

[Collapse All](#) | [Expand All](#)

- › [Home](#)
- › [Archived](#)
- › [Acknowledgements](#)
- [Blog and News](#)
- › [Documentation v2-0-alpha](#)
- › [Documentation v1-6](#)
- › [Documentation v1-5](#)
- › [Documentation v1-1](#)
- › [Download](#)
- ▼ [Developer](#)
- › [Coding conventions](#)
- [Classes design v2-0](#)
- ▼ [Database design v2-0](#)
- [Database 2-0 Tables](#)
- [Ticket Status Guide](#)
- [Browser-compatibility-matrix](#)
- [Installation notes for 1-5 development code](#)
- [Notes on 1-0-1 fixes and stuff](#)
- [International Character Handling](#)
- [Transferring multiple files](#)
- [HTML5](#)
- › [Testing](#)
- ▼ [Specifications](#)
- [Functional design](#)
- ▼ [Technical design](#)
- [Technical design - PHP](#)
- [Technical design - Flex](#)**
- [Technical design - Support folders](#)

- Technical design - Layout and design
- Technical design - Other
- Technical design - Packaging
- Technical design - Javascript
- Technical design - SQL

The Glue

release process

Creating repositories

Developers

Usage statistics module

Icons

Existing installations

Feature requests

Online Demo

Project organisation

Roadmap

Release Schedule

Release status and life cycle

Reviews and prototypes

Support and Mailinglists

Twitter

FileSender's workflows

Installation - Linux Source 2-0-Alpha-from-svn

Technical design - Flex



Version 14, last updated by [ricoshae](#) at 2010-11-17

Flex source files

flex/src/filesender.mxml

Description: default package for the filesender application

Public:

- **main_init()** initialize UI
- **resultInit(event:ResultEvent)** HTTPservice result from main_init

- **getVoucher()** all to javascript to get the current voucher id (vid) from the url if it exists
- **getFile()** HTTP service call for voucher data
- **resultLogon(event:ResultEvent)** Redirect for logon to SimpleSAML
- **aboutLink()** redirect to aboutLink
- **helpLink()** redirect to help link
- **resultFile(event:ResultEvent)** HTTPService result from voucher call
- **resultError(event:FaultEvent)** HTTPService has an error - this is a server error and PHP would log an error
- **checkVoucher(vid:String,gStatus:Boolean)** check if vid from javascript is returned
- **gearsActive(gears:Boolean)** setup gears UI
- **installGears()** redirect for gears install link
- **handleStateChange(event:StateChangeEvent)** whenever a state changes initialise the states UI
- **uploadStopped()** when upload complete - update UI
- **uploadStarted()** when upload started - update UI
- **returnFiles(event:ResultEvent)** Not used - deprecated
- **faultHandler(event:FaultEvent)** FAULT Handler for logging HTTP requests
- **logon()** HTTPRequest by logon button
- **noAuth(msg:String)** not authenticated - update UI and display relevant message
- **logoff(msg:String)** called by logoff button
- **newUpload()** recheck that user is still authenticated before allowing new upload
- **resultAuthAdmin(event:ResultEvent)** HTTPService - returns admin then update UI
- **resultAuth(event:ResultEvent)** HTTPService - returns authenticated then update UI
- **adminClick()** check auth before allowing admin
- **fatalError(error:String)** fatal error display function
- **correctLinux(e:Event)** corrects keyboard characters entered when using linux
- **logProcess(message:String)** log client process
- **resultlog(event:ResultEvent)** HTTPRequest result from log

flex/src/org/ricoshae/core/ff_fix.as

Description: call to javascript DoneLoading() to clear the continuous loading status bar. FireFox issue only

flex/src/org/ricoshae/core/FS_Validation.as

Description: Class FS_Validation with a isValid Email method for validating email addresses.

Directory flex/src/org/ricoshae

File admin.mxml

Public:

- **init()** init UI
- **return_error(event:FaultEvent)** HTTPService return error
- **returnFiles(event:ResultEvent)** HTTPService return files table
- **returnLogs(event:ResultEvent)** HTTPService return logs table
- **returnDriveSpace(event:ResultEvent)** HTTPService drive space
- **check_functions_result(event:ResultEvent)** deprecated
- **convertBytes(bytes:Number)** display filesize in datagrid
- **convertDGToHTMLTable(dg:DataGrid)** excel export function
- **readablizeBytes(bytes:Number)** convert filesize display
- **loadDGInExcel(dg:DataGrid)** display datagrid in php

private:

- **filesizeFunc(item:Object, column:DataGridColumn)** display logfile size in custom cell object
- **filesizeFuncFiles(item:Object, column:DataGridColumn)** display file filesize in custom cell object
- **filterUpload(item:Object)** filters for each datagrid in admin
- **filterDownload(item:Object)** filters for each datagrid in admin

- **filterError(item:Object)** filters for each datagrid in admin
- **filterVoucher(item:Object)** filters for each datagrid in admin
- **filterActive(item:Object)** filters for each datagrid in admin
- **dateFunc(item:Object, column:DataGridColumn)** format date in datagrid

download_std.mxml

Flex - download file using browsers download This code still includes deprecated flash download code left in in-case there is a need to return to flash download Public:

- **init()** init UI
- **cancelDownload()** cancel download - flash *deprecated*
- **fileDownloadComplete(event:Event)** download complete - flash *deprecated*
- **returnMail(event:ResultEvent)** HTTPService return complete - flash *deprecated*
- **return_error(event:FaultEvent)** error event - flash *deprecated*
- **readablizeBytes(bytes:Number)** format file size display
- **testBrowserDownload()** file browser download - CURRENT
- **returnBrowser(event:ResultEvent)** HTTPService return from sendBrowser - flash *deprecated*

private:

- **doEvent(evt:Event)** progress event - flash *deprecated*
- **doProgress(event:ProgressEvent)** download progress - flash *deprecated*
- **downloadFile()** download file - flash *deprecated*

helppop.mxml

Public:

- **init()** initialize UI

private:

- **canceldata()** close pop up

myfiles.mxml

Public:

- **init()** initialise UI
- **completeVoucher(event:ResultEvent)** deprecated - remove after confirm not used elsewhere
- **returnVouchers(event:ResultEvent)** return users files
- **returnInsertedVoucher(event:ResultEvent)** HTTPService - return inserted voucher
- **returnMail(event:ResultEvent)** HTTPService - return email sent
- **return_error(event:FaultEvent)** HTTPService - error accessing service
- **functionNameBtn()** confirm delete a file
- **OK_DeleteVoucher(event:CloseEvent)** file ok to delete
- **functionResendBtn()** re-send confirmation
- **OK_ResendVoucher(event:CloseEvent)** HTTPService - re-send
- **returnResendVoucher(event:ResultEvent)** HTTPService - return from re-send
- **functionAddEmail()** call to pop-up add new recipient
- **deletedVoucher(event:ResultEvent)** HTTPService - return from delete voucher
- **saveVoucher(mailTo:String,VoucherUID:String)** HTTPService - save new file
- **convertBytes(bytes:Number)** convert bytes for filesize display **fromLabel_Func(item:Object, column:DataGridColumn)** change 'from' label in datagrid if current user to display 'Me'

Private: **filesizeFunc(item:Object, column:DataGridColumn)** Display file size - datagrid **dateFunc(item:Object, column:DataGridColumn)** display correct date in datagrid

newEmail.mxml

Public:

- **init()** init UI
- **readableBytes(bytes:Number)** format filesize display
- **addNew()** add new recipient
- **fileSaved(event:ResultEvent)** HTTPService returned save
- **fileError(event:FaultEvent)** HTTPService Error

private:

- **cancelData()** close pop up

upload_gears.mxml

Public:

- **init()** initialise all arrays and set button defaults
- **startUploadGears()** call externalInterface to allow gears to select file
- **doGearsUpload()** start the gears upload process
- **returnStatus(sInfo:String,sType:String)** whenever external interface is called - all return results come back through this function
- **fileMoved(event:ResultEvent)** HTTPService return file moved
- **fileSizeReturn(event:ResultEvent)** HTTPService file size returned from checking temp folder
- **cancelUpload()** upload cancelled confirmation
- **OK_CancelUpload(event:CloseEvent)** continue cancelling upload
- **fileComplete()** file upload complete
- **fileSaved(event:ResultEvent)** HTTPService returns saved file
- **fileError(event:FaultEvent)** HTTPService error
- **readableBytes(bytes:Number)** Format file size

private:

- **fnCheckString(str:String)** show upload status
- **fnCheckExtension(str:String)** check file extension
- **updateSpeed(e:TimerEvent)** update speed display

upload_std.mxml

Public:

- **init()** init UI
- **keepAliveHandler(event:TimerEvent)** sends request to keep alive to log - if logging is off then there is no message recorded
- **validateToEmail()** validate email address
- **doUpload()** start the upload process
- **openHandler(event:Event)** handler for file open
- **progressHandler(event:ProgressEvent)** dispatched during file upload
- **completeHandler(event:Event)** Following dispatched when the file has been given to the server script
- **uploadCompleteHandler(event:DataEvent)** Following dispatched when a file upload has completed
- **uploadIt()** setup file reference handler
- **fileCompleted()** completed file uploaded
- **fileSaved(event:ResultEvent)** HTTPService return from saving file data
- **fileError(event:FaultEvent)** FileReference error
- **httpErrorHandler(event:HTTPStatusEvent)** FileReference HTTP error
- **httpIOErrorHandler(event:IOErrorEvent)** FileReference IO error
- **httpSecurityErrorHandler(event:SecurityErrorEvent)** FileReference security error
- **selectHandler(event:Event)** FileReference select handler
- **cancelUpload()** Cancel Flash Upload
- **readableBytes(bytes:Number)** display bytes correctly

private:

- **updateSpeed(e: TimerEvent)** update upload speed
- **fnCheckString(str:String)** check invalid characters
- **fnCheckExtension(str:String)** check invalid extensions

vouchers.mxml

Public:

- **init()** initialise vouchers UI
- **completeVoucher(event: ResultEvent)** HTTPService return from creating a voucher
- **returnVouchers(event: ResultEvent)** HTTPService return with all current users vouchers
- **returnInsertedVoucher(event: ResultEvent)** HTTPService return completed voucher
- **returnMail(event: ResultEvent)** HTTPService return email sent ok
- **return_error(event: FaultEvent)** HTTPService return error if unable to contact server
- **createVouchers()** create voucher
- **functionNameBtn()** Delete voucher confirmation
- **OK_DeleteVoucher(event: CloseEvent)** delete voucher return confirmation
- **deletedVoucher(event: ResultEvent)** HTTPServer return voucher deleted
- **saveVoucher(mailTo:String, VoucherUID:String)** save voucher

private:

- **dateFunc(item: Object, column: DataGrid Column)** format date in datagrid

3th party libraries

as3corelib

Description: An ActionScript 3 Library that contains a number of classes and utilities for working with ActionScript? 3. These include classes for MD5 and SHA 1 hashing, Image encoders, and JSON serialization as well as general String, Number and Date APIs.

URL: <https://github.com/mikechambers/as3corelib/>

License: BSD

Used modules: * **flex/libs/as3corelib.swc** The as3corelib component library used for inclusion to the flex project during compilation.

- **flex/src/com/adobe/crypto/** Library for performing MD5 hashing
- **flex/src/com/adobe/serialization/json/** Library for performing JSON serialization
- **flex/src/com/adobe/utils/IntUtil.as** Library contains reusable methods for operations pertaining to int values.

Binary flash files

www/lib/swf/playerProductInstall.swf

Description: Flash file used for wrapping a swf. It abstracts implementation details about Plugin detection, embedding, and other features so that you only need to call a single method to embed your SWF file.

More info: http://help.adobe.com/en_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf69084-7b86.html

www/lib/swf/filesender.swf

Description: The binary file, which is the result of the compilation of the flex code. This file is transferred to the client and run client-side.

flex/html-template/filesender.swf

Description: Deprecated - removed from SVN

Comments are disabled for this space. In order to enable comments, Messages tool must be added to project.

You can add Messages tool from Tools section on the Admin tab.