

**TechSvr运维方案**

**卫宁健康科技集团股份有限公司**

**Winning Health Technology Group Co.,Ltd.**

目录

[一、 TechSvr服务端升级方案 3](#_Toc503454405)

[概述 3](#_Toc503454406)

[升级任务表字段 4](#_Toc503454407)

[升级范围设置 5](#_Toc503454408)

[医院范围升级 5](#_Toc503454409)

[服务接口定义 5](#_Toc503454410)

[检查更新接口 5](#_Toc503454411)

[升级完成接口 6](#_Toc503454412)

[二、 TechSvr客户端升级方案 7](#_Toc503454413)

[概述 7](#_Toc503454414)

[升级说明 8](#_Toc503454415)

[三、 服务端中心管理方案 8](#_Toc503454416)

[概述 8](#_Toc503454417)

[原型图展示： 9](#_Toc503454418)

[升级记录页面 9](#_Toc503454419)

[日志分析页面 10](#_Toc503454420)

[功能调用页面 10](#_Toc503454421)

[客户端统计页面 11](#_Toc503454422)

[任务操作页面 11](#_Toc503454423)

[功能点 12](#_Toc503454424)

[监控各客户端TechSvr运行情况 12](#_Toc503454425)

[可以通过该平台设置更新包 12](#_Toc503454426)

[平台支持 12](#_Toc503454427)

[Windows Server 12](#_Toc503454428)

[Linux 13](#_Toc503454429)

[数据表设计 13](#_Toc503454430)

[日志表 Log 13](#_Toc503454431)

[设备表 Device 13](#_Toc503454432)

[功能调用统计表 FuncUsage 14](#_Toc503454433)

[接口调用 14](#_Toc503454434)

[日志上传接口 14](#_Toc503454435)

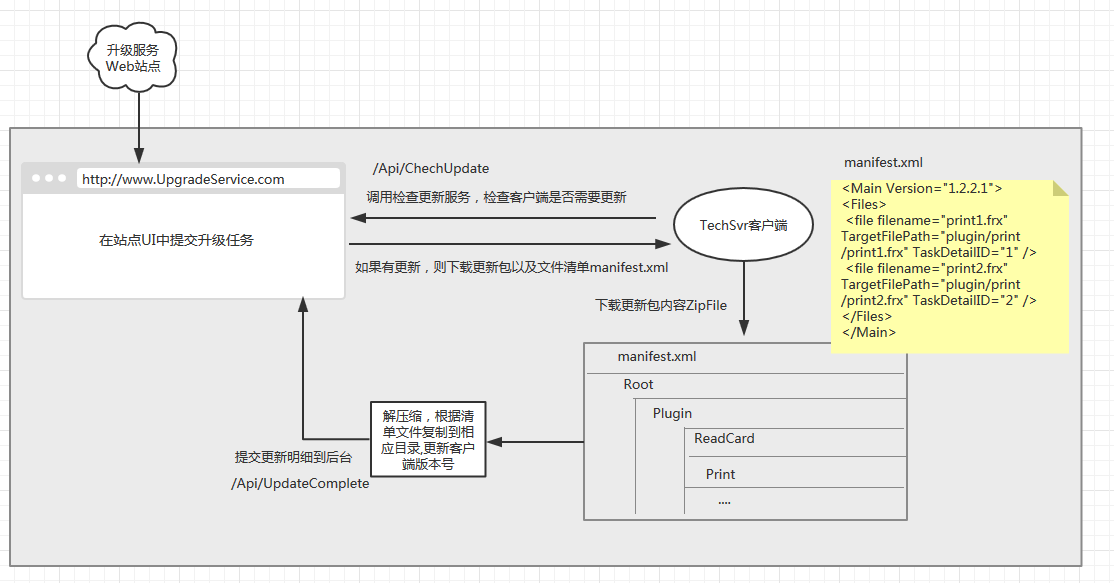
[设备注册 15](#_Toc503454436)

[功能调用统计 16](#_Toc503454437)

# TechSvr服务端升级方案

## 1.1 概述

搭建Web站点，通过站点页面， 创建升级任务，上传待更新的文件，设置单个文件的属性，可以暂存任务，或者启动任务，任务只有在启动之后，才会触发客户端升级 。



* 任务示例：



## 1.2 升级任务表字段

* UpgradeTask 升级任务主表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | 字段列 | 含义 | 类型 | 说明 |
| 1 | ID | 主键ID | Int | 主键ID |
| 2 | CURRENTVERSION | 版本号 | String | 版本号 |
| 3 | HOSPITALCODEStr | 医疗结构代码字符串 | string | 为空则更新所有，机构之间用逗号分隔 |
| 4 | STATUS | 任务状态 | Int | 1表示已启动，2表示暂存 |
| 5 | FORCED | 标识是否强制执行 | bool | 是否强制执行 |
| 6 | REMARK | 升级任务说明 | string | 升级任务说明 |
| 7 | CREATEON | 创建时间 | datetime | 创建时间 |
| 8 | CREATEBY | 创建人 | string | 创建人 |

* UpgradeTaskDetail 任务明细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | 字段列 | 含义 | 类型 | 说明 |
| 1 | ID | 主键 | Int | 主键ID |
| 2 | UPGRADETASKID | 外键 | int | 任务主表ID |
| 3 | NEEDRESTARTCLIENT | 重启客户端 | bool | 是否需要重启客户端 |
| 4 | FILEPATH | 文件路径 | nvarchar | 服务端文件路径 |
| 5 | FILENAME | 文件名 | nvarchar | 不包含路径的文件名 |
| 6 | TARGETFILEPATH | 客户端路径 | nvarchar | 目标客户端待更新的文件路径（相对目录） |
| 7 | REMARK | 说明 | nvarchar | 备注 |
| 8 | CREATEON | 创建时间 | datetime | 创建时间 |
| 9 | CREATEBY | 创建人 | nvarchar | 创建人 |

* FileUpgradeRecord 文件升级记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | 字段列 | 含义 | 类型 | 说明 |
| 1 | ID | 主键 | Int | 主键ID |
| 2 | CLIENTID | 客户端身份标识 | nvarchar | 电脑名+IP 作为客户端身份标识 |
| 3 | UPGRADETASKDETAILID | 外键 | Int | 对应的任务明细ID |
| 4 | FILENAME | 文件名 | nvarchar | 文件名 |
| 5 | UPGRADERESULT | 升级结果 | nvarchar | 升级结果（成功或失败） |
| 6 | REMARK | 失败原因 | nvarchar | 失败原因 |
| 7 | UPGRADEON | 升级时间 | datetime | 升级时间 |
| 8 | OLDCURRENTVERSION | 升级之前的版本号 | nvarchar | 升级之前的版本号 |
| 9 | CURRENTVERSION | 升级之后的版本号 | nvarchar | 升级之后的版本号 |

## 1.3 升级范围设置

### 1.3.1 医院范围升级

当任务的hospitalcodeStr字段为空，则更新所有机构，可指定单个机构，或多个机构，当升级多个机构时，则机构之间用逗号分隔

## 1.4 服务接口定义

### 1.4.1检查更新接口

/Api/ChechUpdate

调用此接口检查是否客户端需要更新，如果需要更新，则设置返回参数HasUpdate为TRUE，并打包一个文件返回到客户端

* 接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名 | 含义 | 类型 | 说明 |
| 1 | CurrentVersion | 客户端版本 | string | 客户端当前版本号 |
| 2 | HospitalCode | 机构代码 | string | 医疗结构代码 |
| 3 | ClientID | 客户端身份标识 | string | 客户端身份标识 |

* 接口返回值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名 | 含义 | 类型 | 说明 |
| 1 | HasUpdate | 是否有更新 | string | 是否有更新 |
| 2 | ZipFile | 更新包 | FileStream | 更新包 |
| 3 | ResultMessage | 接口返回描述 | string | 接口返回描述 |
| 4 | ResultCode | 接口返回代码 | Int | 200 表示成功 500 错误 |

* ZipFile内容包含一个清单文件manifest.xml ，以及带有目录结构的更新的文件列表

<Main Version="1.2.2.1" forced="1">

<Files>

<file filename="print1.frx" TaskDetailID="1" TargetFilePath="plugin/print/print1.frx" />

<file filename="print2.frx" TaskDetailID="2" TargetFilePath="plugin/print/print2.frx" />

<file filename="ReadCard.dll" TaskDetailID="3" TargetFilePath="plugin/ReadCard/ ReadCard.dll " />

</Files>

</Main>

### 1.4.2升级完成接口

/Api/UpdateComplete

客户端升级的之前，先备份客户端文件到临时目录，然后根据清单覆盖目录需要更新的文件，反馈更新结果到服务端

* 接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名 | 含义 | 类型 | 说明 |
| 1 | TaskDetailIDs | 任务明细ID数组 | string | 任务明细ID数组，从下载过来的清单文件中获取 |
| 2 | CurrentVersion | 版本号 | string | 客户端更新之后的版本 |
| 3 | HospitalCode | 客户端医疗结构代码 | string | 客户端医疗结构代码 |
| 4 | ClientID | 身份标识 | string | 客户端身份标识 |

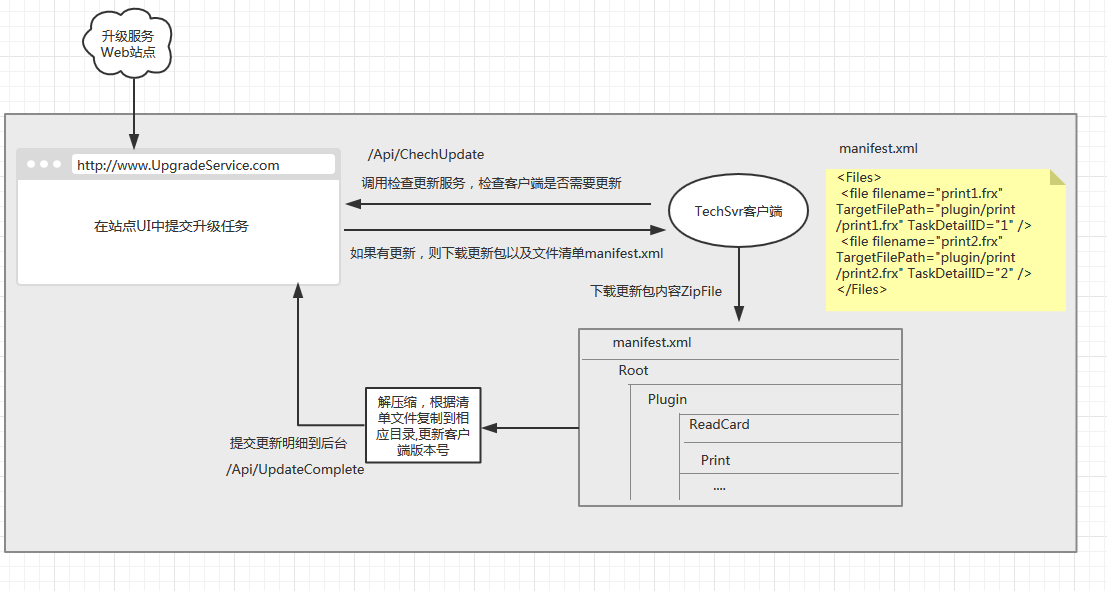
* 接口返回值：

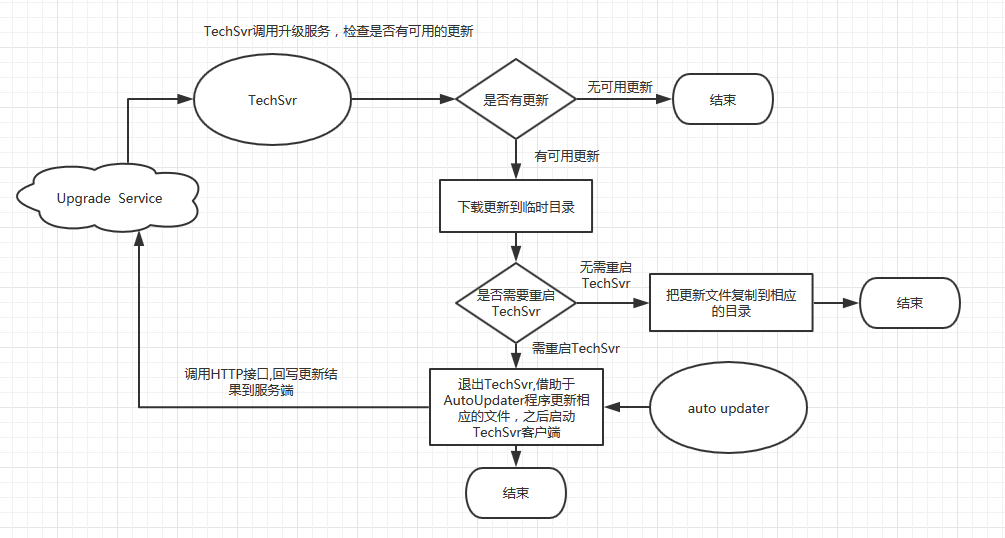
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数名 | 含义 | 类型 | 说明 |
| 1 | ResultMessage | 接口返回描述 | string | 接口返回描述 |
| 2 | ResultCode | 接口返回代码 | Int | 200 表示成功 500 错误 |

# TechSvr客户端升级方案

## 2.1 概述

TechSvr客户端通过定时检查更新或者手动触发更新来完成版本更新检查；





### 2.1.1升级说明

1. 比较两边的版本号判断是否有更新，有更新则下载服务端压缩包到指定的临时目录;
2. 下载完成之后解压缩，备份当前客户端;
3. 根据客户端临时目录结构以及下载的清单文件来覆盖本地的文件;
4. 如果是TechSvr程序本身升级，可能还需要临时关闭TechSvr客户端;
5. 升级完毕之后，修改本地的版本号信息;
6. 最后由AutoUpdater重启TechSvr客户端(如果是TechSvr程序本身升级的情况下);

清单文件manifest.xml ，以及带有目录结构的更新的文件列表

<Main Version="1.2.2.1" forced="1">

<Files>

<file filename="print1.frx" TaskDetailID="1" TargetFilePath="plugin/print/print1.frx" />

<file filename="print2.frx" TaskDetailID="2" TargetFilePath="plugin/print/print2.frx" />

<file filename="ReadCard.dll" TaskDetailID="3" TargetFilePath="plugin/ReadCard/ ReadCard.dll " />

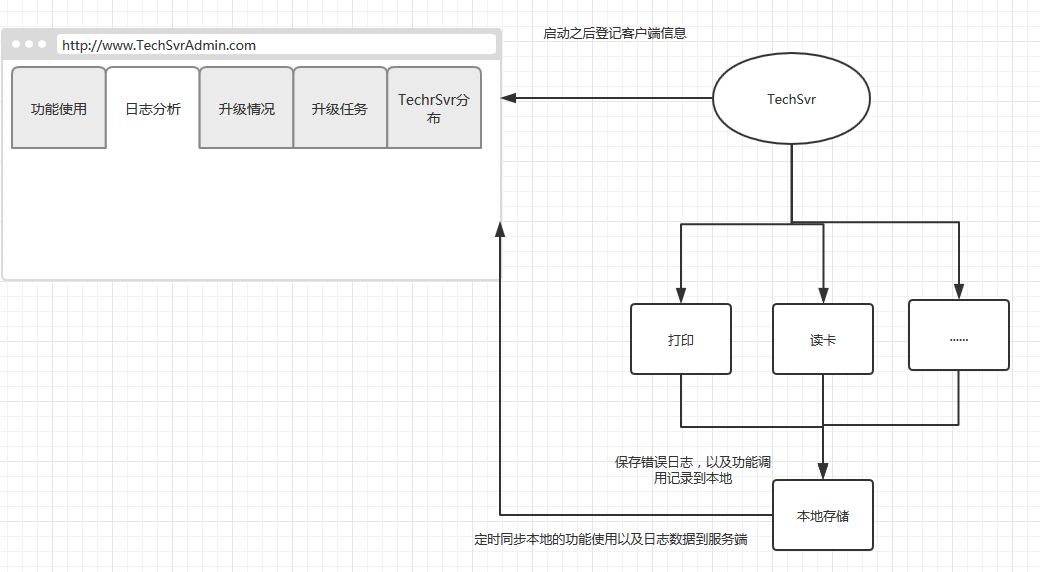
</Files>

</Main>

# 服务端中心管理方案

## 3.1 概述

为了更好的管理和监控TechSvr客户端，需要借助于一个服务端管理系统，



## 3.2 原型图展示：

### 3.2.1 升级记录页面



### 3.2.2 日志分析页面



### 3.2.3 功能调用页面



### 3.2.4 客户端统计页面



### 3.2.5 任务操作页面



## 3.3 功能点

### 3.3.1 监控各客户端TechSvr运行情况

#### 当前局域网内有多少TechSvr登记在册

TechSvr客户端启动的时候自动到服务端登记信息

#### TechSvr客户端的PC信息

TechSvr客户端启动的时候自动到服务端登记客户端的PC信息

包括CPUID,计算机名

#### TechSvr客户端当前版本以及升级记录

TechSvr客户端启动的时候自动到服务端登记客户端的版本信息，切每次升级之后登记一次升级信息

#### 功能使用统计

功能调用以后，上传数据到服务端

#### 错误记录分析

客户端发生错误，则上传错误日志到服务端

## 3.4 可以通过该平台设置更新包

在平台界面直接上传更新包以及更新范围，

## 3.5 平台支持

### 3.5.1 Windows Server

asp.net web + SqlServer

### 3.5.2 Linux

Java Web +MySql

## 3.6 数据表设计

### 3.6.1 日志表 Log

保存客户端错误日志

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 字段列 | 含义 | 类型 | 说明 |
| 1 | ID | 主键 | Int | 接口返回描述 |
| 2 | CLIENTID | 身份标识 | string | 客户端身份标识 |
| 3 | CURRENTVERSION | 客户端版本 | string | 客户端版本号 |
| 4 | ERRORTYPE | 错误类型 | string | 错误类型 |
| 5 | ERRORMSG | 错误内容 | stirng | 错误内容 |
| 6 | FUNCNAME | 插件名称 | String | 被调用的功能名称 |
| 7 | CREATEON | 创建时间 | datetime | 调用时间 |

### 3.6.2 设备表 Device

* 注册设备的PC信息到服务端

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 字段列 | 含义 | 类型 | 说明 |
| 1 | ID | 主键 | Int | 主键ID |
| 2 | CPUID | CPU序列号 | String | CPU序列号 |
| 3 | PCNAME | 计算机名 | String | 计算机名 |
| 4 | LOCALIP | 本地IP地址 | String | 本地IP地址 |
| 5 | MAC | MAC | String | MAC地址 |
| 6 | CREATEON | 创建时间 | Datetime | 登记时间 |
| 7 | CURRENTVERSION | 版本号 | String | 客户端版本号 |
| 8 | HOSPITALCODE | 机构代码 | String | 机构代码 |

### 3.6.3 功能调用统计表 FuncUsage

当某个功能在客户端被调用，则先保存到本地，然后定时上传到服务端数据库

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 字段列 | 含义 | 类型 | 说明 |
| 1 | ID | 主键 | Int | 主键ID |
| 2 | CLIENTID | 身份标识 | String | 客户端身份身份标识 |
| 3 | FUNCNAME | 功能名称 | String | 功能名称 |
| 4 | ISSUCCESS | 调用是否成功 | String | 调用是否成功，如果出错则记录错误 |
| 5 | REMARK | 错误描述 | String | 错误描述 |
| 6 | CREATEON | 创建时间 | Datetime | 调用时间 |

## 3.7 接口调用

/Api/UploadMonitorLogs

上传本地监控日志到服务器，先把监控日志保存到本地，然后定时上传到服务器

### 日志上传接口

* 接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | Data | 日志数据 | List<Log> | 日志数据数组（Log的定义如下） |
| 2 | ClientID | 身份标识 | String | 客户端身份身份标识 |
| 3 | FunName | 功能名称 | String | 功能名称 |
| 4 | CurrentVersion | 版本号 | String | 客户端版本号 |
| 5 | HospitalCode | 机构代码 | String | 机构代码 |

* Log 实体定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | ErrorType | 日志类型 | String | 日志类型 |
| 2 | ErrorMsg | 错误消息 | String | 错误消息 |
| 3 | FuncName | 功能名称 | String | 功能名称 |

* 接口返回值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | ResultMessage | 结果描述 | String | 接口返回描述 |
| 2 | ResultCode | 结果代码 | Int | 2200 表示成功 500 错误 |

### 设备注册

/Api/RegisterDevice

客户端启动之后 自动注册设备信息

* 接口参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NO. | 参数 | | 含义 | | 类型 | | 说明 | |
| 1 | CPUID | CPU序列号 | | String | | CPU序列号 | |
| 2 | PCName | 计算机名 | | String | | 计算机名 | |
| 3 | LocalIP | 本地IP地址 | | String | | 本地IP地址 | |
| 4 | MAC | MAC | | String | | MAC地址 | |
| 5 | CurrentVersion | | 版本号 | | String | | 客户端版本号 | |
| 6 | HospitalCode | | 机构代码 | | String | | 机构代码 | |

* 接口返回值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | ResultMessage | 结果描述 | String | 接口返回描述 |
| 2 | ResultCode | 结果代码 | Int | 2200 表示成功 500 错误 |

### 功能调用统计

/Api/UploadFuncInvoke

客户端插件在使用的同时，把调用信息保存到本地，然后定时调用此接口批量上传到服务端

* 接口参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | Data | 功能调用数组 | List<InvokeItem> | 功能调用数组,包含多条功能调用对象，如下图 |
| 2 | CurrentVersion | 版本号 | String | 客户端版本号 |
| 3 | HospitalCode | 机构代码 | String | 机构代码 |
| 4 | ClientID | 版本号 | String | 客户端版本号 |

* InvokeItem实体定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | InvokeResult | 调用结果 | String | 调用结果 |
| 2 | Remark | 描述 | String | 描述 |
| 3 | FuncName | 功能名称 | String | 被调用的功能名称 |

* 接口返回值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | 参数 | 含义 | 类型 | 说明 |
| 1 | ResultMessage | 结果描述 | String | 接口返回描述 |
| 2 | ResultCode | 结果代码 | Int | 2200 表示成功 500 错误 |