

# 通信装备故障排除电子手册<sup>①</sup>

张弛, 黄孝文, 尹德超

(中国人民解放军 94907 部队, 南昌 330013)

**摘要:** 针对当前部队通信装备排故的现状, 设计了一种基于 .NET 的通信装备故障排除电子手册. 系统使用 .NET 环境、Access 数据库以及 ADO.NET 核心组件, 实现了用户管理、故障信息管理以及数据维护等功能. 首先分析了系统的总体架构和系统功能, 其次对系统关键技术进行了详细阐述, 最后结合日常应用说明了系统的可行性.

**关键词:** 通信装备; 故障排除; 电子手册; .NET; ADO

## Electronic Handbook of Communication Equipment Failure Recovery

ZHANG Chi, HUANG Xiao-Wen, YIN De-Chao

(The Unit 94907 of PLA, Nanchang 330013, China)

**Abstract:** Aimed at the current situation of army's communication equipment failure recovery, a new kind of electronic handbook of communication equipment failure recovery based on .NET is designed. The system uses .NET surroundings and Access data base and the core components of ADO.NET to realize the functions of user management and fault information management and data maintenance and so on. This paper firstly analyses the system's overall framework and system function. Then, it elaborates the system's key technology. Finally, it proves the validity of the system through general use.

**Key words:** communication equipment; failure recovery; electronic handbook; .NET; ADO

## 1 引言

高技术 在通信领域中的广泛应用, 使得部队通信装备越来越复杂, 功能越来越齐全, 同时, 装备的操作及故障排除难度也不断加大, 因此, 加强通信装备的科学管理, 充分发挥其效能, 是当前亟待研究解决的重大课题. 但在实际工作中, 官兵对故障排除经验收集整理还不够重视, 导致很多宝贵经验流失, 影响了部队技术传承、人才培养和战斗力提升. 现行装备故障诊断与排除经验案例主要靠修理人员以笔、纸、口头传承和人工记忆等方式记录, 缺乏资源共享最大化的优势, 在未来战争中, 将影响部队战斗力迅速恢复, 甚至影响整场战争输赢. 针对以上问题, 本文设计了一种基于 .NET 的通信装备故障排除电子手册. 操作者可以通过登陆本电子手册, 进行相关装备的信息查询, 获取排除故障的解决方法, 亦可实时增添新的

故障信息及解决办法或修改删除以往的故障信息, 非常方便实用.

## 2 系统设计

### 2.1 系统总体框架设计

通信装备故障排除电子手册实际运行过程如下:

使用者通过用户名密码登录到系统中, 如果没有分配权限则自动退出系统并需要向管理员申请权限, 否则可进行相应权限事件. 该系统总体结构框架如图 1 所示.

### 2.2 系统各模块设计

本系统共有三个大模块, 分别是用户管理模块、故障信息管理模块、数据维护模块.

<sup>①</sup> 收稿时间:2014-05-13;收到修改稿时间:2014-06-17

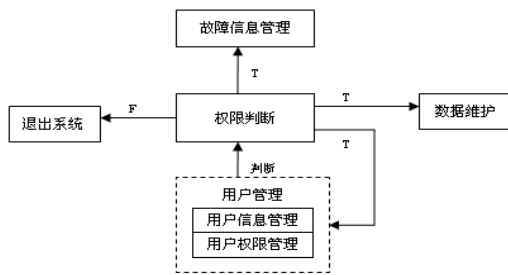


图 1 系统总体结构框架

每个模块的具体功能分别为:

(1)用户管理模块: 主要包括了登陆管理系统、退出管理系统、添加用户信息、注销用户信息、修改用户信息、分配用户权限等功能。

(2)故障信息管理模块: 主要包括了对装备故障信息的查询、添加、删除、修改、刷新、打印等功能, 同时实现用户退出系统的功能。

(3)数据维护模块: 主要包括了对数据的备份、恢复和清空等功能, 同时实现将系统数据导出到 Excel 以及将 Excel 中的数据导入到系统中的功能。

具体的系统功能结构如图 2 所示。

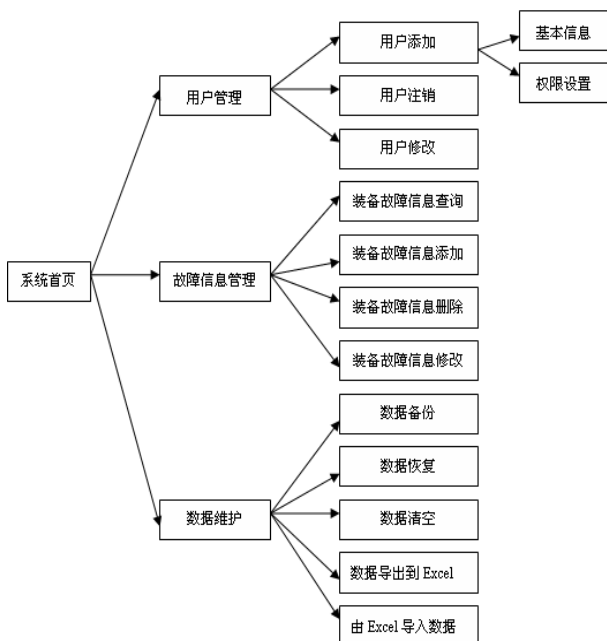


图 2 系统功能结构图

### 3 系统实现

根据系统特点, 我们选用 Visual C#开发前台应用程序, Access 作为数据库服务器, 采用 ADO.NET 方式对数据库进行访问。

#### 3.1 后台数据库

本系统采用 Access 数据库管理系统. 数据库是系统的核心, 针对给定的应用环境, 要构建最优的数据库模式, 建立数据库, 使其满足用户需求. 本系统数据表设计如下:

(1)tbUserInfo 表: 即用户信息表, 用来描述 filedUserName(用户名称), filedUserPwd(用户密码)和是否具有相应授权等信息. 如表 1.

表 1 tbUserInfo 表

名称	数据类型	字段大小	描述
filedUserID	自动编号	长整型	-
filedUserName	文本	50	主键
filedUserPwd	文本	50	-
AddUser	是/否	-	-
DelUser	是/否	-	-
ModiUser	是/否	-	-
AddEquip	是/否	-	-
DelEquip	是/否	-	-
ModiEquip	是/否	-	-
InquireEquip	是/否	-	-
DataBackup	是/否	-	-
DataRestore	是/否	-	-
DataClear	是/否	-	-
DataOut	是/否	-	-
DataIn	是/否	-	-

(2)EquipInfo 表: 即装备信息表, 用来描述各装备的 EquipName(装备名称), Appear(故障现象), Cause(故障可能原因), Way(解决办法)和 Remark(备注)等信息. 如表 2.

表 2 EquipInfo 表

名称	数据类型	字段大小	描述
ID	自动编号	长整型	主键
EquipName	文本	50	-
Appear	备注	-	-
Cause	备注	-	-
Way	备注	-	-
Remark	备注	-	-

#### 3.2 系统关键技术实现

##### 3.2.1 用户权限

因本系统涉及到单位各部门不同身份的人员使用, 故需要对各个用户赋予不同的功能权限. 当用户登录操作时, 因受其权限字段控制, 会使得系统更安全, 有效防止误操作。

在编程实现过程中, 通过建立数据库的成员资格管理方法进行身份验证。

(1)定义权限. 定义权限的 SQL 语句为:

```
sql = "insert into tbUserInfo (filedUserName,
filedUserPwd,AddUser, DelUser, ModiUser, AddEquip,
DelEquip, ModiEquip, InquireEquip, DataBackup,
DataRestore, DataClear, DataOut, DataIn) values (" +
tbUserName.Text.Trim() + "," + tbPwd.Text.Trim() +
"," + 用户添加.Checked + "," + 用户删除.Checked +
"," + 用户修改.Checked + "," + 故障信息添
加.Checked + "," + 故障信息删除.Checked + "," + 故障
信息修改.Checked + "," + 故障信息查询.Checked + ","
+ 数据备份.Checked + "," + 数据恢复.Checked + "," +
数据清空.Checked + "," + 数据导出.Checked + "," +
数据导入.Checked + ")".
```

(2)判断用户身份和权限. 即在用户登录时, 根据用户名从数据库的 tbUserInfo 表中读取用户的资料, 如果某项为 false, 则用户不能访问该项功能, 如果为 true, 则可以操作.

### 3.2.2 装备故障信息的查询、添加、删除和修改

该系统的表格通过 DataGridView 控件实现. Visual Studio.Net 中 DataGridView 显示数据源的指定字段, 通过设置属性和编写代码, 可以自定义数据项的外表和功能. 装备故障信息管理模块主界面如图 3.



图 3 装备故障信息管理模块主界面

(1) 查询. 在界面左边的装备名称下拉框 (cbSpecialty) 中直接选择装备名称或直接输入装备名称的一个或多个字符, 点击查询按钮即可在右边表格中出现相关装备信息. 这两种方法各有优点, 点击下拉框, 若有需要查询的装备, 则直接选择设备类型进行查询, 这不止方便快捷而且可以避免人为输入错误, 如图 4.



图 4 查询装备信息

若装备过多, 在下拉框中寻找对应装备名称则过于花费时间, 此时只要在下拉框中输入装备名称的一个或多个字符, 点击查询, 即可出现相关信息. 如上图 3.

查询功能首先进行数据库连接, 再进行相应字段查询. 部分代码如下:

```
OleDbConnectionconn = new
OleDbConnection(frmMain.odbcConnStr);
conn.Open();
OleDbDataAdapter adp = new OleDbDataAdapter("select
distinct EquipName from EquipInfo", conn);
DataSet ds = new DataSet();
adp.Fill(ds);
cbSpecialty.DataSource = ds.Tables[0].DefaultView;
.....
OleDbDataAdapter adp1 = new OleDbDataAdapter("select *
from EquipInfo where EquipName like '%" +
textBox1.Text.Trim() + "%'", conn);
DataSet ds1 = new DataSet();
adp1.Fill(ds1);
if (ds1.Tables[0].Rows.Count > 0)
{
cbSpecialty.Text = textBox1.Text;
conn.Close();
}
else
{
cbSpecialty.SelectedIndex = -1;
.....
```

若数据库中有需要查询的装备信息, 则右边表格就会出现查询的相关信息, 使用者可以比对存储的故障信息中是否有需要查询的故障现象, 如果有, 则可根据显示的解决办法这一栏排除故障, 例如 XX 手持终端出现了液晶模块无显示的故障, 经过查询, 出现了图 4 中的查询结果, 则可根据表格中的第 1、3、4、5 行的数据排除故障, 非常方便实用, 大大提高了工作效率.

(2)添加、删除和修改. 这三个模块的实现原理都是类似的. 首先自定义函数实现表格绑定数据库的功能, 然后进行数据库连接, 添加(删除、修改)数据记录, 关闭数据库连接, 表格绑定数据源. 添加事件的部分代码如下:

```
if (tbEquipName.Text.Trim() == "" || tbAppear.Text.Trim() ==
    "" || rtbRemark.Text.Trim() == "")
{
    MessageBox.Show("请将基本信息填写完整!", "提示",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
else
{
    OleDbConnection conn = new
        OleDbConnection(frmMain.odbcConnStr);
    conn.Open();
    string sql1, sql2, sql3;
    sql1 = "insert into EquipInfo
        (EquipName,Appear,Cause,Way";
    sql2 = " values (" + tbEquipName.Text.Trim() + "," +
        tbAppear.Text.Trim() + "," + tbCause.Text.Trim() + "," +
        + tbWay.Text.Trim() + """;
    if (rtbRemark.Text.Trim() != "")
    {
        sql1 = sql1 + ",Remark";
        sql2 = sql2 + "," + rtbRemark.Text.Trim() + """;
    }
    sql3 = sql1 + ")" + sql2 + "));
    OleDbCommand cmd = new OleDbCommand(sql3, conn);
    cmd.ExecuteNonQuery();
    MessageBox.Show("添加成功!", "提示",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    tbEquipName.Clear();
    tbAppear.Clear();
    tbCause.Clear();
    tbWay.Clear();
    rtbRemark.Clear();
    conn.Close();
    cmd.Dispose();
}
```

由于篇幅有限, 删除和修改功能这里不再赘述.

### 3.2.3 数据维护

(1)数据备份和恢复. 为了传递和共享资源信息, 也为了防止误操作, 本系统设计了数据备份和恢复功能. 因为考虑到以下几个因素: 1)具备平台无关性, 可

以在不同平台和应用程序之间交换数据; 2)有利于快速检索和查询; 3)便于在网路中传输. 而 XML 文档正好满足这些需求, 故采用 XML 文档来描述数据信息. 数据备份主要是调用 BackupData 函数, 将数据信息存入本地 XML 文件中. 部分相关代码如下:

```
string backupName;
backupName = DateTime.Now.ToString("yyyyMMddHHmm")
    + ".xml";
txtSavePath.Text = Application.StartupPath + "\\backup\\" +
    backupName;
}
.....
string fileName = "";
DialogResult dlgResult;
sFileDialog.Title = "文件另存为";
sFileDialog.InitialDirectory = Application.StartupPath +
    "\\backup\\";
sFileDialog.AddExtension = true;
sFileDialog.Filter = "数据文件 (*.xml)|*.xml";
dlgResult = sFileDialog.ShowDialog();
.....
private bool BackupData(string fileName)
{
    OleDbConnection conn = new
        OleDbConnection(frmMain.odbcConnStr);
    conn.Open();
    DataSet ds = new DataSet();
    string sql = "select
        ID,EquipName,Appear,Cause,Way,Remark from
        EquipInfo";
    OleDbDataAdapter adp = new OleDbDataAdapter(sql,
        conn);
    adp.Fill(ds, "EquipInfo");
    ds.WriteXml(fileName);
    conn.Close();
return true;
.....
```

数据恢复的实现技术与数据备份类似, 它主要调用 RestoreData 函数, 将 XML 文件内容导入系统数据库中. 具体代码这里不再赘述.

(2)系统数据和 Excel 数据的互通. 为了更大一步提高本系统的灵活性和通用性, 扩大使用范围, 本系统增加了可将系统数据导入到 Excel 中, 也可将 Excel 中的数据导入系统中的功能. 数据导出主要通过调用

ExportDt 函数, 若 DataGridView 的行数>0, 则启动此函数. 数据导入主要是通过连接 Excel, 将其数据绑定到系统数据库中来实现. 其部分代码如下:

```
private bool ExportDt(DataGridView dataGridView1,
bool isShowExcle)
{
    Excel.Application excel = new Excel.Application();
    excel.Application.Workbooks.Add(true);
    excel.Visible = isShowExcle;
    for (int i = 1; i < dataGridView1.ColumnCount; i++)
    {
        excel.Cells[1, i] = dataGridView1.Columns[i].HeaderText;
    }
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        for (int j = 1; j < dataGridView1.ColumnCount; j++)
        {
            if (dataGridView1[j, i].ValueType == typeof(string))
            {
                excel.Cells[i + 2, j] = "" + dataGridView1[j,
i].Value.ToString();
            }
            else
            {
                excel.Cells[i + 2, j] = dataGridView1[j,
i].Value.ToString();
            }
        }
    }
}
.....
private void 数据导入ToolStripMenuItem_Click(object sender,
EventArgs e)
{
    string fileName = "";
    DialogResult dlgResult;
    opFileDialog.Title = "选择要导入的文件";
    opFileDialog.Filter = "数据文件 (*.xls)*.xls";
    dlgResult = opFileDialog.ShowDialog();
}
.....
fileName = opFileDialog.FileName;
if (dlgResult == DialogResult.OK)
{
    try
    {
        string StrConn = "Provider=Microsoft.Jet.OLEDB.4.0;" +
"Data Source=" + fileName + ";";
+ "Extended Properties=Excel 5.0;";
        OleDbConnection cnn = new
        OleDbConnection(StrConn);
        cnn.Open();
        string sqlExcel = string.Format("select * from[{0}]",
```

```
"sheet1");
        OleDbDataAdapter adp = new
        OleDbDataAdapter(sqlExcel, cnn);
        DataSet ds = new DataSet();
        adp.Fill(ds, "sheet1");
        OleDbConnection conn = new
        OleDbConnection(odbcConnStr);
        conn.Open();
        foreach (DataRow row in ds.Tables[0].Rows)
        {
            string sql1 = "" + row["EquipName"].ToString() + "," +
            + row["Appear"].ToString() + "," +
            + row["Cause"].ToString() + "," +
            + row["Way"].ToString() + "," +
            + row["Remark"].ToString() + "";
            string sql = "insert into EquipInfo
(EquipName, Appear, Cause, Way, Remark)" + " values(" +
sql1 + ")";
            OleDbCommand cmd = new OleDbCommand(sql,
conn);
            cmd.ExecuteNonQuery();
        }
}
.....
```

#### 4 结语

针对当前部队通信装备排故难、经验缺乏易流失且零散难整理等现状, 设计了一种基于.NET 的通信装备故障排除电子手册. 该系统使用了.NET 中的先进技术, 实现了装备故障信息管理的要求, 提供了可视化的用户操作界面, 使得用户与系统之间具有良好的交互性. 另外, 还实现了办公的无纸化, 速度快、实时性强、运行安全可靠, 在目前试用中运行良好, 取得了不错的效果, 具有广阔的应用前景.

#### 参考文献

- 1 楚王辉. 以项目为导向的 C#编程实用教程. 北京: 机械工业出版社, 2005.
- 2 求是科技. ASP.NET 信息管理系统开发. 北京: 人民邮电出版社, 2006.
- 3 宋依青, 李书旗, 王海滨. 基于 WLAN 的分布式竞赛打分系统的设计与实现. 微电子学与计算机, 2009, 26(11): 165-168.
- 4 张笑青. 基于.NET 的糖尿病管理系统. 计算机系统应用, 2014, 23(2): 70-76.