BIM+服务平台V1.0

概要设计说明书

上海东链博数据科技有限公司

目 录

[一、 概述 5](#_Toc490746898)

[1.1 编写目的 5](#_Toc490746899)

[1.2 背景 5](#_Toc490746900)

[1.3 定义 6](#_Toc490746901)

[1.4 参考资料 6](#_Toc490746902)

[二、 总体设计 7](#_Toc490746903)

[2.1 设计前提和约束条件 7](#_Toc490746904)

[2.2 基本设计思想 7](#_Toc490746905)

[2.3 功能划分及处理流程 7](#_Toc490746906)

[2.3.1 项目管理 8](#_Toc490746907)

[2.3.2 模型管理CRUD 8](#_Toc490746908)

[2.3.3 模型管理🡪解析模型 9](#_Toc490746909)

[2.3.4 模型管理🡪渲染模型 10](#_Toc490746910)

[2.3.5 模型管理🡪持久化 11](#_Toc490746911)

[2.3.6 模型管理🡪备份模型 12](#_Toc490746912)

[2.3.7 模型管理🡪模型合并 13](#_Toc490746913)

[2.3.8 模型管理🡪模型树型结构分析 13](#_Toc490746914)

[2.3.9 模型管理🡪类型 14](#_Toc490746915)

[2.3.10 模型管理🡪部件属性 15](#_Toc490746916)

[2.3.11 模型管理🡪搜索模型部件 16](#_Toc490746917)

[2.3.12 模型管理🡪楼层展示 17](#_Toc490746918)

[2.3.13 模型管理🡪配色功能 18](#_Toc490746919)

[2.3.14 模型管理🡪模型专业配置与展示 19](#_Toc490746920)

[2.3.15 统计分析🡪服务器状态 20](#_Toc490746921)

[2.3.16 统计分析🡪模型统计 20](#_Toc490746922)

[2.3.17 统计分析🡪报表输出 21](#_Toc490746923)

[2.4 系统整体设计 22](#_Toc490746924)

[2.4.1 物理部署架构 22](#_Toc490746925)

[2.4.2 系统的层次模型 24](#_Toc490746926)

[2.5 性能要求 25](#_Toc490746927)

[2.5.1 处理能力 25](#_Toc490746928)

[2.5.2 稳定性 25](#_Toc490746929)

[2.5.3 传输能力 25](#_Toc490746930)

[2.6 运行环境 26](#_Toc490746931)

[2.6.1 硬件环境 26](#_Toc490746932)

[2.6.2 软件环境 27](#_Toc490746933)

[三、 接口设计 28](#_Toc490746934)

[3.1 用户接口 28](#_Toc490746935)

[3.2 外部接口 28](#_Toc490746936)

[3.2.1 几何信息渲染 28](#_Toc490746937)

[3.2.2 格式转换 28](#_Toc490746938)

[3.3 内部接口 28](#_Toc490746939)

[四、 系统数据结构设计 29](#_Toc490746940)

[4.1 逻辑结构设计要点 29](#_Toc490746941)

[4.1.1 项目表 29](#_Toc490746942)

[4.1.2 用户表 29](#_Toc490746943)

[4.1.3 模型描述信息表 30](#_Toc490746944)

[4.1.4 模型构件实体表 30](#_Toc490746945)

[4.1.5 IFC实体类编号表 30](#_Toc490746946)

[4.1.6 专业模板表 30](#_Toc490746947)

[4.1.7 模型专业映射表 30](#_Toc490746948)

[4.1.8 平台版本管理表 30](#_Toc490746949)

[4.2 物理结构设计要点 30](#_Toc490746950)

[4.2.1 项目表 30](#_Toc490746951)

[4.2.2 用户表 31](#_Toc490746952)

[4.2.3 模型描述信息表 31](#_Toc490746953)

[4.2.4 模型构件实体表 32](#_Toc490746954)

[4.2.5 IFC实体类编号表 32](#_Toc490746955)

[4.2.6 专业模板表 32](#_Toc490746956)

[4.2.7 模型专业映射表 33](#_Toc490746957)

[4.2.8 平台版本管理表 33](#_Toc490746958)

[4.3 数据结构与程序的关系 33](#_Toc490746959)

[五、 软件出错处理设计 33](#_Toc490746960)

[5.1 出错信息 33](#_Toc490746961)

[5.2 补救措施 34](#_Toc490746962)

[5.3 系统维护设计 34](#_Toc490746963)

# 概述

## 编写目的

软件概要设计是从总体上把握系统设计框架，包括模块划分、处理流程和接口设计，概要设计说明书对上述内容作了总体描述，体现了用户需求与应用系统实现之间的关系，在设计过程中起到了提纲挈领的作用。

项目名称：东链博BIM+服务管理平台

项目的任务提出者：上海东链博数据科技有限公司

项目的任务开发者：上海东链博数据科技有限公司技术部

本文档的阅读者：上海东链博数据科技有限公司技术部

最终用户：政府，企业，公众

## 背景

BIM建筑信息化模型在设计阶段提供了对建筑的准确数字化表达，让管理人员快速了解建筑功能、结构空间和设计意图，而且其任意的模型剖切及旋转，使得复杂工程结构一目了然，快速实现了对流程以及重难点的深入了解，同时也能快速检测出不同专业领域的结构功能进行整合时否发生碰撞。在施工阶段，该建筑模型也能服务于不同部门不同专业之间的工作协同，以及项目进度，建筑工序调配，为施工提供可靠依据。建筑的运营管理阶段，包括设备管理，物业管理以及其他一些服务，以物业管理为例，通过BIM技术进行三维互动和关联数据等手段，用户将从各个方面了解其物业的状态，物业公司可以利用BIM模型进行可视化管理，快速查询设备信息和物业租赁情况，使应急能力得到了提高，提升了管理水平。而对于政府来说，掌握了建筑的BIM模型，将有极大的社会意义，BIM一旦应用，它就服务于整个管理体系，也受整个管理体系监管。对于政府投资的工程的监管方式必然发生改变，会影响项目立项、概预算审报批、规划报建、消防审批、施工许可、质量与安全监督、工程验收和城建挡案等环节的审批和监管模式，对提高审批流程和行政效率起到很大的作用，目前政府部门也正在加大BIM技术推广的力度。

在这样的背景下，将BIM技术推向应用领域也成为了新热点。而BIM+服务平台未来将服务于政府、企业和公众，同时也依托政府未来对于建筑信息的开放，并在BIM应用中积累沉淀各类建筑信息，如此BIM才能真正成为建筑全生命周期的数字化表达，包涵建筑的所有相关信息和历史变更。此版本的BIM+服务平台是一个BIM数据展示及发布平台，数据模型采用IFC工业基础类提供的一套标准类库。另外该平台也将提供一套开发框架，为BIM面向专业领域的应用提供技术支撑，在此基础上可以整合FM设施管理系统，IBMS智能楼宇系统，亦或结合GIS系统，平台将提供输出轻量化模型用于在GIS系统上成片展示和关联楼宇的BIM信息。

## 定义

●软件、本软件、系统、本系统均指《BIM+服务平台》

●用户指使用本软件的人

●BIM，[建筑信息模型](http://baike.baidu.com/view/1788980.htm)（Building Information Modeling）

●IFC，工业基础类（Industry Fundaction Class）

## 参考资料

●国家软件工程开发标准

●IFC标准

●Open[BIM](https://baike.baidu.com/item/BIM)工作模式

# 总体设计

## 设计前提和约束条件

1. 平台使用IFC格式的BIM模型导入，支持IFC2X3和IFC4。
2. 使用MAVEN对软件项目生命周期进行管理。

## 基本设计思想

总体模块划分和模块之间的关系详见图 2.2‑1：



图 ‑1

## ****功能划分及处理流程****



图 ‑1



### 项目管理

本功能针对用户对不同项目，建立不同的项目进行管理。可对项目进行修改、增加、删除操作，查看项目列表，查看项目信息等操作。



图 2.3‑2

### 模型管理CRUD

本功能是在模型设计制作人员，设计制作完模型导出为IFC文件后，将文件上传到服务器的时所使用的功能。模型列表管理中提供了上传模型、模型信息更改、修改模型、删除模型、查看模型详情与展示模型等功能。流程图关系详见图 2.3‑3：



图 2.3‑3

### 模型管理🡪解析模型

该模块将上传完成的BIM模型进行解析，依赖EMF建模生成的一套JAVA类库，将模型内部的每条文本记录逐个解析为JAVA对象。



图 ‑4

### 模型管理🡪渲染模型

该模块提取出BIM模型内部的几何信息，包括形状、坐标信息以及外轮廓线信息。



图 ‑5

### 模型管理🡪持久化

该模块将解析生成的对象存储到数据库中，需要分布式的存储结构和高可用等特点，建议使用非关系型数据库。



图 ‑6

### 模型管理🡪备份模型

平台除了将上传的BIM模型解析并入库,还将对模型的数据将进行备份,备份时将利用分布式文件系统的高可用的特点。



图 ‑7

### 模型管理🡪模型合并

用户上传的模型很可能只是整体BIM项目的一部分模型，如用户第一次只上传了结构模型，之后又在基础上布置了管线信息，这时候平台将对模型进行合并，作为一个BIM项目进行应用。



图 ‑8

### 模型管理🡪模型树型结构分析

本功能是将IFC文件模型通过树形结构展示给用户，方便用户更加形象通过IFC类型来寻找相应的部件。流程图详见图 2.3‑9



图 2.3‑9

### 模型管理🡪类型

本功能是将模型部件，根据IFC标准划分成IFC类型，用户可根据点击IFC类型来寻找自己要查找的模型部件。流程图详见图 2.3‑10



图 2.3‑10

### 模型管理🡪部件属性

本功能通过鼠标点击3D模型上的部件，来展示模型中此部件的相关属性，例如模型尺寸大小。流程图详见图 2.3‑11。



图 2.3‑11

### 模型管理🡪搜索模型部件

本功能提供用户输入模型名称等关键字，来搜索模型部件。通过搜索得出相关模型部件列表，在列表中寻找用户所需要查看的模型部件。流程图详见下图。



图 2.3‑12

### 模型管理🡪楼层展示

本功能根据楼层分类展示模型部件，点击不同的楼层显示或隐藏当前楼层所有的模型部件。流程图详见图 2.3‑13。



图 2.3‑13

### 模型管理🡪配色功能

本功能提供用户修改模型配色的功能，为了方便用户配色，系统提供了若干种配色方案。流程图详见图 2.3‑14。



图 2.3‑14

### 模型管理🡪模型专业配置与展示

本功能根据用户所在的不同专业，提供了专业配置功能。通过勾选IFC类型，以及IFC类型下不同的结构名称，挑选出符合用户专业领域要求的相关部件结构，用当前部件结构构造出符合用户专业所需的模型，并进行展示。同一用户可配置各种不同的专业。流程图详见图 2.3‑15。

图 2.3‑15

### 统计分析🡪服务器状态

统计服务器的运行状态，包括存储状态和网络状态等。



图 ‑16

### 统计分析🡪模型统计

统计模型数量，上传时间，IFC对象数等关于模型的数据。



图 ‑17

### 统计分析🡪报表输出

图表的多样化展示和导出图表功能。



图 ‑18

## ****系统整体设计****

### ****物理部署架构****



a) 硬件环境：

需要三台机器(racdb,host8,host9)：

数据节点，sharding 2片

每个sharding为Replic Set方式，冗余度为3（主、从、延迟从）

mongos(路由节点)，数量为3

config server(配置服务器节点)，数量为3

arbiter server(表决节点)，数量为6

b) 软件环境：

OS：[Oracle](http://www.2cto.com/database/Oracle/" \t "_blank) [Linux](http://www.2cto.com/os/linux/) 6.5

MongoDB: mongodb3.2.3

### ****系统的层次模型****



* + - 1. **客户端**

客户端指的是访问应用的各类终端，包括浏览器、Android平台、IOS平台和手机Wap。

* + - 1. 展现层

表现层接收客户端的 HTTP 请求，提供项目管理、模型管理和统计分析等功能。

* + - 1. 控制层

对表现层发来的数据格式进行检查判断，根据不同的业务将数据分配到不同的业务处理服务进行处理。

* + - 1. 业务处理层

它接收展示层分发的交易请求，完成业务逻辑的具体实现。对不同的业务数据进行处理，处理完成后，将处理结果返回表现层。

* + - 1. 数据访问层

数据访问层向业务层提供统一的内部和外部资源访问，为业务层的数据访问请求屏蔽不同的数据存储访问技术，以及与外部系统整合技术的差异性。

* + - 1. **数据层**

资源层主要指数据库、文件系统和外部系统。本系统采用MongoDB 作为数据库，GridFS作为分布式文件系统。

## 性能要求

### 处理能力

现阶段系统面对可能达到几十的用户任务并发性，应保证正常的处理能力，后期可能面向大批量的用户，因此要考虑可扩展性和高可用。

### 稳定性

系统资源利用合理，保证系统前后台数据操作效率。

### 传输能力

本系统业务功能包括模型和图片传输，应提高文件传输速度。

## 运行环境

### 硬件环境

* + - 1. 生产环境

|  |  |  |
| --- | --- | --- |
| **服务器** | **最低配置** | **推荐配置** |
| 应用服务器、  数据库服务器 | 2C16G | 4C32G |
| HD：40G | HD：80G |

* + - 1. 开发测试环境

|  |  |  |
| --- | --- | --- |
| **服务器** | **最低配置** | **推荐配置** |
| 应用服务器、  数据库服务器 | 2C8G | 2C16G |
| HD：40G | HD：40G |

### 软件环境

* + - 1. 生产环境

|  |  |  |
| --- | --- | --- |
| **分类** | **名称** | **版本** |
| 服务器操作系统 | Windows Server 2008 | 2008 |
| 应用数据库 | Monogodb | 3.0 |
| Web应用服务器 | Jetty | 9.2 |
| Java | JDK 64bit | 1.8 |

* + - 1. 开发测试环境

|  |  |  |
| --- | --- | --- |
| **分类** | **名称** | **版本** |
| 服务器操作系统 | Windows7/ Windows10/XP | 7/10/XP |
| 应用数据库 | Monogodb | 3.0 |
| 应用服务器 | Jetty | 9.2 |
| Java | JDK | 1.8 |
| 浏览器 | Google Chrome(Windows版本) | 52 |
| Java开发环境 | Eclipse | Mars |

# 接口设计

## 用户接口

进入前端管理界面需要用户验证登录，用户输入标识和口令，只有正确才能进入管理界面。

## 外部接口

### 几何信息渲染

平台用JNA库调用了外部渲染引擎IfcEngineDll来渲染几何信息。

### 格式转换

平台调用了外部Collada2gltf将Dae格式转成gltf格式。

## 内部接口

后端与前端展示层交互的接口如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 接口功能 | 接口方式 | 客户端/服务器端 |
| 添加模型接口 | 上传模型并进行解析，渲染，存档 | Http请求 | 服务器 |
| 转换模型格式接口 | 转换模型为GLB格式 | Http请求 | 服务器 |
| 查询GLB模型接口 | 查询已经转换过的GLB模型 | Http请求 | 服务器 |
| 查询项目下的模型接口 | 该接口将查询出所有属于指定项目的模型 | Http请求 | 服务器 |
| 查询模型的描述信息接口 | 该接口查询单个模型的描述信息 | Http请求 | 服务器 |
| 删除模型接口 | 删除指定模型 | Http请求 | 服务器 |
| 查询模型树接口 | 查询指定模型的内部树状结构 | Http请求 | 服务器 |
| 查询楼层信息接口 | 查询指定模型的楼层信息 | Http请求 | 服务器 |
| 查询对象类型接口 | 查询模型内部部件类别 | Http请求 | 服务器 |
| 查询属性接口 | 查询模型部件属性 | Http请求 | 服务器 |
| 搜索部件接口 | 搜索与关键字匹配的部件 | Http请求 | 服务器 |
| 查询专业模板接口 | 查询出指定模型的专业信息 | Http请求 | 服务器 |
| 查询所有专业模板接口 | 查询出指定模型应用的专业模板 | Http请求 | 服务器 |
| 查询默认专业模板接口 | 查询默认的全模板 | Http请求 | 服务器 |
| 保存专业模板接口 | 保存专业模板到数据库 | Http请求 | 服务器 |
| 更新专业模板接口 | 更新专业模板的内部细节 | Http请求 | 服务器 |
| 删除专业模板接口 | 删除专业模板 | Http请求 | 服务器 |
| 查询模型几何信息接口 | 查询模型内部的几何信息，包括形状和位置 | Websocket | 服务器 |
| 进度推送 | 查询模型几何信息的过程的进度推送到展示层 | Websocket | 服务器 |

# 系统数据结构设计

## 逻辑结构设计要点

本平台数据库用Nonsql的MongoDB，以下表结构其实的准确说法应该是文档结构。

### 项目表

项目（项目编号，作者名，标题，描述，星级，图片）

### 用户表

用户（用户编号，姓，名，公司，email，用户名，密码）

### 模型描述信息表

模型描述（模型编号，模型所属项目编号，模型元数据，模型名，模型类型，文件名，上传日期，文件大小）

### 模型构件实体表

模型构件实体（构件编号，模型编号，IFC类型编号，特征属性列表）

### IFC实体类编目表

实体类编目（类编号，包名加类名，当前最大OID）

### 专业模板表

专业模板（专业编号，专业名，IFC类型选择实例）

### 模型专业映射表

模型专业映射（模型编号，专业编号列表）

### 平台版本管理表

平台版本（平台版本编号，当前最高模型版本编号）

## 物理结构设计要点

### 项目表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | pid | 项目编号 | Long |  | 是 | 否 | 主键 |  |
| 2 | author | 作者名 | String |  |  | 否 |  |  |
| 3 | title | 标题 | String |  |  | 否 |  |  |
| 4 | description | 描述 | String |  |  | 是 |  |  |
| 5 | stars | 星级 | Integer |  |  | 否 |  |  |
| 6 | picUrl | 图片URL | String |  |  | 是 |  |  |

### 用户表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | userId | 用户编号 | Long |  | 是 | 否 | 主键 |  |
| 2 | firstName | 名 | String |  |  | 否 |  |  |
| 3 | lastName | 姓 | String |  |  | 否 |  |  |
| 4 | company | 公司名称 | String |  |  | 是 |  |  |
| 5 | email | 邮箱地址 | String |  |  | 是 |  |  |
| 6 | userName | 用户名 | String |  |  | 否 |  |  |
| 7 | password | 密码 | String |  |  | 否 |  |  |

### 模型描述信息表

父表：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | rid | 模型编号 | Integer |  | 是 | 否 | 主键 |  |
| 2 | pid | 项目编号 | Long |  | 是 | 否 | 外键 |  |
| 3 | modelMetaData | 模型元数据 | ModelMetaData |  |  | 否 |  | 内嵌文档 |
| 4 | name | 模型名 | String |  |  | 是 |  |  |
| 5 | applyType | 模型类型 | Integer |  |  | 是 |  |  |
| 6 | fileName | 文件名 | String |  |  | 否 |  |  |
| 7 | uploadDate | 上传日期 | Date |  |  | 否 |  |  |
| 8 | fileSize | 文件大小 | Long |  |  | 否 |  |  |

其中ModelMetaData属于Nonsql数据库的嵌套类型数据，其嵌套结构如下表所示：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | authorizedUser | 作者名 | String |  | 是 | 是 |  |  |
| 2 | name | 模型名 | String |  | 是 | 是 |  |  |
| 3 | checksum | MD5校验 | byte[] |  |  | 是 |  |  |
| 4 | date | 日期 | Date |  |  | 是 |  |  |
| 5 | revisionId | 模型编号 | Integer |  |  | 是 | 外键 | 手动 |
| 6 | ifcHeader | 模型头部信息 | IfcHeader |  |  | 是 |  | 内嵌 |

### 模型构件实体表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | oid | 构件编号 | Long |  | 是 | 否 | 主键 |  |
| 2 | rid | 模型编号 | Integer |  | 是 | 否 | 外链 |  |
| 3 | eClassId | IFC类型编号 | Short |  | 是 | 否 | 外链 |  |
| 4 | features | 特征属性列表 | Map<String,Object > |  |  | 是 |  |  |

### IFC实体类编目表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | cid | IFC类编号 | Short |  | 是 | 否 | 主键 |  |
| 2 | packageClassName | 包名加类名 | String |  | 是 | 否 |  |  |
| 3 | oid | 当前最大OID | Long |  | 是 | 否 |  |  |

### 专业模板表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | toid | 专业编号 | Long |  | 是 | 否 | 主键 |  |
| 2 | name | 专业名 | String |  |  | 否 |  |  |
| 3 | namespaceSelectorMap | IFC类型选择表 | Map |  |  | 否 |  |  |

### 模型专业映射表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | rid | 模型编号 | Integer |  | 是 | 否 | 主键 |  |
| 2 | otid2Name | 专业编号表 | Map |  |  | 否 |  |  |

### 平台版本管理表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **字段含义** | **类型** | **长度** | **索引** | **允许空** | **主键/外键** | **说明** |
| 1 | platformVersionId | 平台版本编号 | String |  | 是 | 否 | 主键 |  |
| 2 | currentTopRevisionId | 当前最高模型版本编号 | Integer |  |  | 否 |  |  |

## 数据结构与程序的关系

平台采用Nonsql的MongoDB存储数据，并非传统意义的数据库表结构，而是单文档结构，每张表实际上可以看作一个文档数据，这样数据存在在一类文档中存储并非都是同一数据结构，这样对于几百个IFC类对象进行存储自然在数据库层面非常方便。同时MongoDB作为一个分布式的数据库其高可用已扩展的特点正是本平台需要的特性，保证了数据的安全性和面向不同数量级的用户动态扩展系统的效能。

# 软件出错处理设计

## 出错信息

采用弹出窗口方式向用户提示错误，并友好地处理错误。 例如，用户登录失败时，根据具体原因进行提示，用户输入不正确时，进行适当提示。

## 补救措施

a) 定期建立数据库备份，一旦服务器数据库被破坏，可以使用最近的一份数据库副本进行还原。

b) 为防止服务器故障，预备另外一台服务器，只要主服务器出现故障，可以迅速启动预备服务器运行系统。

## 系统维护设计

在系统中专门设置用于系统检查与维护的检测点和专用模块。尽可能地降低系统维护工作量。系统的前台应用的界面风格要一致，提示信息要明确。软件编码的要求风格一致，有详细注释，保证代码易读易懂，以便于系统维护。要求提供完备的设计文档、用户手册、安装文档、在线和联机帮助文档，为系统地维护提供有效的帮助和指导。系统发生变化后要及时更新相关文档内容。