IRIS

Integrated Resource Information System

INTEGRATED RESOURCE INFORMATION SYSTEM

PROJECT HRM PHASE

IRIS HRM General Interface/XML Specifications

Last Update: November 26, 2018

SOA Internal Use Only

TABLE OF CONTENTS

1 TECHNICAL SPECIFICATIONS - INTRODUCTION

The purpose of this document is to provide IRIS Project and State of Alaska (SOA) interface design and development resources with background information needed to understand the overall technical platform used for IRIS interfaces. This document provides:

- A high level overview of the end-to-end interface process.
- Interface file naming conventions, which include system-specific standards for IRIS and departments.
- Guidelines for XML coding of inbound interfaces, as well as common XML conventions and rules.
- Definitions and conventions to aid in understanding terms used throughout the interface design templates.

Although most of this information applies to both IRIS HRM (Human Resource Management) and Financial/Procurement, this document is written specifically for HRM usage.

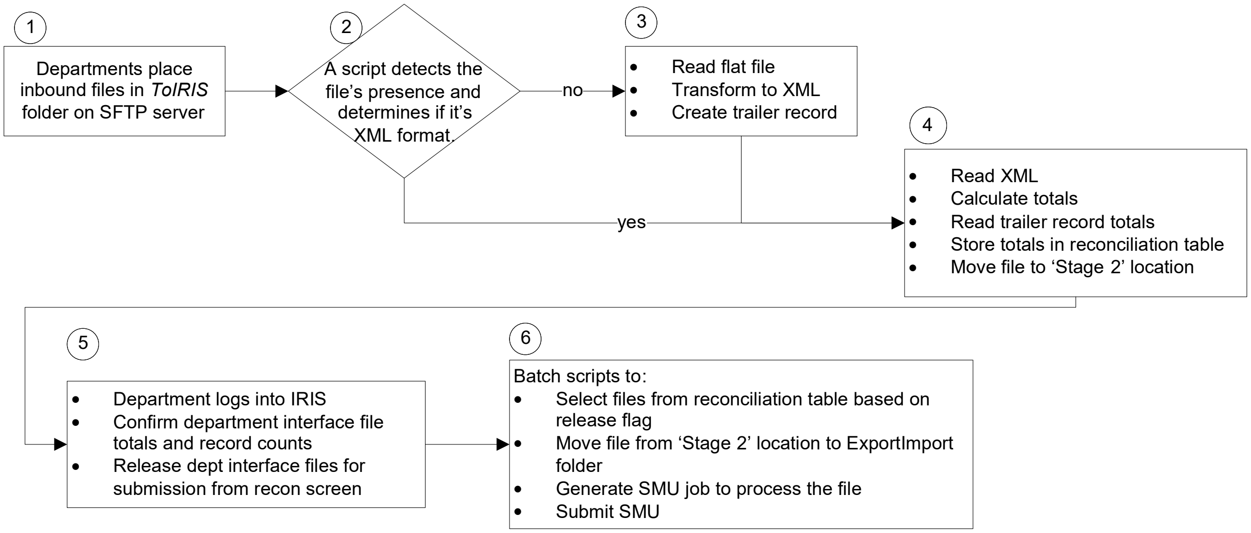
2 PROCESS FLOW

2.1 INBOUND INTERFACE PROCESS FLOW

The following exhibit illustrates the process flow of an inbound interface file, starting from the initiating department and ending with the interface data processed in IRIS.

Note: Instructions for FTP transfer to and from IRIS will be provided separately via email.

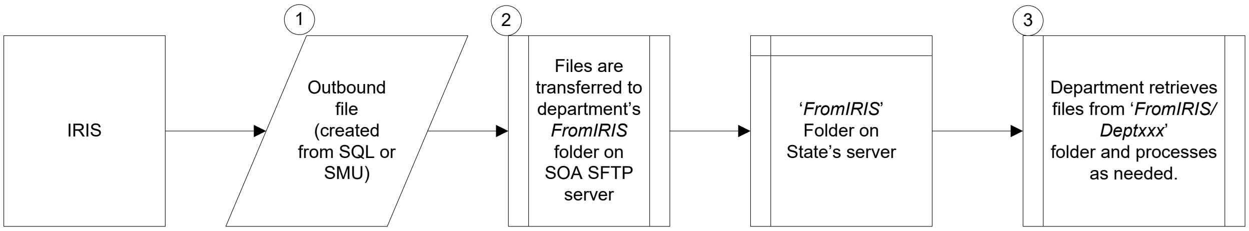
Exhibit 1 IRIS Inbound Interface Process Flow


1
Departments place inbound files in *ToIRIS* folder on SFTP server
2
A script detects the file's presence and determines if it's XML format.
no
yes
3
• Read flat file
• Transform to XML
• Create trailer record
4
• Read XML
• Calculate totals
• Read trailer record totals
• Store totals in reconciliation table
• Move file to 'Stage 2' location
5
• Department logs into IRIS
• Confirm department interface file totals and record counts
• Release dept interface files for submission from recon screen
6
Batch scripts to:
• Select files from reconciliation table based on release flag
• Move file from 'Stage 2' location to ExportImport folder
• Generate SMU job to process the file
• Submit SMU

1. Departments place their inbound interface files in their '*ToIRIS*' folder on the SFTP server.
2. A continuously running batch script will detect the presence of a file and determine if the file is in XML or flat file format. Depending on the file format, the file is processed by step 3 or step 4.
3. For inbound files in the flat file format, the routine will transform it into XML and create a trailer record. The file is then processed by step 4.
4. Inbound XML files will be read and totals (amounts and records) calculated. The trailer records are read and trailer record fields as well as calculated totals are written to a reconciliation table. The trailer record is removed from the file. Lastly, the file is moved to a pre-reconciliation (Stage 2) location.
5. A department user will log into IRIS and navigate to the INTREC reconciliation screen. A list of files available for processing will be presented. The user will confirm the totals and 'release' or authorize the file for processing.
6. A batch script will query the reconciliation table for files that have been released, move them to the IRIS *ExportImport* folder and initiate a System Maintenance Utility (SMU) job to process the file.

2.2 OUTBOUND INTERFACE PROCESS FLOW

Exhibit 2 IRIS Outbound Interface Process Flow


1
2
3
IRIS
Outbound file (created from SQL or SMU)
Files are transferred to department's *FromIRIS* folder on SOA SFTP server
'*FromIRIS*' Folder on State's server
Department retrieves files from '*FromIRIS/ Deptxxx*' folder and processes as needed.

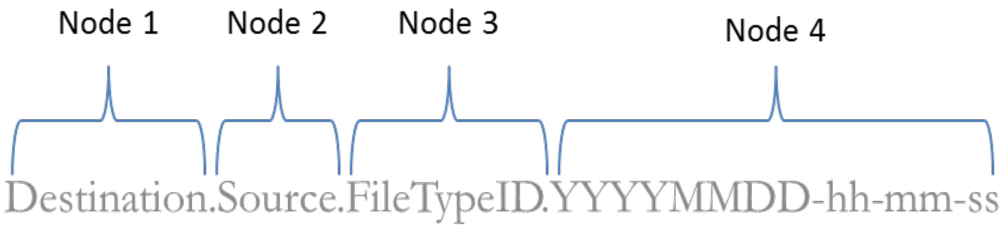
The following describes the Outbound Interface Process:

1. IRIS creates an outbound file.
2. The file is transferred to the department's folder in the '*FromIRIS*' folder on the SFTP server.
3. Departments retrieve their files from their respective folders in the '*FromIRIS*' directory and process the files as needed.

2.3 File Naming Conventions

To process IRIS interface files in an automated fashion, a file naming standard has been developed. All interface files must use the standard to ensure files process correctly.

Exhibit 3 IRIS File Naming Convention



Destination	Preferred length not to exceed 7+1+7+1+8 = 24 Max. 46 characters	refer to section 2.3.1
Source		refer to section 2.3.1
FileTypeID		refer to section 2.3.2
.YYYYMMDD	9 characters	refer to section 2.3.3
-hh-mm-ss	9 characters	refer to section 2.3.3

For example, an inbound interface from DEC's CRITTS to IRIS that was created on June 6, 2016 at 1:20:05 PM would have a file name as follows:

IRISHRM.DECCRIT.TIMESHEET.20160606-13-20-05

2.3.1 Destination and Source Codes

The first two nodes of the naming convention are "Destination" and "Source" and define the destination and source systems. These two values come from a list of standardized system codes created by the IRIS Project team and will be maintained by the Division of Finance (DOF) on an ongoing basis. The list of system codes will be used in both the Destination and Source nodes.

This list of system codes is similar to the current source system ID (SSID) codes used in AKPAY today. Those codes are created and maintained by the DOF's Systems Administration section, working in concert with IT staff in Finance and the interfacing agencies. The IRIS project will work with the agencies to create the comprehensive list of codes to be used in IRIS.

Any agency that requires either an inbound or outbound IRIS interface will be assigned a code. A primary difference from the previous AKSAS & AKPAY SSID is there is a one-to-one correspondence of SSID to interface file. In the new scheme, the same system might have multiple interfaces into and/or out of IRIS. The project team has started the process of identifying the systems and their associated interfaces. The list of department-specific standard codes will be presented to each agency during the Build phase of the IRIS project. This ensures that the destination and source codes for each agency's interface(s), both inbound and outbound, provide an interface file naming standard that is recognizable to their existing systems and associated interfaces.

The following presents the current standard Destination and Source codes for IRIS:

Exhibit 4 IRIS File Naming Convention

Destination / Source Code	Description
IRISLMS	LearnAlaska
IRISFIN	Finance

Destination / Source Code	Description
IRISHRM	Human Resource Management
IRISPAY	Payroll

For departments, the specific standard for destination and source codes used include a three character department code and a 3 to 4 character division or project code. Preferred maximum length for the destination or source codes is 7 characters to keep the over-all file name and FTP folder names to a manageable length.

2.3.2 FILETYPEID

The third portion of the file name is the FileTypeID and provides a descriptive name for the interface between the two systems. The exact content of this portion of the file name will be determined for each interface. The IRIS Project team will work with interfacing agencies to assign names that are meaningful to them, as well as to the project. For example, "TIMESHEET" is the FileTypeID used in the file name example in section 2.3. Departments may suggest names which will be coordinated and approved by the DOF/IRIS team. Preferred maximum length for the FileTypeID is 8 characters to keep the over-all file name and FTP folder names to a manageable length.

2.3.3 DATE TIME

The forth node of the interface file name is the date & time stamp. It has the following format:

YYYYMMDD-hh-mm-ss

This includes a date that is a 4-digit year, a 2-digit month, and a 2-digit day. Time will include two digits each for the hour, minutes and seconds. It is important that the date/time be included on the interface file name so individual interface files can be retained based on their date/time stamp and in accordance with SOA retention policies or agency-specific needs.

3 DATA EXTRACTION FROM IRIS DATABASE FOR OUTBOUND INTERFACES

This section describes how effective date fields are used to access data from the HRM database and is applicable to most **outbound** interfaces.

3.1 EFFECTIVE DATING FOR IRIS TABLES

Multiple data entities in IRIS tables have effective date and expiration date columns. These columns determine when a row is in effect. This section describes how data in effect as of a run date is selected for an outbound interface. In some cases, depending on unique data relationships, additional logic may be required.

To retrieve the records in effect as of the Interface Run Date, from a table with effective dating, the following condition must be true:

Effective_date <= Interface Run Date <= Expiration_date

For example, the following SQL might be used to retrieve EMPL.ASGNMT:

```
SELECT * FROM AHROWN.EMPL_ASGNMT
WHERE
inLAST_RUN_IP_RSLT BETWEEN EFFECTIVE_DT AND EXPIRATION_DT
```

As mentioned above, a table (e.g., Assignment) may have multiple relationships to its parent entity (e.g., Employee), so multiple rows may correctly be in effect. For example, if an employee is assigned to more than one job at a time, then the SQL above could return two assignment rows that are currently in effect. Additional logic would need to be coded into the SQL statement or executed after the SQL executes.

4 GENERAL XML SPECIFICATIONS FOR IRIS INBOUND INTERFACES

Incoming interface files sent to IRIS must be in an XML format that follows specific rules and conventions or it will be rejected by IRIS processing. Essentially, an XML file is a series of values delimited by beginning and ending tags. This chapter presents specifications and examples to illustrate the usage of XML tags in IRIS inbound interface XML files. Note that these XML files will contain one or more IRIS Documents that contain the data to be applied to an entity in IRIS (e.g., an employee, labor distribution, or a work assignment, etc.).

4.1 IRIS XML FILE COMPONENTS

XML files contain multiple components that provide information about the file and the IRIS documents contained in the file. Exhibit 5 presents the five component types in the order they appear in an IRIS incoming interface XML file in the first table column. The second column tells the number of occurrences within a file. The 3rd column presents generic attributes or tags for each component.

Exhibit 5 XML Components

Component Type	Occur-ances	Attributes or XML Tags
File Object	1 per file	<AMS_DOC_XML_IMPORT_FILE VERSION="1.0">

Document Object	1 – many per file	<AMS_DOCUMENT
		DOC_CAT="xx"
		DOC_TYP="xx"
		DOC_CD="xx"
		DOC_DEPT_CD="xx"
		DOC_UNIT_CD="xx"
		DOC_ID="xx"
		DOC_VERS_NO="xx"
		AUTO_DOC_NUM="xx"
		DOC_IMPORT_MODE="xx">

Document Header	1 per Document Object	<XXXX_DOC_HDR AMSDataObject="Y"> <DOC_CAT Attribute="Y"><![CDATA[xx]]></DOC_CAT> <DOC_TYP Attribute="Y"><![CDATA[xx]]></DOC_TYP> <DOC_CD Attribute="Y"><![CDATA[xx]]></DOC_CD> <DOC_DEPT_CD Attribute="Y"><![CDATA[xx]]></DOC_DEPT_CD> <DOC_UNIT_CD Attribute="Y"><![CDATA[xx]]></DOC_UNIT_CD> <DOC_ID Attribute="Y"><![CDATA[xx]]></DOC_ID> <DOC_VERS_NO Attribute="Y"><![CDATA[xx]]></DOC_VERS_NO> <Addtl_Field(s) Attribute="Y"><![CDATA[xx]]>… </XXXX_DOC_HDR>

Component Type	Occur-ances	Attributes or XML Tags
Document Details	1 – many per Document Object	<XXXX_DOC_YYY AMSDataObject="Y"> <DOC_CAT Attribute="Y"><![CDATA[xx]]></DOC_CAT> <DOC_TYP Attribute="Y"><![CDATA[xx]]></DOC_TYP> <DOC_CD Attribute="Y"><![CDATA[xx]]></DOC_CD> <DOC_DEPT_CD Attribute="Y"><![CDATA[xx]]></DOC_DEPT_CD> <DOC_UNIT_CD Attribute="Y"><![CDATA[xx]]></DOC_UNIT_CD> <DOC_ID Attribute="Y"><![CDATA[xx]]></DOC_ID> <DOC_VERS_NO Attribute="Y"><![CDATA[xx]]></DOC_VERS_NO> <Addtl_Field(s) Attribute="Y"><![CDATA[xx]]>… </XXXX_DOC_ YYY >
Closing Tag - Document		</AMS_DOCUMENT>
File Trailer Record	1 per file	<TRAILER_RECORD> <RECORD_TYPE>TRL</RECORD_TYPE> <FILE_NAME>node1.node2.filetypeid.yyyymmdd-hh-mm-ss.xml</FILE_NAME> <RCD_CNT>xx</RCD_CNT> *** Addtl Field(s) depending on document type *** </TRAILER_RECORD>
Closing Tag - File		</AMS_DOC_XML_IMPORT_FILE>

The values for xx, XXXX, YYY, and Addtl_Field(s) will depend on the document associated with the interface.

The specific values to use in these fields are identified in the interface specification templates (e.g., MISC Deductions Interface Document Template).

The **trailer record** is an important component of the IRIS Interface Reconciliation process and must provide accurate document record counts to support the validation that what was sent is what was received. This includes a count of all documents contained in the file, plus one for the trailer record itself.

Additional fields may be required in the trailer record depending on the document type (e.g., MISC Deductions Interface Document Template).

The value for node1.node2.filetypeid.yyyymmdd-hh-mm-ss will depend on the file being transmitted, as described above.

For Document Header and Document Details components, after the beginning XML tag for each field, the string '**Attribute = "Y"**' must be provided where indicated. The **CDATA** section should be used to delimit field data to avoid potential parsing issues with embedded characters that could be misinterpreted as delimiters.

For example, assume a file will be submitted to IRIS which includes a first name and last name. Based on the discussion above, the Document Details component would supply field values formatted like the sample in Exhibit 6.

Exhibit 66 Document Field Specification with Attribute=Y and CDATA section

```
<FIRST_NAME Attribute="Y"><![CDATA[JOHN]]></FIRST_NAME>
<LAST_NAME Attribute="Y"><![CDATA[DOE]]></LAST_NAME>
```

Exhibit 7 below shows the basic XML tag construction with Document Detail lines consisting of one vendor and one accounting line:

Exhibit 7 Interface File with a One Accounting Line CR Document

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<AMS_DOC_XML_IMPORT_FILE>                          ---File Beginning Tag
   <AMS_DOCUMENT>                                  -- Beginning Document Tag
      <CR_DOC_HDR>                                 -- Beginning Header Tag
      </CR_DOC_HDR>                                -- Ending Header Tag
      <CR_DOC_VEND>                                -- Beginning Document Vendor Tag
      </CR_DOC_VEND>                               -- Ending Document Vendor Tag
      <CR_DOC_ACTG>                                -- Beginning Accounting Line Tag
      </CR_DOC_ACTG>                               -- Ending Accounting Line Tag
   </AMS_DOCUMENT>                                 -- Ending Document Tag
   <TRAILER_RECORD>                                -- Beginning File Trailer Tag
   </TRAILER_RECORD>                               -- Ending File Trailer Tag
</AMS_DOC_XML_IMPORT_FILE>                         ---File Ending Tag
```

Exhibit 8 below shows the basic XML tag construction with Document Detail lines consisting of one vendor and two accounting lines:

Exhibit 8 Interface File with a Two Accounting Line CR Document

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<AMS_DOC_XML_IMPORT_FILE>                          ---File Beginning Tag
   <AMS_DOCUMENT>                                  -- Beginning Document Tag
      <CR_DOC_HDR>                                 -- Beginning Header Tag
      </CR_DOC_HDR>                                -- Ending Header Tag
      <CR_DOC_VEND>                                -- Beginning Document Vendor Tag
      </CR_DOC_VEND>                               -- Ending Document Vendor Tag
      <CR_DOC_ACTG>                                -- Beginning Accounting Line 1 Tag
      </CR_DOC_ACTG>                               -- Ending Accounting Line 1 Tag
      <CR_DOC_ACTG>                                -- Beginning Accounting Line 2 Tag
      </CR_DOC_ACTG>                               -- Ending Accounting Line 2 Tag
   </AMS_DOCUMENT>                                 -- Ending Document Tag
   <TRAILER_RECORD>                                -- Beginning File Trailer Tag
   </TRAILER_RECORD>                               -- Ending File Trailer Tag
</AMS_DOC_XML_IMPORT_FILE>                         ---File Ending Tag
```

5 COMMON XML CODING CONVENTIONS:

The following are some coding conventions when constructing XML files:

- There should not be any spaces after "<" or before ">".
- CDATA must be preceded with "<![".
- The data in the CDATA section must be followed by "]]>".
- The special character "<" must be followed by "/" to indicate the end of a tag.
- The attributes can be listed in any order within a data object since use of Document Type Definitions (DTD) is not an IRIS standard.
- Specified single (') or double quotes (") before and after provided values under the **Comments** column in the templates should not be included in corresponding CDATA sections.
- The values "true" and "false" for attributes that can only be set to either value are not case sensitive (i.e., both lower and upper cases are allowed).
- Attributes that are defined as decimal types must include the decimal point and values for digits following the decimal point only if the decimal value is not zero. For example, a value of 50

populated in a field defined with a max size of 15,5 could be represented as 50 or 50.00000 within a CDATA section (i.e., CDATA[50] or CDATA[50.00000]). A value of 50.123 can be specified with or without trailing zeros. These two examples are correct: CDATA[50.12300] or CDATA[50.123].

- Attributes that are defined as decimal types and include negative values must include the "-" sign before the value in a CDATA section (e.g., CDATA[-35.00]).
- Data values contained in the CDATA sections should be trimmed on both sides (i.e., left and right) so the CDATA sections include only the distinct values. Padding fields on the left or right to fill in the remainder of the field size is not necessary as the IRIS Import process is tag oriented rather than fixed field length oriented. However, fields should not exceed the maximum size specified for each attribute in the templates. To further elaborate, assume the following examples:
    - DOC_NM is the XML tag for the Document Name that is defined as VARCHAR2 (40) and has the value "TEST DOCUMENT NAME   ".
    - PYMT_LN_NO is the XML tag for the Payment Line Number that is defined as NUMBER (5) and has the value "   12".
    - DOC_AM is the XML tag for the Document Amount that is defined as DECIMAL (8,2) and has the value "   150.91".

The CDATA section for the above attributes will look like the following:

```
<DOC_NM Attribute="Y"><![CDATA[TEST DOCUMENT NAME]]></DOC_NM>

<PYMT_LN_NO Attribute="Y"><![CDATA[12]]></PYMT_LN_NO>

<DOC_AM Attribute="Y"><![CDATA[150.91]]></DOC_AM>
```

- CDATA sections must be submitted with at least one character between the square parenthesis. Null values, i.e., CDATA[] with no characters will generate an error.
- For attributes that are listed as being **Optional (O)**, there are two options for submitting NULL values:

1) Do not provide the document field at all
2) Specify reserved values, depending on the data type, using these values:

Data Type	Blank Value
CHAR	CDATA[null]
VARCHAR2	CDATA[null]
NUMBER	CDATA[0]
DECIMAL	CDATA[0]
DATE	CDATA[null]

Departments can choose either option to handle "blank" fields. However, option 1 is preferred because it will result in a smaller interface file.

6 DOCUMENT DEFINITIONS AND CONVENTIONS

The following table serves as an aid to better understand the terms used throughout this design.

Term	Definitions	Examples
XML Tag	The tag used in an XML file to define the attribute.	RECORD_TYPE
Caption	Field name of the data element as seen from a console logging into the IRIS system.	
Description	Textual description of the data element.	
R/CR/O/P	Required/Conditional/Optional/Prohibited	**Required (R)** indicates that the data element is required by the interface for transaction to successfully process. **Conditionally Required (CR)** means that when specific conditions are met the data element is required by the interface for transaction to process successfully. When not required the field can be spaces or not included. **Optional (O)** specifies that the data element is optional for the interface and can be populated with spaces or not included. **Prohibited (P)** means that for this transaction the data element is prohibited and must be populated with spaces or not included.

Term	Definitions	Examples
Type	Data type of element	**DATE** – Date Format "YYYY-MM-DD" (with dashes, length is 10). **NUMBER** – A whole number whose value cannot have either positive or negative sign and cannot have a decimal position. An example of a number (4) value is = 9999. In IRIS interfaces, only the significant digits must be supplied. **DECIMAL** – A number whose value can be either positive or negative. Format requirements are specified as "n,d" where "n" is the maximum number of digits, including digits to the left and the right of a decimal point. "n" does NOT include the decimal point or sign. The maximum number of digits to the right of the decimal is "d." Decimal digits to the right of the decimal may be provided but are not required. For example, a decimal type specified as 7,3 with a value of 12.3 may be provided as 12.3, 12.30, or 12.300. Similarly, an integer value of 12 may be provided as 12 (without a decimal point), 12.0, 12.00, or 12.000. The maximum value that can be stored given a 7,3 specification is 9999.999. A value of 12345.1 would be too large to be stored. Negative numbers must be prefixed with a negative sign. Positive numbers may be prefixed with a plus sign; if a sign is not provided, the number is assumed to be positive. An example of a decimal (15,5) field with a negative value is -5.87500. **VARCHAR2** – Alphanumeric (can be a combination of both letters and numbers) field with a maximum character size defined for each field. Data will be left-justified, so VARCHAR2 fields with numeric digits may require all digits be provided. E.g., if UNIT has a length of 4, a value of "0201" would need to be supplied for a unit of 201. **FLAG** –True/False. True conditions = 1, False conditions = 0.
Maximum Size or Length(Len)	The size limit of the data element	Expressed in bytes. For Decimal fields, expressed as 'L,D' where "L" = total number of digits, not including the decimal point or sign, and "D" = maximum number of digits to the right of the decimal. A decimal (7,3) number may have up to 4 digits to the left of the decimal and up to 3 digits to the right.
Comments	Further description of the field or specific comments that relate to the IRIS system.	

Term	Definitions	Examples
Interface	A transfer of data inbound to or outbound from the IRIS databases and/or application. This definition specifically excludes data transferred within the IRIS application itself and users accessing the system remotely.	
Document	An IRIS document such as a MISC (Miscellaneous Deductions) or a GAX (General Accounting Expenditure) document. A document typically contains header and detail pertaining to the purpose of the document (e.g., deduction information, timesheet information, etc.).	
Record	Data specific to a single table or document. (Note: A complete document may need to be separated into multiple Detail Records to allow capture of all necessary information.)	
SMU	System Maintenance Utility, is an IRIS utility for loading documents to the system.	
Transaction	A complete set of Record(s).	
File	One or more transactions. This may refer to one or more transactions all received/sent together from/to a single entity or a composite of one or more transactions received from / sent to one or more entities.	
CVL	Coded Value List. The value of the data element must come from the CVL table specified in IRIS.	