Seeyon A8 OA系统任意文件上传漏洞分析

1. 通过github找到旧版源码

https://github.com/zhf839428881/seeyon_v3x

根据网上的攻击POC，找到项目关键点

2. 查找请求路由，找到关键Servlet

3. 跟踪关键 HtmlOfficeServlet

HtmlOfficeServlet.java

```
package com.seeyon.v3x.common.office;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.web.context.WebApplicationContext;

import com.seeyon.v3x.common.web.login.CurrentUserToSeeyonApp;
import com.seeyon.v3x.common.web.util.ThreadLocalUtil;

public class HtmlOfficeServlet extends HttpServlet {
```

```
    private static Log log = LogFactory.getLog(HtmlOfficeServlet.class);


    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        CurrentUserToSeeyonApp.set(request.getSession());

        ApplicationContext ctx = (ApplicationContext) getServletContext().getAt
tribute(WebApplicationContext.ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE);
        HandWriteManager handWriteManager = (HandWriteManager) ctx.getBean("han
dWriteManager");
        HtmlHandWriteManager htmlHandWriteManager = (HtmlHandWriteManager) ctx.
getBean("htmlHandWriteManager");

        DBstep.iMsgServer2000 msgObj = new DBstep.iMsgServer2000();
        try {
            handWriteManager.readVariant(request, msgObj);

            msgObj.SetMsgByName("CLIENTIP", request.getRemoteAddr());

            String option = msgObj.GetMsgByName("OPTION");

            if ("LOADFILE".equalsIgnoreCase(option)) {
                handWriteManager.LoadFile(msgObj);
            }
            else if("LOADSIGNATURE".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.loadDocumentSinature(msgObj);
            }
            else if("LOADMARKLIST".equalsIgnoreCase(option))
            {
                handWriteManager.LoadSinatureList(msgObj);
            }
            else if("SIGNATRUEIMAGE".equalsIgnoreCase(option))
            {
                handWriteManager.LoadSinature(msgObj);
            }
            else if("SAVESIGNATURE".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.saveSignature(msgObj);
            }
            else if("SAVEHISTORY".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.saveSignatureHistory(msgObj);
```

```
            }
            else if("SIGNATRUELIST".equalsIgnoreCase(option))
            {//调入印章列表
                handWriteManager.LoadSinatureList(msgObj);
            }
            else if("SHOWHISTORY".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.getSignatureHistory(msgObj);
            }

            handWriteManager.sendPackage(response, msgObj);
        }
        catch (Exception e) {
            log.error("",e);
            msgObj = new DBstep.iMsgServer2000();
            msgObj.MsgError("htmoffice operate err");
            handWriteManager.sendPackage(response, msgObj);
        }

        ThreadLocalUtil.removeThreadLocal();
    }

    @Override
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException
{
        this.doGet(request, response);
    }

}
```

发现在 `handWriteManager.readVariant(request, msgObj);` 对数据包里的参数进行了解析

```
public void readVariant(HttpServletRequest request, iMsgServer2000 msgObj) {
        /*byte[] bs = null;
        try {
            InputStream in = request.getInputStream();
            if (in != null) {
                bs = org.apache.commons.io.IOUtils.toByteArray(in);
            }
        }
        catch (IOException e1) {
        }

        if (bs != null) {
```

```
            msgObj.MsgVariant(bs);
        }
        */
        msgObj.ReadPackage(request);

        fileId = new Long(msgObj.GetMsgByName("RECORDID"));
        createDate = com.seeyon.v3x.util.Datetimes.parseDatetime(msgObj.GetMsgB
yName("CREATEDATE"));

        String _originalFileId = msgObj.GetMsgByName("originalFileId");
        String _originalCreateDate = msgObj.GetMsgByName("originalCreateDate");
        needClone = _originalFileId != null && !"".equals(_originalFileId.trim(
));

        needReadFile = Boolean.parseBoolean(msgObj.GetMsgByName("needReadFile")
);

        if(needClone){
            originalFileId = new Long(_originalFileId);
            originalCreateDate = com.seeyon.v3x.util.Datetimes.parseDatetime(_o
riginalCreateDate);
        }
    }
```

其中 `ReadPackage` 函数解析整个POST内容（**能够在iMsgServer2000.java中看到**）

```
public byte[] ReadPackage(HttpServletRequest request) {
        int totalRead = 0;
        int readBytes = 0;
        int totalBytes = 0;
        this.Charset = request.getCharacterEncoding();
        if (this.Charset == null) {
            this.Charset = request.getHeader("charset");
        }
        if (this.Charset == null) {
            this.Charset = "GB2312";
        }
        if (this.FDebug) {
            System.out.println("Charset :" + this.Charset);
        }
        try {
            totalBytes = request.getContentLength();
            this._$904 = new byte[totalBytes];
            while (totalRead < totalBytes) {
                request.getInputStream();
```

```
                readBytes = request.getInputStream().read(this._$904, totalRead
, totalBytes - totalRead);
                totalRead += readBytes;
            }
            if (this._$907 == "") {
                this._$1015();
            }
        } catch (Exception e) {
            System.out.println("ReadPackage:" + e.toString());
        }
        return this._$904;
    }
```

4. 发现 `HtmlOfficeServlet` 并没有保存文件相关操作，继续跟进其他类 `HandWriteManager` 和 `HtmlHandWriteManager`

HandWriteManager.java

```
package com.seeyon.v3x.common.office;

import java.io.File;
import java.io.FileNotFoundException;
import java.sql.Timestamp;
import java.util.Date;
import java.util.List;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.io.FileUtils;
import org.apache.commons.lang.NumberUtils;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import DBstep.iMsgServer2000;

import com.seeyon.v3x.common.SystemEnvironment;
import com.seeyon.v3x.common.authenticate.domain.User;
import com.seeyon.v3x.common.barCode.manager.BarCodeManager;
import com.seeyon.v3x.common.cache.CacheAccessable;
import com.seeyon.v3x.common.cache.CacheFactory;
import com.seeyon.v3x.common.cache.CacheMap;
import com.seeyon.v3x.common.constants.SystemProperties;
```

```java
import com.seeyon.v3x.common.encrypt.CoderFactory;
import com.seeyon.v3x.common.exceptions.BusinessException;
import com.seeyon.v3x.common.filemanager.Constants;
import com.seeyon.v3x.common.filemanager.V3XFile;
import com.seeyon.v3x.common.filemanager.manager.FileManager;
import com.seeyon.v3x.common.filemanager.manager.Util;
import com.seeyon.v3x.common.i18n.ResourceBundleUtil;
import com.seeyon.v3x.common.office.trans.manager.OfficeTransManager;
import com.seeyon.v3x.common.office.trans.util.OfficeTransHelper;
import com.seeyon.v3x.common.utils.UUIDLong;
import com.seeyon.v3x.common.web.login.CurrentUser;
import com.seeyon.v3x.common.web.util.ApplicationContextHolder;
import com.seeyon.v3x.main.MainDataLoader;
import com.seeyon.v3x.organization.domain.V3xOrgMember;
import com.seeyon.v3x.organization.manager.OnLineManager;
import com.seeyon.v3x.organization.manager.OrgManager;
import com.seeyon.v3x.system.signet.domain.V3xDocumentSignature;
import com.seeyon.v3x.system.signet.domain.V3xSignet;
import com.seeyon.v3x.system.signet.manager.SignetManager;
import com.seeyon.v3x.util.Datetimes;
import com.seeyon.v3x.util.Strings;


/**
 *
 *
 * @author <a href="mailto:tanmf@seeyon.com">Tanmf</a>
 * @version 1.0 2006-12-12
 */
public class HandWriteManager {

    private static OnLineManager onLineManager;
    private static OrgManager orgManager;

    private static Log log = LogFactory.getLog(HandWriteManager.class);

    private static String rc="com.seeyon.v3x.common.resources.i18n.SeeyonCommon
Resources";


    private Long fileId;

    private Date createDate;

    private Long originalFileId;
```

```
    private Date originalCreateDate;

    private boolean needClone = false;

    private boolean needReadFile = false;

    private FileManager fileManager;

    private SignetManager signetManager;

    private OfficeTransManager officeTransManager;

    private BarCodeManager barCodeManager;

    public OfficeTransManager getOfficeTransManager() {
        return officeTransManager;
    }

    public void setOfficeTransManager(OfficeTransManager officeTransManager) {
        this.officeTransManager = officeTransManager;
    }
    private synchronized void init() {
        if(onLineManager == null){
            orgManager = (OrgManager) ApplicationContextHolder.getBean("OrgMana
ger");
            onLineManager = (OnLineManager)ApplicationContextHolder.getBean("on
LineManager");
        }
    }
    public HandWriteManager()
    {
        init();
    }

    public void setSignetManager(SignetManager signetManager)
    {
        this.signetManager=signetManager;
    }

    public void setFileManager(FileManager fileManager) {
        this.fileManager = fileManager;
    }

    public void setBarCodeManager(BarCodeManager barCodeManager) {
```

```
        this.barCodeManager = barCodeManager;
    }
    //调入用户有权使用的印章列表
    public boolean LoadSinatureList(iMsgServer2000 msgObj) throws BusinessExcep
tion {
        List <V3xSignet>ls=null;
        Long userId=CurrentUser.get().getId();
        try{
            String mMarkList="";
            ls=signetManager.findSignetByMemberId(userId);
            for(V3xSignet signet:ls)
            {
                mMarkList+=signet.getMarkName()+"\r\n";
            }
            msgObj.SetMsgByName("MARKLIST",mMarkList);
            msgObj.SetMsgByName("SIGNATRUELIST",mMarkList);     //文单印章
            msgObj.MsgError("");                //清除错误信息
        }catch(Exception e)
        {
            log.error(e);
            throw new BusinessException(e.getMessage());
        }
        return true;
    }

    //根据印章名称，密码，调入印章图片
    public boolean LoadSinature(iMsgServer2000 msgObj) throws BusinessException
 {
         String mMarkName=msgObj.GetMsgByName("IMAGENAME");          //取得文档名

         //String mUserName=msgObj.GetMsgByName("USERNAME");         //取得文档名
         String mPassword=msgObj.GetMsgByName("PASSWORD");       //取得文档类型
         msgObj.MsgTextClear();
         V3xSignet signet=null;
         try{
             signet=signetManager.findByMarknameAndPassword(mMarkName, mPasswor
d);
             if(signet!=null)
             {
                byte[] b = signet.getMarkBodyByte();

                 msgObj.SetMsgByName("IMAGETYPE",signet.getImgType());         /
/设置图片类型
                 msgObj.MsgFileBody(b);          //将文件信息打包
                 msgObj.SetMsgByName ("SIGNTYPE",Integer.toString(signet.getMar
```

```
kType());                        // (手写签名0, 单位印章1) 默认值1
                msgObj.SetMsgByName("STATUS",ResourceBundleUtil.getString(rc,"
ocx.alert.opensucceed.label"));      //设置状态信息
                msgObj.SetMsgByName ("ZORDER","5");    //4:在文字上方 5:在文字下
                msgObj.MsgError("");                //清除错误信息
            }
            else
            {
                msgObj.MsgError(ResourceBundleUtil.getString(rc,"ocx.alert.pwd
err.label"));
            }
        }catch(Exception e)
        {
            throw new BusinessException(e.getMessage());
        }
        return true;
    }

    /**
     * 调用文档的签章记录
     * @param msgObj
     * @return
     * @throws BusinessException
     */
    public boolean LoadDocumentSinature(iMsgServer2000 msgObj) throws BusinessE
xception {
        String mRecordId=msgObj.GetMsgByName("RECORDID");
        msgObj.MsgTextClear();
        List<V3xDocumentSignature> ls=null;
        try{
            ls=signetManager.findDocumentSignatureByDocumentId(mRecordId);
            String mMarkName=ResourceBundleUtil.getString(rc,"ocx.signname.labe
l")+"\r\n";
            String mUserName=ResourceBundleUtil.getString(rc,"ocx.signuser.labe
l")+"\r\n";
            String mDateTime=ResourceBundleUtil.getString(rc,"ocx.signtime.labe
l")+"\r\n";
            String mHostName=ResourceBundleUtil.getString(rc,"ocx.clientip.labe
l")+"\r\n";
            String mMarkGuid=ResourceBundleUtil.getString(rc,"ocx.serialnumber.
label")+"\r\n";
            //log.error("DEBUG INFO:签章记录列头数据: ("+mMarkName+")("+mUserName+
")("+mDateTime+")("+mHostName+")("+mMarkGuid+")");
            for(V3xDocumentSignature ds:ls)
            {
```

```
                mMarkName+=ds.getMarkname()+"\r\n";
                mUserName+=ds.getUsername()+"\r\n";
                mDateTime+=Datetimes.formatDatetime(ds.getSignDate())+"\r\n";
                mHostName+=ds.getHostname()+"\r\n";
                mMarkGuid+=ds.getMarkguid()+"\r\n";
            }
            msgObj.SetMsgByName("MARKNAME",mMarkName);
            msgObj.SetMsgByName("USERNAME",mUserName);
            msgObj.SetMsgByName("DATETIME",mDateTime);
            msgObj.SetMsgByName("HOSTNAME",mHostName);
            msgObj.SetMsgByName("MARKGUID",mMarkGuid);
            msgObj.SetMsgByName("STATUS","调入成功!");       //设置状态信息
            msgObj.MsgError("");                  //清除错误信息
        }catch(Exception e)
        {
            throw new BusinessException(e.getMessage());
        }
        return true;
    }

    /**
     * 调入文件之后,直接把数据放到控件服务器对象msgObj中
     * 调入时查询备份文件ID组合后放如控件，提供花脸查看功能
     * @return
     * @throws BusinessException
     */
    public boolean LoadFile(iMsgServer2000 msgObj) throws Exception {
        // 没有创建实践，说明是新建
        if (createDate == null) {
            msgObj.SetMsgByName("STATUS", "打开成功!");
            msgObj.MsgError("");
            return true;
        }

        Long loadFileId = originalFileId != null ? originalFileId : fileId;
        Date loadCreateDate = originalCreateDate != null ? originalCreateDate :
 createDate;
        String filePath = this.fileManager.getFolder(loadCreateDate, true) + Fi
le.separator ;


        V3XFile tempFile=null;
        File ftemp=new File(filePath+loadFileId);
        if(!(ftemp.exists() && ftemp.isFile()))
        {//传入的createDate错误，没找到文件，重新查找数据库;
```

```
            tempFile=fileManager.getV3XFile(loadFileId);
            if(tempFile!=null)
            {
                filePath = this.fileManager.getFolder(tempFile.getCreateDate(),
 true)+ File.separator ;
            }
        }

        filePath += loadFileId;

        if(needReadFile){
            String newfilePath = CoderFactory.getInstance().decryptFileToTemp(f
ilePath);
            if (msgObj.MsgFileLoad(newfilePath)) {
                if(tempFile==null)
                {
                    tempFile=fileManager.getV3XFile(loadFileId);
                }
                //检查备份
                String checkBack=msgObj.GetMsgByName("checkBack");
                if(!"false".equals(checkBack))
                {
                    if(tempFile!=null)
                    {
                        //设置调入文档备份文档的IDs
                        msgObj.SetMsgByName("backupIds",findBackFileIds(tempFil
e.getFilename()));
                    }
                }
                if(tempFile!=null)
                {
                    Date officeUpateTime=tempFile.getUpdateDate();
                    if(officeUpateTime==null){officeUpateTime=tempFile.getCreat
eDate();}
                    if(officeUpateTime!=null)
                    {
                        msgObj.SetMsgByName("OfficeUpdateTime",Long.toString(of
ficeUpateTime.getTime()));
                    }
                }
                msgObj.SetMsgByName("STATUS", "打开成功!");
                msgObj.MsgError("");
                return true;
            }
            else{
```

```
                msgObj.MsgError("打开失败!");
                return false;
            }
        }
        else{
            msgObj.SetMsgByName("STATUS", "打开成功!");
            msgObj.MsgError("");
        }

        return true;
    }
    /**
     * 保存文档签章记录
     * @param msgObj
     * @return
     * @throws BusinessException
     */
    public boolean saveDocumentSignatureRecord(iMsgServer2000 msgObj,HttpServle
tRequest request) throws BusinessException {
        V3xDocumentSignature ds=new V3xDocumentSignature();
        ds.setIdIfNew();
        ds.setRecordId(msgObj.GetMsgByName("RECORDID"));//取得模板编号
        ds.setMarkname(msgObj.GetMsgByName("MARKNAME"));//设置印章名称
        ds.setUsername(msgObj.GetMsgByName("USERNAME"));//盖章用户名称
        ds.setSignDate(new Timestamp(Datetimes.parseDatetime(msgObj.GetMsgByNam
e("DATETIME")).getTime()));
        ds.setMarkguid(msgObj.GetMsgByName("MARKGUID"));
        ds.setHostname(request.getRemoteAddr());
        try{
          signetManager.save(ds);
        }catch(Exception e)
        {
            throw new BusinessException(e);
        }
        return true;
    }

    /**
     * 向客户端插入图片
     * @param msgObj
     * @return
     * @throws Exception
     */
    public void insertImage(iMsgServer2000 msgObj,HttpServletRequest request)th
rows  Exception{
```

```
        User user = CurrentUser.get();
        Long accountId = user.getLoginAccount();
        String path=SystemProperties.getInstance().getProperty(SystemProperties
.CONFIG_APPLICATION_ROOT_KEY);
        path = Strings.getCanonicalPath(path);
        String url = MainDataLoader.getInstance().getLogoImagePath(accountId);
        msgObj.MsgFileLoad(path+url);
    }
    /**
     * 保存文档，如果文档存在，则覆盖，不存在，则添加
     * 清稿保存时，备份原文件，最多备份5份
     * @return
     * @throws BusinessException
     */
    public boolean saveFile(iMsgServer2000 msgObj) throws Exception {
        if (createDate == null) {
            createDate = new Date();
        }

        if(Strings.isNotBlank(msgObj.GetMsgByName("newPdfFileId")))//如果是Word
转PDF，则需要新生成ID
            fileId=Long.parseLong(msgObj.GetMsgByName("newPdfFileId"));

        if(needClone){//需要clone
            //originalCreateDate空指针导致调用Office格式模板发送报错   Mazc 2009-11-
24
            Date loadCreateDate = originalCreateDate;
            if(loadCreateDate == null){
                String _originalCreateDate = msgObj.GetMsgByName("originalCreat
eDate");
                if(Strings.isNotBlank(_originalCreateDate)){
                    loadCreateDate = com.seeyon.v3x.util.Datetimes.parseDatetim
e(_originalCreateDate);
                }
                else{
                    loadCreateDate = createDate;
                }
            }
            try {
                this.fileManager.clone(originalFileId, loadCreateDate, fileId,
createDate);
            }
            catch (FileNotFoundException e) {
            }
```

```
        }

        String filePath = this.fileManager.getFolder(createDate, true)
            + File.separator + fileId;

            //备份物理文件，防止文件丢失。
            bakPhysicalFile(filePath);


        boolean isSuccessSave = false;

        // 标准office的处理
        String stdOffice = msgObj.GetMsgByName("stdOffice");
        Integer category = new Integer(msgObj.GetMsgByName("CATEGORY"));
        String editType=msgObj.GetMsgByName("editType");
        if("clearDocument".equals(editType))
        {//清稿保存，进行备份
            V3XFile tempFile=null;
            List<V3XFile> fs=fileManager.findByFileName("copy"+fileId.toString(
));
            while(fs!=null && fs.size()>=5)
            {
                tempFile=fs.remove(0);
                fileManager.deleteFile(tempFile.getId(),true);
            }
            try{
                fileManager.clone(fileId);
            }
            catch(FileNotFoundException e){
                //ignore e
            }
            catch(Exception e){
                //直接新建，清稿没有原文件不需要备份
                //throw new BusinessException(e.getMessage());
                log.error("",e);
            }
        }

        String tempFile = SystemEnvironment.getSystemTempFolder() + File.separa
tor + UUIDLong.absLongUUID();
        boolean isDraftTaoHong="draftTaoHong".equals(msgObj.GetMsgByName("draft
TaoHong"));//拟文正文套红
        isSuccessSave = msgObj.MsgFileSave(tempFile);
        if(!isSuccessSave){
            log.error("office正文保存失败(msgObj.MsgFileSave),isSuccessSave:"+isS
```

```
uccessSave+",tempFile:"+tempFile);
        }
        //如果需要转换成标准office正文，加密前先转换
        String notJinge2StandardOffice = msgObj.GetMsgByName("notJinge2Standard
Office");
        if(!"true".equals(notJinge2StandardOffice)){
            boolean toJingge =  Util.jinge2StandardOffice(tempFile, tempFile);
            if(isSuccessSave && !toJingge){
                log.error("office正文转为标准office的时候失败( Util.jinge2StandardO
ffice).isSuccessSave:"+isSuccessSave+",toJingge:"+toJingge);
            }
            isSuccessSave = toJingge;
        }
        CoderFactory.getInstance().encryptFile(tempFile, filePath);
        File f = new File(filePath);
        if(f != null && f.exists())
            msgObj.SetMsgByName("fileSize",f.length()+"");

        if (isSuccessSave) {
            // 先删除以保证能触发OFFICE转换
            officeTransManager.clean(fileId, Datetimes.format(createDate, "yyyy
MMdd"));

            if(stdOffice != null && "true".equals(stdOffice)){
                //updateFileNameTo2003(fileId);
                //文档中心历史版本编辑需要插入数据。
                if(!"true".equals(msgObj.GetMsgByName("needInsertToV3XFile")))
return true;
            }

            V3XFile file = new V3XFile();
            file.setId(fileId);
            file.setCategory(category);
            file.setFilename(fileId.toString());
            file.setSize(new Long(msgObj.MsgFileSize()));
            if("pdf".equalsIgnoreCase(msgObj.GetMsgByName("toFileType"))){
                file.setMimeType("application/pdf");
            }else{
                 String realFileType = msgObj.GetMsgByName("realFileType");
                 String mimeType = "msoffice";
                 if (".docx".equals(realFileType))
                     mimeType ="application/vnd.openxmlformats-officedocument.w
ordprocessingml.document";
                 else if (".doc".equals(realFileType))
                     mimeType="application/msword";
```

```
                else if (".xls".equals(realFileType))
                    mimeType = "application/vnd.ms-excel";
                else if (".xlsx".equals(realFileType))
                    mimeType = "application/vnd.openxmlformats-officedocument.
spreadsheetml.sheet";

                file.setMimeType(mimeType);
            }

            file.setCreateDate(createDate);
            String noMillisecondTime = Datetimes.format(new Date(System.current
TimeMillis()),"yyyy-MM-dd HH:mm:ss");
            file.setUpdateDate(Datetimes.parseDate(noMillisecondTime));

            if(!needClone||isDraftTaoHong){
                this.fileManager.deleteFile(fileId, false);
            }
            User user = CurrentUser.get();
            if(user != null){
                file.setCreateMember(user.getId());
                file.setAccountId(user.getAccountId());
            }
            if(this.fileManager.getV3XFile(file.getId())!=null){
                this.fileManager.update(file);
            }else{
                this.fileManager.save(file);
            }

            //更新锁信息
            UserUpdateObject os = useObjectList.get(String.valueOf(fileId));
            if(os!=null){
                if(os.getUserId() == user.getId()){
                    //只能精确到秒，不能精确到毫秒，因为数据库字段保存不了毫秒的值
                    os.setLastUpdateTime(file.getUpdateDate());
                }
            }
            msgObj.SetMsgByName("STATUS", "保存成功!");
            msgObj.MsgError("");

            // 为了适应是否支持转换的判断
            file.setType(Constants.ATTACHMENT_TYPE.FILE.ordinal());
            if (("msoffice").equals(file.getMimeType()))
                file.setFilename(file.getFilename() + ".doc");

            if (OfficeTransHelper.allowTrans(file)) {
```

```
                officeTransManager.generate(fileId, createDate, true);
            }
            return true;
        }

        msgObj.MsgError("saveFaile!");
        log.error("保存offiec正文,isSuccessSave:"+isSuccessSave);
        return false;
    }
    public String ajaxGetOfficeExtension(String fileId) {
        String extension = "";
        if ((Strings.isNotBlank(fileId)) && (NumberUtils.isNumber(fileId))) {
          try {
            V3XFile file = this.fileManager.getV3XFile(Long.valueOf(fileId));
            String mimeType = "";
            if(file!=null){
                mimeType = file.getMimeType();
                if ("application/vnd.openxmlformats-officedocument.wordprocessi
ngml.document".equals(mimeType))
                    extension = "docx";
                else if ("application/msword".equals(mimeType))
                    extension = "doc";
                else if ("application/vnd.ms-excel".equals(mimeType))
                    extension = "xls";
                else if ("application/vnd.openxmlformats-officedocument.spreads
heetml.sheet".equals(mimeType))
                    extension = "xlsx";
            }
          }
          catch (NumberFormatException e) {
            log.error("", e);
          } catch (BusinessException e) {
            log.error("", e);
          }
        }
        return extension;
      }
//  private void  updateFileNameTo2003(Long fileId){
//      V3XFile file = null;
//      try {
//          file = fileManager.getV3XFile(fileId);
//      } catch (BusinessException e) {
//          log.error(e);
//      }
//      if(file!=null){
```

```
//          String fileName = file.getFilename();
//          if(Strings.isNotBlank(fileName)){
//              String[] suffix = fileName.split("[.]");
//              if(suffix!=null && suffix.length>1){
//                  int len = suffix.length;
//                  if("docx".equalsIgnoreCase(suffix[len-1])){
//                      fileName = suffix[len-2]+".doc";
//                  }else if("xlsx".equalsIgnoreCase(suffix[len-1])){
//                      fileName = suffix[len-2]+".xls";
//                  }else if("pptx".equalsIgnoreCase(suffix[len-1])){
//                      fileName = suffix[len-2]+".ppt";
//                  }
//              }
//          }
//          file.setFilename(fileName);
//          fileManager.update(file);
//      }
//  }
    private void bakPhysicalFile(String filePath) {
        // 公文正文备份
        //命名规则：原文件名_时刻（到秒）.bak
        //存放路径：Office正文原始文件路径下
        //比如原始office正文存放在e:\\ufseeyon\group\base\\upload\2010\01\20下，则备份文件也放到这个路径下，主要方便运维同事查找，存在的问题是不能做增量备份
        //todo:流程结束，备份文件删除
        try {
            String now = Datetimes.format(new Date(), "yyyyMMddHHmmss");
            String contentFileBak=filePath+"_"+now+".bak";
            File f=new File(filePath);
            if(f.exists()){
                FileUtils.copyFile(f, new File(contentFileBak));
            }
        }
        catch (Exception e) {
            log.error("公文正文内容备份异常 : " + fileId, e);
        }
    }
    /**
     * 发送处理后的数据包
     *
     * @param response
     */
    public void sendPackage(HttpServletResponse response, iMsgServer2000 msgObj) {
        /*ServletOutputStream out = null;
```

```
        try {
            out = response.getOutputStream();

            out.write(msgObj.MsgVariant());
            out.flush();
        }
        catch (Exception e) {
        }
        finally {
            if (out != null) {
                try {
                    out.close();
                }
                catch (IOException e) {
                }
            }
        }*/
        msgObj.SendPackage(response);
    }

    /**
     * 从request中读取参数，并写道iMsgServer2000中去
     *
     * @param request
     *            由controller传过来
     * @param iMsgServer2000
     */
    public void readVariant(HttpServletRequest request, iMsgServer2000 msgObj)
{
        /*byte[] bs = null;
        try {
            InputStream in = request.getInputStream();
            if (in != null) {
                bs = org.apache.commons.io.IOUtils.toByteArray(in);
            }
        }
        catch (IOException e1) {
        }

        if (bs != null) {
            msgObj.MsgVariant(bs);
        }
        */
        msgObj.ReadPackage(request);
```

```
        fileId = new Long(msgObj.GetMsgByName("RECORDID"));
        createDate = com.seeyon.v3x.util.Datetimes.parseDatetime(msgObj.GetMsgB
yName("CREATEDATE"));

        String _originalFileId = msgObj.GetMsgByName("originalFileId");
        String _originalCreateDate = msgObj.GetMsgByName("originalCreateDate");
        needClone = _originalFileId != null && !"".equals(_originalFileId.trim(
));

        needReadFile = Boolean.parseBoolean(msgObj.GetMsgByName("needReadFile")
);

        if(needClone){
            originalFileId = new Long(_originalFileId);
            originalCreateDate = com.seeyon.v3x.util.Datetimes.parseDatetime(_o
riginalCreateDate);
        }
    }

    //==============================================避免office正文多人同时修改代码开
始==================================
    //用office的处理文件ID做为key保存的修改记录
    //private static Map<String,UserUpdateObject> useObjectList=new Hashtable<S
tring,UserUpdateObject>();
    private final static CacheAccessable cacheFactory = CacheFactory.getInstanc
e(HandWriteManager.class);
    private static CacheMap<String,UserUpdateObject> useObjectList = cacheFacto
ry.createMap("FlowId");

    public static Map<String, UserUpdateObject> getUseObjectList() {
        return useObjectList.toMap();
    }
    public static void setUseObjectList(Map<String, UserUpdateObject> omap) {
        useObjectList.replaceAll(omap);
    }
    //修改对象,放入对象修改列表
    public synchronized UserUpdateObject editObjectState(String objId)
    {
        if(objId==null || "".equals(objId)){return null;}
        User user=CurrentUser.get();
        UserUpdateObject os=null;
        os=useObjectList.get(objId);
        if(os==null)
        {//无人修改
            os=new UserUpdateObject();
```

```
            try{
                V3XFile file=fileManager.getV3XFile(Long.parseLong(objId));
                if(file!=null)
                {
                    os.setLastUpdateTime(file.getUpdateDate());
                }
                else
                {
                    os.setLastUpdateTime(null);
                    //return os;
                }
                os.setObjId(objId);
                os.setUserId(user.getId());
                os.setUserName(user.getName());
                addUpdateObj(os);
            }catch(Exception e)
            {
            }
        }
        else
        {
            if(os.getUserId()==user.getId())
            {
                os.setCurEditState(false);
            }
            else
            {
                //有用户修改时，要判断用户是否在线,如果用户不在线，删除修改状态
                boolean editUserOnline=true;
                V3xOrgMember member = null; //当前office控件编辑用户
                try{
                    member = orgManager.getEntityById(V3xOrgMember.class, os.ge
tUserId());
                    editUserOnline=onLineManager.isOnline(member.getLoginName()
);
                }
                catch(Exception e1){
                    log.warn("检查文档是否被编辑，文档编辑用户不存在[" + os.getUserId
() + "]", e1);
                }
                if(editUserOnline)
                {
                    os.setCurEditState(true);
                }
                else
```

```
                {
                    //编辑用户已经离线，修改文档编辑人为当前用户
                    os.setUserId(user.getId());
                    os.setCurEditState(false);
                }
            }
        }
        return os;
    }
    //检查对象是否被修改
    public synchronized UserUpdateObject checkObjectState(String objId)
    {
        UserUpdateObject os=null;
        os=useObjectList.get(objId);
        if(os==null){os=new UserUpdateObject();}
        return os;
    }

    public  synchronized boolean deleteUpdateObj(String objId ){
        User user=CurrentUser.get();
        if(user==null) return true;
        long userId = user.getId();
        return deleteUpdateObj(objId, String.valueOf(userId));
    }
    /**
     * 解锁
     * @param objId  解锁对象ID , 如公文正文的ID
     * @param userId 当前用户的ID
     * @return
     */
    public synchronized boolean deleteUpdateObj(String objId,String userId )
    {
        UserUpdateObject os=null;
        os=useObjectList.get(objId);

        if(os==null || Strings.isBlank(userId)){return true;}
        if(Long.valueOf(userId).equals(os.getUserId()))
        {
            useObjectList.remove(objId);
            //发送集群通知
//          String[] so= new String[2];
//          so[0] = objId;
//          so[1] = userId;
//          NotificationManager.getInstance().send(NotificationType.EdocUserOff
iceObjectRomoveOffice, so);
```

```
        }
        return true;
    }
    public synchronized boolean addUpdateObj(UserUpdateObject uo)
    {
        useObjectList.put(uo.getObjId(),uo);
//      发送集群通知
//      NotificationManager.getInstance().send(NotificationType.EdocUserOfficeO
bjectAddOffice, uo);
        return true;
    }
    //==========================================避免office正文多人同时修改代码结
束==================================


    private String findBackFileIds(String fn)
    {
        int ic=0;
        String ids="";
        List<V3XFile> fs=fileManager.findByFileName("copy"+fn);
        for(V3XFile tempFile:fs)
        {
            if(ic!=0){ids+=",";}
            ic++;
            ids+=tempFile.getId();
        }
        return ids;
    }

    public boolean taoHong(iMsgServer2000 msgObj) throws BusinessException {
        String path=msgObj.GetMsgByName("TEMPLATE");        //取得文档编号
        //本段处理是否调用文档时打开模版,
        //还是套用模版时打开模版。
        String mCommand=msgObj.GetMsgByName("COMMAND");
        //String templetType=msgObj.GetMsgByName("TEMPLATETYPE");
        if(mCommand.equalsIgnoreCase("INSERTFILE"))
        {
            msgObj.MsgTextClear();
            if(msgObj.MsgFileLoad(path))
            {
                msgObj.SetMsgByName("STATUS","打开模板成功!");        //设置状态信
息
                msgObj.MsgError("");
            }
            else
```

```
            {
                msgObj.MsgError("打开模板失败!");      //设置错误信息
            }
        }
        return true;
    }

    public boolean saveClientFile(iMsgServer2000 msgObj) throws Exception {
        if (createDate == null) {
            createDate = new Date();
        }

        Long fileName = UUIDLong.absLongUUID();
        String filePath = this.fileManager.getFolder(createDate, true)
        + File.separator + fileName;

        boolean isSuccessSave = false;

        Integer category = new Integer(msgObj.GetMsgByName("CATEGORY"));

        String tempFile = SystemEnvironment.getSystemTempFolder() + File.separa
tor + UUIDLong.absLongUUID();
        isSuccessSave = msgObj.MsgFileSave(tempFile);
        if(!isSuccessSave){
            log.error("office上传文件保存失败(msgObj.MsgFileSave),isSuccessSave:"+
isSuccessSave+",tempFile:"+tempFile);
        }

        CoderFactory.getInstance().encryptFile(tempFile, filePath);
        File f = new File(filePath);
        if(f != null && f.exists())
            msgObj.SetMsgByName("fileSize",f.length()+"");

        String ext = msgObj.GetMsgByName("fileExt");
        if (isSuccessSave) {
            V3XFile file = new V3XFile();
            file.setId(fileName);
            file.setCategory(category);
            file.setFilename(fileName.toString());
            file.setSize(new Long(msgObj.MsgFileSize()));

            file.setCreateDate(createDate);
            String noMillisecondTime = Datetimes.format(new Date(System.current
TimeMillis()),"yyyy-MM-dd HH:mm:ss");
            file.setUpdateDate(Datetimes.parseDate(noMillisecondTime));
```

```
            User user = CurrentUser.get();
            if(user != null){
                file.setCreateMember(user.getId());
                file.setAccountId(user.getAccountId());
            }

            this.fileManager.save(file);
        }

        //保存二维码文件信息到bar_code_info
        String saveCodeFile = msgObj.GetMsgByName("saveCodeFile");
        String objectId = msgObj.GetMsgByName("objectId");
        if("true".equals(saveCodeFile) && Strings.isNotBlank(objectId)) {
            this.barCodeManager.saveBarCode(Long.parseLong(objectId), fileName,
ext, category);
        }

        msgObj.SetMsgByName("STATUS", "保存成功!");
        msgObj.MsgError("");
        return true;
    }
}
```

HtmlHandWriteManager.java

```
package com.seeyon.v3x.common.office;

import DBstep.iMsgServer2000;

import com.seeyon.v3x.organization.domain.V3xOrgMember;
import com.seeyon.v3x.organization.manager.OnLineManager;
import com.seeyon.v3x.organization.manager.OrgManager;
import com.seeyon.v3x.system.signet.dao.*;
import com.seeyon.v3x.system.signet.domain.*;
import com.seeyon.v3x.util.Datetimes;
import java.sql.Timestamp;

import java.util.*;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import com.seeyon.v3x.cluster.notification.NotificationManager;
import com.seeyon.v3x.cluster.notification.NotificationType;
```

```
import com.seeyon.v3x.common.authenticate.domain.User;
import com.seeyon.v3x.common.cache.CacheAccessable;
import com.seeyon.v3x.common.cache.CacheFactory;
import com.seeyon.v3x.common.cache.CacheMap;
import com.seeyon.v3x.common.exceptions.BusinessException;
import com.seeyon.v3x.common.filemanager.V3XFile;
import com.seeyon.v3x.common.i18n.ResourceBundleUtil;
import com.seeyon.v3x.common.web.login.CurrentUser;
import com.seeyon.v3x.common.web.util.ApplicationContextHolder;

public class HtmlHandWriteManager {

    private static String rc="com.seeyon.v3x.common.resources.i18n.SeeyonCommon
Resources";

    //private static Map<String,UserUpdateObject> useObjectList=new Hashtable<S
tring,UserUpdateObject>();
    private final static CacheAccessable cacheFactory = CacheFactory.getInstanc
e(HtmlHandWriteManager.class);
    private static CacheMap<String,UserUpdateObject> useObjectList = cacheFacto
ry.createMap("Flow");

    V3xHtmDocumentSignatureDao htmlSignDao;
    HtmlSignatureHistoryDao signHistoryDao;

    private static Log log = LogFactory.getLog(HtmlHandWriteManager.class);
    private static OnLineManager onLineManager;
    private static OrgManager orgManager;


    private synchronized void init() {
        if(onLineManager == null){
            orgManager = (OrgManager) ApplicationContextHolder.getBean("OrgMana
ger");
            onLineManager = (OnLineManager)ApplicationContextHolder.getBean("on
LineManager");
        }
    }

    public HtmlHandWriteManager()
    {
        init();
    }

    public V3xHtmDocumentSignatureDao getHtmlSignDao()
```

```
    {
        return this.htmlSignDao;
    }

    public HtmlSignatureHistoryDao getSignHistoryDao()
    {
        return this.signHistoryDao;
    }

    public void setSignHistoryDao(HtmlSignatureHistoryDao signHistoryDao)
    {
        this.signHistoryDao=signHistoryDao;
    }

    public void setHtmlSignDao(V3xHtmDocumentSignatureDao htmlSignDao)
    {
        this.htmlSignDao=htmlSignDao;
    }

    public boolean loadDocumentSinature(iMsgServer2000 msgObj) throws BusinessE
xception
    {

        List <V3xHtmDocumentSignature> dsList=null;
        V3xHtmDocumentSignature ds=new V3xHtmDocumentSignature();
        ds.setSummaryId(Long.parseLong(msgObj.GetMsgByName("RECORDID")));//取得
文档编号
        ds.setFieldName(msgObj.GetMsgByName("FIELDNAME"));//取得签章字段名称
        ds.setUserName(msgObj.GetMsgByName("USERNAME"));//取得用户名称
        msgObj.MsgTextClear();                                    //清除SetMsgByName
设置的值
        dsList=htmlSignDao.findByIdAndPolicy(ds.getSummaryId(),ds.getFieldName(
));
        if(dsList!=null && dsList.size()>0)
        {
            ds=dsList.get(0);
            msgObj.SetMsgByName("FIELDVALUE",ds.getFieldValue());    //设置签章数
据
            msgObj.SetMsgByName("STATUS","调入成功!");       //设置状态信息
            msgObj.MsgError("");                  //清除错误信息
        }
        else
        {
            msgObj.MsgError("load err!");                //设置错误信息
        }
```

```
        return true;
    }

    public boolean saveSignatureHistory(iMsgServer2000 msgObj) throws BusinessE
xception {
        V3xHtmlSignatureHistory sh=new V3xHtmlSignatureHistory();
        sh.setIdIfNew();
        sh.setSummaryId(Long.parseLong(msgObj.GetMsgByName("RECORDID")));//取得
文档编号
        sh.setFieldName(msgObj.GetMsgByName("FIELDNAME"));//取得签章字段名称
        sh.setMarkName(msgObj.GetMsgByName("MARKNAME"));//取得签章名称
        sh.setUserName(msgObj.GetMsgByName("USERNAME"));         //取得用户名称
        sh.setDateTime(new Timestamp(System.currentTimeMillis()));       //取得签
章日期时间
        sh.setHostName(msgObj.GetMsgByName("CLIENTIP"));                 //取得客
户端IP
        sh.setMarkGuid(msgObj.GetMsgByName("MARKGUID"));              //取得序列号
        msgObj.MsgTextClear();                                  //清除SetMsgByName
设置的值

        try{
          signHistoryDao.save(sh); //保存印章历史信息
        }catch(Exception e)
        {
            msgObj.MsgError("saveerr!");         //设置错误信息
            return false;
        }

        msgObj.SetMsgByName("MARKNAME",sh.getMarkName());        //将签章名称列表打
包
        msgObj.SetMsgByName("USERNAME",sh.getUserName());        //将用户名列表打包
        msgObj.SetMsgByName("DATETIME",Datetimes.formatDatetime(sh.getDateTime(
)));      //将签章日期列表打包
        msgObj.SetMsgByName("HOSTNAME",sh.getHostName());        //将客户端IP列表
打包
        msgObj.SetMsgByName("MARKGUID",sh.getMarkGuid());        //将序列号列表打包
        msgObj.SetMsgByName("STATUS","save ok!");  //设置状态信息
        msgObj.MsgError("");                    //清除错误信息
        return true;
    }

    public boolean getSignatureHistory(iMsgServer2000 msgObj) throws BusinessEx
ception {

        V3xHtmlSignatureHistory dh=new V3xHtmlSignatureHistory();
```

```
        dh.setSummaryId(Long.parseLong(msgObj.GetMsgByName("RECORDID")));
 //取得文档编号
        dh.setFieldName(msgObj.GetMsgByName("FIELDNAME"));       //取得签章字段名称
        dh.setUserName(msgObj.GetMsgByName("USERNAME"));         //取得用户名
        msgObj.MsgTextClear();                              //清除SetMsgByName设置的值

        dh=combStr(signHistoryDao.findByIdAndPolicy(dh.getSummaryId(),dh.getFie
ldName()));

        if (dh!=null)                                  //调入印章历史信息
        {
          msgObj.SetMsgByName("MARKNAME",dh.getMarkName());      //将签章名称列表打
包
          msgObj.SetMsgByName("USERNAME",dh.getUserName());      //将用户名列表打包
          msgObj.SetMsgByName("DATETIME",dh.getDateTimeStr());       //将签章日期
列表打包
          msgObj.SetMsgByName("HOSTNAME",dh.getHostName());      //将客户端IP列表
打包
          msgObj.SetMsgByName("MARKGUID",dh.getMarkGuid());      //将序列号列表打包
          msgObj.SetMsgByName("STATUS","load ok");   //设置状态信息
          msgObj.MsgError("");                 //清除错误信息
        }else{
            msgObj.SetMsgByName("STATUS","load false"); //设置状态信息
            msgObj.MsgError("load fale");        //设置错误信息
        }
        return true;
    }

      /**
       * 查询到的签章记录转变成控件要求格式
       * @param ls
       * @return
       */
      private V3xHtmlSignatureHistory combStr(List <V3xHtmlSignatureHistory>ls)
      {
        V3xHtmlSignatureHistory temp,dh=new V3xHtmlSignatureHistory();
        dh.setMarkName(ResourceBundleUtil.getString(rc,"ocx.signname.label")+"\
r\n");
        dh.setUserName(ResourceBundleUtil.getString(rc,"ocx.signuser.label")+"\
r\n");
        dh.setHostName(ResourceBundleUtil.getString(rc,"ocx.clientip.label")+"\
r\n");
        dh.setDateTimeStr(ResourceBundleUtil.getString(rc,"ocx.signtime.label")
+"\r\n");
```

```
        dh.setMarkGuid(ResourceBundleUtil.getString(rc,"ocx.serialnumber.label"
)+"\r\n");
        int i,len=ls.size();

        for(i=0;i<len;i++)
        {
          temp=ls.get(i);
          dh.setMarkName(dh.getMarkName()+temp.getMarkName()+"\r\n");
          dh.setUserName(dh.getUserName()+temp.getUserName()+"\r\n");
          dh.setHostName(dh.getHostName()+temp.getHostName()+"\r\n");
          dh.setDateTimeStr(dh.getDateTimeStr()+Datetimes.formatDatetime(temp.g
etDateTime())+"\r\n");
          dh.setMarkGuid(dh.getMarkGuid()+temp.getMarkGuid()+"\r\n");
        }
        return dh;
      }

    public boolean saveSignature(iMsgServer2000 msgObj) throws BusinessExceptio
n {
        V3xHtmDocumentSignature hd=new V3xHtmDocumentSignature();
        String clientVer=msgObj.GetMsgByName("Version");
        clientVer=clientVer.replace('.',',');
        if(msgObj.Version().equals(clientVer))
        {
          msgObj.MsgError("ver err");
          msgObj.MsgTextClear();
          msgObj.MsgFileClear();
          return false;
        }
        boolean isUpdate=false;
        Long summaryId=Long.parseLong(msgObj.GetMsgByName("RECORDID"));
        String policy=msgObj.GetMsgByName("FIELDNAME");
        List <V3xHtmDocumentSignature> hsList=htmlSignDao.findByIdAndPolicy(sum
maryId, policy);
        if(hsList!=null && hsList.size()>0)
        {
            hd=hsList.get(0);
            isUpdate=true;
        }
        hd.setIdIfNew();
        hd.setSummaryId(summaryId);//取得文档编号
        hd.setFieldName(policy);//取得签章字段名称
        hd.setFieldValue(msgObj.GetMsgByName("FIELDVALUE"));//取得签章数据内容
        hd.setUserName(msgObj.GetMsgByName("USERNAME"));//取得用户名称
        hd.setDateTime(new Timestamp(System.currentTimeMillis()));//取得签章日期
```

```
时间
        hd.setHostName(msgObj.GetMsgByName("CLIENTIP"));//取得客户端IP
        try{
            if(isUpdate){htmlSignDao.update(hd);}
            else{htmlSignDao.save(hd);}
        }catch(Exception e)
        {
            log.error(e.getMessage(), e);
        }
        return true;
    }
    //读取文单签章，转换成js
    public String getHandWritesJs(Long summaryId,String userName,List<String>op
inionNames)
    {
        StringBuffer sb = new StringBuffer("<Script language='JavaScript'>");
        int i, len;
        List <V3xHtmDocumentSignature>ls = htmlSignDao.findBy("summaryId",summa
ryId);
        V3xHtmDocumentSignature ds = null;
        len = ls.size();

        sb.append("hwObjs=new Array();\r\n");
        for (i = 0; i < len; i++)
        {
            ds = ls.get(i);
            sb.append("hwObjs[").append(i).append("]=new hwObj('").append(summa
ryId).append("','").append(ds.getFieldName()).append("','").append(userName).ap
pend("','").append(ds.getDateTime().getTime()).append("');\r\n");
        }
        sb.append("</Script>\r\n");
        //添加控件菜单响应时间
        /*
        for (i = 0; i < len; i++)
        {
            ds = ls.get(i);
            sb.append(getHandWriteEventJs(ds.getFieldName()));
        }
        */
        if(opinionNames.contains("otherOpinion")==false){opinionNames.add("othe
rOpinion");}
        for (i = 0; i < opinionNames.size(); i++)
        {
            sb.append(getHandWriteEventJs("hw"+opinionNames.get(i)));
        }
```

```
        return sb.toString();
    }

    /**
     * 根据 summmaryId 得到回复该公文签章
     * @param summaryId
     * @return
     */
    public List <V3xHtmDocumentSignature> getHandWrites(Long summaryId){
        if(summaryId!=null){
            return htmlSignDao.findBy("summaryId",summaryId);
        }else{
            return null;
        }
    }
    private String getHandWriteEventJs(String hwName)
    {
        StringBuffer hjen = new StringBuffer();
        hjen.append("<SCRIPT language=javascript for='").append(hwName).append(
            "' event=OnMenuClick(vIndex,vCaption)>\r\n");
        hjen.append("OnMenuHdClick(this,vIndex,vCaption);\r\n");
        hjen.append("</SCRIPT>\r\n");
        return hjen.toString();
    }

    //修改对象,放入对象修改列表
    public synchronized UserUpdateObject editObjectState(String objId)
    {
        if(objId==null || "".equals(objId)){return null;}
        User user=CurrentUser.get();
        UserUpdateObject os=null;
        os=useObjectList.get(objId);
        if(os==null)
        {//无人修改
            os=new UserUpdateObject();
            try{
                String []temp=objId.split("___");
                List <V3xHtmDocumentSignature> dsList=htmlSignDao.findByIdAndPo
licy(Long.parseLong(temp[0]),temp[1]);
                if(dsList!=null && dsList.size()>0)
                {
                    os.setLastUpdateTime(dsList.get(0).getDateTime());
                }
                else
                {
```

```
                    os.setLastUpdateTime(null);
                    os.setCurEditState(false);
                }
                os.setObjId(objId);
                os.setUserId(user.getId());
                os.setUserName(user.getName());
                addUpdateObj(os);
            }catch(Exception e)
            {
            }
        }
        else
        {
//          有用户修改时，要判断用户是否在线,如果用户不在线，删除修改状态
            boolean editUserOnline=true;
            V3xOrgMember member = null; //当前office控件编辑用户
            try{
                member = orgManager.getEntityById(V3xOrgMember.class, os.getUse
rId());
                editUserOnline=onLineManager.isOnline(member.getLoginName());
            }
            catch(Exception e1){
                log.warn("检查文档是否被编辑，文档编辑用户不存在[" + os.getUserId() +
 "]", e1);
            }
            if(editUserOnline && os.getUserId()!=user.getId())
            {
                os.setCurEditState(true);
            }
            else
            {
                //编辑用户已经离线，修改文档编辑人为当前用户
                os.setUserId(user.getId());
                os.setCurEditState(false);
            }
        }
        return os;
    }
    //检查对象是否被修改
    public synchronized UserUpdateObject checkObjectState(String objId)
    {
        UserUpdateObject os=null;
        os=useObjectList.get(objId);
        if(os==null){os=new UserUpdateObject();}
        return os;
```

```
    }
    public synchronized boolean deleteUpdateObj(String objId,Long userId){

        UserUpdateObject os=null;
        os=useObjectList.get(objId);
        if(os==null || userId==null){return true;}
        if(userId.equals(os.getUserId()))
        {
            useObjectList.remove(objId);
            //发送集群通知
            //NotificationManager.getInstance().send(NotificationType.EdocUserO
fficeObjectRomoveHtml, new Object[]{objId,userId});
        }
        return true;
    }
    public synchronized boolean deleteUpdateObj(String objId)
    {
        User user=CurrentUser.get();
        if(user == null) return true;
        Long userId = user.getId();
        return deleteUpdateObj(objId,userId);
    }
    public synchronized boolean addUpdateObj(UserUpdateObject uo)
    {
        useObjectList.put(uo.getObjId(),uo);
//      发送集群通知
        //NotificationManager.getInstance().send(NotificationType.EdocUserOffic
eObjectAddHtml, uo);
        return true;
    }

    public static Map<String, UserUpdateObject> getUseObjectList() {
        return useObjectList.toMap();
    }

    public static void setUseObjectList(Map<String, UserUpdateObject> uo) {
        useObjectList.replaceAll(uo);
    }

}
```

通过跟踪 **HtmlOfficeServlet.java**, 发现些关键操作

HtmlOfficeServlet.java

```
            handWriteManager.readVariant(request, msgObj);

            msgObj.SetMsgByName("CLIENTIP", request.getRemoteAddr());

            String option = msgObj.GetMsgByName("OPTION");

            if ("LOADFILE".equalsIgnoreCase(option)) {
                handWriteManager.LoadFile(msgObj);
            }
            else if("LOADSIGNATURE".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.loadDocumentSinature(msgObj);
            }
            else if("LOADMARKLIST".equalsIgnoreCase(option))
            {
                handWriteManager.LoadSinatureList(msgObj);
            }
            else if("SIGNATRUEIMAGE".equalsIgnoreCase(option))
            {
                handWriteManager.LoadSinature(msgObj);
            }
            else if("SAVESIGNATURE".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.saveSignature(msgObj);
            }
            else if("SAVEHISTORY".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.saveSignatureHistory(msgObj);
            }
            else if("SIGNATRUELIST".equalsIgnoreCase(option))
            {//调入印章列表
                handWriteManager.LoadSinatureList(msgObj);
            }
            else if("SHOWHISTORY".equalsIgnoreCase(option))
            {
                htmlHandWriteManager.getSignatureHistory(msgObj);
            }
```

跟进发现，在 `htmlHandWriteManager.saveSignatureHistory(msgObj)` 时会获取请求传递过来的FILENAME

saveSignatureHistory函数

```
public boolean saveSignatureHistory(iMsgServer2000 msgObj) throws BusinessExcep
tion {
        V3xHtmlSignatureHistory sh=new V3xHtmlSignatureHistory();
        sh.setIdIfNew();
        sh.setSummaryId(Long.parseLong(msgObj.GetMsgByName("RECORDID")));//取得
```

```
文档编号
        sh.setFieldName(msgObj.GetMsgByName("FIELDNAME"));//取得签章字段名称
        sh.setMarkName(msgObj.GetMsgByName("MARKNAME"));//取得签章名称
        sh.setUserName(msgObj.GetMsgByName("USERNAME"));         //取得用户名称
        sh.setDateTime(new Timestamp(System.currentTimeMillis()));       //取得签
章日期时间
        sh.setHostName(msgObj.GetMsgByName("CLIENTIP"));                 //取得客
户端IP
        sh.setMarkGuid(msgObj.GetMsgByName("MARKGUID"));             //取得序列号
        msgObj.MsgTextClear();                                  //清除SetMsgByName
设置的值

        try{
          signHistoryDao.save(sh); //保存印章历史信息
        }catch(Exception e)
        {
            msgObj.MsgError("saveerr!");         //设置错误信息
            return false;
        }

        msgObj.SetMsgByName("MARKNAME",sh.getMarkName());        //将签章名称列表打
包
        msgObj.SetMsgByName("USERNAME",sh.getUserName());        //将用户名列表打包
        msgObj.SetMsgByName("DATETIME",Datetimes.formatDatetime(sh.getDateTime(
)));      //将签章日期列表打包
        msgObj.SetMsgByName("HOSTNAME",sh.getHostName());        //将客户端IP列表
打包
        msgObj.SetMsgByName("MARKGUID",sh.getMarkGuid());        //将序列号列表打包
        msgObj.SetMsgByName("STATUS","save ok!");  //设置状态信息
        msgObj.MsgError("");                    //清除错误信息
        return true;
    }
```

到这里，发现关键点是 **iMsgServer2000** 类，但是发现旧的项目中并没有这个类，我们通过 GitHub 搜索，发现了它

https://github.com/ExllntSuppt/ecology-OA/blob/cb45b44c7c1bf81b0af1cc7a2c21818045b94910/iMsgServer2000.java

iMsgServer2000.java

```
/*
 *
 *
 *
```

```
 *  javax.servlet.ServletInputStream
 *  javax.servlet.ServletOutputStream
 *  javax.servlet.http.HttpServletRequest
 *  javax.servlet.http.HttpServletResponse
 */
package DBstep;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.PrintStream;
import java.io.RandomAccessFile;
import java.io.UnsupportedEncodingException;
import javax.servlet.ServletInputStream;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class iMsgServer2000 {
    private String _$901 = "DBSTEP V3.0";
    private String _$902 = "8.5.0.0";
    private String _$903 = "FxcYg3UZvtEz50Na8G476=mLDI/jVfC9dsoMAiBhJSu2qPKe+QR
bXry1TnkWHlOpw";
    private byte[] _$904;
    private byte[] _$905;
    private String _$906 = "";
    private String _$907 = "";
    private String _$908 = "DBSTEP V3.0";
    private int _$909 = 0;
    private boolean _$910 = false;
    public String Charset = "GB2312";
    public String Md5Value = "";
    public boolean FDebug = false;
    static final int S11 = 7;
    static final int S12 = 12;
    static final int S13 = 17;
    static final int S14 = 22;
    static final int S21 = 5;
    static final int S22 = 9;
    static final int S23 = 14;
    static final int S24 = 20;
    static final int S31 = 4;
    static final int S32 = 11;
```

```
    static final int S33 = 16;
    static final int S34 = 23;
    static final int S41 = 6;
    static final int S42 = 10;
    static final int S43 = 15;
    static final int S44 = 21;
    static final byte[] PADDING = new byte[]{-128, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0};
    private long[] _$931 = new long[4];
    private long[] _$932 = new long[2];
    private byte[] _$933 = new byte[64];
    private String _$934;
    private byte[] _$935 = new byte[16];

    private String _$936(String inbuf) {
        this._$938();
        this._$939(inbuf.getBytes(), inbuf.length());
        this._$941();
        this._$934 = "";
        for (int i = 0; i < 16; ++i) {
            this._$934 = this._$934 + iMsgServer2000._$943(this._$935[i]);
        }
        return this._$934;
    }

    private void _$938() {
        this._$932[0] = 0;
        this._$932[1] = 0;
        this._$931[0] = 1732584193;
        this._$931[1] = 4023233417L;
        this._$931[2] = 2562383102L;
        this._$931[3] = 271733878;
    }

    private long _$944(long x, long y, long z) {
        return x & y | (x ^ -1) & z;
    }

    private long _$948(long x, long y, long z) {
        return x & z | y & (z ^ -1);
    }

    private long _$949(long x, long y, long z) {
```

```
        return x ^ y ^ z;
    }

    private long _$950(long x, long y, long z) {
        return y ^ (x | z ^ -1);
    }

    private long _$951(long a, long b, long c, long d, long x, long s, long ac)
{
        a += this._$944(b, c, d) + x + ac;
        a = (int)a << (int)s | (int)a >>> (int)(32 - s);
        return a += b;
    }

    private long _$958(long a, long b, long c, long d, long x, long s, long ac)
{
        a += this._$948(b, c, d) + x + ac;
        a = (int)a << (int)s | (int)a >>> (int)(32 - s);
        return a += b;
    }

    private long _$959(long a, long b, long c, long d, long x, long s, long ac)
{
        a += this._$949(b, c, d) + x + ac;
        a = (int)a << (int)s | (int)a >>> (int)(32 - s);
        return a += b;
    }

    private long _$960(long a, long b, long c, long d, long x, long s, long ac)
{
        a += this._$950(b, c, d) + x + ac;
        a = (int)a << (int)s | (int)a >>> (int)(32 - s);
        return a += b;
    }

    private void _$939(byte[] inbuf, int inputLen) {
        int i;
        byte[] block = new byte[64];
        int index = (int)(this._$932[0] >>> 3) & 63;
        this._$932[0] = this._$932[0] + (long)(inputLen << 3);
        if (this._$932[0] < (long)(inputLen << 3)) {
            long[] arrl = this._$932;
            arrl[1] = arrl[1] + 1;
        }
        long[] arrl = this._$932;
```

```
        arrl[1] = arrl[1] + (long)(inputLen >>> 29);
        int partLen = 64 - index;
        if (inputLen >= partLen) {
            this._$965(this._$933, inbuf, index, 0, partLen);
            this._$966(this._$933);
            i = partLen;
            while (i + 63 < inputLen) {
                this._$965(block, inbuf, 0, i, 64);
                this._$966(block);
                i += 64;
            }
            index = 0;
        } else {
            i = 0;
        }
        this._$965(this._$933, inbuf, index, i, inputLen - i);
    }

    private void _$941() {
        byte[] bits = new byte[8];
        this._$969(bits, this._$932, 8);
        int index = (int)(this._$932[0] >>> 3) & 63;
        int padLen = index < 56 ? 56 - index : 120 - index;
        this._$939(PADDING, padLen);
        this._$939(bits, 8);
        this._$969(this._$935, this._$931, 16);
    }

    private void _$965(byte[] output, byte[] input, int outpos, int inpos, int
len) {
        for (int i = 0; i < len; ++i) {
            output[outpos + i] = input[inpos + i];
        }
    }

    private void _$966(byte[] block) {
        long a = this._$931[0];
        long b = this._$931[1];
        long c = this._$931[2];
        long d = this._$931[3];
        long[] x = new long[16];
        this._$975(x, block, 64);
        a = this._$951(a, b, c, d, x[0], 7, 3614090360L);
        d = this._$951(d, a, b, c, x[1], 12, 3905402710L);
        c = this._$951(c, d, a, b, x[2], 17, 606105819);
```

```
        b = this._$951(b, c, d, a, x[3], 22, 3250441966L);
        a = this._$951(a, b, c, d, x[4], 7, 4118548399L);
        d = this._$951(d, a, b, c, x[5], 12, 1200080426);
        c = this._$951(c, d, a, b, x[6], 17, 2821735955L);
        b = this._$951(b, c, d, a, x[7], 22, 4249261313L);
        a = this._$951(a, b, c, d, x[8], 7, 1770035416);
        d = this._$951(d, a, b, c, x[9], 12, 2336552879L);
        c = this._$951(c, d, a, b, x[10], 17, 4294925233L);
        b = this._$951(b, c, d, a, x[11], 22, 2304563134L);
        a = this._$951(a, b, c, d, x[12], 7, 1804603682);
        d = this._$951(d, a, b, c, x[13], 12, 4254626195L);
        c = this._$951(c, d, a, b, x[14], 17, 2792965006L);
        b = this._$951(b, c, d, a, x[15], 22, 1236535329);
        a = this._$958(a, b, c, d, x[1], 5, 4129170786L);
        d = this._$958(d, a, b, c, x[6], 9, 3225465664L);
        c = this._$958(c, d, a, b, x[11], 14, 643717713);
        b = this._$958(b, c, d, a, x[0], 20, 3921069994L);
        a = this._$958(a, b, c, d, x[5], 5, 3593408605L);
        d = this._$958(d, a, b, c, x[10], 9, 38016083);
        c = this._$958(c, d, a, b, x[15], 14, 3634488961L);
        b = this._$958(b, c, d, a, x[4], 20, 3889429448L);
        a = this._$958(a, b, c, d, x[9], 5, 568446438);
        d = this._$958(d, a, b, c, x[14], 9, 3275163606L);
        c = this._$958(c, d, a, b, x[3], 14, 4107603335L);
        b = this._$958(b, c, d, a, x[8], 20, 1163531501);
        a = this._$958(a, b, c, d, x[13], 5, 2850285829L);
        d = this._$958(d, a, b, c, x[2], 9, 4243563512L);
        c = this._$958(c, d, a, b, x[7], 14, 1735328473);
        b = this._$958(b, c, d, a, x[12], 20, 2368359562L);
        a = this._$959(a, b, c, d, x[5], 4, 4294588738L);
        d = this._$959(d, a, b, c, x[8], 11, 2272392833L);
        c = this._$959(c, d, a, b, x[11], 16, 1839030562);
        b = this._$959(b, c, d, a, x[14], 23, 4259657740L);
        a = this._$959(a, b, c, d, x[1], 4, 2763975236L);
        d = this._$959(d, a, b, c, x[4], 11, 1272893353);
        c = this._$959(c, d, a, b, x[7], 16, 4139469664L);
        b = this._$959(b, c, d, a, x[10], 23, 3200236656L);
        a = this._$959(a, b, c, d, x[13], 4, 681279174);
        d = this._$959(d, a, b, c, x[0], 11, 3936430074L);
        c = this._$959(c, d, a, b, x[3], 16, 3572445317L);
        b = this._$959(b, c, d, a, x[6], 23, 76029189);
        a = this._$959(a, b, c, d, x[9], 4, 3654602809L);
        d = this._$959(d, a, b, c, x[12], 11, 3873151461L);
        c = this._$959(c, d, a, b, x[15], 16, 530742520);
        b = this._$959(b, c, d, a, x[2], 23, 3299628645L);
```

```
        a = this._$960(a, b, c, d, x[0], 6, 4096336452L);
        d = this._$960(d, a, b, c, x[7], 10, 1126891415);
        c = this._$960(c, d, a, b, x[14], 15, 2878612391L);
        b = this._$960(b, c, d, a, x[5], 21, 4237533241L);
        a = this._$960(a, b, c, d, x[12], 6, 1700485571);
        d = this._$960(d, a, b, c, x[3], 10, 2399980690L);
        c = this._$960(c, d, a, b, x[10], 15, 4293915773L);
        b = this._$960(b, c, d, a, x[1], 21, 2240044497L);
        a = this._$960(a, b, c, d, x[8], 6, 1873313359);
        d = this._$960(d, a, b, c, x[15], 10, 4264355552L);
        c = this._$960(c, d, a, b, x[6], 15, 2734768916L);
        b = this._$960(b, c, d, a, x[13], 21, 1309151649);
        a = this._$960(a, b, c, d, x[4], 6, 4149444226L);
        d = this._$960(d, a, b, c, x[11], 10, 3174756917L);
        c = this._$960(c, d, a, b, x[2], 15, 718787259);
        b = this._$960(b, c, d, a, x[9], 21, 3951481745L);
        long[] arrl = this._$931;
        arrl[0] = arrl[0] + a;
        long[] arrl2 = this._$931;
        arrl2[1] = arrl2[1] + b;
        long[] arrl3 = this._$931;
        arrl3[2] = arrl3[2] + c;
        long[] arrl4 = this._$931;
        arrl4[3] = arrl4[3] + d;
    }

    private void _$969(byte[] output, long[] input, int len) {
        int i = 0;
        for (int j = 0; j < len; j += 4) {
            output[j] = (byte)(input[i] & 255);
            output[j + 1] = (byte)(input[i] >>> 8 & 255);
            output[j + 2] = (byte)(input[i] >>> 16 & 255);
            output[j + 3] = (byte)(input[i] >>> 24 & 255);
            ++i;
        }
    }

    private void _$975(long[] output, byte[] input, int len) {
        int i = 0;
        for (int j = 0; j < len; j += 4) {
            output[i] = iMsgServer2000._$977(input[j]) | iMsgServer2000._$977(i
nput[j + 1]) << 8 | iMsgServer2000._$977(input[j + 2]) << 16 | iMsgServer2000._
$977(input[j + 3]) << 24;
            ++i;
        }
```

```
    }

    private static long _$977(byte b) {
        return b < 0 ? (long)(b & 255) : (long)b;
    }

    private static String _$943(byte ib) {
        char[] Digit = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8',
'9', 'a', 'b', 'c', 'd', 'e', 'f'};
        char[] ob = new char[]{Digit[ib >>> 4 & 15], Digit[ib & 15]};
        String s = new String(ob);
        return s;
    }

    public String MD5Stream(byte[] Value) {
        this._$938();
        this._$939(Value, Value.length);
        this._$941();
        this._$934 = "";
        for (int i = 0; i < 16; ++i) {
            this._$934 = this._$934 + iMsgServer2000._$943(this._$935[i]);
        }
        return this._$934;
    }

    public iMsgServer2000() {
        if (this.FDebug) {
            System.out.println("Run iMsgServer2000 Version " + this._$902 + ".."
.");
        }
    }

    protected String FormatHead(String vString) {
        if (vString.length() > 16) {
            return vString.substring(0, 16);
        }
        for (int i = vString.length() + 1; i < 17; ++i) {
            vString = vString.concat(" ");
        }
        return vString;
    }

    private boolean _$988() {
        int HeadSize = 64;
        int BodySize = 0;
```

```
        int ErrorSize = 0;
        int FileSize = 0;
        int Position = 0;
        try {
            Position = 0;
            BodySize = this._$906.getBytes(this.Charset).length;
            ErrorSize = this._$907.getBytes(this.Charset).length;
            FileSize = this._$909;
            ByteArrayOutputStream mBuffer = new ByteArrayOutputStream(HeadSize
+ BodySize + ErrorSize + FileSize);
            String HeadString = this.FormatHead(this._$908) + this.FormatHead(S
tring.valueOf(BodySize)) + this.FormatHead(String.valueOf(ErrorSize)) + this.Fo
rmatHead(String.valueOf(FileSize));
            mBuffer.write(HeadString.getBytes(), Position, HeadSize);
            Position += HeadSize;
            if (BodySize > 0) {
                mBuffer.write(this._$906.getBytes());
            }
            Position += BodySize;
            if (ErrorSize > 0) {
                mBuffer.write(this._$907.getBytes(this.Charset));
            }
            Position += ErrorSize;
            if (FileSize > 0) {
                mBuffer.write(this._$905);
            }
            Position += FileSize;
            mBuffer.close();
            this._$904 = mBuffer.toByteArray();
            return true;
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            System.out.println(e.toString());
            return false;
        }
    }

    public byte[] MsgVariant() {
        this._$988();
        return this._$904;
    }

    private static int _$1001(byte[] b) {
        int s = 0;
```

```
        for (int i = 3; i > 0; --i) {
            s = b[i] >= 0 ? (s += b[i]) : s + 256 + b[i];
            s *= 256;
        }
        s = b[0] >= 0 ? (s += b[0]) : s + 256 + b[0];
        return s;
    }

    public byte[] ToDocument(byte[] Value) {
        byte[] mIntBuf = new byte[]{0, 0, 0, 0};
        byte[] mFlagBuf = new byte[]{68, 73, 82, 71};
        byte[] mGZipBuf = new byte[]{80, 75, 3, 4};
        byte[] mOutBuf = null;
        boolean HeadFlag = false;
        int Signature = 0;
        int WordSize = 0;
        int PageSize = 0;
        int FlagSize = 0;
        try {
            ByteArrayInputStream mStream = new ByteArrayInputStream(Value);
            mStream.read(mIntBuf, 0, 4);
            Signature = iMsgServer2000._$1001(mIntBuf);
            mStream.read(mIntBuf, 0, 4);
            WordSize = iMsgServer2000._$1001(mIntBuf);
            mStream.read(mIntBuf, 0, 4);
            PageSize = iMsgServer2000._$1001(mIntBuf);
            mStream.read(mIntBuf, 0, 4);
            FlagSize = iMsgServer2000._$1001(mIntBuf);
            if (Signature != iMsgServer2000._$1001(mFlagBuf)) {
                mStream.reset();
                WordSize = mStream.available();
            }
            mOutBuf = new byte[WordSize];
            mStream.read(mOutBuf, 0, WordSize);
            return mOutBuf;
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            System.out.println(e.toString());
            return mOutBuf;
        }
    }

    private boolean _$1015() {
        int HeadSize = 64;
```

```
        int BodySize = 0;
        int ErrorSize = 0;
        int FileSize = 0;
        int Position = 0;
        String Md5Calcu = "";
        this._$910 = false;
        this.Md5Value = "";
        try {
            Position = 0;
            String HeadString = new String(this._$904, Position, HeadSize);
            this._$908 = HeadString.substring(0, 15);
            BodySize = Integer.parseInt(HeadString.substring(16, 31).trim());
            ErrorSize = Integer.parseInt(HeadString.substring(32, 47).trim());
            this._$909 = FileSize = Integer.parseInt(HeadString.substring(48, 6
3).trim());
            Position += HeadSize;
            if (BodySize > 0) {
                this._$906 = new String(this._$904, Position, BodySize);
            }
            Position += BodySize;
            if (ErrorSize > 0) {
                this._$907 = new String(this._$904, Position, ErrorSize);
            }
            Position += ErrorSize;
            this._$905 = new byte[FileSize];
            if (FileSize > 0) {
                for (int i = 0; i < FileSize; ++i) {
                    this._$905[i] = this._$904[i + Position];
                }
                if (this._$904.length >= (Position += FileSize) + 32) {
                    this.Md5Value = new String(this._$904, Position, 32);
                    if (this.FDebug) {
                        System.out.println("From Client:" + this.Md5Value);
                    }
                }
            }
            return true;
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            System.out.println(e.toString());
            return false;
        }
    }
```

```
    public void MsgVariant(byte[] mStream) {
        this._$904 = mStream;
        if (this._$907 == "") {
            this._$1015();
        }
    }

    public boolean SavePackage(String FileName) {
        try {
            FileOutputStream mFile = new FileOutputStream(FileName);
            mFile.write(this._$904);
            mFile.close();
            return true;
        }
        catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }

    public boolean MsgFileSave(String FileName) {
        try {
            FileOutputStream mFile = new FileOutputStream(FileName);
            mFile.write(this._$905);
            mFile.close();
            return true;
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            System.out.println(e.toString());
            return false;
        }
    }

    public boolean MsgFileLoad(String FileName) {
        try {
            File mFile = new File(FileName);
            int mSize = (int)mFile.length();
            this._$905 = new byte[mSize];
            FileInputStream mStream = new FileInputStream(mFile);
            for (int mRead = 0; mRead < mSize; mRead += mStream.read((byte[])th
is._$905, (int)mRead, (int)(mSize - mRead))) {
            }
            mStream.close();
            this._$909 = mSize;
```

```
            return true;
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            System.out.println(e.toString());
            return false;
        }
    }

    public String MsgTextBody() {
        return this._$906;
    }

    public byte[] MsgFileBody() {
        return this._$905;
    }

    public String MsgError() {
        return this._$907;
    }

    public String MsgVersion() {
        return this._$908;
    }

    public void MsgTextBody(String Value) {
        this._$906 = Value;
    }

    public void MsgFileBody(byte[] Value) {
        this._$905 = Value;
        this._$909 = this._$905.length;
    }

    public void MsgError(String Value) {
        this._$907 = this._$910 ? "[01]" + Value : Value;
    }

    public int MsgFileSize() {
        return this._$909;
    }

    public void MsgFileSize(int value) {
        this._$909 = value;
    }
```

```java
public void MsgFileClear() {
    this._$909 = 0;
    this._$905 = null;
}

public void MsgTextClear() {
    this._$906 = "";
}

public void MsgErrorClear() {
    this._$907 = "";
}

public String DecodeBase64(String Value) {
    ByteArrayOutputStream o = new ByteArrayOutputStream();
    String m = "";
    byte[] d = new byte[4];
    try {
        int count = 0;
        byte[] x = Value.getBytes();
        while (count < x.length) {
            for (int n = 0; n <= 3; ++n) {
                if (count >= x.length) {
                    d[n] = 64;
                } else {
                    int y = this._$903.indexOf(x[count]);
                    if (y < 0) {
                        y = 65;
                    }
                    d[n] = (byte)y;
                }
                ++count;
            }
            o.write((byte)(((d[0] & 63) << 2) + ((d[1] & 48) >> 4)));
            if (d[2] == 64) continue;
            o.write((byte)(((d[1] & 15) << 4) + ((d[2] & 60) >> 2)));
            if (d[3] == 64) continue;
            o.write((byte)(((d[2] & 3) << 6) + (d[3] & 63)));
        }
    }
    catch (StringIndexOutOfBoundsException e) {
        this._$907 = this._$907 + e.toString();
        System.out.println(e.toString());
    }
```

```
        try {
            m = o.toString(this.Charset);
        }
        catch (UnsupportedEncodingException ea) {
            System.out.println(ea.toString());
        }
        return m;
    }

    public String EncodeBase64(String Value) {
        ByteArrayOutputStream o = new ByteArrayOutputStream();
        byte[] d = new byte[4];
        try {
            int count = 0;
            byte[] x = Value.getBytes(this.Charset);
            while (count < x.length) {
                byte c = x[count];
                d[0] = (byte)((c & 252) >> 2);
                d[1] = (byte)((c & 3) << 4);
                if (++count < x.length) {
                    c = x[count];
                    d[1] = (byte)(d[1] + (byte)((c & 240) >> 4));
                    d[2] = (byte)((c & 15) << 2);
                    if (++count < x.length) {
                        c = x[count];
                        ++count;
                        d[2] = (byte)(d[2] + ((c & 192) >> 6));
                        d[3] = (byte)(c & 63);
                    } else {
                        d[3] = 64;
                    }
                } else {
                    d[2] = 64;
                    d[3] = 64;
                }
                for (int n = 0; n <= 3; ++n) {
                    o.write(this._$903.charAt(d[n]));
                }
            }
        }
        catch (StringIndexOutOfBoundsException e) {
            this._$907 = this._$907 + e.toString();
            System.out.println(e.toString());
        }
        catch (UnsupportedEncodingException ea) {
```

```
            System.out.println(ea.toString());
        }
        return o.toString();
    }

    public int GetFieldCount() {
        int i = 0;
        int j = 0;
        i = this._$906.indexOf("\r\n", i + 1);
        while (i != -1) {
            ++j;
            i = this._$906.indexOf("\r\n", i + 1);
        }
        return j;
    }

    public String GetFieldName(int Index) {
        int i = 0;
        int j = 0;
        int k = 0;
        int n = 0;
        String mFieldString = "";
        String mFieldName = "";
        String mReturn = "";
        while (i != -1 && j < Index) {
            if ((i = this._$906.indexOf("\r\n", i + 1)) == -1) continue;
            ++j;
        }
        k = this._$906.indexOf("\r\n", i + 1);
        if (i != -1 && k != -1) {
            mFieldString = i == 0 ? this._$906.substring(i, k) : this._$906.sub
string(i + 2, k);
            n = mFieldString.indexOf("=", 0);
            if (n != -1) {
                mReturn = mFieldName = mFieldString.substring(0, n);
            }
        }
        return mReturn;
    }

    public String GetFieldValue(int Index) {
        int i = 0;
        int j = 0;
        int k = 0;
        int n = 0;
```

```
        String mFieldString = "";
        String mFieldValue = "";
        String mReturn = "";
        while (i != -1 && j < Index) {
            if ((i = this._$906.indexOf("\r\n", i + 1)) == -1) continue;
            ++j;
        }
        k = this._$906.indexOf("\r\n", i + 1);
        if (i != -1 && k != -1) {
            mFieldString = i == 0 ? this._$906.substring(i, k) : this._$906.sub
string(i + 2, k);
            n = mFieldString.indexOf("=", 0);
            if (n != -1) {
                mFieldValue = mFieldString.substring(n + 1, mFieldString.length
());
                mReturn = this.DecodeBase64(mFieldValue);
            }
        }
        return mReturn;
    }

    public String GetFieldText() {
        return this._$906.toString();
    }

    public String GetMsgByName(String FieldName) {
        int i = 0;
        int j = 0;
        String mReturn = "";
        String mFieldName = FieldName.trim().concat("=");
        i = this._$906.indexOf(mFieldName);
        if (i != -1) {
            j = this._$906.indexOf("\r\n", i + 1);
            i += mFieldName.length();
            if (j != -1) {
                String mFieldValue = this._$906.substring(i, j);
                mReturn = this.DecodeBase64(mFieldValue);
                return mReturn;
            }
            return mReturn;
        }
        return mReturn;
    }

    public void SetMsgByName(String FieldName, String FieldValue) {
```

```
        String mFieldText = "";
        String mFieldHead = "";
        String mFieldNill = "";
        int i = 0;
        int j = 0;
        boolean f = false;
        if (FieldName.compareToIgnoreCase("STATUS") == 0 && this._$910) {
            FieldValue = "[01]" + FieldValue;
        }
        String mFieldName = FieldName.trim().concat("=");
        String mFieldValue = this.EncodeBase64(FieldValue);
        mFieldText = mFieldName + mFieldValue + "\r\n";
        i = this._$906.indexOf(mFieldName);
        if (i != -1 && (j = this._$906.indexOf("\r\n", i + 1)) != -1) {
            mFieldHead = this._$906.substring(0, i);
            mFieldNill = this._$906.substring(j + 2);
            f = true;
        }
        this._$906 = f ? new StringBuffer().append(mFieldHead).append(mFieldTex
t).append(mFieldNill).toString() : this._$906.concat(mFieldText);
    }

    public boolean MakeDirectory(String FilePath) {
        File mFile = new File(FilePath);
        mFile.mkdirs();
        return mFile.isDirectory();
    }

    public boolean MKDirectory(String FilePath) {
        File mFile = new File(FilePath);
        mFile.mkdirs();
        return mFile.isDirectory();
    }

    public boolean RMDirectory(String FilePath) {
        File mFile = new File(FilePath);
        if (mFile.isDirectory()) {
            mFile.delete();
        }
        return true;
    }

    public boolean DelFile(String FileName) {
        File mFile = new File(FileName);
        if (mFile.exists()) {
```

```
            mFile.delete();
        }
        return true;
    }

    public boolean DelTree(String FilePath) {
        File mFile = new File(FilePath);
        if (mFile.isDirectory()) {
            mFile.delete();
        }
        return true;
    }

    public int LoadFilePoint(String FileName) {
        int i = 0;
        int j = 0;
        int mSize = 0;
        String mText = "";
        String mReturn = "-1";
        String mFieldName = "INDEX=";
        try {
            File mFile = new File(FileName + ".fp");
            mSize = (int)mFile.length();
            byte[] mBuffer = new byte[mSize];
            FileInputStream mStream = new FileInputStream(mFile);
            mStream.read(mBuffer, 0, mSize);
            mStream.close();
            mText = new String(mBuffer);
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            return Integer.parseInt(mReturn);
        }
        i = mText.indexOf(mFieldName);
        if (i != -1) {
            j = mText.indexOf("\r\n", i + 1);
            i += mFieldName.length();
            if (j != -1) {
                mReturn = mText.substring(i, j - i);
                return Integer.parseInt(mReturn);
            }
            return Integer.parseInt(mReturn);
        }
        return Integer.parseInt(mReturn);
    }
```

```
public boolean SaveFilePoint(String FileName, int FCount) {
    boolean i = false;
    boolean j = false;
    int mSize = 0;
    String mFieldName = "INDEX=";
    String mCount = "";
    try {
        FileOutputStream mFile = new FileOutputStream(FileName);
        mCount = mFieldName + FCount + "\r\n";
        byte[] mBuffer = mCount.getBytes();
        mSize = mBuffer.length;
        mFile.write(mBuffer, 0, mSize);
        mFile.close();
        return true;
    }
    catch (Exception e) {
        this._$907 = this._$907 + e.toString();
        System.out.println("SaveFilePoint:" + this._$907);
        return false;
    }
}

public boolean SaveFromStream(String FileName, int Index) {
    String mPkName = "";
    mPkName = FileName + ".fp";
    this.DelFile(mPkName);
    if (Index == 0) {
        this.DelFile(FileName);
    }
    try {
        RandomAccessFile mFile = new RandomAccessFile(FileName, "rw");
        mFile.seek(mFile.length());
        mFile.write(this._$905);
        mFile.close();
    }
    catch (Exception e) {
        this._$907 = this._$907 + e.toString();
        System.out.println("SaveFromStream:" + this._$907);
        return false;
    }
    return true;
}

public boolean DecodeBase64ToFile(String Value, String FileName) {
```

```
        ByteArrayOutputStream o = new ByteArrayOutputStream();
        boolean mResult = false;
        byte[] d = new byte[4];
        try {
            int count = 0;
            byte[] x = Value.getBytes();
            while (count < x.length) {
                for (int n = 0; n <= 3; ++n) {
                    if (count >= x.length) {
                        d[n] = 64;
                    } else {
                        int y = this._$903.indexOf(x[count]);
                        if (y < 0) {
                            y = 65;
                        }
                        d[n] = (byte)y;
                    }
                    ++count;
                }
                o.write((byte)(((d[0] & 63) << 2) + ((d[1] & 48) >> 4)));
                if (d[2] == 64) continue;
                o.write((byte)(((d[1] & 15) << 4) + ((d[2] & 60) >> 2)));
                if (d[3] == 64) continue;
                o.write((byte)(((d[2] & 3) << 6) + (d[3] & 63)));
            }
            FileOutputStream mFile = new FileOutputStream(FileName);
            byte[] mBuffer = o.toByteArray();
            int mSize = mBuffer.length;
            mFile.write(mBuffer, 0, mSize);
            mFile.close();
            mResult = true;
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            mResult = false;
            System.out.println(e.toString());
        }
        return mResult;
    }

    public boolean SaveFromFile(String FileName, int FileCount) {
        int mIndex = 0;
        String mPkName = "";
        mPkName = FileName + ".fp";
        this.DelFile(mPkName);
```

```
        try {
            FileOutputStream mFile = new FileOutputStream(FileName);
            for (mIndex = 0; mIndex <= FileCount; ++mIndex) {
                mPkName = FileName + "." + mIndex;
                File nTemp = new File(mPkName);
                FileInputStream mTemp = new FileInputStream(nTemp);
                byte[] mBuffer = new byte[(int)nTemp.length()];
                mTemp.read(mBuffer, 0, (int)nTemp.length());
                mFile.write(mBuffer, 0, (int)nTemp.length());
                mTemp.close();
                nTemp.delete();
            }
            mFile.close();
        }
        catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            System.out.println("SaveFromFile:" + this._$907);
            return false;
        }
        return true;
    }

    public byte[] ReadPackage(HttpServletRequest request) {
        int totalRead = 0;
        int readBytes = 0;
        int totalBytes = 0;
        this.Charset = request.getCharacterEncoding();
        if (this.Charset == null) {
            this.Charset = request.getHeader("charset");
        }
        if (this.Charset == null) {
            this.Charset = "GB2312";
        }
        if (this.FDebug) {
            System.out.println("Charset :" + this.Charset);
        }
        try {
            totalBytes = request.getContentLength();
            this._$904 = new byte[totalBytes];
            while (totalRead < totalBytes) {
                request.getInputStream();
                readBytes = request.getInputStream().read(this._$904, totalRead
, totalBytes - totalRead);
                totalRead += readBytes;
            }
```

```
            if (this._$907 == "") {
                this._$1015();
            }
        }
        catch (Exception e) {
            System.out.println("ReadPackage:" + e.toString());
        }
        return this._$904;
    }

    public void Load(HttpServletRequest request) {
        int totalRead = 0;
        int readBytes = 0;
        int totalBytes = 0;
        if (this.FDebug) {
            System.out.println("Load request ...");
        }
        this.Charset = request.getCharacterEncoding();
        if (this.Charset == null) {
            this.Charset = request.getHeader("charset");
        }
        if (this.Charset == null) {
            this.Charset = "GB2312";
        }
        if (this.FDebug) {
            System.out.println("Charset :" + this.Charset);
        }
        try {
            totalBytes = request.getContentLength();
            this._$904 = new byte[totalBytes];
            while (totalRead < totalBytes) {
                request.getInputStream();
                readBytes = request.getInputStream().read(this._$904, totalRead
, totalBytes - totalRead);
                totalRead += readBytes;
            }
            if (this._$907 == "") {
                this._$1015();
            }
        }
        catch (Exception e) {
            System.out.println("Load:" + e.toString());
        }
    }
```

```
    public void SendPackage(HttpServletResponse response) {
        try {
            ServletOutputStream OutBinarry = response.getOutputStream();
            OutBinarry.write(this.MsgVariant());
            OutBinarry.flush();
            OutBinarry.close();
        }
        catch (Exception e) {
            System.out.println("SendPackage:" + e.toString());
        }
    }

    public void Send(HttpServletResponse response) {
        if (this.FDebug) {
            System.out.println("Send response ...");
        }
        try {
            ServletOutputStream OutBinarry = response.getOutputStream();
            OutBinarry.write(this.MsgVariant());
            OutBinarry.flush();
            OutBinarry.close();
        }
        catch (Exception e) {
            System.out.println("Send:" + e.toString());
        }
    }

    public static String Version() {
        return "8,5,0,0";
    }

    public static String Version(String SoftwareName) {
        String mVersion = "0,0,0,0";
        if (SoftwareName.equalsIgnoreCase("HandWrite") || SoftwareName.equalsIg
noreCase("")) {
            mVersion = "4,3,44,0";
        }
        if (SoftwareName.equalsIgnoreCase("iWebSignature") || SoftwareName.equa
lsIgnoreCase("iWebRevision")) {
            mVersion = "6,4,0,198";
        }
        if (SoftwareName.equalsIgnoreCase("iWebPDF")) {
            mVersion = "7,2,0,338";
        }
        return mVersion;
```

```
    }

    public static String VersionEx() {
        return "\u5ba2\u6237\u7248\u672c";
    }

    public static String VersionEx(String SoftwareName) {
        String mVersionEx = "\u9519\u8bef\u7248\u672c";
        if (SoftwareName.equalsIgnoreCase("HandWrite") || SoftwareName.equalsIg
noreCase("")) {
            mVersionEx = "\u5ba2\u6237\u7248\u672c";
        }
        if (SoftwareName.equalsIgnoreCase("iWebSignature") || SoftwareName.equa
lsIgnoreCase("iWebRevision")) {
            mVersionEx = "\u6807\u51c6\u7248\u672c";
        }
        if (SoftwareName.equalsIgnoreCase("iWebPDF")) {
            mVersionEx = "\u6807\u51c6\u7248\u672c";
        }
        return mVersionEx;
    }
}
```

此时我们有了 **iMsgServer2000** 文件，定位到关键函数操作 `sh.setFieldName(msgObj.GetMsgByName("FIELDNAME"));//取得签章字段名称`，跟进到 `GetMsgByName` 函数，发现里面还有关键函数 `DecodeBase64`

```
public String GetMsgByName(String FieldName) {
        int i = 0;
        int j = 0;
        String mReturn = "";
        String mFieldName = FieldName.trim().concat("=");
        i = this._$906.indexOf(mFieldName);
        if (i != -1) {
            j = this._$906.indexOf("\r\n", i + 1);
            i += mFieldName.length();
            if (j != -1) {
                String mFieldValue = this._$906.substring(i, j);
                mReturn = this.DecodeBase64(mFieldValue);
                return mReturn;
            }
            return mReturn;
        }
        return mReturn;
```

```
    }

    // ... 省略其他代码

    public boolean DecodeBase64ToFile(String Value, String FileName) {
        ByteArrayOutputStream o = new ByteArrayOutputStream();
        boolean mResult = false;
        byte[] d = new byte[4];
        try {
            int count = 0;
            byte[] x = Value.getBytes();
            while (count < x.length) {
                for (int n = 0; n <= 3; ++n) {
                    if (count >= x.length) {
                        d[n] = 64;
                    } else {
                        int y = this._$903.indexOf(x[count]);
                        if (y < 0) {
                            y = 65;
                        }
                        d[n] = (byte) y;
                    }
                    ++count;
                }
                o.write((byte) (((d[0] & 63) << 2) + ((d[1] & 48) >> 4)));
                if (d[2] == 64) continue;
                o.write((byte) (((d[1] & 15) << 4) + ((d[2] & 60) >> 2)));
                if (d[3] == 64) continue;
                o.write((byte) (((d[2] & 3) << 6) + (d[3] & 63)));
            }
            FileOutputStream mFile = new FileOutputStream(FileName);
            byte[] mBuffer = o.toByteArray();
            int mSize = mBuffer.length;
            mFile.write(mBuffer, 0, mSize);
            mFile.close();
            mResult = true;
        } catch (Exception e) {
            this._$907 = this._$907 + e.toString();
            mResult = false;
            System.out.println(e.toString());
        }
        return mResult;
    }
```

到这里，我们就能直接进行本地调试了，但是最后发现解码不出来 `FILENAME`， 真正的原因是这里的 `_$906` 参数

```
    private String _$902 = "8.5.0.0";
    private String _$903 = "FxcYg3UZvtEz50Na8G476=mLDI/jVfC9dsoMAiBhJSu2qPKe+QRbXry1TnkWHlOpw";
    private byte[] _$904;
```

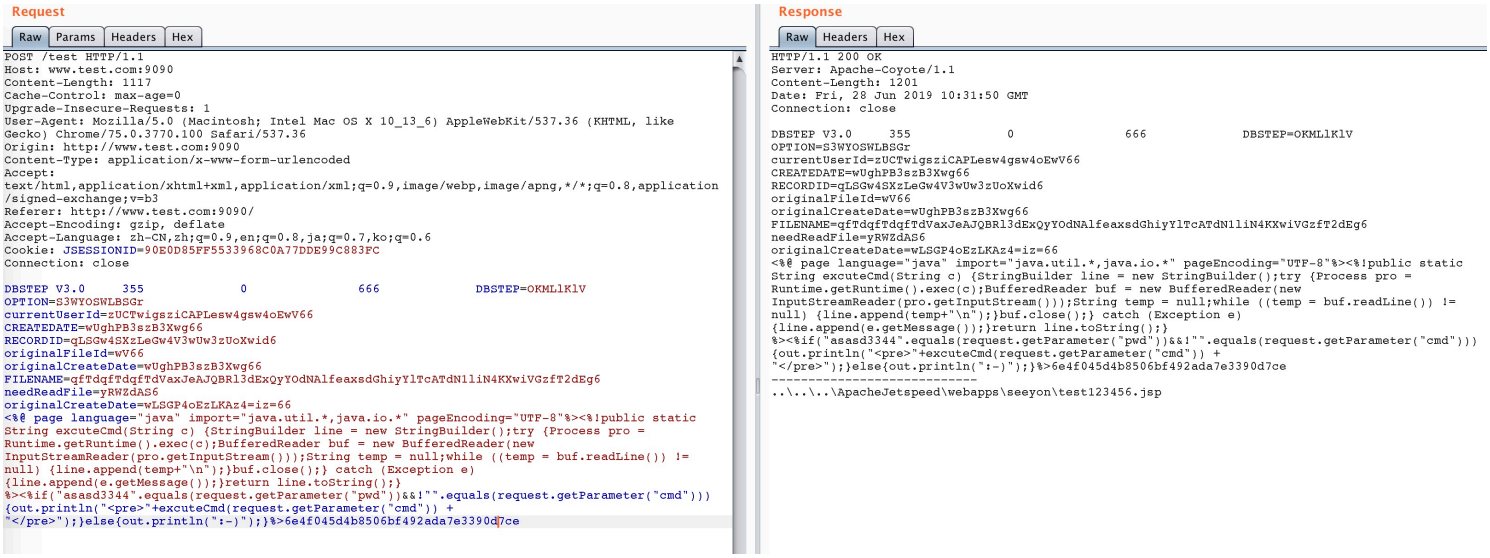
其作用相当于关键的secret key

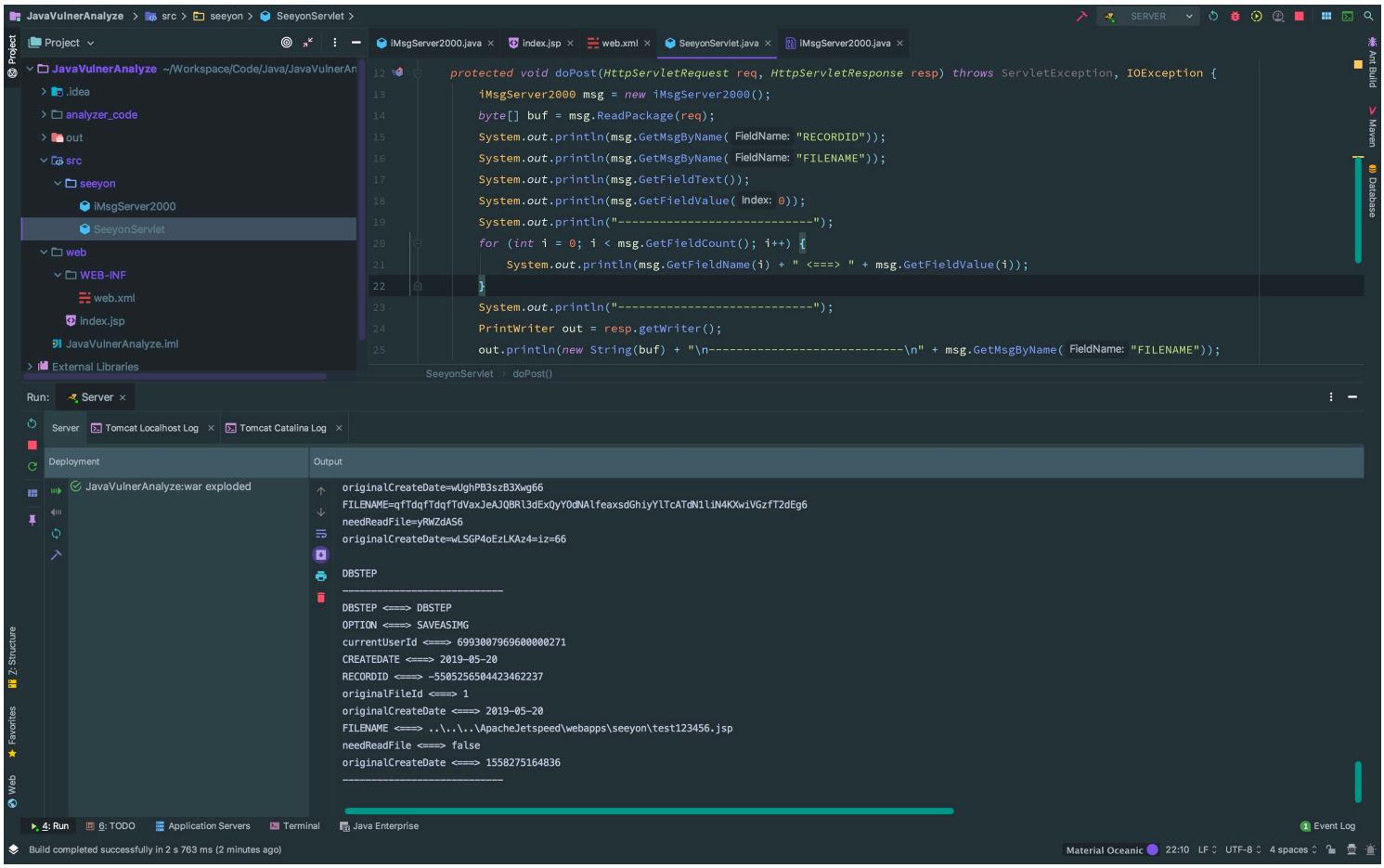
通过 seebug 拿到关键点的 `_$906`

```
gx74KW1roM9qwzPFVOBLS[redacted]05vshXi3GAEuT/m8Dpk6
```

即可实现参数的编码和Payload的解码

5. 如下是解密Payloa的DEMO





6. 漏洞演示环境源码

https://github.com/secoba/JavaVulnerAnalyze

参考链接

- https://paper.seebug.org/964/
- https://bithack.io/forum/350