

# 顶点 ABOSS2 外围接入 开发文档

---

V2.0.1.4

福建顶点软件股份有限公司

2007-05-01

## 目录

1	前言 .....	4
2	修改历史 .....	4
3	文件清单 .....	4
4	概要说明 .....	5
5	函数说明 .....	5
5.1	Fix_Initialize 接口库初始化 .....	5
5.2	Fix_Uninitialize 接口库卸载 .....	5
5.3	Fix_SetAppInfo 设置第三方软件名称和版本 .....	5
5.4	Fix_SetDefaultInfo .....	6
5.5	Fix_Connect .....	6
5.6	Fix_Close .....	6
5.7	Fix_AllocateSession 申请会话句柄 .....	6
5.8	Fix_ReleaseSession 释放会话句柄 .....	7
5.9	Fix_SetTimeOut 设置会话超时 .....	7
5.10	Fix_SetWTFS 设置会话委托方式 .....	7
5.11	Fix_SetFBDM 设置业务的来源营业部 .....	7
5.12	Fix_SetDestFBDM 设置业务的目标营业部 .....	8
5.13	Fix_SetNode 设置业务站点信息 .....	8
5.14	Fix_SetGYDM 设置柜员号 .....	8
5.15	Fix_CreateHead .....	8
5.16	Fix_SetString 设置字符类型请求数据 .....	9
5.17	Fix_SetLong 设置整型类型请求数据 .....	9
5.18	Fix_SetDouble 设置浮点类型请求数据 .....	9
5.19	Fix_Run 业务提交 .....	9
5.20	Fix_AsyncRun 异步提交 .....	10
5.21	Fix_IsReplied 是否完全收到应答 .....	10
5.22	Fix_Cancel 取消应答等待 .....	10
5.23	Fix_GetCode 读取错误编码 .....	10
5.24	Fix_GetErrMsg 读取错误信息 .....	10

---

5.25	Fix_GetCount	读取应答总行数 .....	11
5.26	Fix_GetItem	读取应答数据 .....	11
5.27	Fix_GetLong	读取应答数据(整型) .....	11
5.28	Fix_GetDouble	读取应答数据(浮点).....	12
5.29	Fix_HaveItem	查看指定应答数据.....	12
5.30	Fix_SetToken	设置业务令牌 .....	12
5.31	Fix_GetToken	读取业务令牌.....	12
5.32	Fix_Encode	密码加密函数.....	13
5.33	Fix_AddBackupSvrAddr	新增备份服务器 .....	13

# 1 前言

为了方便外围系统接入ABOSS2架构的后台服务系统中，顶点公司提供基于API方式的技术文档说明。

## 2 修改历史

发布日期	版本号	修改记录
2007.05.09	2.0.0.1	创建发布
2008.08.01	2.0.1.4	新增三个函数 Fix_SetAppInfo Fix_GetToken Fix_SetToken

## 3 文件清单

第三方接口的文件清单

文件名称	版本号	说明
FixApi.dll	2.0.1.4	第三方接入接口动态库
FixApi.lib	—	第三方接入接口连接库
FixApi.h	2.0.1.4	第三方接入接口头文件
Libeay32.dll	0.9.8.0	OpenSSL Shared Library
SSLeay32.dll	0.9.8.0	OpenSSL Shared Library
MSVCP60.dll	6.2.3104.0	Microsoft (R) C++ Runtime Library
TestApi.rar		开发包Demo
FixApi.pdf		开发技术说明文件

## 4 概要说明

系统提供基于 TCP 接入的接口方式，并提供基于 Fix 的 tag 标识符来表示具体的数据信息。(其中具体的 tag 值由顶点其他业务文档实现)系统内部实现自动重连服务器的功能。系统实现两个句柄类型，分别是连接句柄和会话句柄，用户使用该连接句柄分配出相应的会话句柄用于一次的业务请求交互过程。特别要注意的是由应用层分配的句柄，必须有应用层主动释放，以防止内存泄露。为了方便用户进行的快速开发，系统提供基于 C++ 的 Demo 代码。考虑到系统的性能，建议使用长连接进行数据的交互。

## 5 函数说明

### 5.1 Fix\_Initialize 接口库初始化

函数名称	Fix_Initialize 第三方接口库初始化。只有调用一次就可以，不能多次调用。
返回数据	返回True表示初始化成功；False表示失败。
函数原型	BOOL Fix_Initialize()

### 5.2 Fix\_Uninitialize 接口库卸载

函数名称	Fix_Uninitialize 第三方接口库卸载。只能调用一次就可以，不能多次调用。
返回数据	返回True表示初始化成功；False表示失败。
函数原型	BOOL Fix_Uninitialize ()

### 5.3 Fix\_SetAppInfo 设置第三方软件名称和版本

函数名称	设置第三方软件名称和版本； 本函数必须调用的； 设置调用第三方软件接口的名称以及软件版本。
参数说明	pszAppName -- [in] 第三方应用名称。 pszVer -- [in] 第三方软件版本号
返回数据	返回true表示初始化成功；False表示失败

## 5.4 Fix\_SetDefaultInfo

函数名称	Fix_SetDefaultInfo 设置每个业务请求包缺省的头信息。
参数说明	pszUser -- [in] 系统要求的柜员代码(八位字符串)。 pszWTFS -- [in] 委托方式。 pszFBDM -- [in] 发生营业部的代码(四位字符串) pszDestFBDM -- [in] 目标营业部的代码(四位字符串)
返回数据	返回True表示初始化成功；False表示失败。
函数原型	BOOL Fix_SetDefaultInfo ( const char *pszUser, const char *pszWTFS, const char *pszFBDM, const char *pszDestFBDM )

## 5.5 Fix\_Connect

函数名称	Fix_Connect 连接到顶点中间件服务器。
参数说明	pszAddr -- [in]连接的服务器地址；格式为：“ip地址@端口/tcp” pszUser -- [in] 通信用户名称；由客户提供。 pszPwd -- [in] 通信用户的密码；由客户提供。 nTimeOut -- [in] 连接超时的时间。单位为秒。
返回数据	系统返回类型为HANDLE_CONN的连接句柄。 如果连接失败则返回0；成功不为0；
函数原型	HANDLE_CONN Fix_Connect ( const char *pszAddr, const char *pszUser, const char *pszPwd, long nTimeOut )

## 5.6 Fix\_Close

函数名称	Fix_Close 与顶点中间件连接关闭。
参数说明	conn -- [in] 类型为HANDLE_CONN的连接句柄。该句柄由Fix_Connect生成的。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_Close( HANDLE_CONN conn )

## 5.7 Fix\_AllocateSession 申请会话句柄

函数名称	根据一个连接句柄分配一个会话句柄，用于数据交互。由用户自己在应用层通过调用Fix_AllocateSession分配出来的对象必须由用户自己调用Fix_ReleaseSession来释放。反之则不必要。原则上遵循“谁分配谁释放”。这样就可以确保内存不泄露。
参数说明	conn -- [in] 类型为HANDLE_CONN的连接句柄。该句柄由Fix_Connect

	生成的。
返回数据	返回值类型为HANDLE_SESSION的会话对象；如果对象值为空表示对象分配失败。否则表示成功。
函数原型	HANDLE_SESSION Fix_AllocateSession( HANDLE_CONN conn )

## 5.8 Fix\_ ReleaseSession 释放会话句柄

函数名称	Fix_ReleaseSession 释放会话句柄。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话对象。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_ReleaseSession( HANDLE_SESSION sess )

## 5.9 Fix\_ SetTimeOut 设置会话超时

函数名称	设置会话交互的超时时间。
参数说明	conn -- [in] 类型为HANDLE_CONN的连接句柄。该句柄由 Fix_Connect生成的。 timeout -- [in] 业务应答超时时间；单位为秒，系统默认为30秒。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_SetTimeOut ( HANDLE_SESSION sess, long timeout )

## 5.10 Fix\_ SetWTFS 设置会话委托方式

函数名称	Fix_SetWTFS 设置会话的委托方式. (必须在Fix_CreateHead函数之前调用)
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 字符串类型。用于表示客户的接入方式；比如电话委托，磁卡委托，互联网委托等。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_SetWTFS ( HANDLE_SESSION sess, const char *val )

## 5.11 Fix\_ SetFBDM 设置业务的来源营业部

函数名称	Fix_SetFBDM 设置会话的发生营业部代码。 (必须在Fix_CreateHead或者Fix_CreateReq函数之前调用)
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 字符串类型。用于表示客户业务发生的营业代码, 必须是四位的营业部代码；
返回数据	返回值为True表示成功；False表示失败。

函数原型	BOOL Fix_SetFBDM ( HANDLE_SESSION sess, const char *val )
------	---

## 5.12 Fix\_SetDestFBDM 设置业务的目标营业部

函数名称	设置会话的目标营业部代码。 (必须在Fix_CreateHead或者Fix_CreateReq函数之前调用)
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 字符串类型。用于表示客户业务发生的营业代码, 必须是四位的营业部代码;
返回数据	返回值为True表示成功; False表示失败。
函数原型	BOOL Fix_SetDestFBDM ( HANDLE_SESSION sess, const char *val )

## 5.13 Fix\_SetNode 设置业务站点信息

函数名称	Fix_SetNode 设置会话的业务发生的节点信息。 (必须在Fix_CreateHead或者Fix_CreateReq函数之前调用)
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 字符串类型。一般是客户网卡信息或IP地址。
返回数据	返回值为True表示成功; False表示失败。
函数原型	BOOL Fix_SetNode( HANDLE_SESSION sess, const char *val )

## 5.14 Fix\_SetGYDM 设置柜员号

函数名称	设置会话的业务发生的柜员代码信息。 (必须在Fix_CreateHead或者Fix_CreateReq函数之前调用)
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 字符串类型。一般是客户网卡信息或IP地址。
返回数据	返回值为True表示成功; False表示失败。
函数原型	BOOL Fix_SetGYDM ( HANDLE_SESSION sess, const char *val )

## 5.15 Fix\_CreateHead

函数名称	设置会话的业务功能号. 具体的功能号的定义请参照【第三方接入业务接口文档】
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 整型。即为系统提供的业务功能号。
返回数据	返回值为True表示成功; False表示失败。
函数原型	BOOL Fix_CreateHead ( HANDLE_SESSION sess, long nFunc )

注: Fix\_CreateReq() 使用方法和功能域 Fix\_CreateHead()一样



### 5.16 Fix\_SetString 设置字符类型请求数据

函数名称	Fix_SetString 设置该会话要发送给中间件的业务的请求域数据
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 id -- [in] 请求域的tag值；具体的定义值请参考【第三方接入业务接口文档】。 val -- [in] 字符串类型；对应于id的业务数据。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_SetString( HANDLE_SESSION sess, long id, const char *val )

### 5.17 Fix\_SetLong 设置整型类型请求数据

函数名称	Fix_SetLong设置该会话要发送给中间件的业务的请求域数据
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Id -- [in] 请求域的tag值；具体的定义值请参考【第三方接入业务接口文档】。 val -- [in] 整型；对应于id的业务数据。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_SetLong( HANDLE_SESSION sess, long id, long val )

### 5.18 Fix\_SetDouble 设置浮点类型请求数据

函数名称	Fix_SetDouble 设置该会话要发送给中间件的业务的请求域数据
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 id -- [in] 请求域的tag值；具体的定义值请参考【第三方接入业务接口文档】。 val -- [in] 浮点类型；对应于id的业务数据。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_SetItem ( HANDLE_SESSION sess, long id, double val )

### 5.19 Fix\_Run 业务提交

函数名称	Fix_Run 提交业务数据请求，并等待全部的业务结果返回。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。
返回数据	返回值为True表示服务业务处理成功；False表示失败，这个失败是表示业务通信上的失败；失败后，可以通过Fix_GetCode取出错误码（必定是一个负数）。通过Fix_GetErrMsg取出错误的信息
函数原型	BOOL Fix_Run( HANDLE_SESSION sess )

## 5.20 Fix\_AsyncRun 异步提交

函数名称	Fix_AsyncRun 异步提交业务数据，不等待待全部的业务结果返回
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。
返回数据	返回值为True表示服务业务处理成功；False表示失败，这个失败是表示业务通信上的失败；失败后，可以通过Fix_GetCode取出错误码(必定是一个负数)。通过Fix_GetErrMsg取出错误的信息
函数原型	BOOL Fix_AsyncRun ( HANDLE_SESSION sess )

## 5.21 Fix\_IsReplied 是否完全收到应答

函数名称	Fix_IsReplied异步提交业务数据后，检查数据是否全部返回。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 nMsec -- [in] 单位秒，等待时间，如果等于零，立即返回。
返回数据	返回值为True表示应答数据全部返回；False表示需要继续等待。
函数原型	BOOL Fix_IsReplied( HANDLE_SESSION sess, long nMsec )

## 5.22 Fix\_Cancel 取消应答等待

函数名称	Fix_Cancel 直接放弃业务应答数据，结束在Fix_Run中的等待。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。
返回数据	返回值为True表示成功；False表示失败。
函数原型	BOOL Fix_Cancel( HANDLE_SESSION sess )

## 5.23 Fix\_GetCode 读取错误编码

函数名称	Fix_GetCode 读取业务提交失败的错误代码。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。
返回数据	返回整数。
函数原型	BOOL Fix_GetCode( HANDLE_SESSION sess )

## 5.24 Fix\_GetErrMsg 读取错误信息

函数名称	Fix_GetErrMsg 读取业务提交失败的错误信息。
------	------------------------------

参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 out -- [in/out] 用于输出错误信息的字符串。 outlen -- [in] 参数out缓冲区的大小。
返回数据	返回错误信息字符串。
函数原型	char *Fix_GetErrMsg ( HANDLE_SESSION sess, char *out, int outlen )

## 5.25 Fix\_GetCount 读取应答总行数

函数名称	Fix_GetCount 读取中间件返回的业务应答数据的行数。
参数说明	Sess -- [in] 类型为HANDLE_SESSION的会话句柄。
返回数据	返回行数量。
函数原型	long Fix_GetCount( HANDLE_SESSION sess )

## 5.26 Fix\_GetItem 读取应答数据

函数名称	从中间件的返回结果集的指定行中读取业务应答数据。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 id -- [in] 指定域的tag值；具体的定义值请参考【第三方接入业务接口文档】。 out -- [in/out] 用于输出业务信息的字符串。 outlen -- [in] 指定out缓冲区的大小。 row -- [in][option] 指定返回数据结果集中的行数(行数索引值以0开始). 如果不设置这个值，表示读取第一行的数据。
返回数据	返回指定行的的字符串数据。
函数原型	char *Fix_GetItem ( HANDLE_SESSION sess, long id, char *out, int outlen, long row = -1 )

## 5.27 Fix\_GetLong 读取应答数据(整型)

函数名称	从中间件的返回结果集的指定行中读取业务应答数据。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 id -- [in] 指定域的tag值；具体的定义值请参考【第三方接入业务接口文档】。 row -- [in][option] 指定返回数据结果集中的行数(行数索引值以0开始). 如果不设置这个值，表示读取第一行的数据。
返回数据	返回指定行的整型数据。
函数原型	long Fix_GetLong ( HANDLE_SESSION sess, long id, long row = -1 )

### 5.28 Fix\_GetDouble 读取应答数据(浮点)

函数名称	从中间件的返回结果集的指定行中读取业务应答数据。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 id -- [in] 指定域的tag值；具体的定义值请参考【第三方接入 业务接口文档】。 row -- [in][option] 指定返回数据结果集中的行数(行数索引值 以0开始). 如果不设置这个值，表示读取第一行的数据。
返回数据	返回指定行的整型数据。
函数原型	long Fix_GetDouble ( HANDLE_SESSION sess, long id, long row = -1 )

### 5.29 Fix\_HaveItem 查看指定应答数据

函数名称	检查中间件的返回结果集的指定行中是否含有指定域。
参数说明	sess -- [in] 类型为HANDLE_SESSION的会话句柄。 id -- [in] 指定域的tag值；具体的定义值请参考【第三方接入业务接口文档】。 row -- [in][option] 指定返回数据结果集中的行数(行数索引值以0开始). 如果不设置这个值，表示读取第一行的数据。
返回数据	返回True表示, 含有指定的域数据；否则没有。
函数原型	long Fix_HaveItem ( HANDLE_SESSION sess, long id, long row = -1 )

### 5.30 Fix\_SetToken 设置业务令牌

函数名称	设置会话令牌（可选、与后台服务器配置有关）
参数说明	sess-- [in] 类型为HANDLE_SESSION的会话句柄。 Val -- [in] 类型为char* 的令牌字符串
返回数据	返回true表示设置令牌成功；False表示设置令牌失败
函数原型	bool Fix_SetToken( HANDLE_SESSION sess, const char * val)

### 5.31 Fix\_GetToken 读取业务令牌

函数名称	取出对应的令牌字符串（可选、与后台服务器配置有关） 请在客户密码校验的应答中读取令牌。
参数说明	Sess -- [in] 类型为HANDLE_SESSION的会话句柄。 Out -- [in] 用来保存令牌字符串。 outlen -- [in] out字符串的长度
返回数据	返回令牌字符串，如果获取失败，则返回空

函数原型	char * Fix_GetToken( HANDLE_SESSION sess, char * out, int outlen)
------	---

### 5.32 Fix\_ Encode 密码加密函数

函数名称	Fix_ Encode 简单加密函数（用户密码加密）
参数说明	pData-- [in]要求加密的用户密码
返回数据	返回加密后的用户密码
函数原型	char Fix_Encode( char * pData );

### 5.33 Fix\_ AddBackupSvrAddr 新增备份服务器

函数名称	添加备份服务器地址
参数说明	pszAddr -- [in]要添加的备份服务器地址串
返回数据	返回true，表示设置成功；否则失败
函数原型	BOOL Fix_AddBackupSvrAddr( char * pszAddr)