PrintInfo->nBufSize = Size;
        PrintInfo->lpPDTFile = (char *)malloc(sizeof(char) * Size);
        PrintInfo->lpPrtToDskAppFile = (char *)malloc(sizeof(char) * Size);
        PrintInfo->lpPrtToDskSepFile = (char *)malloc(sizeof(char) * Size);
        PrintInfo->lpPrinterName = (char *)malloc(sizeof(char) * Size);
        Rc = pcsGetPrinterSettings('A', PrintInfo, &Error);
    }
}
```

```

if (Rc != PCS_SUCCESSFUL)
    printf("Failure. Return code = %d, Extended Error = 0x%08x\n", Rc, Error);
else {
    if (PrintInfo->bPromptDialog)
        printf("PromptDialog\n");
    else
        printf("No PromptDialog\n");
    if (PrintInfo->bPDTMode)
        printf("PDT Mode\n");
    else
        printf("Not PDT Mode\n");

    switch(PrintInfo->nPrtMode) {

    case PrtToDskAppend:
        printf("Print to Disk-Append Mode\n");
        break;
    case PrtToDskSeparate:
        printf("Print to Disk-Separate Mode\n");
        break;
    case SpecificPrinter:
        printf("Specific Printer Mode\n");
        break;
    case WinDefaultPrinter:
        printf("Windows Default Printer Mode\n");
        break;
    }
    if (PrintInfo->lpPDTFile[0] == '\0')
        printf("No PDT File configured\n");
    else
        printf("PDT File = %s\n", PrintInfo->lpPDTFile);
    if (PrintInfo->lpPrtToDskAppFile[0] == '\0')
        printf("No Disk Append File configured\n");
    else
        printf("DiskAppend File=%s\n", PrintInfo->lpPrtToDskAppFile);
    if (PrintInfo->lpPrtToDskSepFile[0] == '\0')
        printf("No Disk Separate File configured\n");
    else
        printf("DiskSeparate File=%s\n", PrintInfo->lpPrtToDskSepFile);
    if ((PrintInfo->nPrtMode == SpecificPrinter) ||
        (PrintInfo->nPrtMode == WinDefaultPrinter))
        printf("Printer = %s\n", PrintInfo->lpPrinterName);
}
free(PrintInfo->lpPDTFile);
free(PrintInfo->lpPrtToDskAppFile);
free(PrintInfo->lpPrtToDskSepFile);
free(PrintInfo->lpPrinterName);
}
free(PrintInfo);
}

```

pcsSetPrinterSettings

3270	5250	VT
是	是	是

pcsSetPrinterSettings 函数会控制主机会话打印机设置（类似于**文件** → **打印机设置**对话框设置）。

函数类型

ULONG WINAPI pcsSetPrinterSettings(*char cShortSessionID, const PRINTINFO * const pPrintInfo, ULONG * const pErrorInfo*)

参数类型和描述

char cShortSessionID

表示空间短会话标识。

const PRINTINFO * const pPrintInfo

指向 PRINTINFO 结构的指针，其中提到了打印机设置。

nFlags

表示结构中哪些成员有效的位标志组合。这些标志可以单独使用，也可以通过将它们通过 OR 组合在一起来恢复属性页面来使用（在 PCSAPI32.H 中定义）。这些标志以及结构中相应的有效成员如下所示：

表示

结构中的有效成员

PCS_PRINT_PDT

bPDTMode, lpPDTFile

PCS_PRINT_PRINTMODE

nPrtMode、lpPrtToDskAppFile、lpPrtToDskSepFile、lpPrinterName

PCS_PRINT_PROMPT_DIALOG

bPromptDialog

nBufSize

必须设置为 0。这将被忽略。

nSizeNeeded

必须设置为 0。这将被忽略。

bPromptDialog

有效值如下：

- 如果为 TRUE，则在打印前显示“打印机设置”对话框。
- 如果为 FALSE，则在打印前不会显示“打印机设置”对话框。

bPDTMode

有效值如下：

- 如果为 TRUE，则将连接设置为 PDT 方式。
- 如果为 FALSE，则将连接设置为非 PDT 方式（GDI 方式）。

lpPDTFile

仅在将 bPDTMode 设置为 TRUE 时使用。如果将 bPDTMode 设置为 FALSE，将忽略此项。这是一个以 null 结束的字符串，包含 PDT 文件的名称，必须是以下一项：

- NULL
使用连接中当前配置的 PDT 文件。如果连接中尚未配置 PDT 文件，则 API 将失败并出现异常。
- 文件名，不带路径
使用 Z and I Emulator for Windows 安装路径中 PDFPDT 子文件夹中的 lpPDTFile。
- 文件的全限定路径名
如果 lpPDTFile 不存在，API 将失败。

nPrtMode

这是指示连接的 PrintMode 的枚举值。枚举数据类型 PRINTMODE 在 PCSAPI32.H 中定义。nPrtMode 设置必须是以下一项：

- **PrtToDskAppend**（打印到磁盘 - 附加方式）
这相当于在主机会话的**打印机设置** → **打印机** → **打印到磁盘**对话框中选择**追加**选项。
- **PrtToDskSeparate**（打印到磁盘 - 单独方式）
这相当于在主机会话的**打印机设置** → **打印机** → **打印到磁盘**对话框中选择**单独**选项。
- **WinDefaultPrinter**（Windows 缺省打印机方式）
这相当于在主机会话**打印机设置**对话框中选择**使用 Windows 缺省打印机**选项。
- **SpecificPrinter**（特定打印机方式）
这相当于在主机会话**打印机设置**对话框中选择打印机，而不选中**使用 Windows 缺省打印机**选项。

lpPrtToDskAppFile

仅当将 nPrtMode 设置为 PrtToDskAppend 时，才使用此项。这是一个以 null 结束的字符串，包含**打印到磁盘 - 附加**文件的名称，必须是以下一项：

- NULL
使用当前为连接中的 PrtToDskAppend 方式配置的文件。如果连接中尚未配置 PDT 文件，则 API 将失败。
- 文件名，不带路径
用户类应用程序数据目录路径用于查找文件。如果此文件存在，则使用此文件。否则，将在打印完成时创建此文件。

- 文件的全限定路径名
目录必须存在于路径中，否则 API 将失败。文件不需要存在于路径中。

lpPrtToDskSepFile

可能的值为如下所示：

- 会话的**打印到磁盘 - 单独**文件的全限定路径名。
- 如果未为会话配置**打印到磁盘 - 单独**文件，则为空字符串 ("")。
- 缓冲区大小不足时截断的文件名。nSizeNeeded 成员包含所需缓冲区的大小。

lpPrinterName

仅当将 nPrtMode 设置为 SpecificPrinter 时，才使用此项。否则，此项将被忽略。这是包含打印机名的以 null 结束的字符串。如果打印机不存在，此成员将失败。

PrinterName 必须采用以下格式：

```
<Printer name> on <Port Name>
```

例如：

- IBM InfoPrint 40 PS on Network Port
- HP LaserJet 4050 Series PCL 6 on LPT1

ULONG * const pErrorInfo

当 API 失败且返回码为 PCS_FAILURE 时，这将填充扩展错误信息。如果不需要错误的详细信息，则调用方必须将 pErrorInfo 设置为 NULL。

下节介绍了在 PCSAPI32.H 中定义的标志。

PRINTINFO 结构的 pErrorInfo 成员的标志

PCS_PRINT_PDTMODE_ERROR

这可能是由以下一个原因引起的：

- bPDTMode 设置为 TRUE，lpPDTFile 设置为 NULL，并且尚未为主机会话配置 PDT 文件。
- nPrtMode 设置为 PrtToDskAppend 或 PrtToDskSeparate，PCS_PRINT_PDT 未在 nFlags 中设置，并且主机会话尚未处于 PDT 方式。
- nPrtMode 设置为 PrtToDskAppend 或 PrtToDskSeparate，bPDTMode 设置为 FALSE。

PCS_PRINT_PDTFILE_ERROR

未找到在 lpPDTFile 中指定的文件或路径。

PCS_PRINT_PRTTODSK_FILE_ERROR

这可能是由以下一个原因引起的：

- 在字段 IpPrtToDskAppFile 或 IpPrtToDskSepFile 中指定的文件夹不存在或没有写入访问权。
- 在 IpPrtToDskSepFile 字段中指定了扩展名。

PCS_PRINT_PRINTMODE_ERROR

无法成功设置 nPrtMode。这可能是由以下一个原因引起的：

- nPrtMode 的值不是 PRINTMODE 枚举数据类型的枚举常量之一。
- nPrtMode 设置为 PrtToDskAppend, IpPrtToDskAppFile 设置为 NULL, 并且主机会话中尚未配置**打印到磁盘 - 附加文件**。
- nPrtMode 设置为 PrtToDskSeparate, IpPrtToDskSepFile 设置为 NULL, 并且主机会话中尚未配置**打印到磁盘 - 单独文件**。
- nPrtMode 设置为 SpecificPrinter, 未找到在 IpPrinterName 字段中指定的打印机。
- nPrtMode 设置为 WinDefaultPrinter, 并且未在系统中配置缺省 Windows® 打印机。
- bPDTMode 设置为 FALSE, 并且 PCS_PRINT_PRINTMODE 未在 nFlags 中设置, 但主机会话 PrintMode 为 PrtToDskAppend 或 PrtToDskSeparate。

返回码

返回码	价值	意义
PCS_SUCCESSFUL	0	函数已成功结束。
PCS_INVALID_ID	1	指定的会话标识不正确。
PCS_DIALOG_IN_USE	3	失败, 因为主机会话“页面设置”或“打印机设置”对话框正在使用中。
PCS_PRINTING	4	无法更改打印机设置, 因为主机会话正在打印。应用程序必须稍后重试。
PCS_FAILURE	6	未应用任何主机会话打印机设置。这可能是由于 PRINTINFO 结构中的部分或全部字段提供的数据无效。pErrorInfo 包含有关错误的详细信息。
PCS_SYSTEM_ERROR	9	发生了系统错误。

示例

```
{
    ULONG Rc = 0, Error=0;
    PRINTINFO *PrintInfo;
    char PDTFile[] = "epson.pdt";
    char SepFile[] = "DiskSep";

    PrintInfo = (PRINTINFO *) malloc(sizeof(PRINTINFO));
    memset(PrintInfo, 0, sizeof(PRINTINFO));

    PrintInfo->nFlags = PCS_PRINT_PDT | PCS_PRINT_PRINTMODE |
        PCS_PRINT_PROMPT_DIALOG;
    PrintInfo->nBufSize = 0;
    PrintInfo->nSizeNeeded = 0;
}
```

```
PrintInfo->bPDMode = TRUE;
PrintInfo->lpPDTFile =
    (char *)malloc(sizeof(char) * (strlen(PDTFile)+1));
strcpy(PrintInfo->lpPDTFile, PDTFile);
PrintInfo->nPrtMode = PrtToDskSeparate;
PrintInfo->lpPrtToDskSepFile =
    (char *)malloc(sizeof(char) * (strlen(SepFile)+1));
strcpy(PrintInfo->lpPrtToDskSepFile, SepFile);
PrintInfo->bPromptDialog = TRUE;
Rc = pcsSetPrinterSettings('A', PrintInfo, &Error);
if (Rc != PCS_SUCCESSFUL)
    printf("Failure. Return code = %d, Extended Error = 0x%08x\n", Rc, Error);
free(PrintInfo->lpPDTFile);
free(PrintInfo->lpPrtToDskSepFile);
free(PrintInfo);
}
```

第 6 章. 仿真器编程故障诊断

可以使用以下自助信息资源和工具来帮助进行问题故障诊断。

- 请参阅产品的发行信息，以获取已知问题、变通方法和故障诊断信息。
- 检查是否存在用于解决问题的下载或修订。
- 搜索提供的知识库，以查看是否已经说明问题的解决方案。
- 如果仍需要帮助，请联系 HCL 软件支持机构，并报告您的问题。

Z and I Emulator for Windows 主机屏幕上的部分 EHLLAPI 输入

问题

使用 HCL Z and I Emulator for Windows 时，将截断的命令文本发送到主机。

原因

如果 EHLLAPI 应用程序向主机发送 SYSREQ 键，然后尝试在主机屏幕上输入命令，则有时只会向主机发送命令的截断部分。出现此问题的原因是 Z and I Emulator for Windows 主机端的 SYSREQ 处理与 EHLLAPI 应用程序的命令输入之间缺乏同步。

当应用程序向主机发送 SYSREQ 命令时，会出现以下情况：

- OIA 将更新以指示您正在 SSCP-LU 会话中。
- Z and I Emulator for Windows 会话将 AO 命令 (SYSREQ) 发送到 3270 主机。

一旦主机接收到 SYSREQ，它将使用 0x15 或 NL（换行符）代码响应 Z and I Emulator for Windows。当 Z and I Emulator for Windows 处理此 NL 命令时，使用 NULL 填充行的其余部分，并将光标移到下一行的开头。

当 EHLLAPI 应用程序继续在主机屏幕中输入各种命令（通过 SendKeys 函数）时，甚至在 Z and I Emulator for Windows 会话从主机收到 NL 命令并对其进行处理之前，就会出现问题。因此，首先将输入命令的一部分输入到屏幕上，同时处理 NL 命令并将光标移到下一行。然后在下一行输入命令的其余部分。因此，只有被截断的命令的第二部分被发送到主机，从而导致错误的结果。

分辨率

此问题的解决方法是强制 EHLLAPI 应用程序等待 NL 命令被接收和处理，然后再继续将命令输入主机屏幕。一旦会话通知 EHLLAPI 应用程序 SYSREQ 的主机响应已被处理，EHLLAPI 应用程序就可以继续其输入（因为会话现在处于接受新输入的正确状态）。为此，使用以下 EHLLAPI 函数调用：

```
Start_Host_Notification (23)
Pause (18)
Set_Session_Parameters (9)
Query_Host_Update (24).
```

EHLLAPI 应用程序中的可能代码如下所示：

- 调用 Sendkeys(@A@H)。这会将 SYSREQ 命令发送到会话。
- 使用输入 B 调用 StartHostNotify, 其中 B 表示 OIA 和 PS 的通知。这会告诉会话在主机更新会话的 OIA 和/或 PS 时通知 EHLLAPI 应用程序。
- 调用暂停, 指定足够的超时时间。这会导致 EHLLAPI 应用程序等待, 直到会话通知它会话的 OIA 和/或 PS 的主机更新。当会话收到等待时间最长的 SYSREQ 命令主机响应时, 就会发生这种情况。请注意, 如果超过了超时值, 并且未收到主机通知, 则仍会返回“暂停”功能调用。

此外, 要使此“暂停”调用生效, 必须使用 Set_Session_Parameters (9) 功能调用启用 IPAUSE 选项。这是必需的, 因为它告诉暂停 API 调用在主机通知会话 OIA 和/或 PS 更新时返回。

如果由于 OIA/PS 更新 (主机通知) 而返回“暂停”, 则返回值为 26。如果是这种情况, 则可以发送主机命令。否则, 必须再次等待主机响应。

当 EHLLAPI 应用程序知道主机已更新 OIA 或表示空间 (或两者) 时, 它可以继续执行命令。QueryHostUpdate 用于检查更新了哪些内容: 即, 是仅更新了 OIA (返回码 21), 还是仅更新了 PS (返回码 22), 还是同时更新了 OIA 和 PS (返回码 23)。

例如, EHLLAPI 代码可能类似于以下部分:

```
Send Keys(@A@H) /* Send SYSREQ command to the host */

Start Host Notification with 'B' in byte 2 /* Enable notification to EHLLAPI application
                                             when session's OIA and/or PS are updated */

Set Session Parms with IPAUSE option /* Allow Pause to be interrupted */

Label WW:

Pause for 15 seconds /* 15 secs is a sample time-out value */

retVal = Query Host Update /* Store return value of QueryHostUpdate() into retVal */

If (retVal = 21 or 22 or 23) /* OIA and/or PS was updated */

Send Keys("Your Input Command to host") /* Send input command to host */

else

goto (Label WW)

Stop Host Notification /* Disable host notification */
```

这是最适合解决此问题的解决方案, 因为 EHLLAPI 应用程序在发送其命令输入之前, 等待允许会话接收和处理 SYSREQ 主机响应所需的最短时间。

另一个解决方案是在 EHLLAPI 应用程序中的 SYSREQ 命令和后续命令之间添加一个延迟 [例如, Sleep(1000)], 以便会话有足够的时间接收和处理主机响应。但是, 这种解决方案不是最好的, 因为延迟可能会太短或过长。

有关 3270 SYSREQ 功能的详细信息, 请参阅 RFC 2355 (TN3270 增强)。

HCL Z and I Emulator for Windows VBHLLAPI 样本无法在 FDCC Windows Vista 中运行

问题

HCL Z and I Emulator for Windows VBHLLAPI 样本使用 comdlg32.ocx 提供的控件，而 Microsoft Windows Vista 的 Federal Desktop Core Configuration (FDCC) 中未安装这些控件。

原因

VBHLLAPI 使用由 Microsoft comdlg32.ocx 模块提供的 ActiveX 和 Common Dialog 控件。出于安全性考虑，Windows Vista 的 FDCC 不包含此特定模块。

分辨率

Windows Vista 的 FDCC 版本是定制的，建议不要进行更改。

如果需要运行包含 VBHLLAPI 的 HLLAPI 样本，则必须将 comdlg32.ocx 模块从标准 Windows Vista 机器复制到 FDCC Windows Vista 安装的 `\Windows\System32\` 目录中。

接着重新引导系统，以使更改生效。

附录 A. 受 EHLLAPI 支持的查询应答数据结构

本附录列出并定义 PC/3270 的 EHLLAPI 结构化字段接口支持的查询应答结构。请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*; 如果是 IBM 许可程序, 请参阅特定许可程序的文档。



注:

1. EHLLAPI 必须扫描查询应答缓冲区, 才能找到目标/来源标识 (DOID) 定制参数 (SDP), 以便结构化字段支持正常工作且可靠。接着, 在 DOID 字段中填充分配的标识。
2. 应用程序应在应用程序的专用内存中构建查询应答数据结构。
3. 只对查询应答数据执行粗略检查。仅检查结构的标识和长度是否有效。
4. 每个查询应答开头的 2 字节长度字段**不是字节反向**。
5. 每个主机会话仅允许一个分布式数据管理 (DDM) 基本类型连接。如果 DDM 连接支持 DOID 的 SDP, 则允许多个连接。
6. 如果接收到非零返回码, 表示应用程序已连接到选定会话 (RC 32 或 39), 请谨慎使用该表示空间。可能会导致与文件传输和其他 EHLLAPI 应用程序发生冲突。

DDM 查询应答

支持多种 DDM 查询应答格式。以下是其中一些格式:

表 15. DDM 查询应答基本格式

偏移量	Length	内容	意义
0	1 个字	Length	结构长度
2	1 个字节	X'81'	查询应答标识
3	1 个字节	X'95'	查询应答类型
4-5	2 字节	FLAGS	保留
6-7	2 字节	LIMIN	进站传输中允许的最大 DDM 字节数
8-9	2 字节	LIMOUT	出站传输中允许的最大 DDM 字节数
10	1 个字节	NSS	子集标识的数量
11	1 个字节	DDMSS	DDM 子集标识

DDM 应用程序名称自定义参数

DDM 应用程序名称自定义参数为主机应用程序提供包含 DDM 辅助设备控制的应用程序的名称。控制应用程序由直接访问自定义参数中的 DOID 标识。

此自定义参数是可选参数, 但如果主机应用程序要在远程工作站上存在多个应用程序时标识不同的 DDM 辅助设备, 则需要使用此参数。

表 16. DDM 应用程序名称自定义参数

偏移量	Length	内容	意义
0	1 个字节	Length	参数长度
1	1 个字节	X'02'	DDM 应用程序名称
2-n	n-2 字节	NAME	远程应用程序的名称

NAME

该名称由 8 个或更少字符组成，是主机应用程序与远程工作站中的应用程序相关的方式。主机和远程应用程序用户有责任确保每端的应用程序都能理解该名称

PCLK 协议控制自定义参数

PCLK 协议控制自定义参数指明，PCLK 协议控制结构化字段 ID = X'1013' 可用于发往或来自 DDM 辅助设备处理器的数据流中的入站和出站。

表 17. DDM PCLK 辅助设备自定义参数

偏移量	Length	内容	意义
0	1 个字节	X'04'	参数长度
1	1 个字节	X'03'	PCLK 协议控制
2-3	2 字节	VERS	协议版本

VERS

VERS 中提供的值用于指示返回查询应答时终端中安装的 PCLK 版本。例如，X'0001' 表示 PCLK V1.1。

有关此查询应答的字段定义，请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*。

基本 DDM 查询应答格式

以下查询应答格式是一些可能的基本 + SDP（自定义参数）组合的示例。并非所有组合都会显示。

表 18. 具有名称和直接访问自定义参数的基本 DDM 查询应答格式

偏移量	Length	内容	意义
0	1 个字	Length	结构长度（包括自定义参数）
2	1 个字节	X'81'	查询应答标识
3	1 个字节	X'95'	查询应答类型
4-5	2 字节	FLAGS	保留
6-7	2 字节	LIMIN	入站传输中允许的最大 DDM 字节数
8-9	2 字节	LIMOUT	出站传输中允许的最大 DDM 字节数
10	1 个字节	NSS	支持的子集数
11	1 个字节	DDMSS	DDM 子集标识

表 18. 具有名称和直接访问自定义参数的基本 DDM 查询应答格式 (续)

偏移量	Length	内容	意义
12	1 个字节	长度 (n+2)	参数长度
13	1 个字节	X'02'	DDM 应用程序名称
14- (13+n)	n 个字节	姓名	远程应用程序的名称
14+n	1 个字节	X'04'	参数长度
15+n	1 个字节	X'01'	直接访问标识
16+n - 17+n	2 字节	DOID	由子系统分配的目标/来源标识

自定义参数从偏移量 12 和 (14 + n) 开始, 其中 n 是偏移量 14 处提供的应用程序名称的长度。

有关此查询应答的字段定义, 请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*。

表 19. 具有直接访问和名称自定义参数的基本 DDM 查询应答格式

偏移量	Length	内容	意义
0	1 个字	Length	结构长度 (包括自定义参数)
2	1 个字节	X'81'	查询应答标识
3	1 个字节	X'95'	查询应答类型
4-5	2 字节	FLAGS	保留
6-7	2 字节	LIMIN	进站传输中允许的最大 DDM 字节数
8-9	2 字节	LIMOUT	出站传输中允许的最大 DDM 字节数
10	1 个字节	NSS	支持的子集数
11	1 个字节	DDMSS	DDM 子集标识
12	1 个字节	X'04'	参数长度
13	1 个字节	X'01'	直接访问标识
14-15	2 字节	DOID	由子系统分配的目标/来源标识
16	1 个字节	长度 (n+2)	参数长度
17	1 个字节	X'02'	DDM 应用程序名称
16+n - 17+n	n 个字节	姓名	远程应用程序的名称

自定义参数从偏移量 12 和 16 开始。

有关此查询应答的字段定义, 请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*。

IBM 辅助设备查询应答

辅助设备查询应答用于向主机应用程序指示支持 IBM 辅助设备使用 IBM 定义的数据流。有关更多详细信息, 请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*。

此功能受支持时，将入站传输查询应答，以应答“读取分区”结构化字段，从而指定查询或查询列表（QCODE List = X'9E'、Equivalent 或 All）。

当工作站支持多个辅助设备时，必须为每个设备发送 IBM 辅助设备查询应答。

任选参数

查询应答的基本部分中显示的所有参数都必须存在。未使用的参数设置为 X'00'。

必须至少存在一个自定义参数。

表 20. 带直接访问自定义参数的 IBM 辅助设备基本格式

偏移量	Length	内容	意义
0-1	1 个字	Length	结构长度（包括自定义参数）
2	1 个字节	X'81'	查询应答标识
3	1 个字节	X'9E'	IBM 辅助设备应答
4	1 字节	FLAGS	保留
	BIT 0	QUERY B'1'	读取部分（查询、查询列表） 辅助设备支持查询
	1-7	RES	保留，必须为 B'0'
5	1 个字节	FLAGS	保留
6-7	2 字节	LIMIN	入站传输中允许的最大 DDM 字节数
8-9	2 字节	LIMOUT	出站传输中允许的最大 DDM 字节数
10	1 个字节	TYPE	支持的辅助设备类型
		X'01'	IBM 辅助设备显示
		X'02'	IBM 辅助设备打印机
		其他	保留
11	1 个字节	X'04'	参数长度
12	1 个字节	X'01'	直接访问
13-14	1 个字	DOID	由子系统分配的目标/来源标识

QUERY	对于所有 IBM 辅助设备，必须将此位设置为 B'1'，以表明它支持接收读取分区（查询、查询列表）。然后，主机应用程序可以使用定向到辅助设备的读取分区来确定其特性。目标/来源结构化字段用于将读取分区结构化字段定向到辅助设备。 IBM 辅助设备的最低支持级别是返回 Null 查询应答，以响应读取分区。
LIMIN	陈述入站传输中可发送的最大字节数。LIMIN 值 X'0000' 表示对入站传输的字节数没有实现限制。
LIMOUT	陈述出站传输中可发送到 IBM 辅助设备的最大字节数。LIMOUT 值 X'0000' 表示出站传输的字节数没有实现限制。

TYPE	标识受支持的辅助设备。两个值有效。一个标识辅助显示，另一个标识辅助打印机。所有其他值均为保留值。
-------------	--

IBM 辅助设备处理器支持两个自定义参数 01 和 03。这些可在 [表 21: IBM 辅助设备直接访问自定义参数 \(on page 201\)](#) 中定义。

直接访问自定义参数

直接访问自定义参数提供用于 IBM 辅助设备直接访问中的目标/来源结构化字段的标识。

基本查询应答始终需要此 SDP。

表 21. IBM 辅助设备直接访问自定义参数

偏移量	Length	内容	意义
0	1 个字节	X'04'	参数长度
1	1 个字节	X'01'	直接访问标识
2-3	2 字节	DOID	目标/来源 ID

DOID

这些字节中的值用于目标/来源结构化字段的标识字段，用于将辅助设备标识为后续数据的目标或来源。

PCLK 协议控制自定义参数

PCLK 协议控制自定义参数的存在表明，PCLK 协议控制结构化字段 ID = X'1013' 可用于发往或来自 IBM 辅助设备处理器的数据流中的入站和出站。

表 22. IBM 辅助设备 PCLK 自定义参数

偏移量	Length	内容	意义
0	1 个字节	X'04'	参数长度
1	1 个字节	X'03'	PCLK 协议控制
2-3	2 字节	VERS	协议版本

VERS

VERS 中提供的值用于指示返回查询应答时终端中安装的 PCLK 版本。例如，X'0001' 表示 PCLK V1.1。

有关此查询应答的字段定义，请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*。

产品定义的查询应答

IBM 产品使用在 X'9C' 数据结构中注册的子标识来使用此查询应答。产品定义的数据流查询应答指示支持使用 IBM 产品定义的数据流的 3270DS 工作站辅助设备。数据流不是由具有可标识控制点（如体系结构审查板）的格式体系结构文档定义的。

当辅助设备支持 IBM 产品定义的数据流时，此查询应答将以入站方式传输，以应答查询列表（QCODE List = X'9C' 或 All）。

任选参数

查询应答的基本部分中显示的所有参数和直接访问自定义参数都必须存在。

产品定义的查询应答的格式如下所示：

表 23. IBM 产品定义的查询应答基本格式

偏移量	Length	内容	意义
0-1	1 个字	Length	结构长度（包括自定义参数）
2	1 个字节	X'81'	查询应答标识
3	1 个字节	X'9C'	IBM 产品定义的数据流
4-5	2 字节	FLAGS	保留
6	1 个字节	REFID	引用标识
7	1 个字节	SSID	子集标识
8	1 个字节	X'04'	参数长度
9	1 个字节	X'01'	直接访问
10-11	1 个字	DOID	由子系统分配的目标/来源标识

产品定义的查询应答的 REFID（偏移量 6）和 SSID（偏移量 7）的有效值如下所示：

表 24. IBM 产品定义的查询应答的有效 REFID 和 SSID 值

REFID	SSID	产品和数据流文档
X'01'		5080 图形系统： 此引用标识表示辅助设备支持 5080 图形系统数据流。5080 图形体系结构、结构化字段、子集标识、DOID 和关联功能集的描述在 <i>IBM 5080 Graphics System Principles of Operation</i> 中定义
	X'01' X'02'	5080 HGFD 图形子集 5080 RS232 端口子集
X'02'		WHIP API（写入时替换为 SRL 名称） 此引用标识表示辅助设备支持 WHIP API 数据流。WHIP API 体系结构的描述在 <i>IBM RT PC Workstation Host Interface Program Version 1.1 User's Guide and Reference Manual</i> 中定义
	X'01'	WHIP 子集 1
X'03' 至 X'FF'		所有其他值均为保留值。

IBM 产品定义的处理器仅支持直接访问自定义参数。它在 [表 25: IBM 产品定义的直接访问自定义参数 \(on page 203\)](#) 中定义。

直接访问自定义参数

直接访问 ID 自定义参数的存在表示可以使用目标/来源结构化字段直接访问辅助设备。当支持多个使用产品定义的数据流的辅助设备时，必须提供单独的产品定义的数据流查询应答，并且每个应答都有唯一的 DOID。

表 25. IBM 产品定义的直接访问自定义参数

偏移量	Length	内容	意义
0	1 个字节	X'04'	参数长度
1	1 个字节	X'01'	直接访问标识
2-3	2 字节	DOID	目标/来源 ID

DOID

这些字节中的值用于目标/来源结构化字段的标识字段，用于将辅助设备标识为后续数据的目标或来源。

文档交换体系结构查询应答

此查询应答指示支持的文档交换体系结构 (DIA) 功能集。DIA 查询应答的格式如下所示：

表 26. IBM DIA 基本格式

偏移量	Length	内容	意义
0	1 个字	Length	结构长度（包括自定义参数）
2	1 个字节	X'81'	查询应答标识
3	1 个字节	X'97'	IBM DIA
4-5	2 字节	FLAGS	保留
6-7	2 字节	LIMIN	进站传输中允许的最大 DDM 字节数
8-9	2 字节	LIMOUT	出站传输中允许的最大 DDM 字节数
10	1 个字节	NFS	随后的 3 字节功能集标识数
11-13	3 字节	DIAFS	DIA 功能集标识
14- (13+(N*3))	N*3 字节	DIAFSs	其他 DIA 功能集标识
14+(N*3)	1 个字节	X'04'	参数长度
15+(N*3)	1 个字节	X'01'	直接访问
16+(N*3)	1 个字	DOID	由子系统分配的目标/来源标识

DIA 辅助设备处理器仅支持直接访问自定义参数。它在 [表 27: IBM 产品定义的直接访问自定义参数 \(on page 204\)](#) 中定义。

直接访问标识自定义参数的存在表示可以使用目标/来源结构化字段直接访问辅助设备。

表 27. IBM 产品定义的直接访问自定义参数

偏移量	Length	内容	意义
0	1 个字节	X'04'	参数长度
1	1 个字节	X'01'	直接访问标识
2-3	2 字节	DOID	目标/来源 ID

DOID

这些字节中的值用于目标/来源结构化字段的标识字段，用于将辅助设备标识为后续数据的目标或来源。

有关此查询应答的字段定义，请参阅 *IBM 3270 Information Display System Data Stream Programmer's Reference*。

附录 B. 与 Communication Manager/2 EHLLAPI 的差异

本附录介绍 Z and I Emulator for Windows 的 EHLLAPI 和 Communication Manager/2 的 EHLLAPI 之间的差异。

以下 EHLLAPI 功能与 Communication Manager/2 中同名的功能不同。在使用这些功能时，您需要了解它们的区别：

- 设置会话参数 (9)
- 复制 OIA (13)
- 将字符串复制到 PS (15)
- 存储管理器 (17)
- 将字符串复制到字段 (33)
- 获取键 (51)
- 窗口状态 (104)
- 查询会话 (10)
- 结构化字段的连接 (120)
- 分配通信缓冲区 (123)
- ASCII 助记符

设置会话参数 (9)

设置选项

Z and I Emulator for Windows 不提供 Communication Manager 提供的以下设置选项：

OLDOIA, NEWOIA
COMPCASE, COMPICASE
OLD5250OIA, NEW5250OIA

返回参数

终止**设置会话参数** (9) 功能时，Communication Manager 会返回有效数据字符串的长度作为第三个参数，即数据字符串长度。但是，Z and I Emulator for Windows 会返回多个有效设置选项作为数据字符串长度。

EAB 选项

在 Communication Manager/2 中，当在**设置会话参数** (9) 功能中指定 EAB 选项时，颜色重新映射会影响由**复制 PS** (5) 或**将 PS 复制到字符串** (8) 功能复制的 EAB 属性中的字符颜色值。

但在 Z and I Emulator for Windows 中，EAB 属性中的字符颜色值取决于表示空间的内容，而不考虑颜色重新映射，并且不受颜色重新映射的影响。

复制 OIA (13)

复制 OIA (13) 功能在 Communication Manager/2 和 Z and I Emulator for Windows 之间存在以下不同之处。有关组和列位置的更多信息，请参阅 [复制 OIA \(13\) \(on page 45\)](#)。

- 字节位置 21
 - Z and I Emulator for Windows 返回 X'F6'。
 - Communication Manager/2 返回 X'20'。
- 字节位置 61–63
 - Z and I Emulator for Windows 不返回打印机信息。
 - Communication Manager/2 返回打印机信息。
- 组 3: Shift 状态

Communication Manager/2 不返回位 2 的值。位 2 已保留，位 0 包含 Upper Shift 和 Caps Lock。
- 组 8 字节 1: 输入被禁止
 - Z and I Emulator for Windows 不返回位 6 (设备无法正常运行)。
 - Communication Manager/2 可以返回位 6。
- 组 8 字节 3: 输入被禁止
 - Z and I Emulator for Windows 不返回位 1 (操作员未授权) 和位 2 (操作员未授权 -f)。
 - Communication Manager/2 可以返回位 1 和 2。
- 组 8 字节 4: 输入被禁止
 - Z and I Emulator for Windows 不返回位 2 (系统等待)。
 - Communication Manager/2 可以返回位 2。
- 组 10: 突出显示组 2
 - Z and I Emulator for Windows 不返回位 0 (已选择)。
 - Communication Manager/2 可以返回位 0。
- 组 11: 颜色组 2
 - Z and I Emulator for Windows 不返回位 0 (已选择)。
 - Communication Manager/2 可以返回位 0。
- 组 13: 打印机状态
 - 在 Z and I Emulator for Windows 中，此组被保留。
 - Communication Manager/2 可以返回此组。
- 组 14: 图形

Communication Manager/2 不返回位 0 (图形光标)。

将字符串复制到 PS (15)

在 Communication Manager/2 中，**设置会话参数 (9)** 功能的 EAB 选项会影响**将字符串复制到 PS** 功能。指定 EAB 选项时，将与文本数据大小相同的属性数据传递给具有文本数据的函数。

但在 Z and I Emulator for Windows 中，要传递的数据仅为文本数据，而不考虑 EAB 选项。如果要将同一接口用于 Communication Manager/2，请使用**设置会话参数 (9)** 的 `PUTEAB` 选项。

存储管理器 (17)

Communication Manager/2 提供的**存储管理器 (17)** 功能不受 Z and I Emulator for Windows 支持。使用 Windows® 提供的 API 为应用程序分配内存。

将字符串复制到字段 (33)

在 Communication Manager/2 中，指定**设置会话参数 (9)** 功能的 EAB 选项时，属性数据将作为数据的一部分传递到该功能。因此，指定 EAB 选项时，将与文本数据大小相同的属性数据传递给具有文本数据的函数。

但在 Z and I Emulator for Windows 中，EAB 选项不会影响**将字符串复制到字段 (33)** 功能的数据内容。要传递的数据不是属性数据，而只是文本数据。如果要同一接口用于 Communication Manager/2，请使用**设置会话参数 (9)** 的 `PUTEAB` 选项。

获取键 (51)

如果所传递键的 Shift 状态不是仿真器会话识别的键或功能，Communication Manager/2 会使用 @A、@S 或 @r 返回 Shift 状态。Z and I Emulator for Windows 不支持这些 ASCII 助记符。

窗口状态 (104)

EHLLAPI 函数 104 (PM_WINDOW_STATUS)“query extended status” 命令 (0x03) 将返回仿真器表示空间窗口的句柄。这与函数的定义和 Communication Manager/2 实现一致。但是，Z and I Emulator for Windows EHLLAPI 返回框架窗口的句柄。使用此函数为 Z and I Emulator for Windows 编写的 EHLLAPI 应用程序需要使用返回的窗口句柄的父项。

查询会话 (10)

在 Communication Manager/2 中，返回个人计算机的描述符。但是，未在 Z and I Emulator for Windows 中返回描述符。

结构化字段的连接 (120)

Communication Manager/2 提供的通信连接状态的事件对象不在 Z and I Emulator for Windows 中。

分配通信缓冲区 (123)

在 Communication Manager/2 中，请求的缓冲区大小的最大值为 64 KB 减 8 字节 (X'FFF8')。

但在 Z and I Emulator for Windows 中，这是 64 KB 减 256 字节 (X'FF00')。

ASCII 助记符

以下 ASCII 助记符在 Z and I Emulator for Windows 中不受支持：

助记符	意义
@A@N	获取游标
@A@O	定位游标
@A@X	十六进制
@A@Y	Cmd (功能) 键
@A@a	破坏性退格
@S@A	擦除 EOL
@S@B	字段前进
@S@C	字段退格
@S@D	有效字符退格
@S@P	POR (仅用于发送)
@S@T	跳转至任务管理器
@/	队列超限 (仅限在 获取键 功能中)

获取请求完成 (125)

Z and I Emulator for Windows 不支持空白或 Null 会话标识。

附录 C. 声明

本信息是为在美国提供的产品和服务编写的。HCL 可能在其他国家或地区不提供本资料中讨论的产品、服务或功能。请咨询您当地的 HCL 代表，以获取有关您所在区域当前可获得的产品和服务的信息。任何对 HCL 产品、程序或服务的提及并非意在明示或默示只能使用 HCL 产品、程序或服务。只要不侵犯 HCL 的知识产权，就可以改用任何具有同等功能的产品、程序或服务。但是，评估和验证任何非 HCL 产品、程序或服务，则由用户自行负责。

HCL 可能已拥有或正在申请与本资料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证。您可以用书面方式将许可查询寄往：

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
注意：Office of the General Counsel

HCL TECHNOLOGIES LTD. “按现状” 提供本出版物，不附有任何种类的（无论是明示的还是默示的）保证，包括但不限于默示的有关不侵权、适销和适用于某特定用途的保证。某些管辖区域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。

本资料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。HCL 可随时对本资料中描述的产品和/或程序进行改进和/或更改，恕不另行通知。

本资料中对非 HCL 文档或非 HCL Web 站点的任何提及只是为方便起见才提供，不以任何方式充当对那些文档或 Web 站点的保证。那些 Web 站点中的资料不是本 HCL 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

HCL 可以按它认为适当的任何方式使用或分发您所提供的任何信息，而无须对您承担任何责任。

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：

HCL
330 Potrero Ave.
Sunnyvale, CA 94085
USA
注意：Office of the General Counsel

只要遵守适当的条件和条款，包括某些情形下支付一定数额的费用，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 HCL 依据 HCL 客户协议、HCL 国际软件许可协议或任何同等协议中的条款提供。

此处讨论的性能数据是在特定操作条件下得出的。实际结果可能会有差异。

涉及非 HCL 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。HCL 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 HCL 产品的声明。有关非 HCL 产品性能的问题应当向这些产品的供应商提出。

本资料包括日常业务运作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，若实际人员或业务企业与此相似，纯属巧合。

商标

HCL、HCL 徽标和 hcl.com 是 HCL Technologies Ltd Corp., 在全球许多管辖区域的商标或注册商标。其他产品和服务名称可能是 IBM® 或其他公司的商标。