

Zserver 说明目录中的库全部是多平台支持, 并且 FPC/Delphi 兼容, 这些库可以在手机, Linux, 各种操作系统下使用

Zserver4D 库说明目录

Cadencer.pas	2
CommunicationFramework.pas	2
CommunicationFrameworkDataStoreServiceCommon.pas	2
CommunicationFrameworkDataStoreService.pas	2
CommunicationFrameworkDataStoreService_NoAuth.pas	2
CommunicationFrameworkDataStoreService_VirtualAuth.pas	3
CommunicationFrameworkDoubleTunnelIO.pas	3
CommunicationFrameworkDoubleTunnelIO_NoAuth.pas	3
CommunicationFrameworkDoubleTunnelIO_VirtualAuth.pas	3
CommunicationFrameworkDoubleTunnelIO_ServMan.pas	3
CommunicationFrameworkIO.pas	4
CommunicationTest.pas	4
CoreCipher.pas	4
CoreClasses.pas	4
CoreCompress.pas	4
DataFrameEngine.pas	4
DBCompressPackageForFile.pas	4
DoStatusIO.pas	5
Geometry2DUnit.pas	5
Geometry3DUnit.pas	5
GeometryLib.pas	5
GeometryRotationUnit.pas	5
ItemStream.pas	5
JsonDataObjects.pas	5
LibraryManager.pas	6
StreamList.pas	6
ListEngine.pas	6
MemoryStream64.pas	6
MH.pas	6
MH_1.pas, MH_2.pas, MH_3.pas	6
MH_delphi.inc, MH_fpc.inc	7
MH_ZDB.pas	7
NotifyObjectBase.pas	7
NumberBase.pas	7
ObjectData.pas	7
ObjectDataManager.pas	7
Optimize.Move.Win32.pas	7
PascalStrings.pas	8

TextDataEngine.pas	8
TextParsing.pas	8
TextTable.pas	8
UnicodeMixedLib.pas	8
ZDBEngine.pas	8
ZDBLocalManager.pas	8

Cadencer.pas

Cadencer 是对时间的节奏控制器，Cadencer 可以在每秒触发数百次时间事件，并且在事件中附加了当前 DeltaTime，我们多用于计时事件，渲染器刷帧，物理引擎刷帧，逻辑系统刷帧。

CommunicationFramework.pas

ZServer4D 的通讯核心框架，CommunicationFramework 主要为通讯中间件提供底层支持，诸如处理来自网络的碎片数据，处理命令事件等等

CommunicationFrameworkDataStoreServiceCommon.pas

ZServer4D 的高级数据库框架叫做 ZDB，这是 ZDB 网络数据库的公共单元，里面是一些 ZDB 网络服务的声明和定义

CommunicationFrameworkDataStoreService.pas

ZServer4D 的高级数据库框架叫做 ZDB，CommunicationFrameworkDataStoreService 提供了基于网络身份验证数据库处理服务，该框架基于 CommunicationFrameworkDoubleTunnelIO 搭建

CommunicationFrameworkDataStoreService_NoAuth.pas

ZServer4D 的高级数据库框架叫做 ZDB，CommunicationFrameworkDataStoreService_NoAuth 提供了基于网络的数据库处理服务，该框架不会验证用户身份，多用于隔断了公网在后台间的数据库交换和处理，我们使用它时一定要注意不要暴露在公网下，否则会让我们的数据非常危险，该框架基于 CommunicationFrameworkDoubleTunnelIO_NoAuth 搭建

CommunicationFrameworkDataStoreService_VirtualAuth.pas

ZServer4D 的高级数据库框架叫做 ZDB，CommunicationFrameworkDataStoreService_VirtualAuth 提供了基于网络的数据库处理服务，使用该框会验证用户身份，并且验证用户身份的机制为默许，该框架多用于对用户身份验证机制以及对安全有较高要求做二次开发时使用，该框架基于 CommunicationFrameworkDoubleTunnelIO_VirtualAuth 搭建

CommunicationFrameworkDoubleTunnelIO.pas

ZServer4D 的核心双通道框架，该框架内置了一套基于加密文件的用户身份验证系统，并且支持各个用户拥有自己的私人空间，良好支持文件私人传输与存储，共享文件传输与存储。该框架多用于在公网中面向用户服务器开发和使用，维护该框架的用户数据库需要自行通过编程解决。建议用户承载量保持在 20 万注册用户以下。具体使用请参考相关演示

CommunicationFrameworkDoubleTunnelIO_NoAuth.pas

ZServer4D 的核心双通道框架，该框架没有用户身份验证系统，不支持用户私人文件空间，但是支持共享文件空间，该框架多用于服务器间的交互。具体使用请参考相关演示

CommunicationFrameworkDoubleTunnelIO_VirtualAuth.pas

ZServer4D 的核心双通道框架，该框架有用户身份验证系统，但是默许为全部通过，使用该框架需要自行开发一套用户身份系统，诸如自行搭建用户身份数据库，然后再使用该框架在数据库进行验证。该框架不支持用户私人文件空间，该框架支持共享文件空间。该框架主要用于制作在公网下面向终端用户的服务器，用户承载量根据你自己设计而定，可以数百万，也可以是数亿，属于大型后台框架。暂时没有该框架的演示

CommunicationFrameworkDoubleTunnelIO_ServMan.pas

该框架主要作用是面向云服务器的调度，在云服务器系统都有 ManagerServer 的调度系统，因为 ManagerServer 是一个独立程序，过去我们使用都是直接修改，现在，ManagerServer 被直接封装成了一个引用库，我们制作云服务器的调度服务只需要引用它，简单敲下几行代码就能将 ManagerServer 驱动起来

CommunicationFrameworkIO.pas

这是单通道的框架，类似 http，全都是单向式的请求和响应，不能交互。如果要做交互性质的长连接服务，请用双通道框架。因为 http 非常普及，该框架很少使用，如果你在使用该框架，请告诉我，我会帮你维护。

CommunicationTest.pas

ZServer4D 的测试单元，基于网络通讯层的测试，不是压测，主要是模拟日常收发命令，处理命令

CoreCipher.pas

ZServer4D 的加密库，内置的加解密算法都是对称算法，支持并行化

CoreCipher 有自己独立开源项目地址

<https://github.com/PassByYou888/CoreCipher>

CoreClasses.pas

ZServer4D 的多平台支持基类，CoreClasses 主要是重新定义了一些底层类，诸如我们常用的 TList, TStream, TStrings，便于多平台兼容，便于 FPC 兼容。

CoreCompress.pas

ZServer4D 的压缩库，内置了两种压缩算法，纯 pascal 实现，不包含外部引用库。支持 Stream 压缩操作。

DataFrameEngine.pas

ZServer4D 的数据结构引擎，我们在 ZServer4D 的通讯中，所有的数据收发，都会经过它的打包和编码，DataFrameEngine 支持三种压缩算法，支持几乎所有的 Delphi 和 FPC 数据结构，解码和编码使用的内存拷贝很小。在 Zserver4D 对该库都有针对性的优化。

DBCompressPackageForFile.pas

基于祖传数据结构 ObjectData 对文件进行打包，查询，压缩处理的库，只掉一个函数即可实现文件的压缩和打包，打包文件格式不兼容 FilePackageWithZDB 这类工具，也不兼容 ZDB，它只是提供一种更便捷的文件打包处理。

DoStatusIO.pas

运行的状态输出库，有多线程无关性机制，兼容 MH，全局中心化的状态输出，支持在 Delphi 的 IDE 中输出运行状态，如果你写 Console 的应用，DoStatusIO 自动支持 Console 输出。内部提供了各种外部钩子，可以勾多次，勾在各种 Form 中。

Geometry2DUnit.pas

Delphi/FPC 圈最强大的 2D 几何库，相交判断，折线，凹包，凸包，相对坐标系，Texture Atlas(Rect Package)，Clip(非 cut 裁剪)，几乎覆盖我们常用所有 2D 几何计算。该库已经许多项目考验，高速且无 Bug。在 ZServer4D 引入该库是因为 DataFrameEngine 内置了该库的数据结构。

Geometry3DUnit.pas

基于 GeometryLib 封装的高级 3D 计算库，不支持汇编加速，但是使用方便。在 ZServer4D 引入该库是因为 DataFrameEngine 内置了该库的数据结构。

GeometryLib.pas

底层 3D 计算库，不支持汇编加速。在 ZServer4D 引入该库是因为 DataFrameEngine 内置了该库的数据结构。

GeometryRotationUnit.pas

3D 正反逆向矩阵库，因为几乎所有的 3D 引擎的矩阵都是不可逆的，诸如模型矩阵，我们在给定了欧拉角或则矩阵转角后产生矩阵，然后直接参与渲染的矩阵变换。逆矩阵主要用于 AR，VR 等等工业需求，因为精确的逆矩阵十分难解，以前我花了许多时间才逆向出来，一直舍不得抛弃它。在 ZServer4D 引入该库是因为 DataFrameEngine 内置了该库的数据结构。

ItemStream.pas

基于祖传数据结构 ObjectData 对 Stream 支持和交换的库

JsonDataObjects.pas

引用于开源项目，不兼容 FPC，但是在 Delphi 中因为速度快，稳定，比 EMB 的 Json 官方更快，所以我将它纳入了 ZServer4D 中的 ZDB 子工程。开源项目地址如下
<https://github.com/ahausladen/JsonDataObjects>

LibraryManager.pas

基于祖传数据结构 `ObjectData` 的一级目录加速管理库，建议少用，`ZServer4D` 工程以后几乎不会再维护这套一级目录加速的库。因为 `ObjectData` 在开发 `ZDB` 的需求时，实现了内置 `Cache`，我们几乎不再需要另外再做一次一级目录加速了，现在还没有放弃它是因为在 `MediaCenter` 这类中心管理器中还有使用它。

StreamList.pas

基于祖传数据结构 `ObjectData` 的单目录文件的加速管理库，建议少用，`ZServer4D` 工程以后几乎不会再维护这套一级目录加速的库。

ListEngine.pas

因为 `hash` 查找加速是有针对性的，不可以做成一个类然后一锅端。在 `ListEngine` 中，`Hash` 分别针对了 `Pointer`, `String`, `Int64`, `NativeInt`, `UInt64`, `Object` 进行了特殊 `Hash` 加速优化。`ListEngine` 这套库非常大，接近上万行代码，属于 `ZServer4D` 的链表核心加速库。

MemoryStream64.pas

比 `MemoryStream` 更快的内存 `Stream` 支持，`MemoryStream64` 库额外还内置了压缩，和计数器（对 `Stream` 进行读写的计数）

MH.pas

`MH` 简称 `memoryHook`，这是 `MH_1`, `MH_2`, `MH_3`, `MH_ZDB` 四个钩子的综合库，`MH` 可以勾住内存分配 `API`，监听你的所有内存分配，然后使用内存回收方法来进行释放。`MH` 同样也可以分析内存泄漏，帮助系统决策环节管理内存 `Cache` 开销。`MH` 和 `FastMM` 这类内存管理单元不会冲突，`MH` 不用像 `FastMM` 那样放在应用程序第一个 `Uses` 位置，`MH` 在任何地方引用它都能使用。

MH_1.pas, MH_2.pas, MH_3.pas

`MH` 简称 `memoryHook`，`MH_1` 是指 1 号钩子，因为钩子 `API` 一定启用就不能复用，要么就启动内存监听钩子，要么就停止监听，为了在监听时，还可以再监听，所有在 `ZServer4D` 中有 `MH_1`, `MH_2` 这类多个钩子库。比如，你在 `form.show` 中打开 1 号钩子监听，这时，你就不能再用 1 号钩子了，你只能使用 `MH_2`，依次重复。

MH_delphi.inc, MH_fpc.inc

MH 钩子在 Delphi 和 FPC 的不同功能实现实例

MH_ZDB.pas

MH 针对 ZDB 的专用钩子，在 ZDB 内部管理 Cache 开销时使用，勿在外部使用，也不要引用

NotifyObjectBase.pas

延迟事件引擎的实现库，它比 Timer 的开销更小，触发事件为一次性。可以大量用于 ZServer4D 的后台处理业务，它容错性比 Timer 更高。注意：使用它必须头脑清醒，因为跟踪它的事件流程是反人类的。

NumberBase.pas

数值引擎的基类，在游戏，消费，用户升级这些后台环节，我们都会用到数值，NumberBase 提供了大规模数值的存储，管理，叠加计算，延迟计算这些基础功能。

ObjectData.pas

祖传数据结构，类似虚拟硬盘，objectData 的历史非常久远，诞生于 TurboPascal 时代，后续经过无数维护和升级，现在 ObjectData 不光支持 Delphi/fpc，还支持各种移动平台。ObjectData 有 Pascal 和 c++两套版本。在 ZServer4D 的开源项目中，是优化过的 Pascal 版本

ObjectDataManager.pas

ObjectData 是纯 API 的数据结构，使用起来非常反人类，要写很多代码才能驱动它，ObjectDataManager 是将祖传数据结构封装成了 OO 的接口，便于使用。ObjectDataManager 内置了 ObjectData 的 Cache 接口。

Optimize.Move.Win32.pas

基于 windows x86 平台对 move 的优化支持库，它有基于 cpu1,2,3 级硬件 Cache 的推进型 copy 机制，支持 SSE，MMX，只要你引用它，它会自动 Hook 你的 move api，自动化加速。内部已经做了平台判断，如果你在 x64 编译器引用它，可以编译通过，但是并不能加速，它只支持 windows x86 的 move 加速。因为我使用的 Optimize.Move.Win32 来自不明开源社区，无法查证作者是谁。

PascalStrings.pas

Delphi/fpc 的 string 封装单元，默认数据结构为 UTF8。如果你将自己的 string 替代成 TPascalString，就不存在手机和 windows，不存在 FPC 和 Delphi。

TextDataEngine.pas

INI 的块数据引擎，和 INI 的区别是 TextDataEngine 是千万级 INI 条目的支持，内置了 Hash 查找加速。支持高速读取和缓存。

TextParsing.pas

文本解析库，文本词法库，专业处理字符串和文本，处理常用字符串，如切割，查找，分析，不要使用它，如果处理的是，http 头，email 头，脚本引擎，句法，你可以使用它，它不会让你失望。

TextTable.pas

我做国际化翻译项目时写的文本表结构，除非你做文本翻译，否则可以不用。

UnicodeMixedLib.pas

Unicode 支持库，它不光支持 Unicode，还支持了内置的小块文件预读和回写 cache，许多字符串支持，可以让你替代掉 SysUtils 这类库，跨平台更安全和省心。

ZDBEngine.pas

基于祖传数据结构 ObjectData 所开发的 ZDB 数据引擎基类，内置了内存 Cache 和硬件 IO 优化。支持 KeyValue,Json,Stream,DataFrameEngine,String 等等数据条目格式。

ZDBLocalManager.pas

基于 ZDB 数据引擎基类所开发的本地数据库