



**State of Florida  
Project Aspire  
Agency Interface File Processing,  
Standards, and Load Error  
Notification  
Version 2**

---

## TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION.....</b>	<b>3</b>
<b>2.0</b>	<b>INTERFACE FILE PROCESSING STEPS .....</b>	<b>4</b>
<b>3.0</b>	<b>ASPIRE INTERFACE NOMENCLATURE STANDARDS .....</b>	<b>5</b>
<b>4.0</b>	<b>ERROR HANDLING SCENARIOS.....</b>	<b>10</b>
4.1	INTERFACE DATA LOAD ERRORS .....	10
<b>5.0</b>	<b>INTERFACE ERROR HANDLING STANDARDS .....</b>	<b>10</b>
5.1	ERROR HANDLING FOR BATCH INTERFACES .....	10
5.2	ERROR HANDLING FOR EXCEL TO CI .....	12
<b>APPENDIX A</b>	<b>.....</b>	<b>14</b>

## **1.0 INTRODUCTION**

This paper contains information about interface file processing and error notification solution. The paper covers file processing from the point that the interface file is transferred to Aspire via FTP (File Transfer Protocol) or SSH (Secure Schell) protocols; or that Aspire has created the outbound interface file and/or interface load report. The operational details for each interface are covered in the Interface Operations Guide (IOG). The Interface Operations Guide contains information about interface load reports, business process area, runtime, frequency, error types and notifications.

This paper is divided by sections:

- **Section 2 Interface File Processing Steps**. Describes how Aspire inbound and outbound interfaces will be processed. This section should bring together all sections in the document.
- **Section 3 Aspire Interface Nomenclature Standards**. Describes general standards to which the agencies should adhere.
- **Section 4 Error Handling Scenarios**. Describes the various types of interface data load errors that can be created by the Aspire system.
- **Section 5 Interface Error Handling Standards**. Describes the process that the agencies will utilize to retrieve the interface load reports.

## 2.0 INTERFACE FILE PROCESSING STEPS

<b>Inbound Interfaces</b>	<b>Outbound Interfaces</b>
<ol style="list-style-type: none"><li>1. Agencies will send interface files via FTP or SSH to Aspire.</li><li>2. An Aspire process will validate the files are coming from a valid source.</li><li>3. An Aspire process will move the agency inbound files from the Aspire File Server to the interface processing directory.</li><li>4. After processing the file, the interface load reports will be transmitted via HTTP to the Aspire Report Manager.<ol style="list-style-type: none"><li>a. The reports will be located in folders representing a description of the interface via Aspire Report Manager online. <b><u>Report Folder Name Example:</u></b> <b>Interface: ADML 173</b> <b><u>Aspire Report Manager</u></b> <b>Folder Name: Billing Interface</b></li></ol></li></ol> <p>b. Agency designated personnel will log on to Report Manager via Aspire to view interface load reports.</p> <p><b>[For more information see SECTION 5.0 and APPENDIX A - FIGURE 1]</b></p>	<ol style="list-style-type: none"><li>1. Aspire outbound interface programs will execute and produce an output file.</li><li>2. An Aspire process will move the output files to the appropriate Agency directory.</li><li>3. Agencies will access the files via FTP or SSH.</li></ol> <p><b>[For more information see APPENDIX A - FIGURE 2]</b></p> <p><b>Note:</b> Please refer to the <i>Interface Operations Guide</i> for operational details.</p>

### 3.0 ASPIRE INTERFACE NOMENCLATURE STANDARDS

This section describes nomenclature interface standards for Aspire.

#### Inbound File Name Standard

<Instance>.<interface>.<GLBU>.<DEPTID>.<YYMMDDHHmmss>

**Note: The open and closed angled brackets <> are not part of the file name.**

#### Example

##### *Instance*

*Description:* The nomenclature for Instance is composed of five (5) characters representing an Aspire database instance.

Aspire database instance names:

- FDEVA - for Development Environment
- FTSTA - for Testing Environment
- FQATA - for Quality Assurance Testing
- FUATA - for User Acceptance Testing
- FPROD – for Production

Example File Name:

**FPROD.** <interface>.<GLBU>.<DEPTID>.<YYMMDDHHmmss>

##### *Interface*

*Description:* The nomenclature for Interface is composed of eight (8) characters;

- The first of which is an alpha. The alpha character will be < I > for Interface program.
- The four (4) numeric characters refer to the ADML number.
- The last three (3) characters are alphanumeric to be used only to handle special situations as they occur. The default characters are XXX if not needed.

Example File Name: **FPROD.I0045XXX.**<GLBU>.<DEPTID>.<YYMMDDHHmmss>

### Example (continued)

#### **GLBU**

**Description:** The nomenclature for GLBU is composed of five (5) characters representing the General Ledger Business Unit for an agency. For example: DFS – 43000 represents the DFS GLBU.

Example File Name: FPROD.I0045XXX.**43000**.<DEPTID>.<YYMMDDHHmmss>

#### **DEPTID**

**Description:** The nomenclature for DEPTID (department id) is composed of four (4) characters, the first of which is an alpha. The DEPTID will be derived from the agency department sending the file. If DEPTID is not available a default value of NA00 (Capital N, Capital A, Zero, Zero) will be required.

Example: Treasury - TREA

**Example File Name:** FPROD.I0045XXX.43000.**TREA**.<YYMMDDHHmmss>

#### **YYMMDDHHmmss**

**Description:** The nomenclature for YYMMDDHHmmss (date time stamp) is composed of twelve (12) characters.

YY – Year

MM – Month

DD – Day

HH – Hours

mm – Minutes

ss - Seconds

Example File Name: FPROD.I0045XXX.43000.TREA.**050519121234**

**NOTE:** The timestamp node of the naming standard should not be hard-coded. If the timestamp is hard-coded then there is a risk that interface files may be processed incorrectly.

## Outbound File Name Standard

<Instance>.<interface>.<GLBU>.<DEPTID>.<YYMMDDHHmmss>

**Note: The open and closed angled brackets <> are not part of the file name. Refer to Section 3.0 Inbound File Name Standards for a detailed definition.**

Agencies can rename files utilizing their own standards prior to receiving a file from Aspire servers. See examples below:

**FTP:**

get filename filename

Example:

get FPROD.I0045XXX.43000.TREA.050519121234 file.dat

**SSH:**

scp remote\_host:/path/filename path/filename

Example:

scp rmtsys:/user/home/userfile.txt file1.txt

## Interface Load Report Standard

<Inbound File Name>\_log.txt

**Note:** The open and closed angled brackets <> are not part of the file name. For Inbound File Name Standards refer to Section 3.0.

**Description:** The Interface Load Report is generated per inbound file. In cases where Aspire must deviate from the standard; the exception will be documented in the Interface Operations Guide.

### Interface Load Report Example

FPROD.I0045XXX.43000.TREA.050519121234\_log.txt

<b>Interface Processing Standards</b>	
<b>Header Records</b>	Header records will only be included as needed by specific interface and are noted as such in the record layouts.
<b>Trailer Records</b>	<p>The trailer record will be the last data in the file. The trailer record is required to verify that the entire file was successfully received from the source system.</p> <p>Trailer record format:</p> <ul style="list-style-type: none"> <li>- open angled bracket &lt;</li> <li>- <b>END</b></li> <li>- space</li> <li>- <b>OF</b></li> <li>- space</li> <li>- <b>FILE</b></li> <li>- closed angled bracket &gt;</li> </ul> <p><b>The resulting record would be shown as: &lt;END OF FILE&gt;</b></p>
<b>File Formats</b>	Aspire interfaces are designed to process and create ASCII files with all fields formatted as documented in the record layouts.
<b>File Delimiters</b>	The Aspire standard is to send and receive fixed format files. This means that the files will contain no delimiters.
<b>Signed Numeric Fields</b>	Aspire interfaces will expect signed numeric fields which reserve a byte for the negative sign and a byte for an explicit decimal. For example, a signed numeric field of 5.2 will be 9 bytes long on the file, as follows: -99999.99.
<b>File Processing</b>	Aspire will process multiple agency files per interface. The agency files must be received by the interface processing time defined in the IOG. Agency interface files will be stored on the server for 8 days and will be stored on tape for 90 days.

## 4.0 ERROR HANDLING SCENARIOS

This section describes interface data load errors that can occur in the Aspire system.

### 4.1 INTERFACE DATA LOAD ERRORS

Interface data loading errors occur during staging table population process. Staging tables are repositories of data where validation and processing can occur on interface data prior to the data being processed by Aspire delivered processes. The majority of inbound Aspire interfaces load data into staging tables. The inbound custom interfaces perform edits on the data by placing them in staging tables to minimize the errors created when running downstream processes.

The edits on staging tables have been determined by the Aspire business requirements and are documented in the functional design specifications. When an error situation occurs, the erroneous transactions will be written to Aspire interface load reports and no further processing of the transaction will occur. The sending agency will review their interface load reports, as described in Section 5.1.1.

## 5.0 INTERFACE ERROR HANDLING STANDARDS

The information described here are the standards the Aspire Technical team will implement for uniformity; however, there will be instances where Aspire will deviate from the standard based on the functional requirement. These instances where Aspire must deviate from the standard will be evaluated on a case by case basis.

### 5.1 ERROR HANDLING FOR BATCH INTERFACES

This section describes how State of Florida agencies will receive their interface load reports for batch interfaces. Batch interfaces are those that are regularly scheduled and initiated programmatically.

#### 5.1.1 Inbound File Processing Errors

##### *Error Types*

This section describes the types of errors that may be created by interface processing. The standard error types will be resolved by different processes which are outlined below. The functional specifications determine the type of error processing a particular interface will utilize.

**5.1.1.1 File Transfer Error:** A program runs every five minutes that checks the various files under the subfolders of the ftp parent directory. Currently, two conditions will prompt files to be moved to an “error” directory.

Condition 1: If the file does not contain the trailer record <END OF FILE> and ages past 20 minutes, it is moved to the error directory and the appropriate agency production control is notified.

Condition 2: If the GLBU (3rd node of the file name) does not match the sub-directory that it is in, the file is moved to the error directory, renamed to have an extension identifying the folder it came from and the appropriate agency production control is notified.

For example:

1. A file received on the Aspire File Server is  
FPROD.I0045XXX.43000.TREA.050519121234
2. The program extracts the GLBU (43000).
3. The program reads the sub-directory name of the file location.
4. If the directory name and the GLBU does not match.
5. Then Condition 2 has been met.

**5.1.1.2 File Level Error:** Interfaces will programmatically analyze each transaction contained in the file. However, when an error situation is encountered, the transaction which caused the error will be documented in the interface load report; but, the entire file will not be processed. The following ADMLs create File Level Errors:

- i. **ADML 7 - Journal Entries for Agency Use**
- ii. **ADML 92 - DCF - Public Assistance Cancellations**
- iii. **ADML 1422 – Interface Security**

**5.1.1.3 Transaction Level Error:** The majority of interfaces will evaluate and process errors at the transaction level. A Transaction is one or more records that complete a data entry. This means that the interface will programmatically analyze each transaction contained in the file, based on the error conditions documented in the functional specification. When an error situation is encountered, the transaction which caused the error will be documented in the interface load report and no further processing will occur on that particular transaction. The interface will move on to the next transaction and continue processing. The transaction level errors must be corrected and resubmitted in a file.

### **Error Retrieval Process**

An Aspire interface load report will be created for each specific interface run. These reports will contain the entire transaction which caused the error and a description of why the particular transaction caused an error. If processing completes with no erroneous transactions, the report will contain ‘**All Records Processed Successfully**’.

The interface load reports will be named using the following standard: <inbound file name>\_log.txt. The interface load report standard is explained in section 3.0, Aspire Interface Nomenclature Standards.

Agencies will view their interface load reports in Aspire via Report Manager. Using the Report Manager, agency designated personnel will view reports provisioned by security roles. The reports are grouped by folders. Agency personnel can view reports for their agency only.

## **5.2 ERROR HANDLING FOR EXCEL TO CI**

Excel to Component Interface (CI) interfaces gives State of Florida agencies the ability to enter data into an Excel spreadsheet and validate the data utilizing PeopleSoft’s business rules. If the transactions pass the business rules the data in the spreadsheet will be loaded into PeopleSoft. If the transactions do not pass the business rules the end-user will receive immediate and specific information on the error. The end-user then has the capability to correct the erroneous transaction and re-submit the data in the spreadsheet. The Excel to CI macros will be available for download via Aspire.

### **5.2.1 Understanding the Excel to Component Interface Utility**

Use the Excel to Component Interface utility to upload data from Microsoft Excel into your Aspire, using component interfaces. Each source workbook contains both worksheets and Excel Visual Basic code modules that execute business logic for each transaction.

Use the Microsoft Excel workbooks as a template to create worksheets that are specific to the business logic that you need to use when you are uploading data to Aspire. The code formats spreadsheet data into an Aspire readable Document Object Model (DOM) structure, and submits it to the Aspire database. Next a program parses the DOM structure and uses the component interface to create entries in the Aspire database, validating the data submitted against the business logic built into the component. Because the component interface is a wrapper around the component, all logic applied during data entry is applied when loading data through this tool.

The component interface executes all the necessary events and the field-level edits. Based upon results from saving the component interface, another DOM is created that returns success, warnings, and/or errors to the Microsoft Excel document. Records in error can be corrected and resubmitted.

### Microsoft Excel Column and Row Restrictions

A Microsoft Excel spreadsheet has a physical limitation of 252 columns and 65,000 rows. When you are creating a template and submitting data, you will need to keep these restrictions in mind.

When a template has more than 252 properties, you can work around this limitation by deselecting an appropriate number of columns to meet the restriction before submitting to the database. You can determine in advance which fields should be selected as input cells, which cells should be included for submission, and which cells need not be included at all.

To work around Microsoft Excel's row limitations, you may need to stage your data in batches so that the number of rows on the Staging and Submission sheet will not exceed 65,000.

### 5.2.2 Correcting and Resubmitting Data

After you submit the data to Aspire, results of the process appear on the Staging and Submission sheet. If a submission had an error, the error status appears on the Staging and Submission sheet. Use the Data Input page to correct the data and then resubmit it to Aspire. Continue this process of correcting errors and resubmitting until there are no error records on the Data Input page.

#### Submission Statuses

Errors received for that record appear in a comment field by moving the cursor over the error flag. The records marked OK in green cannot be restaged for submission and can be kept as a record of work completed.

There are three statuses that can appear when you have submitted the data to Aspire.

**Ok** The submission to Aspire completed successfully. The field will be highlighted in green.

**Warning** The data was submitted to Aspire successfully, but a warning was generated in the process. The field will be highlighted in yellow.

**Error** The data was not saved to Aspire due to an error. This field will be highlighted in red. To see the error message that the component interface generated, place your cursor over the Status field. This message indicates how the data needs to be corrected.

**Note.** Data that was not submitted because the error threshold was reached will have no status. When the data that created the error is corrected on the Data Input sheet, the data that was not submitted will be staged to the database.

APPENDIX A

Interface Inbound Process

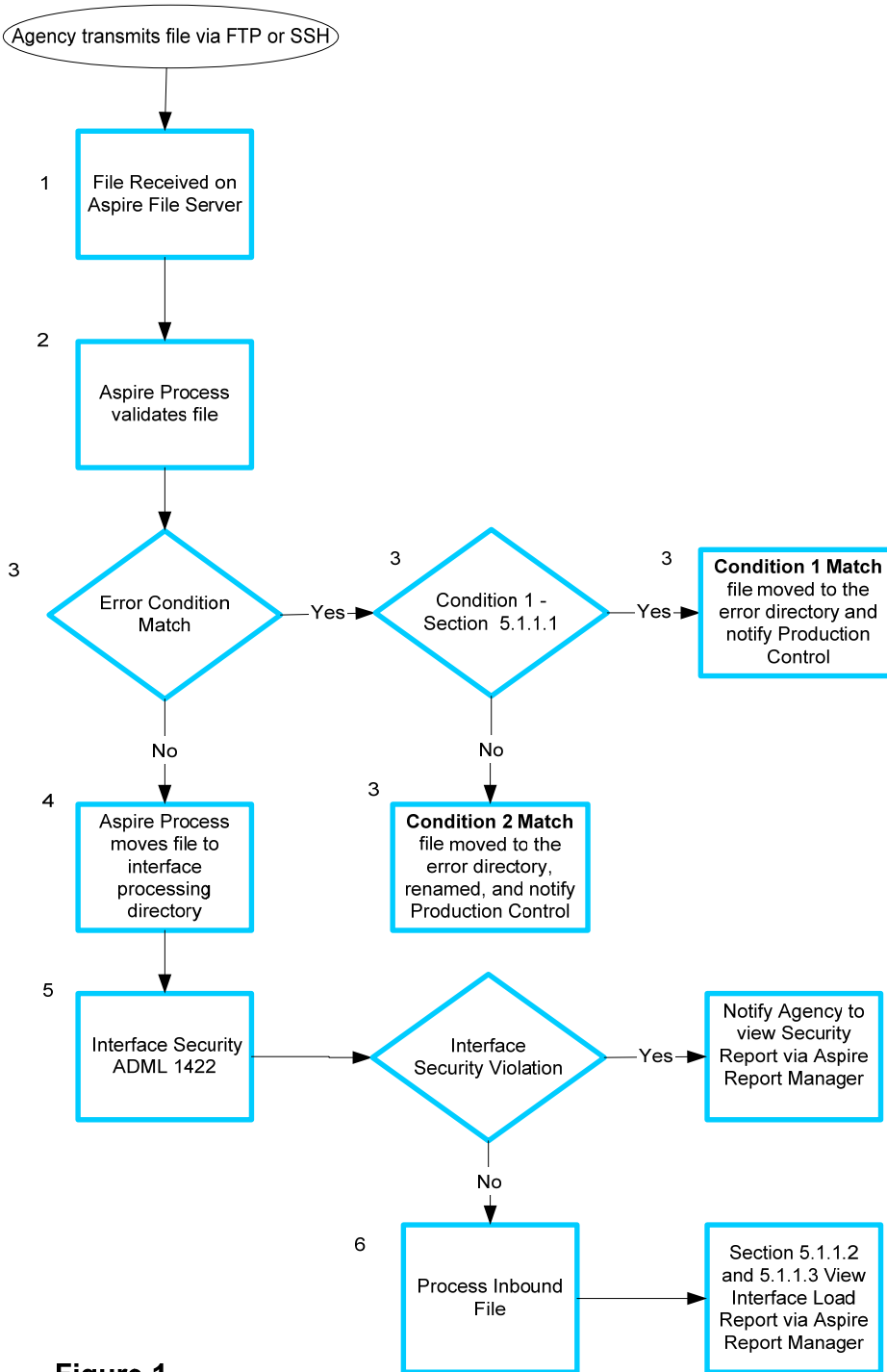


Figure 1

Interface Outbound Process

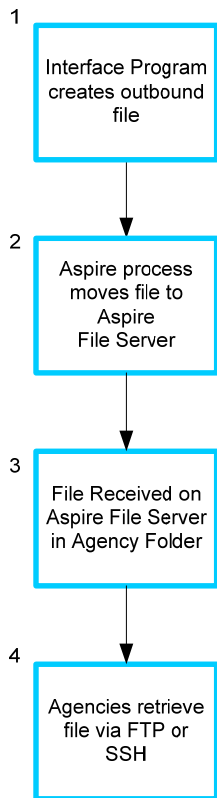


Figure 2