AUTOMATED EXAM PROGRAM
FINOP DATA USER GUIDE
FOR MEMBER FIRMS
VERSION 5.12

Table of Contents

1 Overview

1.1 Purpose/Objective

In an effort to improve its ability to assess risk and the efficiency of the financial and operational examinations of carrying and clearing member firms ("firms"), the Financial Industry Regulatory Authority (FINRA) has implemented an automation initiative to obtain firms' Stock Record, Chart of Accounts and SEA Rule 15c3-3 Reserve Formula Allocation data in a standardized format, using eXtensible Markup Language ("XML")1.

FINRA has included detailed data definitions and specifications for the information being requested. Please use the information in this AEP Guide to assist you in fulfilling this request

1.2 FINRA Team

This section lists the FINRA team members you may contact for assistance:

Name	Title	Department	Contact Information
FINRA Firm Support Distribution List, FINRA Team			FINOPS_FIRM_SUPPORT@finra.org
Arlene Harvey	Project Manager, Examination Development	Business	arlene.harvey@finra.org (646) 315-8644
Sarine Altan	Project Manager, Examination Development	Business	sarine.altan@finra.org (646) 315-8492
Louanne Forbes-Elcock	Project Manager, Examination Development	Business	louanne.forbes-elcock@finra.org (646) 315-8456
Jake Ciuccio	Software Development Lead	Technology	jake.ciuccio@finra.org (240) 386-5581

[1] See *What Is XML?* section in the Appendix for a brief discussion of the XML format.

2 Glossary of Terms

TERM	DEFINITION
ACAT	Allocation Categories – A file containing the Allocation Codes used by the firm.
ACSM	Allocation Category Summary/Allocation Category Totals - A file that summarizes the allocation codes by category.
Allocation Category Code	Term may be used interchangeably with "Allocation Code", "Category Code", "Allocation Category."
APOH	Allocation Hierarchy (Allocation Pair Off) - A file detailing the sequence of the firm's Allocation Pair Off Codes.
APOS	Allocation Pair Off Summary - A file summarizing the Allocation Pair offs.
COA	Firm's Chart of Accounts – A file listing the firm's accounts with definitions/descriptions of each account or range of account(s).
Data Element	A specified piece of data. For example, *Account Number* is a data element that reflects the number or alphanumeric identifier assigned to a specific account.
FinOp	Financial & Operational
FINRA	Financial Industry Regulatory Authority
FINRA SCOA	Same as FINRA's Standard Chart of Accounts (Appendix A).
FINRA's Standard Chart of Accounts (Appendix A)	A listing of categories defined by FINRA used to identify the types of accounts on the Stock Record. The following terms may be used interchangeably to reflect the FINRA Standard Chart of Accounts: "FINRA COA", "FINRA Chart of Accounts", "SCOA Codes".
Member Firm	A carrying and clearing FINRA member organization.
Positions/Quantity	These terms may be used interchangeably to reflect the quantity of securities in an account, the quantity of securities assigned to an allocation code, etc.
Required	Required if the data exists in the firm's records or systems. If the requested data element is reflected on the firm's records or used by the firm in its business process, then that data element must be provided. This will hold true regardless of whether or not the *technical* specification indicates that a data element is required[2]: if the data exists on the firm's records or systems, the data *must* be provided. As an example, consider the "Memo" field on the Stock Record: In the technical specification, the Memo field is listed as optional. However, if the firm's Stock Record contains a "Memo" field or if the firm uses a "Memo" field to record information as part of its business process, that data must be included in the Stock Record submission.

[2] In an XSD, the attribute *minOccurs="0"* indicates that the element is optional from a technical perspective.

SADT	Security Allocation Details - A file that details how each CUSIP is allocated.
SR	Full Stock Record- A file in XML format, identifying all securities held by the firm as of a specified date. See Section 6.1of this AEP Guide.
SRX	The firm's Full Stock Record as of a specified date, in text format.
Unallocated positions	Security positions that are assigned to an allocation code on the Stock Record file , but that do not get paired off during the pair off process and/or remaining in the assigned allocation category after the airoff process is completed.
Unmapped positions	Security positions that are not assigned to an allocation code on the Stock Record file and are not part of the allocation process.
XML schema ("XSD")	A file specification that describes the structure of an XML document. See the *What Is XML?* And *XML Schema (XSD) and Firm-Side Validation* sections of the Appendix.

3 Requested Information

FINRA is requesting several sets of files from its carrying and clearing member firms. The requests for these files will be made via FINRA's Secure Request Manager (SRM) application. Member Firms will receive e-mails informing them of new requests from FINRA. All information will be requested as of a specified date and the submitted data is expected to reflect all information as of the close of business for such date(s). The files requested are listed below:

Required XML files

- Full Stock Record along with the FINRA Standard Chart of Accounts Codes and the Firm's Allocation Codes – SR
- Chart of Accounts – COA
- Allocation Categories – ACAT
- Allocation Hierarchy (Allocation Pair Off)– APOH
- Allocation Summary/Allocation Category Totals – ACSM
- Allocation Pair Off Summary – APOS
- Security Allocation Details – SADT

Non XML files

- Full Stock Record (text)
- Firm Profile – FINRA will collect information regarding the firm's Stock Record and related processes; this information will comprise each firm's Firm Profile. Information of interest to the Stock Record process includes:

    - Number of Stock Records – indicate if the firm holds all security positions on one Stock Record or multiple Stock Records (e.g., one Stock Record for for equities and another for fixed income products).

    - Reporting Method – indicate if the firm's Stock Record and allocation is reported on a gross or netted basis(e.g., fails reported net on stock record, broad on allocation).

    - Pricing of debt securities – indicate if the firm divides price by 100 or market value by 100 in deriving market value of debt securities.. The method used should be applied consistently for Security Types 1-10 and anywhere debt securities are reported. (Refer to Appendix B in the document for a complete list of FINRA's Security Types.)

    - Account Type – Provide a listing and description of each account type used by the firm (e.g., cash, margin, short, etc.) and included in the Stock Record file.

    - Allocation of Options – indicate if the firm allocates options positions.

    - Identification of Certain Allocation Pair offs (e.g. identifying all pairoffs that would be included in Stock Borrow vs. Box (Control), or Fail to Deliver vs. Box (Control) etc.)

    - Grand Total algorithm – when providing grand total information in the SR file, indicate whether or not the firm provides the Grand Total on each XML file that contains that field.A mapping of each of the firm's DTCC account numbers on its Stock Record to the corresponding account number on DTCC's records.

4 Overarching Guidelines

This section lists some basic guidelines that should be observed when generating your firm's Stock Record and Allocation data files in order to ensure a successful data submission process.

GUIDELINE	DESCRIPTION
Understanding the meaning of *Optional* vs. *Required*	The XSDs that govern the Stock Record data sets have some elements flagged as *optional* and some as required. However, *if the data exists in the firm's systems, then that data* must *be provided.* As an example, consider the "Memo" field on the Stock Record: In the schema, the Memo field is listed as optional. However, if the firm's Stock Record contains a "Memo" field or if the firm uses a "Memo" field to record information as part of its business process, that data must be included in the Stock Record submission. If the firm's Stock Record does not contain a Memo field, then this field may be left blank.
Data must be provided in a consistent manner across the different file types	As further discussed in Section 6 of this AEP Guide, the Stock Record process adds data validations to check that the submitted data is consistent throughout the different files submitted. The validations include many such "crossfile" checks. It is critical that the data reflected in one file (e.g., SR file) be consistent with related data in another file (e.g. ACSM). Examples: • FINRA's validations will check to make sure that the total allocated quantity for a given allocation code (as listed on the ACSM) is less than or equal to the total quantity specified in the SR (and APOS or SADT files) for that same allocation category. • FINRA's file processing **will have a problem** if a debt instrument in the SR data reflects a Market value that uses the "divide-price-by-100" convention to determine the market value of the debt instruments, but the SADT data for that same security does NOT. (For debt instruments, the firm may use the divide-price-by-100 convention or the divide-market-value-by-100 convention. Whichever approach is used for one file must be used consistently throughout all of the files.)
Rules for relating a specific security between SR and SADT	The SR and SADT are the only two file types where the process must be able to link a given security in one file type to that same security in the other file type. The specification offers two fields for identifying a security: *CUSIP* and *SecurityID*. In order for the FINRA process to link securities across SR and SADT files, the SADT file must either: a) provide SecurityId on *every* SADT record, or b) provide a NULL or no SecurityId element on *every* SADT record. If the SecurityId is provided on every SADT record, then the FINRA process will use SecurityId to link SR data to SADT data. Therefore, the SecurityId information provided in the SR and SADT files must support that linking. If the firm does not provide SecurityId (i.e, firm provides a NULL SecurityId) for every record, then the FINRA process will use CUSIP to link SR data to SADT data. Of course, in that scenario, the CUSIP information provided in the SR and SADT files must support that linking. The same holds true for the SR record. The SR file must either: a) provide SecurityId on *every* SR record, or b) provide a NULL SecurityId element on *every* SR record. If the firm provides SecurityId on every SR record, then the FINRA process will use SecurityId to link

	SR data to SADT data. Therefore, the SecurityId information provided in the SR and SADT files must support that linking. If the firm does not provide SecurityId (i.e., firm provides a NULL SecurityId) for every record, then the FINRA process will use CUSIP to link SR data to SADT data. Again, in that scenario, the CUSIP information provided in the SR and SADT files must support that linking.
Do not round – provide exact values	FINRA performs a series of calculations and aggregations on the data submitted. In most validations, when comparing one value to another, the application expects to find an exact match. For that reason, please do not use any rounding during the process of creating the files to be submitted to FINRA. Please provide exact values with multiple decimal places as necessary.
Do not use scientific notation for any values	For example, provide a market value of 223400000.00, **not** 2.2234E8.
Use data types that provide *exact* representations for decimal values	The *float* (and *double*) data types cannot represent *exactly* many numeric values, and should be used only with great caution in financial applications.[3] The new eAnalytics platform and the latest schemas do not use the *float* data type – they use *decimal* types that provide exact representations of decimal values. If firms are not already using *decimal* or *big decimal* types in their processes (as opposed to *float* or *double* types), they should do so.

Rules for handling whitespace	"Whitespace" refers to spaces, tabs, returns, and line feeds within the submitted data. FINRA's processes handle whitespace as follows:
	
	• Whitespace *within a data value* will be collapsed to one space. For example, a Security Description value of:
	
	`"U.S. TREASURY BOND  @ 6.25% DUE"`
	
	will be processed as if it read:
	
	`"U.S. TREASURY BOND @ 6.25% DUE"`
	
	(Note how the multiple spaces between the word BOND and the @ sign are collapsed into one space.)
	
	• Whitespace *at the beginning and/or end of a data value* will be eliminated. In technical terms, data values will be "trimmed". For example, a Security Description value of:
	
	`"  U.S. TREASURY BOND @ 6.25% DUE  "`
	
	will be processed as if it read
	
	`"U.S. TREASURY BOND @ 6.25% DUE"`
	
	(Note how the leading and trailing spaces have been removed.)

No Negative or Positive signs should be used	Do not include negative or positive (plus) signs in the Quantity or Market Values fields anywhere in the submission.

[3] Please see http://en.wikipedia.org/wiki/Floating_point or http://download.oracle.com/docs/cd/E19957-01/806-3568/ncg_goldberg.html for details, or search for "floating point accuracy".

All position values and the corresponding prices in all file types should be specified in USD	Position values and prices should always be specified in U.S. dollars. For positions held in a foreign currency, the schema offers elements that must be used to specify the currency and exchange rate for the position; however, the position value (and price) must be provided in USD.

5 XML Schemas

The following section provides an element-by-element description of all of the data that FINRA is requesting. The firm's submission must adhere to FINRA's standard XML formats for Stock Record requests – those formats are defined by XML Schemas (XSDs). The individual data elements found in those schemas are detailed in the tables in this section. Each schema has a corresponding diagram that provides all of the information found in the tables in this section and more, including a visual representation of the hierarchical nature of the data. In most cases, the schema diagram is included in the corresponding schema's section, below; however, some of the diagrams were too big to fit these pages. All of the schemas and *all* of the diagrams, large and small, are available at http://www.finra.org/finops/stockrecord.

The tables in this section sometimes list an element as having a Type of "Container". A Container element is just that: an element that is used to group (or contain) other elements.

5.1 Full Stock Record Data (SR)

Definition: A record reflecting each security long position and each security short position, held by the firm for all accounts and showing the location of all securities long and the offsetting position to all securities short as of a specified date. See also as a reference: SEA Rule 17a-3(a)(5) and this AEP Guide for the specifications of all data elements required for the Full Stock Record.

FINRA is requesting the Full Stock Record along with additional data elements that may or may not be part of the Full Stock Record (i.e., Security Type, Price Indicator, Factor, etc.). The data will be requested as of a specified date, to be provided at the time the request is made.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	StockRecord		[00000] In addition to the individual Security sub-elements, this container holds summary level information for the entire stock record.
attribute	StockRecord/@AsOfDate	xs:date	[00010] As Of Date: The date of the data being provided - should be the same as the AsOf date of the Full Stock Record requested.
attribute	StockRecord/@FirmCrdNumber	xs:int	[00030] Firm CRD Number: Firm's Central Registration Depository (CRD) number.
element	StockRecord/FileControl		[00040] File Control Data: The attributes associated with this element provide some metadata surrounding the SR files submitted - number of files, XSD version used, etc.
attribute	StockRecord/FileControl/@FileNumber	xs:int	[00050] File Number: For submissions that contain multiple SR files, this value represents the index of this file within that set. For example, the second of ten files would have a "2" as the value of this attribute. If there is only one SR file in the submission, then this attribute should be set to "1".
attribute	StockRecord/FileControl/@FileTotal	xs:int	[00060] Total Number Files: For submissions that contain multiple SR files, this value represents the total number of SR files in the submission. If there is only one SR file in the submission, then this attribute should be set to "1".
attribute	StockRecord/FileControl/@Version	xs:normalizedString	[00070] Schema Version: Should reflect the schema version used for this submission - may provide either the entire name of the schema document (i.e., "AEP_FinOPs_SR_V9.xsd" or just the numeric portion (i.e., "9").
element	StockRecord/GrandTotalLongSD	DQS.NonNegativeDecimal	[00080] Grand Total Long Quantity on Settlement Date: Aggregate (sum) for all Settlement Date Long Positions across all securities. If the Stock Record data is split across multiple SR files, then there are three ways for the submitter to provide the true Grand Total across all SR files: 1) "SUM" approach A: the Grand Total element in each SR file reflects the Total *for that one file* - in this

			case, the consuming app will sum the Grand Total elements across all of the SR files, and that sum will reflect the true Grand Total across all SR files; 2) "SUM" approach B: the Grand Total element in each SR file except one is zero; the one non-zero entry reflects the true Grand Total across ALL SR files; 3) "EVERY" approach: each and every SR file contains the same Grand Total number, and that Grand Total number reflects the true Grand Total across ALL SR files. This value should never be a negative number. During firm setup process, the firm must tell FINRA whether it uses the "SUM" or "EVERY" approach.
element	StockRecord/GrandTotalShortSD	DQS.NonNegativeDecimal	[00090] Grand Total Short Quantity on Settlement Date: Aggregate (sum) for all Settlement Date Short Positions across all securities. See description for Grand Total Long Quantity on Settlement Date element. This value should never be a negative number.
element	StockRecord/GrandTotalLongTD	DQS.NonNegativeDecimal	[00100] Grand Total Long Quantity on Trade Date: Aggregate (sum) for all Trade Date Long Positions across all securities. See description for Grand Total Long Quantity on Settlement Date element. This value should never be a negative number.
element	StockRecord/GrandTotalShortTD	DQS.NonNegativeDecimal	[00110] Grand Total Short Quantity on Trade Date: Aggregate (sum) for all Trade Date Short Positions across all securities. See description for Grand Total Long Quantity on Settlement Date element. This value should never be a negative number.
element	StockRecord/TotalNumberOfSecurities	DQS.NonNegativeDecimal	[00120] Total Number of Securities: The distinct count of all securities. See description for Grand Total Long Quantity on Settlement Date element. This value should never be a negative number.
element	StockRecord/Security		[00130] Security: Detail and summary information for each security that appears in the Full Stock Record.
element	StockRecord/Security/Cusip	xs:normalizedString	[00140] CUSIP: (Committee on Uniform Securities Identification Procedures) Universal identifer for each security. See important notes under Security ID element, below.
element	StockRecord/Security/SecurityId	DQS.SecurityDetail	[00150] Security ID: Firm's internal identifier for the Security, if applicable. In order for FINRA's process to link securities across the SR and SADT files, the SR file must either (a) provide Security ID on every SR record OR (b) provide a NULL on every SR record. In other words, Security ID must be present on every SR record or on NO SR records. If the firm provides Security ID FINRA's process will use Security ID to link SR data to SADT data. If the firm provides a NULL Security ID FINRA's process will use CUSIP to link SR data to SADT data. Maximum field length shall not exceed 30 characters.
element	StockRecord/Security/SecuritySymbol	DQS.SecurityDetail	[00160] Security Symbol: Security's trading symbol. Maximum field length shall not exceed 30 characters.
element	StockRecord/Security/SecurityDescription	DQS.Description	[00170] Security Description: Complete security name; include coupon description where applicable. Maximum field length shall not exceed 200 characters.
element	StockRecord/Security/SecurityType	SecurityTypes	[00180] Security Type: The type of the security - see the Security Type codes included in the Appendix to the Data Guide (or see the Security Type codes enumerated in the SecurityType type definition in this same XSD file).
element	StockRecord/Security/MaturityDate	OptionalDate	[00190] Maturity Date: Payable date of debt instrument. Do not provide the element at all if not applicable.
element	StockRecord/Security/Factor	DQS.NonNegativeDecimal	[00200] Factor: The multiplier used to determine the outstanding principal balance in a mortgage pool. If not applicable, provide a value of 1.
element	StockRecord/Security/Price	Price	[00210] Price: The Market Price of the security as of the date for this filing. [Note change from xs:float in V8 to Price type (based on xs:decimal) in V9.]
attribute	StockRecord/Security/Price/@PriceIndicator	xs:string	[00220] Price Indicator: Flag that indicates how the pricing for this security was derived. “A” indicates security was priced from an automated pricing system; “M” indicates security price was unavailable from automated pricing services and price was entered manually (e.g., from trader's desk).
element	StockRecord/Security/Currency	xs:normalizedString	[00250] Currency: If applicable, specify the monetary unit that the security is traded in. Use currency symbols as defined in ISO 4217 Type Currency Code List. If no value is provided, the application will assume "USD". (All

			prices and values provided in the submitted data must be in USD.)
element	StockRecord/Security/ExchangeRate	xs:normalizedString	[00260] Exchange Rate: The rate at which the currency was converted to USD. If not applicable, provide a value of 1.
element	StockRecord/Security/AllocatedPrice	DQS.NonNegativeDecimal	[00270] Allocated Price: The price of each security in USD as provided in the Security Allocation Details file. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalLongSD	DQS.NonNegativeDecimal	[00280] Total Long Quantity on Settlement Date by Security: The sum of all Long positions in this security as of the Settlement Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalShortSD	DQS.NonNegativeDecimal	[00290] Total Short Quantity on Settlement Date by Security: The sum of all Short positions in this security as of the Settlement Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalLongTD	DQS.NonNegativeDecimal	[00300] Total Long Quantity on Trade Date by Security: The sum of all Long positions in this security as of the Trade Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalShortTD	DQS.NonNegativeDecimal	[00310] Total Short Quantity on Trade Date by Security: The sum of all Short positions in this security as of the Trade Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/SegRequiredBySecurity	DQS.NonNegativeDecimal	[00320] Segregated Quantity Required by Security: The number of shares required to be segregated for each account holding this security, pursuant to SEA Rule 15c3-3 (segregation instructions for fully paid and excess margin securities). [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalLongMKSD	DQS.NonNegativeDecimal	[00330] Total Long Market Value on Settlement Date by Security: The Long Market Value as of the Settlement Date. The Market Value should comprehend debt pricing and factor calculations where applicable. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalShortMKSD	DQS.NonNegativeDecimal	[00340] Total Short Market Value on Settlement Date by Security: The Short Market Value as of the Settlement Date. The Market Value should comprehend debt pricing and factor calculations where applicable. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalLongMKTD	DQS.NonNegativeDecimal	[00350] Total Long Market Value on Trade Date by Security: The Long Market Value as of the Trade Date. The Market Value should comprehend debt pricing and factor calculations where applicable. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	StockRecord/Security/TotalShortMKTD	DQS.NonNegativeDecimal	[00360] Total Short Market Value on Trade Date by Security: The Short Market Value as of the Trade Date. The Market Value should comprehend debt pricing and factor calculations where applicable. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account		[00400] Account information for each/every account in which the parent element security has a position in the Full Stock Record as of the AsOf date specified.
element	Account/AccountNumber	DQS.AccountDetail	[00410] Account Number: Alphanumeric identifier used by firm to identify each account. Maximum field length shall not exceed 100 characters.
element	Account/BranchCode	DQS.BranchCode	[00420] Branch Code: Code used to identify the Branch where the account is located. Maximum field length shall not exceed 50 characters.
element	Account/AccountName	DQS.Description	[00430] Account Name: The name on the account. Maximum field length shall not exceed 200 characters.
element	Account/AccountType	DQS.AccountType	[00440] Account Type: Code used to identify the type of the account.
element	Account/LongPosSettle	DQS.NonNegativ	[00450] Long Quantity on Settlement Date by Account:

		eDecimal	The sum of all Long positions for current security in this account as of the Settlement Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/ShortPosSettle	DQS.NonNegativeDecimal	[00460] Short Quantity on Settlement Date by Account: The sum of all Short positions for current security in this account as of the Settlement Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/LongMKSettle	DQS.NonNegativeDecimal	[00470] Long Market Value on Settlement Date by Account: The Long Market Value of current security in this account as of the Settlement Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/ShortMKSettle	DQS.NonNegativeDecimal	[00480] Short Market Value on Settlement Date by Account: The Short Market Value of current security in this account as of the Settlement Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/LongPosTrade	DQS.NonNegativeDecimal	[00490] Long Quantity on Trade Date by Account: The sum of all Long positions for current security in this account as of the Trade Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/ShortPosTrade	DQS.NonNegativeDecimal	[00500] Short Quantity on Trade Date by Account: The sum of all Short positions for current security in this account as of the Trade Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/LongMKTrade	DQS.NonNegativeDecimal	[00510] Long Market Value on Trade Date by Account: The Long Market Value of current security in this account as of the Trade Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/ShortMKTrade	DQS.NonNegativeDecimal	[00520] Short Market Value on Trade Date by Account: The Short Market Value of current security in this account as of the Trade Date. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/SegRequiredByAccount	DQS.NonNegativeDecimal	[00530] Segregated Quantity Required by Account: For this security/account, the number of shares required to be segregated for each long position in each account pursuant to SEA Rule 15c3-3 (segregation instructions for fully paid and excess margin securities). [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based on xs:decimal) in V9.]
element	Account/LastActivitySettleDate	OptionalDate	[00540] Last Activity Settlement Date: Settlement date of the last movement for this security/account. If not applicable, omit the element entirely. This value must be earlier than or equal to the request date of the filing.
element	Account/LastActivityTradeDate	OptionalDate	[00550] Last Activity Trade Date: Trade date of the last movement for this security/account. If not applicable, omit the element entirely. This value must be earlier than or equal to the request date of the filing.
element	MemoFields		[00560] Memo Fields: Container element; holds 0 or more MemoField elements. If not applicable, omit this element entirely.
element	MemoField		[00570] Memo Field: Container that supports the inclusion of Memo information. Each Memo Field container is comprised of a Memo Field Quantity and a Memo Field Type.
element	MemoField/MemoFieldType	DQS.Memo	[00580] Memo Field Type: Alphanumeric codes that indicate the location of securities for non-streetside accounts (e.g., TF [Transfer], TTRF/FTRF [To Transfer/From Transfer], SK [Safekeeping], TSKC/FSKC [To Safekeeping/From Safekeeping]). Maximum field length shall not exceed 4000 characters.
element	MemoField/MemoFieldQuantity	xs:normalizedString	[00590] Memo Field Quantity: The long or short quantity for the corresponding Memo Field code.
element	Account/AllocationCode	DQS.AllocationCode	[00600] Allocation Code: The Allocation Code used by the firm in its Reserve Formula Allocation. The Allocation Code is assigned to securities in an account. FINRA plans to deprecate the separate DALLOC and MAPP files at some point in the future - all allocation codes will have to be provided using this element on the SR file. The

			Allocation Code is referred to in some places as "Allocation Category Code" - those terms can be used interchangeably. Maximum field length shall not exceed 50 characters.
element	Account/COACode	xs:int	[00610] FINRA Standard Chart of Accounts Code: the FINRA Standard Chart of Accounts Code assigned to this account. FINRA plans to deprecate the separate SCOA file at some point in the future - all FINRA Standard Chart of Accounts codes will have to be provided using this element on the SR file. The FINRA Standard Chart of Accounts Code is referred to in some places as "FINRA Category Number" or as "SCOA Code" - those terms can be used interchangeably.

Please see http://www.finra.org/finops/stockrecord for the image file for the SR. As mentioned, that image file provides a diagrammatic view of the SR data structure. The diagram contains the same information as provided in the table, above; for some readers, the diagram may offer a clearer picture of the file content. The diagram can be viewed using most image tools and/or using a browser.

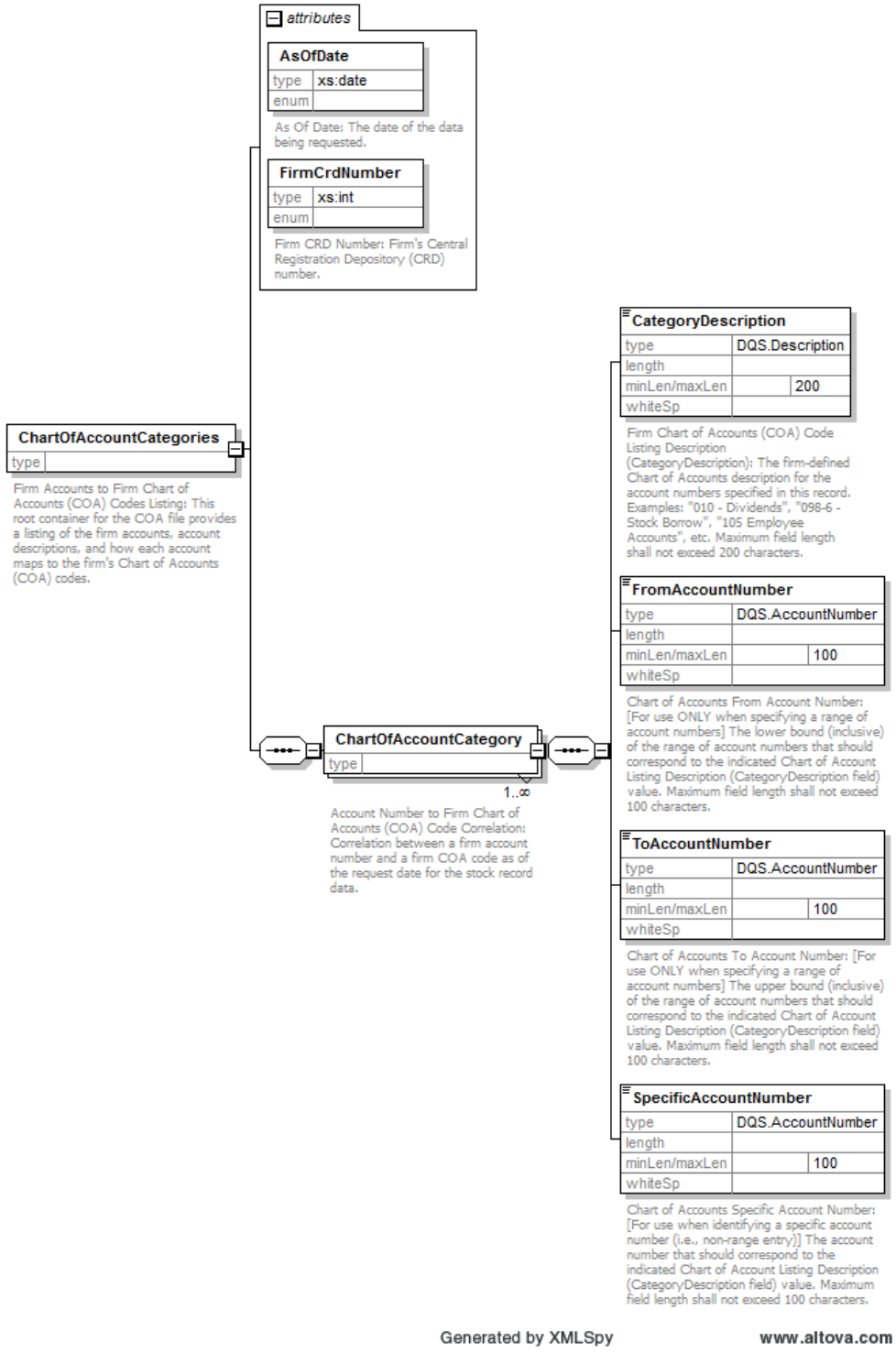
5.2 Chart of Accounts (COA)

Definition: A file containing a listing of all accounts and/or account ranges on the firm's Stock Record, along with their corresponding definitions.

FINRA is requesting the Chart of Accounts utilized by the Firm in order to understand the Full Stock Record as well as the Full Stock Record Allocation Process. The data provided should be as of the same date as the Full Stock Record data file.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	ChartOfAccountCategories	Container	Firm Accounts to Firm Chart of Accounts (COA) Codes Listing: This root container for the COA file provides a listing of the firm accounts, account descriptions, and how each account maps to the firm's Chart of Accounts (COA) codes.
attribute	ChartOfAccountCategories/@AsOfDate	xs:date	As Of Date: The date of the data being requested.
attribute	ChartOfAccountCategories/@FirmCrdNumber	xs:int	Firm CRD Number: Firm's Central Registration Depository (CRD) number.
element	ChartOfAccountCategories/ChartOfAccountCategory	Container	Account Number to Firm Chart of Accounts (COA) Code Correlation: Correlation between a firm account number and a firm COA code as of the request date for the stock record data.
element	ChartOfAccountCategories/ChartOfAccountCategory/CategoryDescription	DQS.Description	Firm Chart of Accounts (COA) Code Listing Description (CategoryDescription): The firm-defined Chart of Accounts description for the account numbers specified in this record. Examples: "010 - Dividends", "098-6 - Stock Borrow", "105 Employee Accounts", etc. Maximum field length shall not exceed 200 characters.
element	ChartOfAccountCategories/ChartOfAccountCategory/FromAccountNumber	DQS.AccountNumber	Chart of Accounts From Account Number: [For use ONLY when specifying a range of account numbers] The lower bound (inclusive) of the range of account numbers that should correspond to the indicated Chart of Account Listing Description (CategoryDescription field) value. Maximum field length shall not exceed 100 characters.
element	ChartOfAccountCategories/ChartOfAccountCategory/ToAccountNumber	DQS.AccountNumber	Chart of Accounts To Account Number: [For use ONLY when specifying a range of account numbers] The upper bound (inclusive) of the range of account numbers that should correspond to the indicated Chart of Account Listing Description (CategoryDescription field) value. Maximum field length shall not exceed 100 characters.
element	ChartOfAccountCategories/ChartOfAccountCategory/SpecificAccountNumber	DQS.AccountNumber	Chart of Accounts Specific Account Number: [For use when identifying a specific account number (i.e., non-range entry)] The account number that should correspond to the indicated Chart of Account Listing Description (CategoryDescription field) value. Maximum field length shall not exceed 100 characters.

The image, below, provides the same information shown in the tabular layout, above; for some readers, the diagram may offer a clearer picture of the file content. This image is available as a separate .png file (see http://www.finra.org/finops/stockrecord), which can be viewed using most image tools and/or using a browser. (This may be useful for viewing images that do not show up well on the one page display available within this document.)



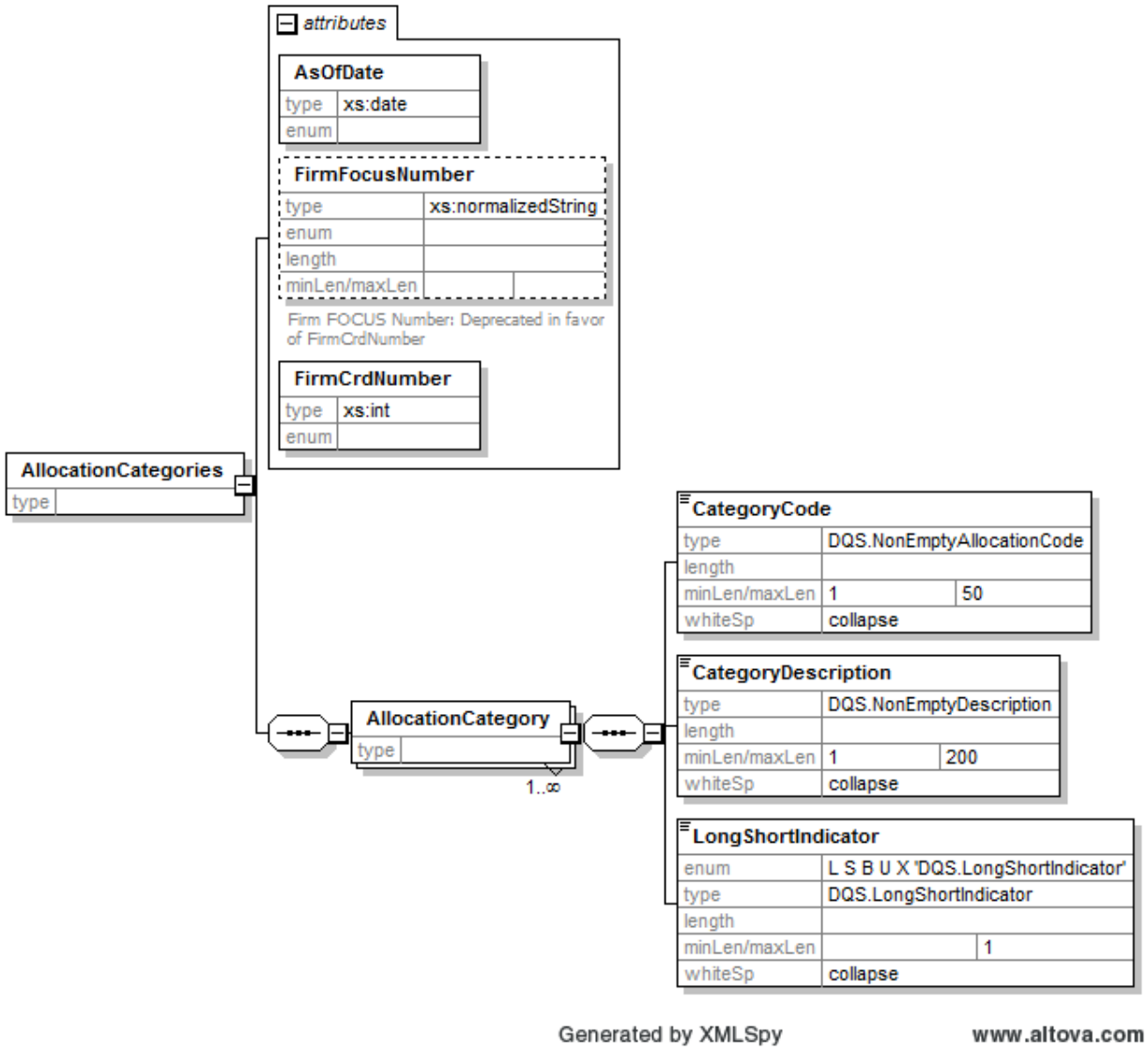
5.3 Allocation Categories (ACAT)

Definition: A file containing the list of classification codes used in the preparation of the SEA Rule 15c3-3 Allocation, along with their corresponding definitions.

FINRA is requesting the Allocation Categories utilized by the Firm in order to understand the Full Stock Record Allocation Process along with the Chart of Accounts. Note this information should be submitted as of the same date as the Full Stock Record data file.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	AllocationCategories	Container	Allocation Categories: This root container for the ACAT file contains a list of Allocation Category code and description definitions.
attribute	AllocationCategories/@AsOfDate	xs:date	As Of Date: The date of the requested data.
attribute	AllocationCategories/@FirmCrdNumber	xs:int	Firm CRD Number: Firm's Central Registration Depository (CRD) number.
element	AllocationCategories/AllocationCategory	Container	Allocation Category: The firm's classification code used as part of SEA Rule 15c3-3 Allocations, with its corresponding description. The Allocation Category Codes used in all of the allocation files (APOH, ACSM, APOS, SADT) must be defined in one of these elements.
element	AllocationCategories/AllocationCategory/CategoryCode	DQS.NonEmptyAllocationCode	Allocation Code: The firm's unique identifier for the allocation category. Maximum field length shall not exceed 50 characters.
element	AllocationCategories/AllocationCategory/CategoryDescription	DQS.NonEmptyDescription	Allocation Description: The firm's description for the corresponding allocation category code. Examples include: "Firm Long", "Firm Short", "DVP Long", "RVP Short", etc. Maximum field length shall not exceed 200 characters.
element	AllocationCategories/AllocationCategory/LongShortIndicator	DQS.LongShortIndicator	Long (L) or Short (S) Allocation Code Indicator Type: identifies a position as long or short in the allocation files. New values available as of V6: B (Both - code may be used as Long and Short), U (Unallocated - special code for use by the very few firms that explicitly identify unallocated positions, often using 99*, ZZ*, or XX* Allocation Category Codes. Those "unallocated" Allocation Category Codes should be identified by setting the LongShortIndicator to "U".), X (SR Position to be Left Out of Allocation Process - special code for use by the few firms that explicitly identify SR positions that should be left out of the allocation pairoff process. Those SR positions should be assigned an Allocation Category Code whose LongShortIdndicator is "X".)

The image, below, provides the same information shown in the tabular layout, above; for some readers, the diagram may offer a clearer picture of the file content. This image is available as a separate .png file (see http://www.finra.org/finops/stockrecord), which can be viewed using most image tools and/or using a browser. (This may be useful for viewing images that do not show up well on the one page display available within this document.)



5.4 Allocation Hierarchy (Allocation Pair Off) – (APOH)

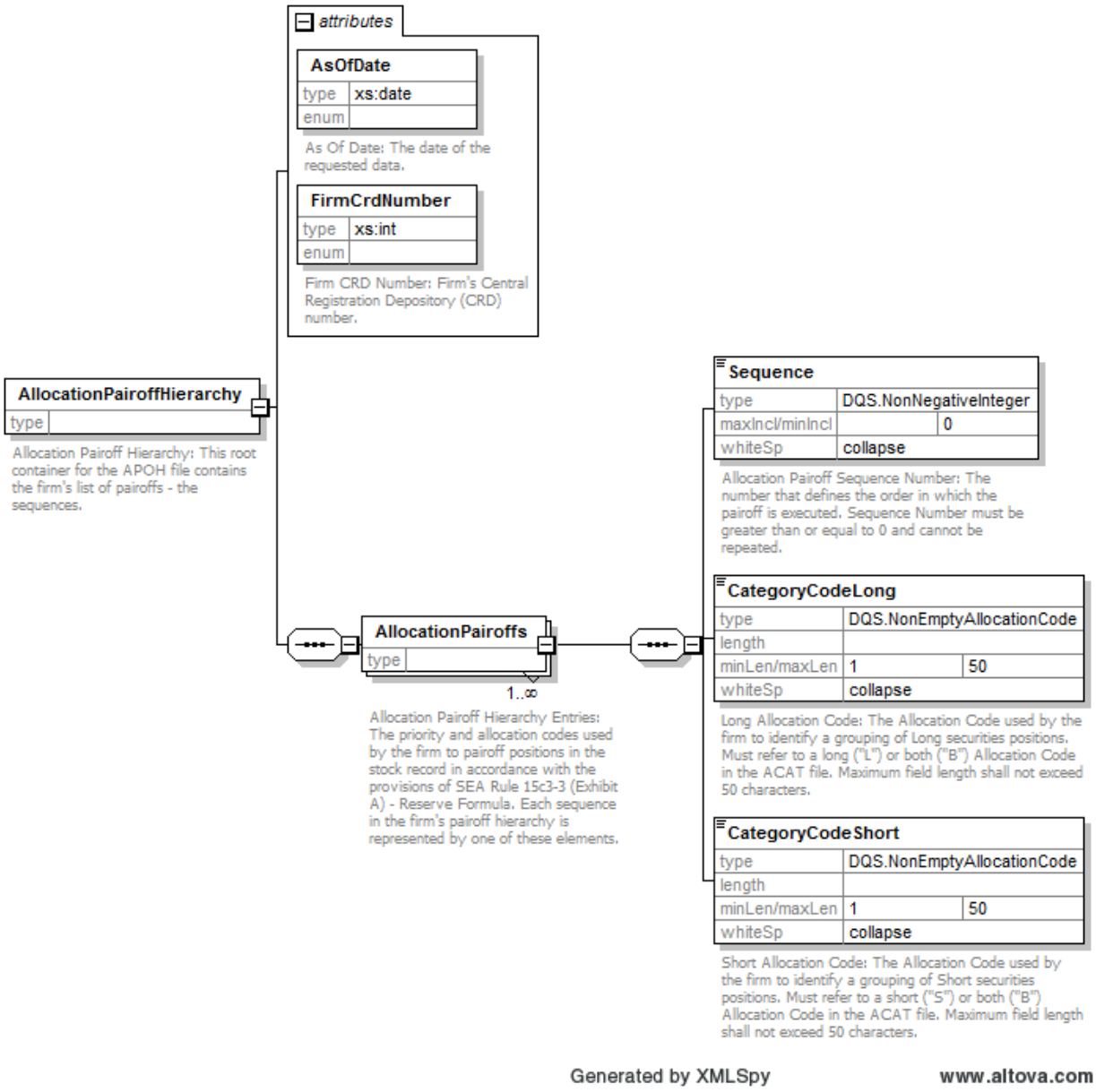
Definition: File describing the priority order used by the firm to pair off positions within the stock record in accordance with the provisions of SEA Rule 15c3-3 (Exhibit A) – Reserve Formula.

FINRA is requesting the Allocation Hierarchy (Allocation Pair Off) utilized by the Firm in order to understand the Full Stock Record Allocation Process. Note this information should be submitted as of the same date as the Full Stock Record data file.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	AllocationPairoffHierarchy	Container	Allocation Pairoff Hierarchy: This root container for the APOH file contains the firm's list of pairoffs - the sequences.
attribute	AllocationPairoffHierarchy/@AsOfDate	xs:date	As Of Date: The date of the requested data.
attribute	AllocationPairoffHierarchy/@FirmCrdNumber	xs:int	Firm CRD Number: Firm's Central Registration Depository

			(CRD) number.
element	AllocationPairoffHierarchy/AllocationPairoffs	Container	Allocation Pairoff Hierarchy Entries: The priority and allocation codes used by the firm to pairoff positions in the stock record in accordance with the provisions of SEA Rule 15c3-3 (Exhibit A) - Reserve Formula. Each sequence in the firm's pairoff hierarchy is represented by one of these elements.
element	AllocationPairoffHierarchy/AllocationPairoffs/Sequence	DQS.NonNegativeInteger	Allocation Pairoff Sequence Number: The number that defines the order in which the pairoff is executed. Sequence Number must be greater than or equal to 0 and cannot be repeated.
element	AllocationPairoffHierarchy/AllocationPairoffs/CategoryCodeLong	DQS.NonEmptyAllocationCode	Long Allocation Code: The Allocation Code used by the firm to identify a grouping of Long securities positions. Must refer to a long ("L") or both ("B") Allocation Code in the ACAT file. Maximum field length shall not exceed 50 characters.
element	AllocationPairoffHierarchy/AllocationPairoffs/CategoryCodeShort	DQS.NonEmptyAllocationCode	Short Allocation Code: The Allocation Code used by the firm to identify a grouping of Short securities positions. Must refer to a short ("S") or both ("B") Allocation Code in the ACAT file. Maximum field length shall not exceed 50 characters.

The image, below, provides the same information shown in the tabular layout, above; for some readers, the diagram may offer a clearer picture of the file content. This image is available as a separate .png file (see http://www.finra.org/finops/stockrecord), which can be viewed using most image tools and/or using a browser. (This may be useful for viewing images that do not show up well on the one page display available within this document.)



5.5 Allocation Category Summary (ACSM)

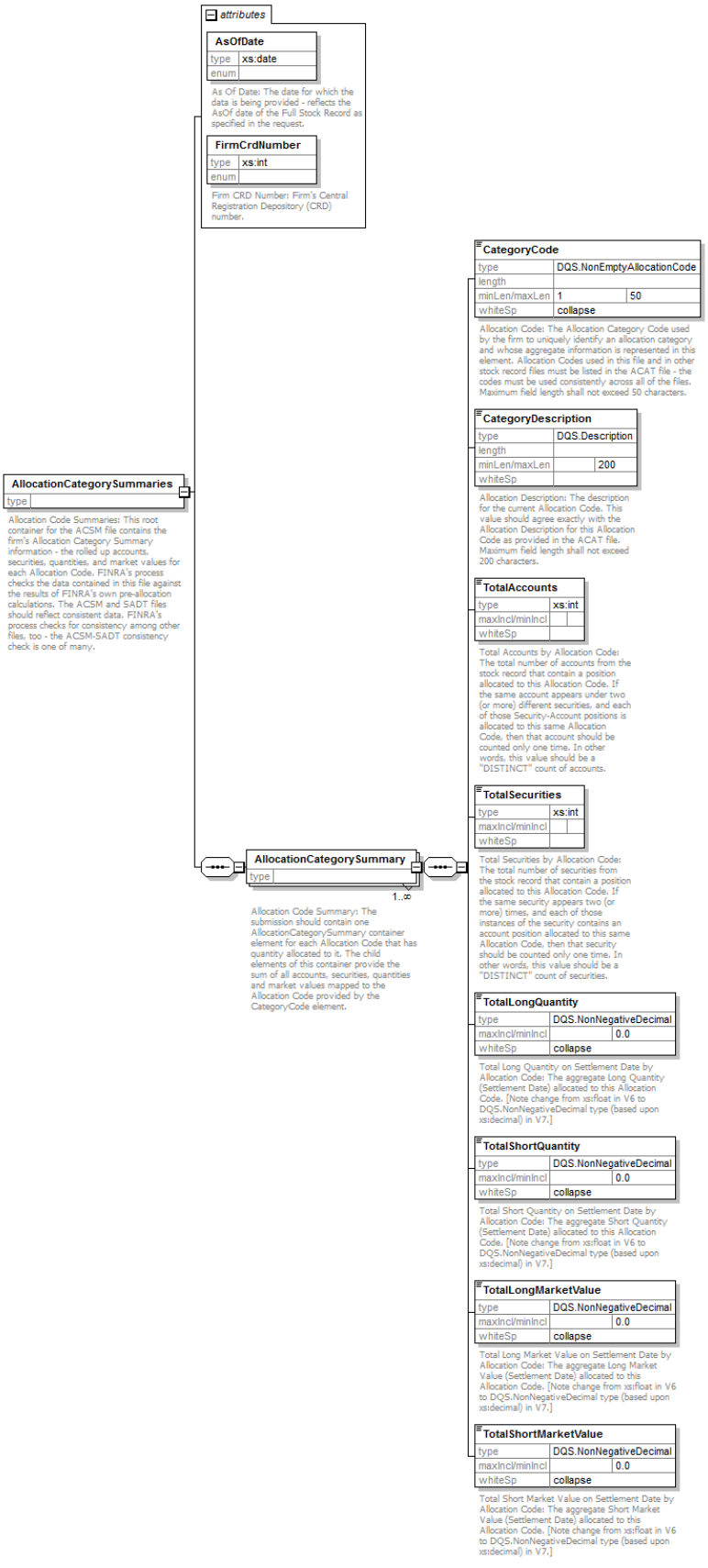
Definition: File that provides the total of all accounts, securities, quantities, and values by individual Allocation Category (e.g., Customer Long, Securities Borrows, Fails to Deliver, Box, etc.)

FINRA is requesting the Allocation Category Summary utilized by the Firm in order to understand the Full Stock Record Allocation Process. Note: this information should be submitted as of the same date as the Full Stock Record data file.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	AllocationCategorySummaries	Container	Allocation Code Summaries: This root container for the ACSM file contains the firm's Allocation Category Summary information - the rolled up accounts, securities, quantities, and market values for each Allocation Code. FINRA's process checks the data contained in this file

			against the results of FINRA's own pre-allocation calculations. The ACSM and SADT files should reflect consistent data. FINRA's process checks for consistency among other files, too - the ACSM-SADT consistency check is one of many.
attribute	AllocationCategorySummaries/@AsOfDate	xs:date	As Of Date: The date for which the data is being provided - reflects the AsOf date of the Full Stock Record as specified in the request.
attribute	AllocationCategorySummaries/@FirmCrdNumber	xs:int	Firm CRD Number: Firm's Central Registration Depository (CRD) number.
element	AllocationCategorySummaries/AllocationCategorySummary	Container	Allocation Code Summary: The submission should contain one AllocationCategorySummary container element for each Allocation Code that has quantity allocated to it. The child elements of this container provide the sum of all accounts, securities, quantities and market values mapped to the Allocation Code provided by the CategoryCode element.
element	AllocationCategorySummaries/AllocationCategorySummary/CategoryCode	DQS.NonEmptyAllocationCode	Allocation Code: The Allocation Category Code used by the firm to uniquely identify an allocation category and whose aggregate information is represented in this element. Allocation Codes used in this file and in other stock record files must be listed in the ACAT file - the codes must be used consistently across all of the files. Maximum field length shall not exceed 50 characters.
element	AllocationCategorySummaries/AllocationCategorySummary/CategoryDescription	DQS.Description	Allocation Description: The description for the current Allocation Code. This value should agree exactly with the Allocation Description for this Allocation Code as provided in the ACAT file. Maximum field length shall not exceed 200 characters.
element	AllocationCategorySummaries/AllocationCategorySummary/TotalAccounts	xs:int	Total Accounts by Allocation Code: The total number of accounts from the stock record that contain a position allocated to this Allocation Code. If the same account appears under two (or more) different securities, and each of those Security-Account positions is allocated to this same Allocation Code, then that account should be counted only one time. In other words, this value should be a "DISTINCT" count of accounts.
element	AllocationCategorySummaries/AllocationCategorySummary/TotalSecurities	xs:int	Total Securities by Allocation Code: The total number of securities from the stock record that contain a position allocated to this Allocation Code. If the same security appears two (or more) times, and each of those instances of the security contains an account position allocated to this same Allocation Code, then that security should be counted only one time. In other words, this value should be a "DISTINCT" count of securities.
element	AllocationCategorySummaries/AllocationCategorySummary/TotalLongQuantity	DQS.NonNegativeDecimal	Total Long Quantity on Settlement Date by Allocation Code: The aggregate Long Quantity (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
element	AllocationCategorySummaries/AllocationCategorySummary/TotalShortQuantity	DQS.NonNegativeDecimal	Total Short Quantity on Settlement Date by Allocation Code: The aggregate Short Quantity (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
element	AllocationCategorySummaries/AllocationCategorySummary/TotalLongMarketValue	DQS.NonNegativeDecimal	Total Long Market Value on Settlement Date by Allocation Code: The aggregate Long Market Value (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
element	AllocationCategorySummaries/AllocationCategorySummary/TotalShortMarketValue	DQS.NonNegativeDecimal	Total Short Market Value on Settlement Date by Allocation Code: The aggregate Short Market Value (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]

The image, below, provides the same information shown in the tabular layout, above; for some readers, the diagram may offer a clearer picture of the file content. This image is available as a separate .png file (see http://www.finra.org/finops/stockrecord), which can be viewed using most image tools and/or using a browser. (This may be useful for viewing images that do not show up well on the one page display available within this document.)


attributes
AsOfDate
type xs:date
enum
As Of Date: The date for which the data is being provided - reflects the AsOf date of the Full Stock Record as specified in the request.
FirmCrdNumber
type xs:int
enum
Firm CRD Number: Firm's Central Registration Depository (CRD) number.
AllocationCategorySummaries
type
Allocation Code Summaries: This root container for the ACSM file contains the firm's Allocation Category Summary information - the rolled up accounts, securities, quantities, and market values for each Allocation Code. FINRA's process checks the data contained in this file against the results of FINRA's own pre-allocation calculations. The ACSM and SADT files should reflect consistent data. FINRA's process checks for consistency among other files, too - the ACSM-SADT consistency check is one of many.
AllocationCategorySummary
type
1..∞
Allocation Code Summary: The submission should contain one AllocationCategorySummary container element for each Allocation Code that has quantity allocated to it. The child elements of this container provide the sum of all accounts, securities, quantities and market values mapped to the Allocation Code provided by the CategoryCode element.
CategoryCode
type DQS.NonEmptyAllocationCode
length
minLen/maxLen 1 50
whiteSp collapse
Allocation Code: The Allocation Category Code used by the firm to uniquely identify an allocation category and whose aggregate information is represented in this element. Allocation Codes used in this file and in other stock record files must be listed in the ACAT file - the codes must be used consistently across all of the files. Maximum field length shall not exceed 50 characters.
CategoryDescription
type DQS.Description
length
minLen/maxLen 200
whiteSp
Allocation Description: The description for the current Allocation Code. This value should agree exactly with the Allocation Description for this Allocation Code as provided in the ACAT file. Maximum field length shall not exceed 200 characters.
TotalAccounts
type xs:int
maxIncl/minIncl
whiteSp
Total Accounts by Allocation Code: The total number of accounts from the stock record that contain a position allocated to this Allocation Code. If the same account appears under two (or more) different securities, and each of those Security-Account positions is allocated to this same Allocation Code, then that account should be counted only one time. In other words, this value should be a "DISTINCT" count of accounts.
TotalSecurities
type xs:int
maxIncl/minIncl
whiteSp
Total Securities by Allocation Code: The total number of securities from the stock record that contain a position allocated to this Allocation Code. If the same security appears two (or more) times, and each of those instances of the security contains an account position allocated to this same Allocation Code, then that security should be counted only one time. In other words, this value should be a "DISTINCT" count of securities.
TotalLongQuantity
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Total Long Quantity on Settlement Date by Allocation Code: The aggregate Long Quantity (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
TotalShortQuantity
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Total Short Quantity on Settlement Date by Allocation Code: The aggregate Short Quantity (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
TotalLongMarketValue
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Total Long Market Value on Settlement Date by Allocation Code: The aggregate Long Market Value (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
TotalShortMarketValue
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Total Short Market Value on Settlement Date by Allocation Code: The aggregate Short Market Value (Settlement Date) allocated to this Allocation Code. [Note change from xs:float in V6 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V7.]
Generated by XMLSpy
www.altova.com

5.6 Allocation Pair off Summary (APOS)

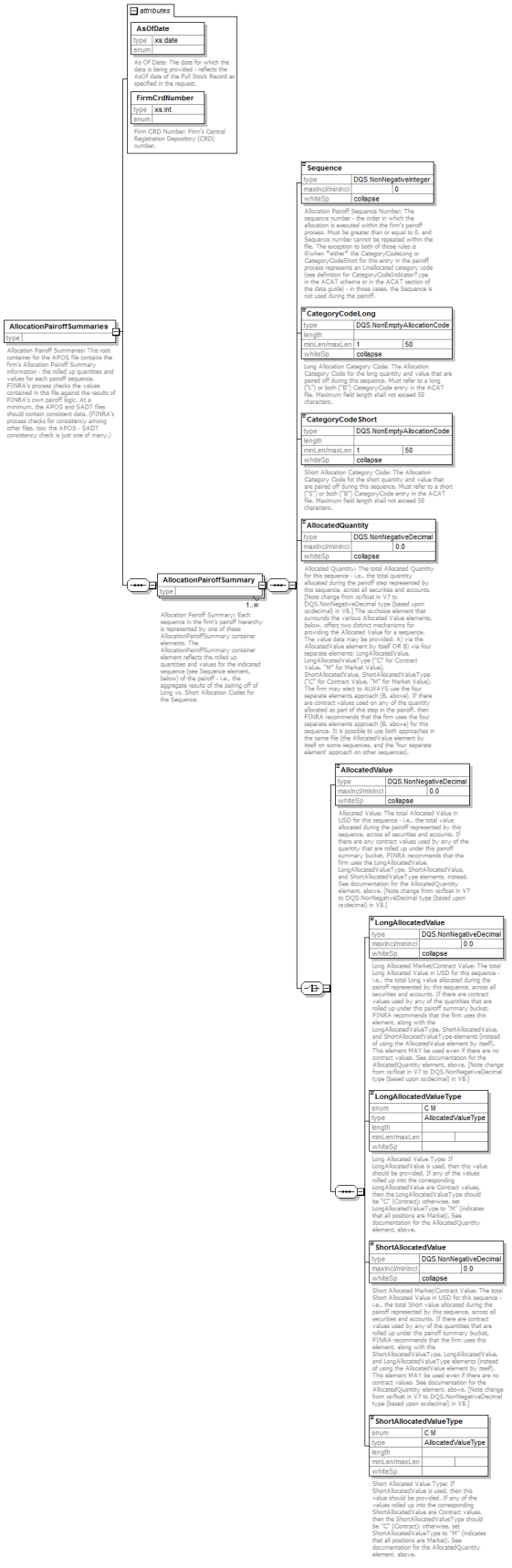
Definition: File that contains a summary of the long and short pairing of securities by Allocation Category (e.g., Customer Long vs. Securities Loaned).

FINRA is requesting the Allocation Pair Off Summary utilized by the Firm in order to understand the Full Stock Record Allocation process, and the Allocation Category Summary data. Note: this information should be submitted as of the same date as the Full Stock Record data file.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	AllocationPairoffSummaries	Container	Allocation Pairoff Summaries: This root container for the APOS file contains the firm's Allocation Pairoff Summary information - the rolled up quantities and values for each pairoff sequence. FINRA's process checks the values contained in this file against the results of FINRA's own pairoff logic. At a minimum, the APOS and SADT files should contain consistent data. (FINRA's process checks for consistency among other files, too; the APOS - SADT consistency check is just one of many.)
attribute	AllocationPairoffSummaries/@AsOfDate	xs:date	As Of Date: The date for which the data is being provided - reflects the AsOf date of the Full Stock Record as specified in the request.
attribute	AllocationPairoffSummaries/@FirmCrdNumber	xs:int	Firm CRD Number: Firm's Central Registration Depository (CRD) number.
element	AllocationPairoffSummaries/AllocationPairoffSummary	Container	Allocation Pairoff Summary: Each sequence in the firm's pairoff hierarchy is represented by one of these AllocationPairoffSummary container elements. The AllocationPairoffSummary container element reflects the rolled up quantities and values for the indicated sequence (see Sequence element, below) of the pairoff - i.e., the aggregate results of the pairing off of Long vs. Short Allocation Codes for the Sequence.
element	AllocationPairoffSummaries/AllocationPairoffSummary/Sequence	DQS.NonNegativeInteger	Allocation Pairoff Sequence Number: The sequence number - the order in which the allocation is executed within the firm's pairoff process. Must be greater than or equal to 0, and Sequence number cannot be repeated within the file. The exception to both of those rules is if/when *either* the CategoryCodeLong or CategoryCodeShort for this entry in the pairoff process represents an Unallocated category code (see definition for CategoryCodeIndicatorType in the ACAT schema or in the ACAT section of the data guide) - in those cases, the Sequence is not used during the pairoff.
element	AllocationPairoffSummaries/AllocationPairoffSummary/CategoryCodeLong	DQS.NonEmptyAllocationCode	Long Allocation Category Code: The Allocation Category Code for the long quantity and value that are paired off during this sequence. Must refer to a long ("L") or both ("B") CategoryCode entry in the ACAT file. Maximum field length shall not exceed 50 characters.
element	AllocationPairoffSummaries/AllocationPairoffSummary/CategoryCodeShort	DQS.NonEmptyAllocationCode	Short Allocation Category Code: The Allocation Category Code for the short quantity and value that are paired off during this sequence. Must refer to a short ("S") or both ("B") CategoryCode entry in the ACAT file. Maximum field length shall not exceed 50 characters.
element	AllocationPairoffSummaries/AllocationPairoffSummary/AllocatedQuantity	DQS.NonNegativeDecimal	Allocated Quantity: The total Allocated Quantity for this sequence - i.e., the total quantity allocated during the pairoff step represented by this sequence, across all securities and accounts. [Note change from xs:float in V7 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V8.] The xs:choice element that surrounds the various Allocated Value elements, below, offers two distinct mechanisms for providing the Allocated Value for a sequence. The value data may be provided: A) via the AllocatedValue element by itself OR B) via four separate elements: LongAllocatedValue, LongAllocatedValueType ("C" for Contract Value, "M" for Market Value), ShortAllocatedValue, ShortAllocatedValueType ("C" for Contract Value, "M" for Market Value). The firm may elect to ALWAYS use the four separate elements approach (B, above). If there are contract values used on any of the

			quantity allocated as part of this step in the pairoff, then FINRA recommends that the firm uses the four separate elements approach (B, above) for this sequence. It is possible to use both approaches in the same file (the AllocatedValue element by itself on some sequences, and the 'four separate element' approach on other sequences).
element	AllocationPairoffSummaries/AllocationPairoffSummary/AllocatedValue	DQS.NonNegativeDecimal	Allocated Value: The total Allocated Value in USD for this sequence - i.e., the total value allocated during the pairoff represented by this sequence, across all securities and accounts. If there are any contract values used by any of the quantity that are rolled up under this pairoff summary bucket, FINRA recommends that the firm uses the LongAllocatedValue, LongAllocatedValueType, ShortAllocatedValue, and ShortAllocatedValueType elements, instead. See documentation for the AllocatedQuantity element, above. [Note change from xs:float in V7 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V8.]
element	AllocationPairoffSummaries/AllocationPairoffSummary/LongAllocatedValue	DQS.NonNegativeDecimal	Long Allocated Market/Contract Value: The total Long Allocated Value in USD for this sequence - i.e., the total Long value allocated during the pairoff represented by this sequence, across all securities and accounts. If there are contract values used by any of the quantities that are rolled up under this pairoff summary bucket, FINRA recommends that the firm uses this element, along with the LongAllocatedValueType, ShortAllocatedValue, and ShortAllocatedValueType elements (instead of using the AllocatedValue element by itself). This element MAY be used even if there are no contract values. See documentation for the AllocatedQuantity element, above. [Note change from xs:float in V7 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V8.]
element	AllocationPairoffSummaries/AllocationPairoffSummary/LongAllocatedValueType	AllocatedValueType	Long Allocated Value Type: If LongAllocatedValue is used, then this value should be provided. If any of the values rolled up into the corresponding LongAllocatedValue are Contract values, then the LongAllocatedValueType should be "C" (Contract); otherwise, set LongAllocatedValueType to "M" (indicates that all positions are Market). See documentation for the AllocatedQuantity element, above.
element	AllocationPairoffSummaries/AllocationPairoffSummary/ShortAllocatedValue	DQS.NonNegativeDecimal	Short Allocated Market/Contract Value: The total Short Allocated Value in USD for this sequence - i.e., the total Short value allocated during the pairoff represented by this sequence, across all securities and accounts. If there are contract values used by any of the quantities that are rolled up under this pairoff summary bucket, FINRA recommends that the firm uses this element, along with the ShortAllocatedValueType, LongAllocatedValue, and LongAllocatedValueType elements (instead of using the AllocatedValue element by itself). This element MAY be used even if there are no contract values. See documentation for the AllocatedQuantity element, above. [Note change from xs:float in V7 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V8.]
element	AllocationPairoffSummaries/AllocationPairoffSummary/ShortAllocatedValueType	AllocatedValueType	Short Allocated Value Type: If ShortAllocatedValue is used, then this value should be provided. If any of the values rolled up into the corresponding ShortAllocatedValue are Contract values, then the ShortAllocatedValueType should be "C" (Contract); otherwise, set ShortAllocatedValueType to "M" (indicates that all positions are Market). See documentation for the AllocatedQuantity element, above.

The image, below, provides the same information shown in the tabular layout, above; for some readers, the diagram may offer a clearer picture of the file content. This image is available as a separate .png file (see http://www.finra.org/finops/stockrecord), which can be viewed using most image tools and/or using a browser. (This may be useful for viewing images that do not show up well on the one page display available within this document.)


attributes
AsOfDate
type xs:date
enum
As Of Date: The date for which the data is being provided - reflects the AsOf date of the Full Stock Record as specified in the request.
FirmCrdNumber
type xs:int
enum
Firm CRD Number: Firm's Central Registration Depository (CRD) number.
AllocationPairoffSummaries
type
Allocation Pairoff Summaries: This root container for the APOS file contains the firm's Allocation Pairoff Summary information - the rolled up quantities and values for each pairoff sequence. FINRA's process checks the values contained in this file against the results of FINRA's own pairoff logic. At a minimum, the APOS and SADT files should contain consistent data. (FINRA's process checks for consistency among other files, too: the APOS - SADT consistency check is just one of many.)
AllocationPairoffSummary
type
1..∞
Allocation Pairoff Summary: Each sequence in the firm's pairoff hierarchy is represented by one of these AllocationPairoffSummary container elements. The AllocationPairoffSummary container element reflects the rolled up quantities and values for the indicated sequence (see Sequence element, below) of the pairoff - i.e., the aggregate results of the pairing off of Long vs. Short Allocation Codes for the Sequence.
Sequence
type DQS.NonNegativeInteger
maxIncl/minIncl 0
whiteSp collapse
Allocation Pairoff Sequence Number: The sequence number - the order in which the allocation is executed within the firm's pairoff process. Must be greater than or equal to 0, and Sequence number cannot be repeated within the file. The exception to both of those rules is if/when "either" the CategoryCodeLong or CategoryCodeShort for this entry in the pairoff process represents an Unallocated category code (see definition for CategoryCodeIndicatorType in the ACAT schema or in the ACAT section of the data guide) - in those cases, the Sequence is not used during the pairoff.
CategoryCodeLong
type DQS.NonEmptyAllocationCode
length
minLen/maxLen 1 50
whiteSp collapse
Long Allocation Category Code: The Allocation Category Code for the long quantity and value that are paired off during this sequence. Must refer to a long ("L") or both ("B") CategoryCode entry in the ACAT file. Maximum field length shall not exceed 50 characters.
CategoryCodeShort
type DQS.NonEmptyAllocationCode
length
minLen/maxLen 1 50
whiteSp collapse
Short Allocation Category Code: The Allocation Category Code for the short quantity and value that are paired off during this sequence. Must refer to a short ("S") or both ("B") CategoryCode entry in the ACAT file. Maximum field length shall not exceed 50 characters.
AllocatedQuantity
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Allocated Quantity: The total Allocated Quantity for this sequence - i.e., the total quantity allocated during the pairoff step represented by this sequence, across all securities and accounts. [Note change from xsi:float in V7 to DQS.NonNegativeDecimal type [based upon xsi:decimal] in V8.] The xs:choice element that surrounds the various Allocated Value elements, below, offers two distinct mechanisms for providing the Allocated Value for a sequence. The value data may be provided: A) via the AllocatedValue element by itself OR B) via four separate elements: LongAllocatedValue, LongAllocatedValueType ("C" for Contract Value, "M" for Market Value), ShortAllocatedValue, ShortAllocatedValueType ("C" for Contract Value, "M" for Market Value). The firm may elect to ALWAYS use the four separate elements approach (B, above). If there are contract values used on any of the quantity allocated as part of this step in the pairoff, then FINRA recommends that the firm uses the four separate elements approach (B, above) for this sequence. It is possible to use both approaches in the same file (the AllocatedValue element by itself on some sequences, and the four separate element approach on other sequences).
AllocatedValue
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Allocated Value: The total Allocated Value in USD for this sequence - i.e., the total value allocated during the pairoff represented by this sequence, across all securities and accounts. If there are any contract values used by any of the quantity that are rolled up under this pairoff summary bucket, FINRA recommends that the firm uses the LongAllocatedValue, LongAllocatedValueType, ShortAllocatedValue, and ShortAllocatedValueType elements. Instead. See documentation for the AllocatedQuantity element, above. [Note change from xsi:float in V7 to DQS.NonNegativeDecimal type (based upon xsi:decimal) in V8.]
LongAllocatedValue
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Long Allocated Market/Contract Value: The total Long Allocated Value in USD for this sequence - i.e., the total Long value allocated during the pairoff represented by this sequence, across all securities and accounts. If there are contract values used by any of the quantities that are rolled up under this pairoff summary bucket, FINRA recommends that the firm uses this element, along with the LongAllocatedValueType, ShortAllocatedValue, and ShortAllocatedValueType elements (instead of using the AllocatedValue element by itself). This element MAY be used even if there are no contract values. See documentation for the AllocatedQuantity element, above. [Note change from xsi:float in V7 to DQS.NonNegativeDecimal type (based upon xsi:decimal) in V8.]
LongAllocatedValueType
enum C M
type AllocatedValueType
length
minLen/maxLen
whiteSp
Long Allocated Value Type: If LongAllocatedValue is used, then this value should be provided. If any of the values rolled up into the corresponding LongAllocatedValue are Contract values, then the LongAllocatedValueType should be "C" (Contract); otherwise, set LongAllocatedValueType to "M" (indicates that all positions are Market). See documentation for the AllocatedQuantity element, above.
ShortAllocatedValue
type DQS.NonNegativeDecimal
maxIncl/minIncl 0.0
whiteSp collapse
Short Allocated Market/Contract Value: The total Short Allocated Value in USD for this sequence - i.e., the total Short value allocated during the pairoff represented by this sequence, across all securities and accounts. If there are contract values used by any of the quantities that are rolled up under this pairoff summary bucket, FINRA recommends that the firm uses this element, along with the ShortAllocatedValueType, LongAllocatedValue, and LongAllocatedValueType elements (instead of using the AllocatedValue element by itself). This element MAY be used even if there are no contract values. See documentation for the AllocatedQuantity element, above. [Note change from xsi:float in V7 to DQS.NonNegativeDecimal type (based upon xsi:decimal) in V8.]
ShortAllocatedValueType
enum C M
type AllocatedValueType
length
minLen/maxLen
whiteSp
Short Allocated Value Type: If ShortAllocatedValue is used, then this value should be provided. If any of the values rolled up into the corresponding ShortAllocatedValue are Contract values, then the ShortAllocatedValueType should be "C" (Contract); otherwise, set ShortAllocatedValueType to "M" (indicates that all positions are Market). See documentation for the AllocatedQuantity element, above.
Generated by XMLSpy
www.altova.com

5.7 Security Allocation Details (SADT)

Definition: File that identifies how each security on the Full Stock Record is allocated according to the Firm's Allocation Hierarchy.

FINRA is requesting the Security Allocation Details utilized by the Firm in order to understand the Full Stock Record Allocation process, the Allocation Category Summary, and the Allocation Pair Off Summary data. Note this information should be submitted as of the same date as of the Full Stock Record data file.

Attribute or Element	Attribute or Element Name (XPath)	Type	Description
element	SecurityAllocationDetails	Container	Security Allocation Details: This root container for the SADT file contains the firm's Security Allocation Details information. The data in this file describes how each security on the firm's full Stock Record was allocated, based upon the firm's Allocation Hierarchy. FINRA's process checks the values contained in this file against the results of FINRA's own pairoff logic. At a minimum, the APOS, SADT, and ACSM files should contain consistent data. (FINRA's process checks for consistency among other files, too; the APOS/SADT/ACSM consistency checks are just a few of the many checks performed.)
attribute	SecurityAllocationDetails/@AsOfDate	xs:date	As Of Date: The date for which the data is being provided - reflects the AsOf date of the Full Stock Record as specified in the request.
attribute	SecurityAllocationDetails/@FirmCrdNumber	xs:int	Firm CRD Number: Firm's Central Registration Depository (CRD) number.
element	SecurityAllocationDetails/FileControl	Container	File Control Data: The attributes associated with this element provide some metadata surrounding the SADT files submitted - number of files, XSD version used, etc. See individual attributes, below, for details.
attribute	SecurityAllocationDetails/FileControl/@FileNumber	xs:int	File Number: For submissions that contain multiple SADT files, this value represents the index of the current file within that set of multiple files. For example, the second of ten files would have a "2" as the value of this attribute. If there is only one SADT file in the submission, then this attribute should be set to "1".
attribute	SecurityAllocationDetails/FileControl/@FileTotal	xs:int	Total Number Files: For submissions that contain multiple SADT files, this value represents the total number of SADT files in the submission. If there is only one SADT file in the submission, then this attribute should be set to "1".
attribute	SecurityAllocationDetails/FileControl/@Version	xs:normalizedString	Schema Version: Should reflect the schema version used for this submission - may provide either the entire name of the schema document (i.e., "AEP_FinOPs_SADT_V9.xsd" or just the numeric portion (i.e., "9").
element	SecurityAllocationDetails/SecurityAllocationDetail	Container	Security Allocation Detail: Provides details regarding how each security on the full Stock Record is allocated according to the firm's allocation hierarchy. Each security that has some quantity paired off during a step in the pairoff process is represented here - there will be one of these elements for each security and sequence. In other words, if security ABC quantity is "used" during pairoff sequences 12 and 84, then there will be one entry here for security ABC/sequence 12, and another entry for security ABC/sequence 84. The element reflects the quantities and values paired off for the indicated security during the relevant pairoff step (i.e., sequence).
element	SecurityAllocationDetails/SecurityAllocationDetail/Cusip	xs:normalizedString	CUSIP: Committee on Uniform Securities Identification Procedures number of the security or registered bond. See important notes under SecurityId element, below.
element	SecurityAllocationDetails/SecurityAllocationDetail/SecurityId	DQS.SecurityId	Security ID: Firm's internal security identifier for the relevant Security, if applicable. In order for FINRA's process to link securities across SR and SADT files, the SADT file must either a) provide SecurityId on every SADT record...or b) provide a NULL or no SecurityId element on every SADT record. In other words, it is an all-or-nothing data element - SecurityId must be present

			on every SADT record or on NO SADT records. If the firm provides SecurityId on every record, then FINRA's process will use SecurityId to link SR data to SADT data (and therefore, SecurityId information provided in the SR and SADT files must support that linking). If the firm leaves SecurityId off of (or provides a NULL SecurityId for) every record, then FINRA's process will use CUSIP to link SR data to SADT data (and, therefore, CUSIP information provided in the SR and SADT files must support that linking). Maximum field length shall not exceed 30 characters.
element	SecurityAllocationDetails/SecurityAllocationDetail/SecurityDescription	DQS.Description	Security Description: Complete security name; include coupon description where applicable. Should agree with SecurityDescription for each instrument as provided in the SR file. Maximum field length shall not exceed 200 characters.
element	SecurityAllocationDetails/SecurityAllocationDetail/Sequence	xs:int	Sequence Number: The order in which the pairoff is executed within the firm's pairoff process. Must be greater than or equal to 0, and Sequence Number cannot be repeated per security (i.e., CUSIP + Security ID). The exception to both of those rules is if/when either the CategoryCodeLong or CategoryCodeShort for this entry in the pairoff process represents an Unallocated category code (see the description for CategoryCodeIndicatorType in the ACAT schema or in the ACAT section of the data guide) - in those cases, the Sequence is not used during the pairoff.
element	SecurityAllocationDetails/SecurityAllocationDetail/CategoryCodeLong	DQS.NonEmptyAllocationCode	Long Allocation Category Code: The long side Allocation Category Code for the sequence and security. Must refer to a long ("L") or both ("B") CategoryCode entry in the ACAT file. Maximum field length shall not exceed 50 characters.
element	SecurityAllocationDetails/SecurityAllocationDetail/CategoryCodeShort	DQS.NonEmptyAllocationCode	Short Allocation Category Code: The short side Allocation Category Code for the sequence and security. Must refer to a short ("S") or both ("B") CategoryCode entry in the ACAT file. Maximum field length shall not exceed 50 characters.
element	SecurityAllocationDetails/SecurityAllocationDetail/LongQuantity	DQS.NonNegativeDecimal	Long Quantity: Reflects the quantity not-yet-allocated or available to be allocated for the combination of Security and Long Allocation Code (as specified in the CategoryCodeLong element, above) upon entry to this pairoff step. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V9.]
element	SecurityAllocationDetails/SecurityAllocationDetail/ShortQuantity	DQS.NonNegativeDecimal	Short Quantity: Reflects the quantity not-yet-allocated or available to be allocated for the combination of Security and Short Allocation Code (as specified in the CategoryCodeShort element, above) upon entry to this pairoff step. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V9.]
element	SecurityAllocationDetails/SecurityAllocationDetail/AllocatedQuantity	DQS.NonNegativeDecimal	Allocated Quantity by Sequence: The quantity of the security allocated during the pairoff step (sequence). [Note change from xs:double in V8 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V9.] The xs:choice element that surrounds the various Allocated Value elements, below, offers two distinct mechanisms for providing the Allocated Value for a sequence and security. The value data may be provided: A) via the AllocatedValue element by itself OR B) via four separate elements: LongAllocatedValue, LongAllocatedValueType ("C" for Contract Value, "M" for Market Value), ShortAllocatedValue, ShortAllocatedValueType ("C" / "M"). The firm may elect to ALWAYS use the four separate elements approach (B, above). If there are contract values used on any of the quantity allocated as part of the step for the security in the pairoff, then FINRA recommends that the firm uses the four separate elements approach (B, above) for the sequence/security. It is possible to use both approaches in the same file (the AllocatedValue element by itself on some sequences, and the 'four separate element' approach on other sequences).
element	SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValue	DQS.NonNegativeDecimal	Allocated Market Value by Sequence: The Allocated Market Value in USD for the sequence/security

			combination - i.e., the total value allocated during the pairoff represented by the sequence, for the security. If there are contract values reflected in the security/sequence bucket, then FINRA recommends that the firm uses the LongAllocatedValue, LongAllocatedValueType ("C" for Contract Value, "M" for Market Value), ShortAllocatedValue, and ShortAllocatedValueType ("C" / "M") for Market Value) elements, instead. See description for the AllocatedQuantity element, above. [Note change from xs:float in V8 to xs:decimal in V9.]
element	SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValueType	AllocValueType	Allocated Value Type by Sequence: If any of the values rolled up into the corresponding AllocatedValue are Contract values, then FINRA recommends that the firm uses the four separate elements approach (approach B as described under AllocatedQauntity, above). If the firm must use approach A and if any of the values rolled up into the corresponding AllocatedValue are Contract values, then the AllocatedValueType should be set to "C" (Contract); otherwise, set AllocatedValueType to "M" (indicates that all positions are Market). See description for AllocatedQuantity element, above.
element	SecurityAllocationDetails/SecurityAllocationDetail/LongAllocatedValue	DQS.NonNegativeDecimal	Long Allocated Market/Contract Value by Sequence: The Long Allocated Value in USD for the sequence/security combination - i.e., the total Long value allocated during the pairoff represented by the sequence, for the security. If there are contract values used by any of the quantities that are rolled up under the security/sequence bucket, FINRA recommends that the firm uses this element, along with the LongAllocatedValueType ("C" for Contract Value, "M" for Market Value), ShortAllocatedValue, and ShortAllocatedValueType elements (instead of using the AllocatedValue element by itself). This element MAY be used even if there are no contract values. See description for the AllocatedQuantity element, above. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V9.]
element	SecurityAllocationDetails/SecurityAllocationDetail/LongAllocatedValueType	AllocValueType	Long Allocated Value Type by Sequence: If any of the values rolled up into the corresponding LongAllocatedValue are Contract values, then the LongAllocatedValueType should be set to "C" (Contract); otherwise, set LongAllocatedValueType to "M" (indicates that all positions are Market). See description for the AllocatedQuantity element, above.
element	SecurityAllocationDetails/SecurityAllocationDetail/ShortAllocatedValue	DQS.NonNegativeDecimal	Short Allocated Market/Contract Value by Sequence: The Short Allocated Value in USD for the sequence/security combination - i.e., the total Short value allocated during the pairoff represented by the sequence, for the security. If there are contract values used by any of the quantities that are rolled up under the security/sequence bucket, FINRA recommends that the firm uses this element, along with the ShortAllocatedValueType ("C" for Contract Value, "M" for Market Value), LongAllocatedValue, and LongAllocatedValueType elements (instead of using the AllocatedValue element by itself). This element MAY be used even if there are no contract values. See documentation for the AllocatedQuantity element, above. [Note change from xs:float in V8 to DQS.NonNegativeDecimal type (based upon xs:decimal) in V9.]
element	SecurityAllocationDetails/SecurityAllocationDetail/ShortAllocatedValueType	AllocValueType	If ShortAllocatedValue is used, then this value should be provided. "C" (Contract) or "M" (Market). See documentation for the AllocatedQuantity element, above.

Please see http://www.finra.org/finops/stockrecord for the image file for the SADT. As mentioned, that image file provides a diagrammatic view of the SADT data structure. The diagram contains the same information as provided in the table, above; for some readers, the diagram may offer a clearer picture of the file content. The diagram can be viewed using most image tools and/or using a browser.

6 The Validations

Summary of Data Quality Scorecard (DQS) Validations

FINRA's ingest process enforces checks and validations on the Stock Record and Allocation files received from the firm- SR, SADT, APOH, APOS, ACSM and ACAT. By enforcing validations at the data file submission stage, FINRA is able to disqualify early in the process those data submissions that do not meet the specified criteria. This will also minimizes the manual data validation and analysis of errors by both FINRA and firm personnel.

The Data Quality Scorecard ("DQS") page shows the results of every validation performed during the ingest process. Any validation failures is accompanied by clear explanations as to the cause of the failure and examples of the invalid data. This process enables the firm to identify the errors, correct the erroneous data, and resubmit corrected files.

The DQS supports a variety of display sorting and grouping functions, which makes it easy to identify data issues. For more information, please see A Brief Preview of the DQS.

A Closer Look at the Validations

The data validations are divided into two classes: Hard and Soft. *Hard* validations are data integrity checks that are deemed critical for a successful file load. Hard validation exceptions will cause the processing of a file to fail. When this happens, the firm must correct the issues and resubmit the data. *Soft* validations are data integrity checks that require manual evaluation. Soft validation exceptions will not automatically cause the processing of the data file to fail. In some cases the firms will be required to correct and resubmit the data. In other cases, where the data issue represents a legitimate business scenario, the data may be allowed to stand. For more information, please see the Validations section, below.

Both Hard and Soft validations are performed at different points during the data ingest process. The first tier of validations focuses on making sure that each *individual* file contains all requisite data and is formatted correctly. Later validations check to ensure that data *between* files is in agreement ("cross-file validations"). If an individual file is found to be invalid, it usually does not make sense to continue on to the cross-file validations that utilize that file. The reason: those problems with the individual file would likely result in problems with the cross-file checks – and those cross-file errors would, in most cases, be "false failures" (i.e., the *real* problem was with the individual file, not with the cross-file check). The eAnalytics validation process is designed to minimize the possibility of these kinds of false failures – in the scenario under discussion, the ingest would stop and provide feedback on the individual file failures and *not* go on to the cross-file validations. Therefore, it is possible for an early Hard failure to "mask" a Hard failure later in the process – again, using this scenario, that problem with the individual file could mask the fact that there are additional (unrelated) problems with the cross-file checks. As a result, firms may have to correct and resubmit their data several times in order to identify and address all Hard failures.

Note that the eAnalytics platform is designed to "go as far as it can" when performing the validations – even if a Hard failure occurs, it will continue to run through as many additional validations as possible before it gets to a point where the kinds of false failures mentioned above become more likely. This allows the submitter to gain as much insight as possible into any issues with the data with each load attempt.

6.1 Validation Types

Validations are categorized by the following types:

1. **Hard Validations** – Hard validations are data integrity checks that are deemed critical for a successful load. Hard validations will cause the processing to fail. The firm must correct the issue[s] and resubmit the data.

   Examples of Hard validations include:
   - Missing required fields.
   - Use of an Allocation Category that is not found in the ACAT file.

2. **Soft Validations** – Soft Validations are data integrity checks that require manual evaluation. Soft validations will not automatically cause the processing to fail. In some cases the firms will be required to correct and resubmit the data. In other cases, where the data issue represents a legitimate business scenario, the data may be allowed to stand.

   Examples of Soft validations include:
   - Accounts where both Long and Short Position are zero.
   - Instances where the ACAT Allocation Category Description does not match the ACSM Allocation Category Description.

3. **XSD Validations** – XSD validations are handled by the XML schema. These validations can easily be performed by the Firm – see *XML Schema (XSD) and Firm-Side Validation* section of the Appendix.

4. **Defaults** – Defaults are performed by the system before validations are run.

   Examples of Defaults include:
   - Currency Code null or 'N/A' defaults to 'USD'.
   - CUSIP 'N/A' defaults to null.

6.2 List of Validations

Validation Req Nbr	Validation Name	DQS Section	DQS Label	Description	Validation Type	Last Updated Version
VAL-7	ACAT - Duplicate Allocation Code	ACAT	Duplicate Allocation Code	ACAT Allocation Code (CategoryCode) shall be unique for the file.	Hard	5.0
VAL-540	ACAT - Invalid Allocation Code Length	ACAT	Basic Validations	ACAT - Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-210	ACAT - Missing Allocation Code	ACAT	Basic Validations	ACAT Allocation Code (CategoryCode) shall not be null or blank.	XSD	5.0
VAL-8	ACAT - Invalid Long-Short Indicator	ACAT	Basic Validations	ACAT Long-Short Indicator shall be 'L', 'S', 'U', 'X' or 'B' (i.e., not null or blanks.)	XSD	5.1
VAL-541	ACAT - Invalid Allocation Description Length	ACAT	Basic Validations	ACAT - Allocation Description shall be less than or equal to 200.	XSD	5.8
VAL-211	ACAT - Missing Allocation Description	ACAT	Basic Validations	ACAT Allocation Description (Category Description) shall not be null or blank.	XSD	5.0

VAL-439	ACAT - ACAT Allocation Description Not Match ACSM	ACAT	ACAT Allocation Description Not Match ACSM	ACAT Allocation Description (CategoryDescription) shall match ACSM Allocation Description (CategoryDescription), ignoring any case differences.	Soft	5.0
VAL-3.1	ACAT - Invalid As Of Date	ACAT	Invalid As Of Date	ACAT As Of Date shall match the As Of Date for the Request, based on the Request ID in the file name.	Hard	5.0
VAL-2.1	ACAT - Invalid XML	ACAT	Invalid XML	ACAT XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4.9	ACAT - Invalid Firm CRD Number	ACAT	Invalid Firm CRD Number	ACAT Firm CRD number shall match the Firm CRD Number of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-37	ACSM - Allocation Code Not In ACAT	ACSM	Allocation Code Not In ACAT	ACSM Allocation Code (CategoryCode) shall exist in ACAT Allocation Code (CategoryCode) where Long-Short Indicator is not 'U' or 'X'.	Hard	5.5
VAL-542	ACSM - Invalid Allocation Code Length	ACSM	Basic Validations	ACSM - Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-318	ACSM - Missing Allocation Code	ACSM	Basic Validations	ACSM Allocation Code (CategoryCode) shall not be null or blank.	XSD	5.0
VAL-482	ACSM Duplicate Allocation Code	ACSM	Duplicate Allocation Code	ACSM Allocation Code (CategoryCode) shall be unique (i.e., one record per file).	Hard	5.0
VAL-333	APOS - Missing Long Allocation Code	ACSM	Basic Validations	APOS Long Allocation Code shall not be null or blank.	XSD	5.0
VAL-543	ACSM - Invalid Allocation Description Length	ACSM	Basic Validations	ACSM - Allocation Description shall be less than or equal to 200.	XSD	5.8
VAL-319	ACSM - Missing Allocation Description	ACSM	Missing Allocation Description	ACSM Allocation Description (CategoryDescription) shall not be null or blank.	Soft	5.0
VAL-3.2	ACSM - Invalid As Of Date	ACSM	Invalid As Of Date	ACSM As Of Date shall match the As of Date of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-2.2	ACSM - Invalid XML	ACSM	Invalid XML	ACSM XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4.8	ACSM - Invalid Firm CRD Number	ACSM	Invalid Firm CRD Number	ACSM Firm CRD Number shall match the Firm CRD Number of the Request ID, based on the file name.	Hard	5.0
VAL-441	Sum of SR Long Quantity on Settlement Date by Account Not Match ACSM	ACSM	SR Allocation Code or Allocated Sum Not Match ACSM	For each Allocation Code in SR or ACSM files, the sum of SR Long Quantity on Settlement Date by Account shall equal ACSM Total Long Quantity (i.e., implicitly checks for missing Allocation Codes in both files).	Hard	5.4
VAL-442	Sum of SR Short Quantity on Settlement Date by Account Not Match ACSM	ACSM	SR Allocation Code or Allocated Sum Not Match ACSM	For each Allocation Code in SR or ACSM files, the sum of SR Short Quantity on Settlement Date by Account shall equal ACSM Total Long Quantity (i.e., implicitly checks for missing Allocation Codes in both files).	Hard	5.4

VAL-35	ACSM - Total Accounts - Securities Mismatch	ACSM	Total Accounts - Securities Mismatch	A) If ACSM Total Accounts is greater than zero then ACSM Total Securities shall be greater than zero. B) If ACSM Total Securities is greater than zero then ACSM Total Accounts shall be greater than zero.	Soft	5.0
VAL-325	ACSM - Negative Or Non-Numeric Total Long Quantity	ACSM	Basic Validations	ACSM Total Long Quantity shall be non-negative numeric.	XSD	5.0
VAL-329	ACSM - Negative Or Non-Numeric Total Long Value	ACSM	Basic Validations	ACSM Total Long Value shall be non-negative numeric.	XSD	5.0
VAL-36	ACSM - Quantity - Market Value Mismatch (Long)	ACSM	Quantity - Market Value Mismatch	If ACSM Total Long Value is greater than zero then ACSM Total Long Quantity shall be greater than zero.	Soft	5.0
VAL-36.5	ACSM - Quantity - Market Value Mismatch (Short)	ACSM	Quantity - Market Value Mismatch	If ACSM Total Short Value is greater than zero then ACSM Total Short Quantity shall be greater than zero.	Soft	5.0
VAL-327	ACSM - Negative or Non-Numeric Total Short Quantity	ACSM	Basic Validations	ACSM Total Short Quantity shall be non-negative numeric.	XSD	5.0
VAL-331	ACSM - Negative or Non-Numeric Total Short Value	ACSM	Basic Validations	ACSM Total Short Value shall be non-negative numeric.	XSD	5.0
VAL-392	APOH - Allocation Code Not In ACAT	APOH	Allocation Code Not In ACAT	APOH Allocation Code (CategoryCode) shall exist in ACAT Allocation Code (CategoryCode) where Long-Short Indicator is not 'X'.	Hard	5.3
VAL-11	APOH - Duplicate Long and Short Allocation Code	APOH	Duplicate Long and Short Alloc Code	The combination of APOH Long Allocation Code (CategoryCodeLong) and Short Allocation Code (CategoryCodeShort) shall be unique for the submission.	Hard	5.0
VAL-471	APOH - Missing Long Allocation Code	APOH	Basic Validations	APOH Long Allocation Code (CategoryCodeLong) shall not be null or blank.	XSD	5.0
VAL-473	APOH - Missing Short Allocation Code	APOH	Basic Validations	APOH Short Allocation Code (CategoryCodeShort) shall not be null or blank.	XSD	5.0
VAL-9	APOH - Duplicate Allocation Pair Off Sequence	APOH	Duplicate Allocation Pair Off Sequence	APOH Allocation Pair Off Sequence Number (Sequence) shall be unique for the submission, unless one or both of the Allocation Codes for that sequence is defined as 'U' in ACAT.	Hard	5.3
VAL-223	APOH - Invalid Allocation Pair Off Sequence Number	APOH	Basic Validations	APOH Allocation Pair Off Sequence Number (Sequence) shall be numeric and greater than or equal to zero.	XSD	5.0
VAL-3.4	APOH - Invalid As Of Date	APOH	Invalid As Of Date	APOH As Of Date shall match the As of Date of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-2.3	APOH - Invalid XML	APOH	Invalid XML	APOH XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4.7	APOH - Invalid Firm CRD Number	APOH	Invalid Firm CRD Number	APOH Firm CRD Number shall match the Firm CRD Number of the Request, based on the Request ID in the file name.	Hard	5.0

VAL-544	APOH - Invalid Long Allocation Code Length	APOH	Basic Validations	APOH - Long Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-393	APOH - Allocation Code Indicator Not Match ACAT (Long)	APOH	Allocation Code Indicator Not Match ACAT	For an Allocation Code if an ACAT Long-Short Indicator equals 'L', then APOH Long-Short Indicator shall equal 'L' or 'B' or 'U' (i.e., not 'S' or 'X').	Hard	5.3
VAL-394	APOH - Allocation Code Indicator Not Match ACAT (Short)	APOH	Allocation Code Indicator Not Match ACAT	For an Allocation Code if an ACAT Long-Short Indicator equals 'S', then APOH Long-Short Indicator shall equal 'S' or 'B' or 'U' (i.e., not 'L' or 'X').	Hard	5.3
VAL-545	APOH - Invalid Short Allocation Code Length	APOH	Basic Validations	APOH - Short Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-335	APOS - Negative Or Non-Numeric Allocated Quantity	APOS	Basic Validations	APOS Allocated Quantity shall be non-negative numeric.	XSD	5.0
VAL-336	APOS - Negative Or Non-Numeric Allocated Value	APOS	Basic Validations	If APOS Allocated Value is not null, then APOS Allocated Value shall be non-negative numeric.	XSD	5.0
VAL-337	APOS - Negative Or Non-Numeric Long Allocated Value	APOS	Basic Validations	If APOS Long Allocated Value is not null, then APOS Allocated Value shall be non-negative numeric.	XSD	5.0
VAL-340	APOS - Negative Or Non-Numeric Short Allocated Value	APOS	Basic Validations	If APOS Short Allocated Value is not null, then APOS Short Allocated value shall be non-negative numeric.	XSD	5.0
VAL-74	APOS - Duplicate Allocation Code	APOS	Duplicate Allocation Code	Combination of APOS Long Allocation Code (CategoryCodeLong) and APOS Short Allocation Code (CategoryCodeShort) shall be unique for the file.	Hard	5.0
VAL-334	APOS - Missing Short Allocation Code	APOS	Basic Validations	APOS Short Allocation Code shall not be null or blank.	XSD	5.0
VAL-12	APOS - Duplicate Allocation Pair Off Sequence	APOS	Duplicate Allocation Pair Off Sequence	APOS Allocation Pair Off Sequence Number shall be unique for the file, except where Long Allocation Code or Short Allocation Code is Unallocated (U) as defined in ACAT.	Hard	5.0
VAL-220	APOS - Invalid Allocation Pair Off Sequence	APOS	Basic Validations	APOS Allocation Pair Off Sequence Number shall be numeric and greater than or equal to zero.	XSD	5.0
VAL-3.5	APOS - Invalid As Of Date	APOS	Invalid As Of Date	APOS As of Date shall match the As Of Date of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-2.8	APOS - Invalid XML	APOS	Invalid XML	APOS XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4.6	APOS - Invalid Firm CRD Number	APOS	Invalid Firm CRD Number	APOS Firm CRD Number shall match the Firm CRD Number of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-546	APOS - Invalid Long Allocation Code Length	APOS	Basic Validations	APOS - Long Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-452	APOS Sequence + Long Code + Short Code Not In APOH	APOS	APOS Sequence - Long Code - Short Code Not In APOH	The combination of APOS Allocation Pair Off Sequence Number, Long Allocation Code and Short Allocation Code shall exist in APOH file, except where Allocation Code is Unallocated (U).	Hard	5.0

VAL-547	APOS - Invalid Short Allocation Code Length	APOS	Basic Validations	APOS - Short Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-15	COA - Missing Account Number	COA	Missing or Ambiguous Account Number	A) If COA Specific Account Number is populated, then From Account Number shall be null. B) If COA From Account Number is populated, then Specific Account Number shall be null. C) COA Specific Account Number and COA From Account Number shall not both be null.	Soft	5.2
VAL-16	COA - Missing To/From Account Number	COA	Missing or Ambiguous Account Number	A) If COA From Account Number is populated, then To Account Number shall be populated. B) If COA To Account Number is populated, then From Account Number shall be populated.	Soft	5.2
VAL-3.6	COA - Invalid As Of Date	COA	Invalid As Of Date	COA As Of Date shall match the As Of Date of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-548	COA - Invalid Category Description Length	COA	Basic Validations	COA - Category Description length shall be less than or equal to 200.	XSD	5.8
VAL-302	COA - Missing Category Description	COA	Missing Category Description	COA Category Description shall not be null or blank.	Soft	5.0
VAL-2.4	COA - Invalid XML	COA	Invalid XML	COA XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4.5	COA - Invalid Firm CRD Number	COA	Invalid Firm CRD Number	COA Firm CRD Number shall match the Firm CRD Number of the Request, based on the Request ID of the file name.	Hard	5.0
VAL-549	COA - Invalid From Account Number Length	COA	Basic Validations	COA - From Account Number length shall be less than or equal to 100.	XSD	5.8
VAL-564	COA - Invalid Specific Account Number Length	COA	Basic Validations	COA - Specific Account Number length shall be less than or equal to 100.	XSD	5.8
VAL-550	COA - Invalid To Account Number Length	COA	Basic Validations	COA - To Account Number length shall be less than or equal to 100.	XSD	5.8
VAL-514	Firm Profile - Missing Account Type Mapping - Margin	FirmSettings	Missing Account Type Mapping - Margin	In the Firm Profile, at least one firm account shall be mapped to FINRA Account Type 'MARGIN'.	Soft	5.4
VAL-527	Firm Setings - Missing Allocation Files	FirmSettings	Missing Allocation Files	If any allocation files are submitted, then all allocation files must be submitted, including ACAT, SADT, APOS, APOH and ACSM.	Hard	5.5
VAL-526	Firm Settings - File Submission Error	FirmSettings	File Submission Error	All files shall be submitted without error to be processed.	Hard	5.5
VAL-525	Firm Settings - No SR Files Submitted	FirmSettings	No SR Files Submitted	At least one SR file shall be submitted for a Request to be processed.	Hard	5.5
VAL-519	Firm Profile - Missing Account Type Mapping - Cash	FirmSettings	Missing Account Type Mapping - Cash	In the Firm Profile, at least one firm account shall be mapped to FINRA Account Type 'CASH'.	Soft	5.4
VAL-515	Firm Profile - Missing Grand Total Algorithm	FirmSettings	Missing Grand Total Algorithm	In the Firm Profile, the Grand Total Algorithm must be specified.	Hard	5.4

VAL-275	SR Account - Missing Long Market Value on Settlement Date by Account	N/A	N/A	SR Account - Long Market Value on Settlement Date by Account (LongMKSettle) shall not be null or blank.	XSD	5.0
VAL-277	SR Account - Missing Short Market Value on Settlement Date by Account	N/A	N/A	SR Account - Short Market Value on Settlement Date by Account (ShortMKSettle) shall not be null or blank.	XSD	5.0
VAL-285	SR Account - Missing Short Market Value on Trade Date by Account	N/A	N/A	SR Account - Short Market Value on Trade Date by Account (ShortMKTrade) shall not be null or blank.	XSD	5.0
VAL-271	SR Account - Missing Long Quantity on Settlement Date by Account	N/A	N/A	SR Account - Long Quantity on Settlement Date by Account (LongPosSettle) shall not be null or blank.	XSD	5.0
VAL-273	SR Account - Missing Short Quantity on Settlement Date by Account	N/A	N/A	SR Account - Short Quantity on Settlement Date by Account (ShortPosSettle) shall not be null or blank.	XSD	5.0
VAL-279	SR Account - Long Position on Trade Date by Account Required	N/A	N/A	SR Account - Long Position on Trade Date by Account (LongPosTrade) shall not be null or blanks.	XSD	5.0
VAL-281	SR Account - Missing Short Quantity on Trade Date by Account	N/A	N/A	SR Account - Short Quantity on Trade Date by Account (ShortPosTrade) shall not be null or blank.	XSD	5.0
VAL-248	SR Security - Negative Or Non-Numeric Allocated Price	N/A	N/A	SR Security - Allocated Price shall be non-negative numeric.	XSD	5.0
VAL-338	APOS - Missing Allocated Quantity	N/A	N/A	APOS Allocated Quantity shall not be null or blank.	XSD	5.0
VAL-339	APOS - Missing Allocated Value	N/A	N/A	A) If APOS Long Allocated Value and Short Allocated Value are null or blank, then APOS Allocated Value shall not be null or blank. B) If APOS Long Allocated Value and Short Allocated Value are not null or blank then APOS Allocated Value shall be null or blank.	XSD	5.0
VAL-341	APOS - Invalid Long Allocated Value Type	N/A	N/A	If APOS Long Allocated Value Type is not null or blank, then APOS Long Allocated Value Type shall be 'C' or 'M'.	XSD	5.0
VAL-342	APOS - Invalid Short Allocated Value Type	N/A	N/A	If APOS Short Allocated Value Type is not null or blank, then APOS Short Allocated Value Type shall be 'C' or 'M'.	XSD	5.0
VAL-304	SADT - Missing Allocation Pair Off Sequence Number	N/A	N/A	SADT Allocation Pair Off Sequence Number shall not be null or blank.	XSD	5.0
VAL-343	APOS Allocation Pair Off Sequence Number Required	N/A	N/A	APOS Allocation Pair Off Sequence Number shall not be null or blank.	XSD	5.0
VAL-472	APOH Allocation Pair Off Sequence Number Required	N/A	N/A	APOH Allocation Pair Off Sequence Number (Sequence) shall not be null or blank.	XSD	5.0
VAL-3.15	ACAT - Missing As Of Date	N/A	N/A	A) ACAT As Of Date shall not be null or blank. B) ACAT As Of Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0

VAL-3.25	ACSM - Missing As Of Date	N/A	N/A	A) ACSM As Of Date shall not be null or blank. B) ACSM As Of Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-3.45	APOH - Missing As Of Date	N/A	N/A	A) APOH As Of Date shall not be null or blank. B) APOH As Of Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-3.55	APOS - Missing As Of Date	N/A	N/A	A) APOS As Of Date shall not be null or blank. B) APOS As Of Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-3.65	COA - Missing As Of Date	N/A	N/A	A) COA As Of Date shall not be null or blank. B) COA As Of Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-3.95	SADT - Missing As of Date	N/A	N/A	A) SADT As Of Date shall not be null or blank. B) SADT As Of Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-3.05	As of Date Required - SR	N/A	N/A	A) SR As Of Date shall not be null or blank. B) SR As Of Date shall be a valid date with format YYYY-MM-DD.	Hard	5.0
VAL-244.5	SR Security - Currency Code Default	N/A	N/A	If SR Security Currency Code equals null, 'N/A', 'n/a', 'NA' or 'na' then system shall default Currency Code to 'USD'.	Default	5.0
VAL-246	SR Security - Negative or Non-Numeric Exchange Rate Default	N/A	N/A	If SR Security Currency Code is 'USD' and Exchange Rate is zero, negative or non-numeric, then the system shall default Exchange Rate to 1.	Default	5.0
VAL-246.5	SR Security - Missing Exchange Rate Default (USD)	N/A	N/A	If SR Security Currency Code is 'USD' and Exchange Rate is null or blank, then system shall default Exchange Rate to 1.	Default	5.0
VAL-380	SR Security - CUSIP Not N/A	N/A	N/A	If SR CUSIP equals 'N/A', 'n/a', 'NA' or 'na' then the system shall default CUSIP to Null.	Default	5.0
VAL-469	SADT CUSIP - N/A Default	N/A	N/A	If SADT CUSIP equals 'N/A', 'n/a', 'NA' or 'na' then the system shall default CUSIP to Null.	Default	5.0
VAL-240	SR Security - NonNumeric Factor	N/A	N/A	If SR Security - Factor is non-numeric, then the system shall default Factor to 1.	Default	5.8
VAL-52	SR Account - Both Sides Check (SR)	N/A	N/A	For an SR Account Number if Long Quantity and Short Quantity are both greater than zero for a single record, the system shall split these into separate records for processing.	Default	5.0
VAL-474	Collapse Whitespace Default	N/A	N/A	The system shall collapse whitespace within any text field for all XML and text files (i.e., replace contiguous whitespace with a single whitespace character).	Default	5.0
VAL-4.05	SRFile - Missing Firm CRD Number	N/A	N/A	SR Firm CRD Number shall not be null or blank.	XSD	5.0
VAL-4.25	SADT - Missing Firm CRD Number	N/A	N/A	SADT Firm CRD Number shall not be null or blank.	XSD	5.0

VAL-4.55	COA - Missing Firm CRD Number	N/A	N/A	COA Firm CRD Number shall not be null or blank.	XSD	5.0
VAL-4.65	APOS - Missing Firm CRD Number	N/A	N/A	APOS Firm CRD Number shall not be null or blank.	XSD	5.0
VAL-4.75	APOH - Missing Firm CRD Number	N/A	N/A	APOH Firm CRD Number shall not be null or blank.	XSD	5.0
VAL-4.85	ACSM - Missing Firm CRD Number	N/A	N/A	ACSM Firm CRD Number shall not be null or blank.	XSD	5.0
VAL-4.95	ACAT - Missing Firm CRD Number	N/A	N/A	ACAT Firm CRD Number shall not be null or blank.	XSD	5.0
VAL-225	SR Summary - Missing Grand Total Long Quantity on Settlement Date	N/A	N/A	SR Summary - Grand Total Long Quantity on Settlement Date (GrandTotalLongSD) shall not be null or blank.	XSD	5.0
VAL-227	SR Summary - Missing Grand Total Short Quantity on Settlement Date	N/A	N/A	SR Summary - Grand Total Short Quantity on Settlement Date (GrandTotalShortSD) shall not be null or blank.	XSD	5.0
VAL-229	SR Summary - Missing Grand Total Long Quantity on Trade Date	N/A	N/A	SR Summary - Grand Total Long Quantity on Trade Date (GrandTotalLongTD) shall not be null or blank.	XSD	5.0
VAL-231	SR Summary - Missing Grand Total Short Quantity on Trade Date	N/A	N/A	SR Summary - Grand Total Short Quantity on Trade Date (GrandTotalShortTD) shall not be null or blank.	XSD	5.0
VAL-233	SR Summary - Missing Total Number of Securities	N/A	N/A	SR Summary - Total Number of Securities in the Full Stockc Record shall not be null or blank.	XSD	5.0
VAL-289	SR Account - Last Activity Trade Date Valid	N/A	N/A	If SR Account - Last Activity Trade Date is not null or blank, then value shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-290	SR Account - Last Activity Settlement Date Valid	N/A	N/A	If SR Account - Last Activity Settlement Date is not null or blank, then value shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-283	SR Account - Missing Long Market Value on Trade by Account Required	N/A	N/A	SR Account - Long Market Value on Trade Date Long by Account (LongMKTrade) shall not be null or blank.	XSD	5.0
VAL-238	SR Security - Invalid Maturity Date	N/A	N/A	If SR Security - Maturity Date is not null or blank, Maturity Date shall be a valid date with format YYYY-MM-DD.	XSD	5.0
VAL-241	SR Security - Missing Price	N/A	N/A	SR Security - Price shall not be null or blank.	XSD	5.0
VAL-255	SR Security - Missing Total Short Quantity on Trade Date by Security	N/A	N/A	SR Security - Total Short Quantity on Trade Date by Security (TotalShortTD) shall not be null or blank.	XSD	5.0
VAL-305	SADT - Missing Long Quantity	N/A	N/A	SADT Long Quantity shall not be null or blank.	XSD	5.0
VAL-307	SADT - Missing Short Quantity	N/A	N/A	SADT Short Quantity shall not be null or blank.	XSD	5.0
VAL-309	SADT - Missing Allocated Quantity	N/A	N/A	SADT Allocated Quantity shall not be null or blank.	XSD	5.0
VAL-440	SR Securities Missing Accounts	N/A	N/A	Each Security in SR shall be mapped to at least one Account.	XSD	5.0
VAL-466	SR Security ID - N/A Default	N/A	N/A	If SR Security ID equals 'N/A', 'n/a', 'NA' or 'na' then the system shall default Security ID to Null.	Default	5.0

VAL-500	SADT - Security ID Default N/A	N/A	N/A	If SADT Security ID equals 'N/A', 'n/a', 'NA' or 'na' then the system shall default Security ID to Null.	Default	5.0
VAL-259	SR Security - Missing Total Long Market Value on Settlement Date by Security	N/A	N/A	SR Security - Total Long Market Value on Settlement Date by Security (TotalLongMKSD) shall not be null or blank.	XSD	5.0
VAL-264	SR Security - Missing Total Short Market Value on Settlement Date by Security	N/A	N/A	SR Security - Total Short Market Value on Settlement Date by Security (TotalShortMKSD) shall not be null or blank.	XSD	5.0
VAL-265	SR Security - Missing Total Long Market Value on Trade Date by Security	N/A	N/A	SR Security - Total Long Market Value on Trade Date by Security (TotalLongMKTD) shall be non-negative numeric.	XSD	5.0
VAL-266	SR Security - Missing Total Short Market Value on Trade Date by Security	N/A	N/A	SR Security - Total Short Market Value on Trade Date by Security (TotalShortMKTD) shall not be null or blank.	XSD	5.0
VAL-249	SR Security - Missing Total Long Quantity on Settlement Date by Security	N/A	N/A	SR Security - Total Long Quantity on Settlement Date by Security (TotalLongSD) shall not be null or blank.	XSD	5.0
VAL-251	SR Security - Missing Total Short Quantity on Settlement Date by Security	N/A	N/A	SR Security - Total Short Quantity on Settlement Date by Security (TotalShortSD) shall not be null or blank.	XSD	5.0
VAL-253	SR Security - Missing Total Long Quantity on Trade Date by Security	N/A	N/A	SR Security - Total Long Quantity on Trade Date by Security (TotalLongTD) shall not be null or blank.	XSD	5.0
VAL-237	SR Security - Invalid Security Type	N/A	N/A	SR Security - Security Type shall be a valid FINRA Security Type Code per Appendix B of this guide.	XSD	5.0
VAL-236	SR Security - Missing Security Type	N/A	N/A	SR Security - Security Type shall not be null or blank.	Hard	5.0
VAL-257	SR Security - Missing Segregation Required by Security	N/A	N/A	SR Security - Segregation Required by Security (SegRequiredbySecurity) shall not be null or blank.	XSD	5.0
VAL-258	SR Security - Non-Numeric Segregation Required by Security	N/A	N/A	SR Security - Segregation Required by Security (SegRerquiredbySecurity) shall be numeric.	Hard	5.7
VAL-287	SR Account - Segregation Required by Account Missing	N/A	N/A	SR Account - Segregation Required by Account shall not be null or blank.	XSD	5.0
VAL-288	SR Account - Non-Numeric Segregation Required by Account	N/A	N/A	SR Account - Segregation by Account (SegRequiredbyAccount) shall be numeric.	Hard	5.7
VAL-437	Trim Whitespace	N/A	N/A	The system shall trim all leading and trailing whitespace on text fields for all XML and text files.	Default	5.0
VAL-320	ACSM - Missing Total Accounts	N/A	N/A	ACSM Total Accounts shall not be null or blank.	XSD	5.0
VAL-321	ACSM - Non-Numeric Total Accounts	N/A	N/A	ACSM Total Accounts shall be numeric.	XSD	5.0
VAL-324	ACSM - Missing Total Long Quantity	N/A	N/A	ACSM Total Long Quantity shall not be null or blank.	XSD	5.0
VAL-328	ACSM - Missing Total Long Quantity	N/A	N/A	ACSM Total Long Value shall not be null or blank.	XSD	5.0
VAL-322	ACSM - Missing Total Securities	N/A	N/A	ACSM Total Securities shall not be null or blank.	XSD	5.0

VAL-323	ACSM - Non-Numeric Total Securites	N/A	N/A	ACSM Total Securities shall be numeric.	XSD	5.0
VAL-326	ACSM - Missing Total Short Quantity	N/A	N/A	ACSM Total Short Quantity shall not be null or blank.	XSD	5.0
VAL-330	ACSM - Missing Total Short Value	N/A	N/A	ACSM Total Short Value shall not be null or blank.	XSD	5.0
VAL-317	SADT - Invalid Short Allocated Value Type	N/A	N/A	If SADT Short Allocated Value Type is not null or blank, SADT Short Allocated Value Type shall be 'C' or 'M'.	XSD	5.0
VAL-352	SADT - Missing Allocated Value	N/A	N/A	A) If SADT Allocated Value is null, then SADT Long Allocated Value shall not be null or blank. B) If SADT Allocated Value is not null then SADT Long Allocated Value shall be null.	XSD	5.0
VAL-355	SADT - Missing Allocated Value	N/A	N/A	A) If SADT Allocated Value is null, then SADT Short Allocated Value shall not be null or blank. B) If SADT Allocated Value is not null then SADT Short Allocated Value shall be null.	XSD	5.0
VAL-314	SADT - Invalid Allocated Value Type	N/A	N/A	If SADT Allocated Value Type is not null, then SADT Allocated Value Type shall be 'C' or 'M' .	XSD	5.0
VAL-314.1	SADT - Allocated Value Type Default	N/A	N/A	If SADT Allocated Value Type is null or blank and SADT Allocated Value is not null, then SADT Allocated Value Type shall default to 'M'.	Default	5.2
VAL-315	SADT - Invalid Long Allocated Value Type	N/A	N/A	If SADT Long Allocated Value Type is not null or blank, SADT Long Allocated Value Type shall be 'C' or 'M'.	XSD	5.0
VAL-315.1	SADT - Long Allocated Value Type Default	N/A	N/A	If SADT Long Allocated Value Type is null or blank and SADT Long Allocated Value is not null, then SADT Long Allocated Value Type shall default to 'M'.	Default	5.2
VAL-316.1	SADT - Short Allocated Value Type Default	N/A	N/A	If SADT Short Allocated Value Type is null or blank and SADT Short Allocated Value is not null, then SADT Short Allocated Value Type shall default to 'M'.	Default	5.2
VAL-428	SADT - APOS Allocated Quantity Quantity Mismatch	SADT	SADT-APOS Allocated Quantity or Value Mismatch	For an Allocation Pair Off Sequence in SADT or APOS, the sum of SADT Allocated Quantity shall be equal to APOS Allocated Quantity (i.e., for Long and Short Allocation Code).	Hard	5.0
VAL-429	SADT - APOS Long Allocated Value Mismatch	SADT	SADT-APOS Allocated Quantity or Value Mismatch	For an Allocation Pair Off Sequence in SADT or APOS the sum of SADT Long Allocated Value plus SADT Allocated Value shall be equal to the sum of APOS Long Allocated Value plus APOS Allocated Value plus APOS Allocated Value.	Hard	5.4

VAL-430	SADT - APOS Short Allocated Value Mismatch	SADT	SADT-APOS Allocated Quantity or Value Mismatch	For an Allocation Pair Off Sequence in SADT or APOS the sum of SADT Short Allocated Value plus SADT Allocated Value shall be equal to the sum of APOS Short Allocated Value plus APOS Allocated Value plus APOS Allocated Value.	Hard	5.4
VAL-18	SADT - Missing Allocation Code	SADT	Basic Validations	A) SADT Long Allocation Code (CategoryCodeLong) shall not be null or blank. B) SADT Short Allocation Code (CategoryCodeShort) shall not be null or blank.	XSD	5.0
VAL-396	SADT - SADT Allocation Code Not In ACSM (Long)	SADT	SADT Allocation Code Not In ACSM	If SADT Long Allocation Code (CategoryCodeLong) does not have ACAT Long-Short Indicator equals 'U' then SADT Long Allocation Code (CategoryCodeLong) shall exist in ACSM Allocation Code (CategoryCode).	Hard	5.0
VAL-397	SADT - SADT Allocation Code Not In ACSM (Short)	SADT	SADT Allocation Code Not In ACSM	If SADT Short Allocation Code (CategoryCodeLongShort) does not have ACAT Long-Short Indicator equals 'U' then SADT Short Allocation Code (CategoryCodeShort) shall exist in ACSM Allocation Code (CategoryCode).	Hard	5.0
VAL-398	SADT - ACSM Allocation Code Not In SADT	SADT	ACSM Allocation Code Not In SADT	ACSM Allocation Code (CategoryCode) shall exist in SADT Long Allocation Code (CategoryCodeLong) or SADT Short Allocation Code (CategoryCodeShort), excluding Allocation Codes where balances are zero.	Soft	5.2
VAL-17	SADT - Allocation Pair Off Sequence Less Than Zero	SADT	Allocation Pair Off Sequence Less Than Zero	SADT Allocation Pair Off Sequence Number shall be greater than or equal to zero.	Soft	5.0
VAL-449	SADT - Duplicate Security - Sequence	SADT	Duplicate Security - Sequence	SADT Match Key (i.e., Security ID or CUSIP as per *"Rules for relating a security between SR and SADT"*) shall be unique for an Allocation Pair Off Sequence Number, except where Long Allocation Code or Short Allocation Code is Unallocated (U) as defined in ACAT.	Hard	5.0
VAL-3.9	SADT - Invalid As Of Date	SADT	Invalid As Of Date	SADT As Of Date shall match the As Of Date of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-125	SR Security Not in SADT - No Options	SADT	SR Security (No Option) Not In SADT	Every security in SR that is not Security Type Option (16) shall match to a security in SADT per *"Rules for relating a security between SR and SADT",* excluding securities where Settlement Date data is not reported.	Soft	5.2
VAL-126	SR Security Not in SADT	SADT	SR Security Not In SADT	Every security in SR (i.e., including options) shall match to a security in SADT per "*Rules for relating a security between SR and SADT*", excluding securities where Settlement Date data is not reported.	Soft	5.2
VAL-222	SADT - Missing CUSIP and Security ID	SADT	Missing CUSIP and Security ID	SADT CUSIP and Security ID shall not both be null or blank.	Hard	5.0

VAL-477	SADT CUSIP Match Key Mapped to Multiple Security ID	SADT	CUSIP Match Key Mapped to Multiple Security IDs	If Match Key is CUSIP as per *"Rules for relating a security between SR and SADT"*, SADT CUSIP shall be mapped to only one distinct SADT Security ID or to one or more null Security IDs (e.g., CUSIP may not be mapped to two Security IDs or one Security ID and one null but may be mapped to two nulls).	Hard	5.0
VAL-476	SADT Security ID Match Key Mapped to Multiple CUSIP	SADT	Security ID Match Key Mapped to Multiple CUSIPs	If Match Key is Security ID as per *"Rules for relating a security between SR and SADT"*, SADT Security ID shall be mapped to only one distinct SADT CUSIP or to one or more null CUSIPs (e.g., Security ID may not be mapped to two CUSIPS or one CUSIP and one null but may be mapped to two nulls.).	Hard	5.0
VAL-64	SADT Security Not in SR or Unmapped in SR	SADT	SADT Security Not In SR Or Unmapped In SR	Every security in SADT shall match to a mapped (i.e., associated with an allocation code) security in SR per *"Rules for relating a security between SR and SADT"*.	Hard	5.7
VAL-2.6	SADT - Invalid XML	SADT	Invalid XML	SADT XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4.2	SADT - Invalid Firm CRD Number	SADT	Invalid Firm CRD Number	SADT Firm CRD Number shall match the Firm CRD Number of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-552	SADT - Invalid Long Allocation Code Length	SADT	Basic Validations	SADT - Long Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-310	SADT - Negative Or Non-Numeric Allocated Quantity	SADT	Basic Validations	SADT Allocated Quantity shall be non-negative numeric.	XSD	5.0
VAL-306	SADT - Negative Or Non-Numeric Long Quantity	SADT	Basic Validations	SADT Long Quantity shall be non-negative numeric.	XSD	5.0
VAL-308	SADT - Negative Or Non-Numeric Short Quantity	SADT	Basic Validations	SADT Short Quantity shall be non-negative numeric.	XSD	5.0
VAL-399	SADT - SADT Total Long Quantity Greater Than ACSM	SADT	SADT Total Qty Greater Than ACSM	For each SADT Long Allocation Code (CategoryCodeLong), excluding any Sequence with an Unallocated (U) code as defined in ACAT, evaluate the following steps: 1) Sum the Allocated Quantity across all securities. 2) Result of Step 1 shall be less than or equal to ACSM Total Long Quantity for the same Allocation Code.	Hard	5.2

VAL-400	SADT - SADT Total Short Quantity Greater Than ACSM	SADT	SADT Total Qty Greater Than ACSM	For each SADT Short Allocation Code (CategoryCodeShort), excluding any Sequence with an Unallocated (U) code as defined in ACAT, evaluate the following steps: 1) Sum the Allocated Quantity across all securities. 2) Result of Step 1 shall be less than or equal to ACSM Total Short Quantity for the same Allocation Code.	Hard	5.2
VAL-535	SADT - Quantity-Value Mismatch (Long)	SADT	Allocated Quantity - Allocated Value Mismatch	If SADT - Allocated Quantity is equal to zero then: a) If SADT Allocated Value is not null then SADT Allocated Value shall be equal to zero and b) If SADT Allocated Long Value is not null then SADT Allocated Long Value shall be equal to zero.	Soft	5.9
VAL-536	SADT - Quantity-Value Mismatch (Short)	SADT	Allocated Quantity - Allocated Value Mismatch	If SADT - Allocated Quantity is equal to zero then: a) If SADT Allocated Value is not null then SADT Allocated Value shall be equal to zero and b) If SADT Allocated Short Value is not null then SADT Allocated Long Value shall be equal to zero.	Soft	5.9
VAL-551	SADT - Invalid Security Description Length	SADT	Basic Validations	SADT - Security Description length shall be less than or equal to 200.	XSD	5.8
VAL-303	SADT - Missing Security Description	SADT	Missing Security Description	SADT Security Description shall not be null or blank.	Soft	5.0
VAL-411	SADT - Inconsistent Security ID	SADT	Inconsistent Security ID	If SADT Security ID is null for any record, it shall be null for all records in the file.	Hard	5.0
VAL-562	SADT - Invalid Security ID Length	SADT	Basic Validations	SADT - Security ID length shall be less than or equal to 30.	XSD	5.8
VAL-450	SADT Sequence + Long Code + Short Code Not In APOH	SADT	SADT Sequence - Long Code - Short Code Not In APOH	The combination of SADT Allocation Pair Off Sequence Number, Long Allocation Code and Short Allocation Code shall exist in APOH except where Allocation Code is Unallocated (U).	Hard	5.0
VAL-553	SADT - Invalid Short Allocation Code Length	SADT	Basic Validations	SADT - Short Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-312	SADT - Negative Or Non-Numeric Allocated Value	SADT	Basic Validations	If SADT Allocated Value is not null, SADT Allocated Value shall be non-negative numeric.	XSD	5.0
VAL-313	SADT - Negative Or Non-Numeric Long Allocated Value	SADT	Basic Validations	If SADT Long Allocated Value is not null, SADT Long Allocated Value shall be non-negative numeric.	XSD	5.0
VAL-316	SADT - Negative Or Non-Numeric Short Allocated Value	SADT	Basic Validations	If SADT Short Allocated Value is not null, then SADT Short Allocated Value shall be non-negative numeric.	XSD	5.0
VAL-537	SR Account - Multiple Account Names for Account	SRAccount	Multiple Account Names for Account	For each SR account (i.e., Account Number and Account Type) there shall only be one distinct Acccount Name.	Hard	5.7

VAL-443	SRAccount - Missing Account Type Mapped to Cash or Margin	SRAccount	Missing Account Type Mapped to Cash or Margin	SR Account - Account Type mapped in Firm Profile - Account Type Mappings to CASH or MARGIN shall exist in SR Account data.	Soft	5.10
VAL-565	SR - Missing Allocation Codes with SADT	SRAccount	Missing Allocation Codes with SADT	If SADT file is submitted, one or more Allocation Codes shall be provided on either the SR file.	Hard	5.8
VAL-467	SR - Derived ACAT Allocation Code Wrong Side (Long)	SRAccount	Allocation Code L-S Side Not Match ACAT/APOH	If SR Allocation Code is not null and SR Long Quantity on Settlement Date by Account (LongPosSettle) is greater than zero, then Allocation Code shall exist in ACAT or APOH Allocation Code where Long-Short Indicator equals 'L' or 'B' or 'X'.	Soft	5.1
VAL-468	SR - Derived ACAT Allocation Code Wrong Side (Short)	SRAccount	Allocation Code L-S Side Not Match ACAT/APOH	If SR Allocation Code is not null and SR Short Quantity on Settlement Date by Account (ShortPosSettle) is greater than zero, then Allocation Code shall exist in ACAT or APOH Allocation Code where ACAT Long-Short Indicator equals 'S' or 'B' or 'X'.	Soft	5.1
VAL-360	SRAccount - Allocation Code Not In ACAT/APOH	SRAccount	Allocation Code Not In ACAT/APOH	If SR Allocation Code (Category Code) is not null then Allocation Code shall exist in ACAT where Long-Short Indicator is not Unallocated (U).	Hard	5.2
VAL-345	SRAccount - Zero Long and Short Position	SRAccount	Zero Long and Short Position	For an SR security (CUSIP + Security ID) and SR account (Account Number + Account Type) combination, Long Quantity on Settlement Date by Account (LongPosSettle), Short Quantity on Settlement Date by Account (ShortPosSettle), Long Quantity on Trade Date by Account (LongPosTrade) and Short Quantity on Trade Date by Account (ShortPosTrade) shall not all be zero.	Soft	5.0
VAL-49	SR Account - Settlement Date Quantity-Market Value Mismatch (Long)	SRAccount	Quantity - Market Value Mismatch	If SR Account - Long Quantity on Settlement Date by Account (LongPosSettle) is equal to zero then SR Account - Long Market Value on Settlement Date by Account (LongMKSettle) shall be equal to zero.	Soft	5.0
VAL-404	SR Account - Trade Date Quantity-Market Value Mismatch (Long)	SRAccount	Quantity - Market Value Mismatch	If SR Account - Long Quantity on Trade Date by Account (LongPosTrade) is equal to zero then SR Account - Long Market Value on Trade Date by Account (LongMKTrade) shall not be greater than zero.	Soft	5.0
VAL-405	SR Account - Trade Date Quantity-Market Value Mismatch (Short)	SRAccount	Quantity - Market Value Mismatch	If SR Account - Short Quantity on Trade Date by Account (ShortPosTrade) is equal to zero then SR Account - Short Market Value on Trade Date by Account (ShortMKTrade) shall not be greater than zero.	Soft	5.0
VAL-50	SR Account - Settlement Date Quantity-Market Value Mismatch (Short)	SRAccount	Quantity - Market Value Mismatch	If SR Account - Short Quantity on Settlement Date by Account (ShortPosSettle) is equal to zero then SR Account - Short Market Value on Settlement Date by Account (ShortMKSettle) shall be equal to zero.	Soft	5.0

VAL-291	SRAccount - Invalid SCOA Code	SRAccount	Invalid SCOA Code	If SR SCOA Code is not null, SCOA Code (COACode) shall exist in FINRA Standard Chart of Accounts.	Hard	5.0
VAL-89	SRAccount - Missing SCOA Code	SRAccount	Missing SCOA Code	SR SCOA Code (COACode) shall not be null or blank.	Hard	5.0
VAL-359	SRAccount - Multiple SCOA Codes for Account	SRAccount	Multiple SCOA Codes for Account	If SR SCOA Code is not null, each account (i.e., Account Number + Account Type) shall be mapped to only one SCOA Code except for those accounts mapped to the following SCOA Code combinations: a) two or more codes in 19, 20, 21 or 22, b) two or more codes in 23, 24, 25, 26, 27, 28 or 29, c) 30 and 32, or d) 31 and 34.	Hard	5.12
VAL-538	SR Account - Negative Segregation Required by Account	SRAccount	Negative Segregation Required by Account	SR Account - Segregation by Account shall be non-negative.	Hard	5.7
VAL-276	SR Account - Negative or Non-Numeric Long Market Value on Settlement Date by Account	SRFile	Basic Validations	SR Account - Long Market Value on Settlement Date by Account (LongMKSettle) shall be non-negative numeric.	XSD	5.0
VAL-284	SR Account - Negative or Non-Numeric Long Market Value on Trade Date by Account	SRFile	Basic Validations	SR Account - Long Market Value on Trade Date by Account (LongMKTrade) shall be non-negative numeric.	XSD	5.0
VAL-278	SR Account - Negative or Non-Numeric Short Market Value on Settlement Date by Account	SRFile	Basic Validations	SR Account - Short Market Value on Settlement Date by Account (ShortMKSettle) shall be non-negative numeric.	XSD	5.0
VAL-286	SR Account - Negative or Non-Numeric Short Market Value on Trade Date by Account	SRFile	Basic Validations	SR Account - Total Short Market Value on Trade Date by Account (ShortMKTrade) shall be non-negative numeric.	XSD	5.0
VAL-555	SR Account - Invalid Account Name Length	SRFile	Basic Validations	SR Account - Account Name length shall be less than or equal to 200.	XSD	5.8
VAL-554	SR Account - Invalid Account Number Length	SRFile	Basic Validations	SR Account - Account Number length shall be less than or equal to 100.	XSD	5.8
VAL-267	SR Account - Missing Account Number	SRFile	Basic Validations	SR Account - Account Number shall not be null or blank.	Hard	5.0
VAL-270	SR Account - Missing Account Type	SRFile	Basic Validations	SR Account - Account Type shall not be null or blank.	XSD	5.0
VAL-272	SR Account - Negative or Non-Numeric Long Quantity on Settlement Date by Account	SRFile	Basic Validations	SR Account - Long Quantity on Settlement Date by Account (LongPosSettle) shall be non-negative numeric.	XSD	5.0
VAL-280	SR Account - Negative or Non-Numeric Long Quantity on Trade Date by Account	SRFile	Basic Validations	SR Account - Long Quantity on Trade Date by Account (LongPosTrade) shall be non-negative numeric.	XSD	5.0
VAL-274	SR Account - Negative or Non-Numeric Short Quantity on Settlement Date by Account	SRFile	Basic Validations	SR Account - Short Quantity on Settlement Date by Account (ShortPosSettle) shall be non-negative numeric.	XSD	5.0

VAL-282	SR Account - Negative or Non-Numeric Short Quantity on Trade Date by Account	SRFile	Basic Validations	SR Account - Short Quantity on Trade Date by Account (ShortPosTrade) shall be non-negative numeric.	XSD	5.0
VAL-563	SR Account - Invalid Account Type Length	SRFile	Basic Validations	SR Account - Account Type length shall be less than or equal to 100.	XSD	5.8
VAL-557	SR Account - Invalid Allocation Code Length	SRFile	Basic Validations	SR Account - Allocation Code length shall be less than or equal to 50.	XSD	5.8
VAL-3	SRFile - Invalid As Of Date	SRFile	Invalid As Of Date	SR As Of Date shall match the As Of Date of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-556	SR Account - Invalid Branch Code Length	SRFile	Basic Validations	SR Account - Branch Code length shall be less than or equal to 50.	XSD	5.8
VAL-2	SRFile - Invalid XML	SRFile	Invalid XML	SR XML file shall comply with the XML Schema (XSD). Note: FINRA will support the latest XML Schema plus one prior version.	Hard	5.0
VAL-4	SRFile - Invalid Firm CRD Number	SRFile	Invalid Firm CRD Number	SR Firm CRD Number shall match the Firm CRD Number of the Request, based on the Request ID in the file name.	Hard	5.0
VAL-54.5	SRFile - Grand Total Mismatch - Settlement Date Long - Settlement Date Long	SRFile	Grand Total Mismatch - Settlement Date Long	A) If Firm Profile - Grand Totals Algorithm = "FIRST" then the sum of SR Security - Total Long Quantity on Settlement Date by Security (TotalLongSD) for the filing shall be equal to SR Summary - Grand Total Long Quantity on Settlement Date (GrandTotalLongSD) on the first file of the filing. B) If Firm Profile - Grand Totals Algorithm = "SUM" then the sum of SR Security - Total Long Quantity on Settlement Date by Security (TotalLongSD) for the filing shall be equal to the sum of SR Summary - Grand Total Long Quantity on Settlement Date (GrandTotalLongSD) across all files for the filing.	Soft	5.0

VAL-54.6	SRFile - Grand Total Mismatch - Settlement Date Short	SRFile	Grand Total Mismatch - Settlement Date Short	A) If Firm Profile - Grand Totals Algorithm = "FIRST" then the sum of SR Security - Total Short Quantity on Settlement Date by Security (TotalShortSD) for the filing shall be equal to SR Summary - Grand Total Short Quantity on Settlement Date (GrandTotalShortSD) on the first file of the filing. B) If Firm Profile - Grand Totals Algorithm = "SUM" then the sum of SR Security - Total Short Quantity on Settlement Date by Security (TotalShortSD) for the filing shall be equal to the sum of SR Summary - Grand Total Short Quantity on Settlement Date (GrandTotalShortSD) across all files for the filing.	Soft	5.0
VAL-54.7	SRFile - Grand Total Mismatch - Trade Date Long	SRFile	Grand Total Mismatch - Trade Date Long	A) If Firm Profile - Grand Totals Algorithm = "FIRST" then the sum of SR Security - Total Long Quantity on Trade Date by Security (TotalLongTD) for the filing shall be equal to SR Summary - Grand Total Long Quantity on Trade Date (GrandTotalLongTD) on the first file of the filing. B) If Firm Profile - Grand Totals Algorithm = "SUM" then the sum of SR Security - Total Long Quantity on Trade Date by Security (TotalLongTD) for the filing shall be equal to the sum of SR Summary - Grand Total Long Quantity on Trade Date (GrandTotalLongTD) across all files for the filing.	Soft	5.0
VAL-54.8	SRFile - Grand Total Mismatch - Trade Date Short	SRFile	Grand Total Mismatch - Trade Date Short	A) If Firm Profile - Grand Totals Algorithm = "FIRST" then the sum of SR Security - Total Short Quantity on Trade Date by Security (TotalShortTD) for the filing shall be equal to SR Summary - Grand Total Short Quantity on Trade Date (GrandTotalShortTD) on the first file of the filing. B) If Firm Profile - Grand Totals Algorithm = "SUM" then the sum of SR Security - Total Short Quantity on Trade Date by Security (TotalShortTD) for the filing shall be equal to the sum of SR Summary - Grand Total Short Quantity on Trade Date (GrandTotalShortTD) across all files for the filing.	Soft	5.0

VAL-234.1	SR Summary - Total Number of Securities Sum Greater Than Zero	SRFile	Grand Total Mismatch - Number of Securities	A) If Firm Profile - Grand Totals Algorithm = "FIRST" then SR Summary - Total Number of Securities in the first file of the Full Stock Record shall be greater than zero. B) If Firm Profile - Grand Totals Algorithm = "SUM" then SR Summary - Total Number of Securities summed across all files in the Full Stock Record shall be greater than zero.	Hard	5.0
VAL-226	SRFile - Negative Or Non-Integer Grand Total Long Quantity on Settlement Date	SRFile	Basic Validations	SR Summary - Grand Total Long Quantity on Settlement Date by Security (GrandTotalLongSD) shall be non-negative integer.	XSD	5.0
VAL-230	SRFile - Negative Or Non-Integer Grand Total Long Quantity on Trade Date	SRFile	Basic Validations	SR Summary - Grand Total Long Quantity on Trade Date by Security (GrandTotalLongTD) shall be non-negative integer.	XSD	5.0
VAL-228	SRFile - Negative Or Non-Integer Grand Total Short Quantity on Settlement Date	SRFile	Basic Validations	SR Summary - Grand Total Short Quantity on Settlement Date by Security (GrandTotalShortSD) shall be non-negative integer.	XSD	5.0
VAL-232	SRFile - Negative Or Non-Integer Grand Total Short Quantity on Trade Date	SRFile	Basic Validations	SR Summary - Grand Total Short Quantity on Trade Date by Security (GrandTotalShortTD) shall be non-negative integer.	XSD	5.0
VAL-234	SRFile - Negative Or Non-Numeric Total Number of Securities	SRFile	Basic Validations	SR Summary - Total Number of Securities shall be non-negative numeric.	XSD	5.0
VAL-558	SR Memo - Invalid Memo Field Type Length	SRFile	Basic Validations	SR Account - Memo Field Type length shall be less than or equal to 4000.	XSD	5.8
VAL-559	SR Security - Invalid Security Description Length	SRFile	Basic Validations	SR Security - Security Description length shall be less than or equal to 200.	XSD	5.8
VAL-561	SR Security - Invalid Security ID Length	SRFile	Basic Validations	SR Security - Security ID length shall be less than or equal to 30.	XSD	5.8
VAL-260	SR Security - Negative Or Non-Numeric Total Long Market Value on Settlement Date by Security	SRFile	Basic Validations	SR Security - Total Long Market Value on Settlement Date by Security (TotalLongMKSD) shall be non-negative numeric.	XSD	5.0
VAL-262	SR Security - Negative Or Non-Numeric Total Long Market Value on Trade Date by Security	SRFile	Basic Validations	SR Security - Total Long Market Value on Trade Date by Security (TotalLongMKTD) shall be non-negative numeric.	XSD	5.0
VAL-261	SR Security - Negative Or Non-Numeric Total Short Market Value on Settlement Date by Security	SRFile	Basic Validations	SR Security - Total Short Market Value on Settlement Date by Security (TotalShortMKSD) shall be non-negative numeric.	XSD	5.0
VAL-263	SR Security - Negative Or Non-Numeric Total Short Market Value on Trade Date by Security	SRFile	Basic Validations	SR Security - Total Short Market Value on Trade Date by Security (TotalShortMKTD) shall be non-negative numeric.	XSD	5.0
VAL-250	SR Security - Negative Or Non-Numeric Total Long Quantity on Settlement Date by Security	SRFile	Basic Validations	SR Security - Total Long Quantity on Settlement Date by Security (TotalLongSD) shall be non-negative numeric.	XSD	5.0
VAL-254	SR Security - Negative Or Non-Numeric Total Long Quantity on Trade Date by Security	SRFile	Basic Validations	SR Security - Total Long Quantity on Trade Date by Security (TotalLongTD) shall be non-negative numeric.	XSD	5.0

VAL-252	SR Security - Negative Or Non-Numeric Total Short Quantity on Settlement Date by Security	SRFile	Basic Validations	SR Security - Total Short Quantity on Settlement Date by Security (TotalShortSD) shall be non-negative numeric.	XSD	5.0
VAL-256	SR Security - Negative Or Non-Numeric Total Short Quantity on Trade Date by Security	SRFile	Basic Validations	SR Security - Total Short Quantity on Trade Date by Security (TotalShortTD) shall be non-negative numeric.	XSD	5.0
VAL-560	SR Security - Invalid Security Symbol Length	SRFile	Basic Validations	SR Security - Security Symbol length shall be less than or equal to 30.	XSD	5.8
VAL-244	SRSecurity - Invalid Currency Code	SRSecurity	Invalid Currency Code	After the default (i.e., VAL-244.5) has been applied, SR Security - Currency Code shall be a valid currency code as specified by ISO 4217 Type Currency Code List.	Soft	5.0
VAL-245	SRSecurity - Missing Exchange Rate	SRSecurity	Missing Exchange Rate	If SR Security Currency Code is not 'USD' then Exchange Rate shall not be null or blank.	Soft	5.0
VAL-245.5	SRSecurity - Invalid Exchange Rate	SRSecurity	Invalid Exchange Rate	If SR Security Currency Code is not 'USD' then Exchange Rate shall be valid numeric greater than zero.	Soft	5.0
VAL-246.7	SRSecurity - Exchange Rate Not Equal to 1 for USD	SRSecurity	Exchange Rate Not Equal to 1 for USD	If SR Security Currency Code is 'USD' then Exchange Rate shall be equal to 1.	Soft	5.0
VAL-475	SR CUSIP Match Key Mapped to Multiple Security ID	SRSecurity	CUSIP Match Key Mapped to Multiple Security IDs	If Match Key is CUSIP as per *"Rules for relating a security between SR and SADT"*, then SR CUSIP shall be mapped to only one distinct SR Security ID or one null Security ID.	Hard	5.0
VAL-39	SRSecurity - Duplicate Security ID After Rollup	SRSecurity	Duplicate Security ID After Rollup	After security rollup as per VAL-481, SR Security ID (i.e., not null) shall be unique for the submission (i.e., Security ID may not exist for more than one CUSIP or the same CUSIP on more than one record).	Hard	5.5
VAL-38	SRSecurity - Missing CUSIP and Security ID	SRSecurity	Missing CUSIP and Security ID	SR Security - CUSIP and Security ID shall not both be null or blank.	Hard	5.0
VAL-46	SRSecurity - Unbalanced Market Value by Security	SRSecurity	Unbalanced Quantity or Market Value by Security	For each security SR Total Market Value on Settlement Date by Security (TotalLongMKSD) shall be equal to SR Total Short Market Value on Settlement Date by Security (TotalShortMKSD).	Soft	5.0
VAL-45	SRSecurity - Unbalanced Quantity by Security	SRSecurity	Unbalanced Quantity or Market Value by Security	For each security SR Total Long Quantity on Settlement Date by Security (TotalLongSD) shall be equal to SR Total Short Quantity on Settlement Date by Security (TotalShortSD).	Soft	5.0
VAL-41	SR Security - Market Price Mismatch (Long)	SRSecurity	Price-Value Mismatch	For SR Security with Security Type not in ('1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '16') the Total Long Market Value on Settlement Date shall equal Price * Factor * Total Long Quantity on Settlement Date (TotalLongSD).	Soft	5.0
VAL-109	SR Security - Market Price Mismatch (Short)	SRSecurity	Price-Value Mismatch	For SR Security with Security Type not in ('1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '16') the Total Short Market Value on Settlement Date shall equal Price * Factor * Total Short Quantity on Settlement Date by Security (TotalLongSD).	Soft	5.0

VAL-465	SR Security - Option Pricing Mismatch (Long)	SRSecurity	Option Price-Value Mismatch	For SR Security with Security Type 16 the Total Long Market Value on Settlement Date (TotalLongMKSD) shall equal Price * Factor * Total Long Quantity on Settlement Date (TotalLongSD) * 100.	Soft	5.0
VAL-478	SR Security - Option Pricing Mismatch (Short)	SRSecurity	Option Price-Value Mismatch	For SR Security with Security Type 16 the Total Short Market Value on Settlement Date (TotalShortMKSD) shall equal Price * Factor * Total Short Quantity on Settlement Date (TotalShortSD) * 100.	Soft	5.0
VAL-243	SRSecurity - Missing Price Indicator	SRSecurity	Missing Price Indicator	If SR Security - Price is greater than zero then Price Indicator shall not be null or blank.	Hard	5.0
VAL-242	SRSecurity - Negative or Non-Numeric Price	SRSecurity	Negative or Non-Numeric Price	SR Security - Price shall be non-negative numeric.	Hard	5.0
VAL-350	SRSecurity - Settlement Date Market Value Mismatch (Long)	SRSecurity	Security - Account Quantity or Value Mismatch Settlement Date	SR Security - Total Long Market Value on Setlement Date by Security (TotalLongMKSD) shall be equal to the sum of SR Account - Long Market Value on Settlement Date by Account (LongMKSettle) for a given security.	Hard	5.0
VAL-351	SRSecurity - Settlement Date Market Value Mismatch (Short)	SRSecurity	Security - Account Quantity or Value Mismatch Settlement Date	SR Security - Total Short Market Value on Settlement Date by Security (TotalShortMKSD) shall be equal to the sum of SR Account - Long Market Value on Settlement Date by Account (ShortMKSettle) for a given security.	Hard	5.0
VAL-406	SRSecurity - Trade Date Market Value Mismatch (Long)	SRSecurity	Security - Account Quantity or Value Mismatch Trade Date	SR Security -Total Long Market Value on Trade Date by Security (TotalLongMKTD) shall be equal to the sum of SR Account - Long Market Value on Trade Date by Account (LongMKTrade) for a given security.	Hard	5.0
VAL-407	SRSecurity - Trade Date Market Value Mismatch (Short)	SRSecurity	Security - Account Quantity or Value Mismatch Trade Date	SR Security -Total Short Market Value on Trade Date by Security (TotalShortMKTD) shall be equal to the sum of SR Account - Short Market Value on Trade Date by Account (ShortMKTrade) for a given security.	Hard	5.0
VAL-43	SRSecurity - Security - Account Quantity on Settlement Date Mismatch (Long)	SRSecurity	Security - Account Quantity or Value Mismatch Settlement Date	SR Security - Total Long Quantity on Settlement Date by Security (TotalLongSD) shall be equal to the sum of SR Account - Long Quantity on Settlement Date by Account (LongPosSettle) for a given security.	Hard	5.0
VAL-44	SRSecurity - Security - Account Quantity on Settlement Date Mismatch (Short)	SRSecurity	Security - Account Quantity or Value Mismatch Settlement Date	SR Security - Total Short Quantity on Settlement Date by Security (TotalShortSD) shall be equal to the sum of SR Account - Short Quantity on Settlement Date by Account (ShortPosSettle) for a given security.	Hard	5.0
VAL-408	SRSecurity - Trade Date Quantity Mismatch (Long)	SRSecurity	Security - Account Quantity or Value Mismatch Trade Date	SR Security - Total Long Quantity on Trade Date by Security (TotalLongTD) shall be equal to the sum of SR Account - Long Quantity on Trade Date by Account (LongPosTrade) for a given security.	Hard	5.0
VAL-409	SRSecurity - Trade Date Quantity Mismatch (Short)	SRSecurity	Security - Account Quantity or Value Mismatch Trade Date	SR Security - Total Short Quantity on Trade Date by Security (TotalShortTD) shall be equal to the sum of SR Account - Short Quantity on Trade Date by Account (ShortPosTrade) for a given	Hard	5.0

				security.		
VAL-235	SR Security - Missing Security Description	SRSecurity	Missing Security Description	SR Security - Description shall not be null or blank.	Hard	5.0
VAL-412	SRSecurity - Inconsistent Security ID	SRSecurity	Inconsistent Security ID	If SR Security ID is null for any record, it shall be null for all records in the filing.	Hard	5.0
VAL-42	SRSecurity - Missing Security Symbol	SRSecurity	Missing Security Symbol	If SR Security Type is not 16 (Options) then Security Symbol shall not be null or blanks.	Soft	5.2
VAL-356	SRSecurity - Security - Account Segregation Required Mismatch	SRSecurity	Security - Account Segregation Required Mismatch	For a security the sum of SR Segregation Required by Account (SegRequiredbyAccount) shall be equal to SR Segregation Required By Security (SegRequiredbySecurity).	Soft	5.0
VAL-539	SR Security - Negative Segregation Required by Security	SRSecurity	Negative Segregation Required by Security	SR Security - Segregation by Security shall be non-negative.	Hard	5.7
VAL-456	SRSecurity - Allocated Quantity Less Than SADT (Long)	SRSecurity	Allocated Quantity Less Than SADT	For each security (i.e., CUSIP + Security ID) and Allocation Code (CategoryCode), the sum of SR Total Long Quantity on Settlement Date by Security shall be greater than or equal to SADT Allocated Quantity, excluding the following conditions: a) Allocation Code is defined in ACAT as Unallocated (U), b) Allocation Code is defined in ACAT as Left Out of Allocation Process (X), or c) Quantity is zero on one side (i.e., SR or SADT) and missing on the other side.	Hard	5.2
VAL-457	SRSecurity - Allocated Quantity Less Than SADT (Short)	SRSecurity	Allocated Quantity Less Than SADT	For each security (i.e., CUSIP + Security ID) and Allocation Code (CategoryCode), the sum of SR Total Short Quantity on Settlement Date by Security shall be greater than or equal to SADT Allocated Quantity, excluding the following conditions: a) Allocation Code is defined in ACAT as Unallocated (U), b) Allocation Code is defined in ACAT as Left Out of Allocation Process (X), or c) Quantity is zero on one side (i.e., SR or SADT) and missing on the other side.	Hard	5.2

7 A Brief Preview of the Data Quality Scorecard (DQS)

The validation home screen (the "DQS" screen) provides the status of the firm's flings, as seen below:

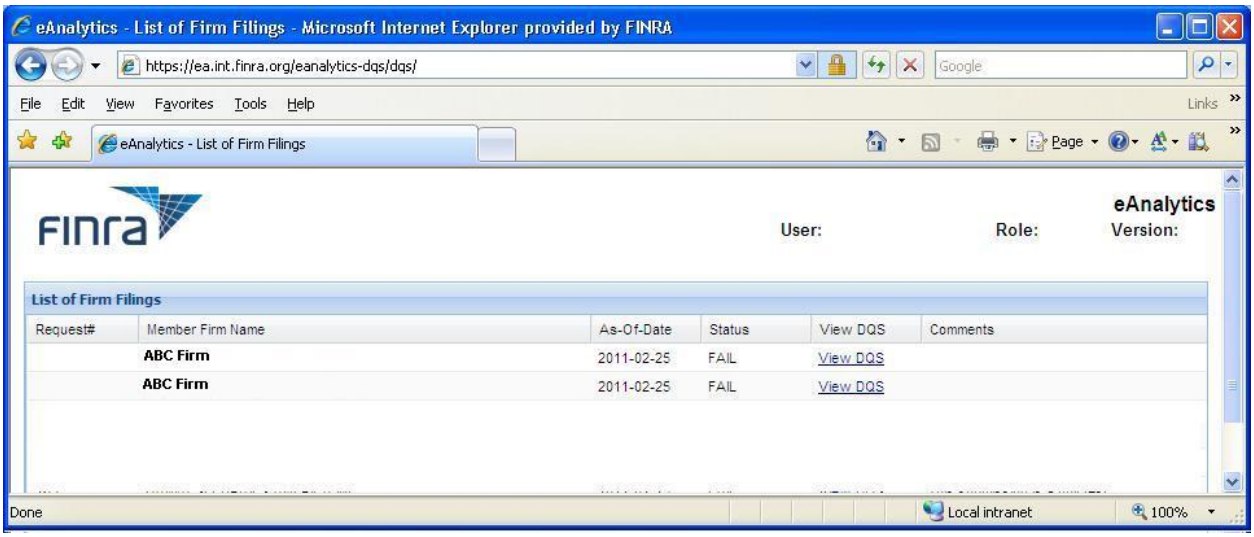


Figure 1

Filings that encounter any validation issues (whether XML or business validations) will be flagged with a status of 'FAIL' as seen below:

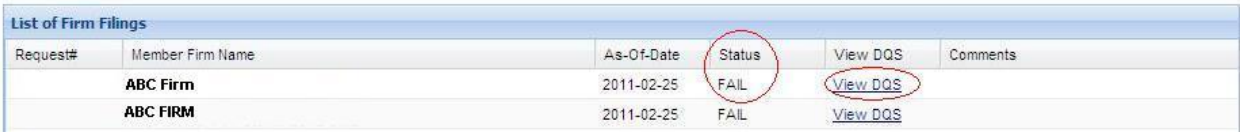


Figure 2

At this point a firm user can click on the 'View DQS' Link  next to the status to investigate the cause of the validation failure. The failures will be displayed in a new window that provides a description of the validation failure, the validation type (hard or soft), the number of instances of the failure, and the timestamp of the processing of the validation. See the example screen, below:

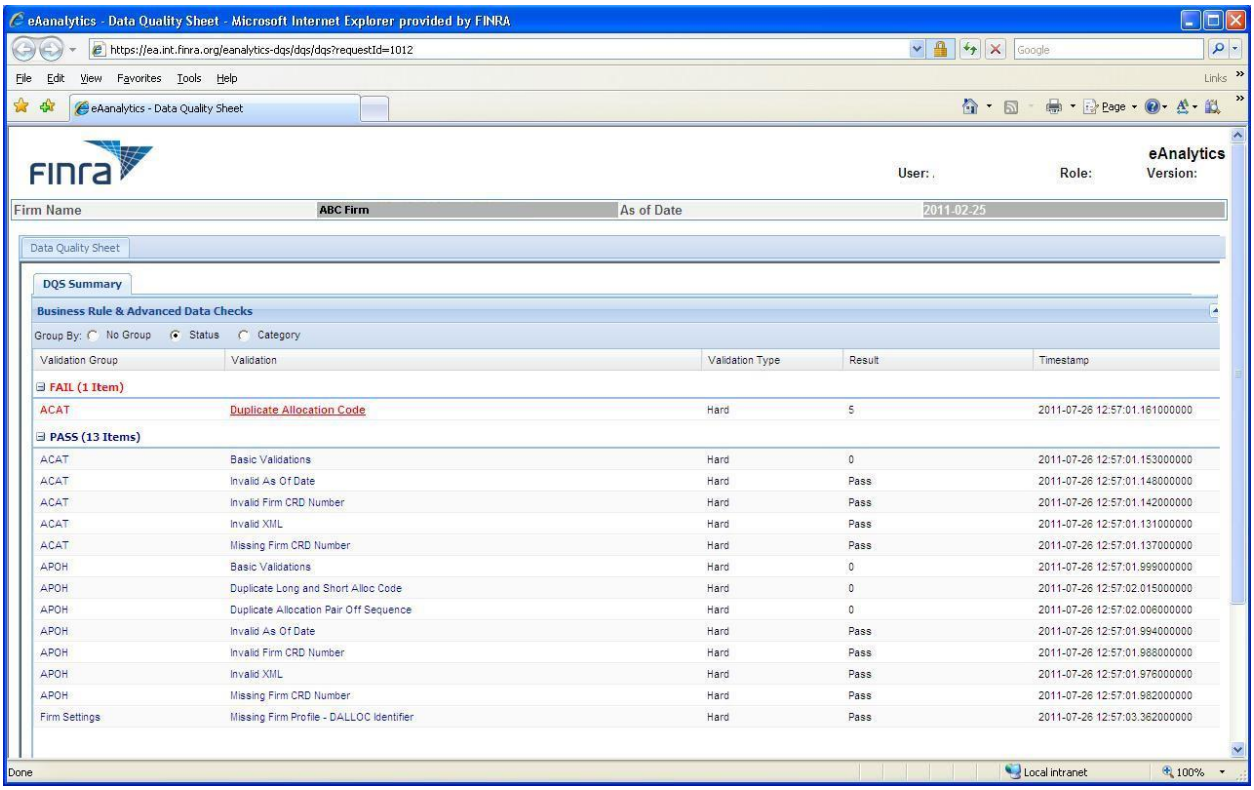


Figure 3

As seen above this particular filing has one hard failure due to 'Duplicate Allocation Code' in the ACAT File. To view the details of Duplicate allocation codes, click on the link highlighted in the Red underlined text Duplicate Allocation Code in the FAIL Items section. This will display a new tab next to the DQS Summary tab; the new tab will be named with the title of the validation. The display shows the total number of records in the file that caused this error and that need to be corrected by the firm before re-submission.

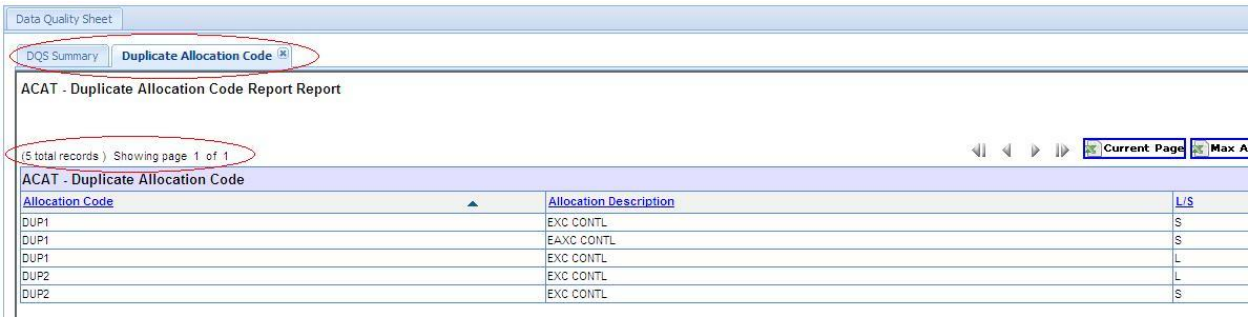


Figure 4

The DQS Summary screen (see Figure 3 above) offers several additional features:

1. The user may elect to group the display by *Category* (file type) or by *Status* (Success or Fail) by clicking on the radio button as seen below. (Selecting the *No Group* option displays the rows based upon the timing of the validations – i.e., when each validation was run.)

Figure 5

2. The user may elect to read the description of the validation failure: by rolling the cursor over the validation link (red underline text) Duplicate Allocation Code you can read a more detailed description for this validation (see example below):

Figure 6

8 Preparing the Files for Submission

When the data files are ready and the firm has ensured that the XML files are valid according to their corresponding XML schemas (XSDs[4]), the firm should submit the files to FINRA for processing.

8.1 Submission Methods and File Size Limitations

FINRA accepts file submissions through a Secure FTP transmission or a secure HTTP browser upload mechanism.

- Secure FTP file Transmission: FINRA's process is able to consume very large files. However, we have found that SFTP clients used by the firms can be inconsistent when transferring very large files, suffering occasional file transfer timeouts. If your firm's SFTP client experiences timeouts when transferring very large files, try breaking the file up into multiple smaller files. We typically do not see these problems with files in the 500MB size range.

- Secure Website/Browser/ HTTP upload: The size limit for browser upload is 55MB; any files larger than the specified size limit shall be submitted via the Secure FTP process.

8.2 Compressing the File

FINRA currently accepts compressed files in the following format:

- zip - Winzip 9.0 SR-1.

Only one file type (e.g., SR, SADT, ACAT) may be included in a single zip file.

A zip file cannot contain another zip file; FINRA will be unable to process compressed files with multiple levels.

8.3 Secure Request Manager (SRM) and the FINRA Request ID

FINRA will request the package of these files via the *Secure Request Manager (SRM)* application. FINRA Business shall publish this Request ID within two days from the announcement made by the Regulatory Coordinator for the Exam date to the firm. The firm is given a minimum two weeks to submit the data after the request is generated.

The Request IDs will be sent via an automated SRM system notification to the firm contacts that have established access accounts with FINRA Entitlements. A unique FINRA Request Number will identify each request for a given as-of-date. The Firm must use this unique Request ID in all file names when submitting the files to FINRA. File name formats are defined in the next section. Any files without a Request ID or files with an invalid Request ID will be routed to an invalid directory and will not be processed.

---

[4] See *Appendix D - XML Schema (XSD) and Firm-Side Validation.*

8.4 Naming the File

FINRA systems recognize the files by the FINRA-provided Request ID and routes each to the appropriate Firm Requests for further processing. Please ensure that each file (i.e., compressed container or single data file) shall be named as follows and contains the accurate FINRA Request ID and file Sequence number.

Files must be named with the convention:
<RequestNumber>_<SequenceNumber>_<FileName>. <FileExtension>.

Naming components are defined as follows:

- **<RequestNumber>** = FINRA provided Request Number. This is a 7 digit numeric Request ID identifying a unique exam request.

- **<SequenceNumber>** = The file sequence number for that date and file type.

  For example: If a single data file is broken into 10 smaller files, then the sequence number represents each file in sequential order. 1,2,3, …. 10. On the other hand if you provide only one single file then the sequence number will be defaulted to one (1).

- **<FileName>** = The schema file name (e.g., SR, SADT) for an XML or zipped XML file. For a text file text file this may be any free text used to identify the file.

- **<FileExtension>** = The file extension, such as ".xml", ".txt" or ".zip"

Data File	Schema File Name	Naming Convention of Container File (if zipped)	Naming Convention of Data Files
Full Stock Record	SR	<RequestNumber>_<SequenceNumber>_SR.zip	<RequestNumber>_<SequenceNumber>_SR.xml
Chart of Accounts	COA	<RequestNumber>_<SequenceNumber>_COA.zip	<RequestNumber>_<SequenceNumber>_COA.xml
Allocation Categories	ACAT	<RequestNumber>_<SequenceNumber>_ACAT.zip	<RequestNumber>_<SequenceNumber>_ACAT.xml
Allocation Hierarchy (Allocation Pair Off)	APOH	<RequestNumber>_<SequenceNumber>_APOH.zip	<RequestNumber>_<SequenceNumber>_APOH.xml
Allocation Category Summary	ACSM	<RequestNumber>_<SequenceNumber>_ACSM.zip	<RequestNumber>_<SequenceNumber>_ACSM.xml
Allocation Pair Off Summary	APOS	<RequestNumber>_<SequenceNumber>_APOS.zip	<RequestNumber>_<SequenceNumber>_APOS.xml
Security Allocation Details	SADT	<RequestNumber>_<SequenceNumber>_SADT.zip	<RequestNumber>_<SequenceNumber>_SADT.xml

9 Connections to FINRA for File Submission

Based on the file size defined above, the firm can choose to either submit files via the Secure FTP Transmission or the secure website HTTP Upload mechanism. This section provides information on how to connect to FINRA systems and submit your filings via SFTP or HTTP. This section also provides instructions to request new user accounts and password assistance in both Test and Production environments. Finally, this section contains URLs for the website to view Request IDs in SRM and the Data Quality Scorecard (DQS) to view results.

9.1 Establishing a Secure FTP Connection

In order to establish a secure FTP connection with FINRA, please ensure the following guidelines are met.

9.1.1 Obtain FTP User Account and Password from FINRA Entitlement

This section provides details on how to request a new FTP account from FINRA Entitlement and receive assistance with password or account resets, in case of password lockout.

The same FTP accounts may be used in both Production and Test environments.

9.1.1.1 New FTP Account Instructions

- Please fill out the FINRA Entitlement form:
  http://www.finra.org/web/groups/industry/@ip/@comp/documents/industry/p126932.pdf
- Get it signed by your Firm designated Super Administrator.
- Fax the form to FINRA at the number provided on the form.
- Log in here to change the FINRA-provided temporary Password:
  https://accountmgmt.finra.org/myews/changepassword.jsp
- You are now ready to log in to FINRA FTP with the changed password.

9.1.1.2 FTP Password Reset and Account Recovery Assistance

- Please call **FINRA Entitlements at (240) 386-4185** to request a new temporary Password.
- Log in here to change the FINRA provided temporary Password:
  https://accountmgmt.finra.org/myews/changepassword.jsp
- You are now ready to log in to FINRA FTP with the changed password.

****New Account Request:** Please note that you will require separate accounts for FTP and Website access. These cannot be used interchangeably. For Website accounts see instruction listed under section 9.2 below.*

9.1.2 Configure Source and Destination IP Addresses

In order to allow smooth and uninterrupted file transmission between Firm and FINRA, both the firm and FINRA Network teams will be required to configure the source (Firm) IP Addresses and destination (FINRA) IP Addresses into the network firewall configuration list.

FINRA IP addresses in the Production environment are provided in the following table along with Port and DNS Name. Please have your firm network team configure these settings to establish FTP connection. Also

please contact FINRA at FINOPS_FIRM_SUPPORT@finra.org to provide FINRA with your firm's source IP Addresses from which the files will be transmitted to the FINRA Server.

The following table details DNS Name, Port and FINRA IP Addresses for Secure FTP Connections.

SFTP Connections:					
File Transfers in this FINRA environment ….	…Coming From This Source Server at the Firm …	…Via…	… Should Be Configured to be able to Connect to this FINRA Destination Server		
			CONNECTION DETAILS		
			FINRA Destination IP Address	DNS Name	Port
SRM (Production)	Production SFTP	Internet	75.98.61.45	filetransfer.finra.org	22
	Production S/FTP	SAVVIS4	150.123.246.45	filetransfere.finra.org	21 or 22
SRM (QC/UAT)	UAT SFTP	Internet	198.202.241.38	filetransfer.ct.finra.org	22
	UAT S/FTP	SAVVIS4	150.123.247.38	filetransfere.ct.finra.org	21 or 22

You may also connect to FTP via HTTPS login:

File Transfers in this Environment	**HTTPS**
SRM (Production)	https://filetransfer.finra.org/

9.2 Submit Files via Website HTTP Upload

If your file size is less than or equal to 55 mb, you may upload the files using the FINRA Secure Request Manager (SRM) application accessible via a web browser. The same account may be used in Prod and QC/UAT.

Website HTTP Uploads:		
Environment	URLs	To get access to the Website
SRM (Prod)	https://aep.finra.org/srm/all/Srm/SrmRequestList.do	Contact your firm Super Account Administrator (SAA). Note: User account must have entitlements to "Secure Request Manager" and "eAnalytics" to access SRM.
SRM (QC/UAT)	https://aep.qc.finra.org/srm/all/Srm/SrmRequestList.do	

9.3 View Validation Results in Data Quality Scorecard (DQS)

To view the eAnalytics - Data Quality Scorecard (DQS) results for your submissions, following are the URLs. The same account may be used in Prod and QC/UAT.

Data Quality Scorecard:		
Environment	URLs	To get access to the Website
DQS (Prod)	https://ea.finra.org/eanalytics-dqs/dqs/	Contact your firm Super Account Administrator (SAA). Note: User account must have entitlements to "eAnalytics" to access the Data Quality Scorecard.
DQS (QC/UAT)	https://ea.uat.finra.org/eanalytics-dqs/dqs/	

Appendix A – FINRA Standard Chart of Accounts

Category Number	Name	Definition/Description
1	CUSTOMER (DVP/RVP)	Long and Short Security Positions in accounts that are awaiting delivery to or receipt from banks /broker dealers, other than those accounts of affiliated non-Broker Dealers
2	CUSTOMER (MANAGED ACCOUNTS)	Long and Short Security Positions in investment accounts managed by financial consultants who provide advisory services.
3	CUSTOMER (IRA ACCOUNTS)	Long and Short Security Positions in individual retirement accounts that allows the customer to contribute a defined dollar amount and have various investments.
4	CUSTOMER (EMPLOYEE)	Long and Short Security Positions in Employee accounts
5	CUSTOMER (PRIME BROKER)	Long and Short Security Positions in Prime Broker accounts where the Firm is the Prime Broker
6	CUSTOMER (AFFILIATE – NON BD)	Long and Short Security Positions in accounts of Affiliates that are not Broker Dealers
7	CUSTOMER (OMNIBUS)	Long and Short Security Positions in customer accounts grouped to form a unified omnibus account
8	CUSTOMER	Long and Short Security Positions not included in Other Customer Categories
9	NON-CUSTOMER (OFFICER/DIRECTOR)	Long and Short Security Positions in accounts of Officers/Directors of the Firm
10	NON-CUSTOMER (OFFICER/DIRECTOR – FAMILY MEMBERS)	Long and Short Security Positions in accounts of household members of Officers/Directors of the Firm
11	NON-CUSTOMER (AFFILIATE – BD)	Long and Short Security Positions in accounts of Foreign Broker Dealer Affiliates of the Firm
12	NON-CUSTOMER (OMNIBUS)	Long and Short Security Positions in Omnibus Non Customer accounts
13	NON CUSTOMER	Long and Short Security Positions of Non Customers not included in Other Non Customer Categories
14	PAIB	Long and Short Security Positions in proprietary accounts of Introducing Brokers and other Broker Dealers, including domestic affiliated Broker Dealers
15	FIRM INVENTORY	Long and Short Security Positions in proprietary accounts of the Firm
16	CNS FAILS	Open Long and Short Positions at NSCC
17	OTHER CLEARING CORP FAILS	Open Long and Short Positions at Clearing Corporations such as GSCC, EMCC, MBSCC and other clearing corps
18	EXECUTING PRIME BROKER ACCOUNTS	Open Long and Short positions in accounts where Firm acts as Executing Broker in a Prime Broker arrangement
19	BROKER FAILS TO RECEIVE	Open Short Positions awaiting receipt from other Broker Dealers.
20	BROKER FAILS TO DELIVER	Open Long Positions awaiting delivery to other Broker Dealers.
21	BROKER FAILS	Open Long and Short positions awaiting delivery or receipt to/from other Broker Dealers. **For Firms that use one dealer account to record fails to deliver and fails to receive.*
22	FAIL TO DELIVER GREATER THAN 30 DAYS	Open Long Positions awaiting delivery to other Broker Dealers aged more than 30 days

FINRA Standard Chart of Accounts continued		
Category Number	Name	Definition/Description
23	SECURITIES BORROWED	Securities borrowed for cash
24	SECURITIES BORROWED NON CASH	Securities borrowed for consideration other than cash, e.g. – other securities, LOC.
25	SECURITIES BORROWED CONDUIT	Securities borrowed for the purpose of lending to other Broker Dealers
26	SECURITIES LOANED	Securities loaned for cash
27	SECURITIES LOANED NON CASH	Securities Loaned for consideration other than cash, e.g. – other securities, LOC.
28	SECURITIES LOANED CONDUIT	Securities Loaned to other Broker Dealers that had been previously borrowed
29	SECURITIES LOANED CNS	Securities Loaned for cash to NSCC
30	REVERSE REPO	Securities purchased with an agreement to resell
31	REVERSE REPO (TRI-PARTY)	Securities purchased with the agreement to resell where the securities are held in segregation at a Custodian Bank
32	REPO	Securities sold with the agreement to repurchase
33	REPO (HOLD IN CUSTODY)	Securities sold with the agreement to repurchase where the securities are held in segregation at the Broker Dealer's Custodian Bank
34	REPO (TRI-PARTY)	Securities sold with the agreement to repurchase where the securities are held in segregation at a Custodian Bank
35	BANK LOAN FIRM	Proprietary Securities pledged to a bank for a cash loan
36	BANK LOAN CUSTOMER	Customer Securities pledged to a bank for a cash loan
37	BANK LOAN NON-CUSTOMER	Non Customer Securities pledged to a bank for a cash loan
38	BANK LOAN PAIB	Proprietary Securities of an Introducing Broker pledged to a bank for a cash loan
39	SUSPENSE	Long and Short Security Positions in accounts used to hold securities pending the determination of the proper account
40	REORGANIZATIONS	Long and Short Security Positions pending conversion related to corporate action(s) e.g. –Mergers, Tender Offers, Bond Conversions, Name Change, etc.
41	STOCK DIVIDENDS	Positions of dividend payments made in the form of additional shares of stock instead of cash.
42	ABANDONED PROPERTY	Securities that remain unclaimed after the appropriate escheatment period and the beneficial owner cannot be contacted
43	PHYSICAL BOX	Physical locations where securities are stored on premises e.g. – Vaults, Cage
44	LEGAL BOX	Securities in customer name awaiting legal papers
45	SAFEKEEPING	Securities physically held by the Firm in Customer Name
46	15C3-3 DEPOSIT	Collateral held to meet the Customer Reserve Formula Deposit Requirements as defined in SEA Rule 15c3-3
47	DOMESTIC DEPOSITORY – CONTROL	A domestic financial institution that acts as custodian for securities held by the Firm and its customers which has been approved by the SEC as a control location and meets the definition of a control location as per SEA Rule 15c3-3I
48	FOREIGN DEPOSITORY – CONTROL	A foreign financial institution that acts as custodian for securities held by the Firm and its customers which has been approved by the SEC as a control location and meets the definition of a control location as per SEA Rule 15c3-3I

FINRA Standard Chart of Accounts continued		
Category Number	Name	Definition/Description
49	DOMESTIC DEPOSITORY – NON-CONTROL	A domestic financial institution used as a depository for securities of the Firm and its customers but which has not been approved by the SEC as a control location or does not meet the definition of a control location as per SEA Rule 15c3-3I
50	FOREIGN DEPOSITORY – NON-CONTROL	A foreign financial institution used as a depository for securities of the Firm and its customers but which has not been approved by the SEC as a control location or does not meet the definition of a control location as per SEA Rule 15c3-3I
51	TRANSFER – CONTROL	Securities held by a transfer agent (including DTC) aged less than 40 days or aged greater than 40 days and confirmed by the transfer agent
52	TRANSFER – NON-CONTROL	Securities held by a transfer agent aged greater than 40 days and not confirmed by the transfer agent
53	TRANSFER – LEGAL	Transaction that requires documentation other than the standard stock or bond power to validate the transfer of stock certificates from a seller to a buyer, e.g. – securities registered to a corporation or deceased person. It is the selling broker's responsibility to supply proper documentation to the buying broker in a legal transfer
54	IN-TRANSIT – CONTROL	Securities in transit between Firm locations aged 5 business days or less
55	IN-TRANSIT – NON-CONTROL	Securities in transit between Firm locations aged greater than 5 business days
56	CLEARING DOMESTIC	Securities held in a domestic financial institution that are not control locations, as defined in 15c3-3I e.g. – Government securities clearance at JPMC or BONY etc., Options Clearance at OCC
57	CLEARING FOREIGN	Securities held in a foreign financial institution that are not control locations, as defined in 15c3-3I
58	TEST ACCOUNT	Long and Short Security positions in accounts used by Broker Dealer to perform testing
59	OUT OF BALANCE	Long and Short Security positions in stock record break or out of balance accounts
60	CONVERSION	Long and Short Security Positions in accounts used to record and monitor system or other conversions (e.g. – Switching data processing systems, addition of new correspondent)
61	DEPOSITS – OCC MARGIN	Securities pledged to the Options Clearing Corp to meet margin requirements
62	DEPOSITS – OTHER	Securities pledged to meet deposit requirements (Does not include OCC margin requirement deposits)
63	OTHER	Long and Short Security Positions in any accounts not identified in the categories listed should be included here

Appendix B – FINRA Security Type List

Security Type Code	Security Type Description
1	Bankers Acceptances
2	Certificates of Deposit
3	Commercial Paper
4	US Treasury Securities
5	US Government Agency Securities
6	Canadian Government Obligations
7	State and Municipal Government Obligations
8	Corporate Obligations
9	Asset Backed Securities
10	Private Label CMOs
11	Common Stock
12	Preferred Stock
13	Warrants
14	Mutual Funds
15	Money Market Instruments
16	Options
17	Other Securities
18	Currencies

Appendix C – What is XML?

XML is a mature technology standard for representing data in a manner that can be read easily by man or machine. Individual data elements in an XML file are surrounded by "tags" that identify the meaning of the element, and the elements are arranged in a hierarchical fashion that provides immediate visibility into the relationships among the elements.[5] Here is a simplified example from the eAnalytics Stock Record file:

```
<StockRecord AsOfDate="2011-06-03" FirmCrdNumber="36118" >
        <Security>
                <Cusip>000375.@*</Cusip>
                <SecurityDescription>ABB LTD JUN 18,2011 @ 27 CALL 100 MULTIPLIER</SecurityDescription>
                <SecurityType>16</SecurityType>
                <Price PriceIndicator="A">0.275</Price>
                <Currency>USD</Currency>
                <TotalLongSD>10</TotalLongSD>
                <TotalShortSD>10</TotalShortSD>
                <TotalLongTD>10</TotalLongTD>
                <TotalShortTD>10</TotalShortTD>
                <SegRequiredBySecurity>0</SegRequiredBySecurity>
                <TotalLongMKSD>275</TotalLongMKSD>
                <TotalShortMKSD>275</TotalShortMKSD>
                <TotalLongMKTD>275</TotalLongMKTD>
                <TotalShortMKTD>275</TotalShortMKTD>
                <Accounts>
                        <Account>
                                <AccountNumber>89165000</AccountNumber>
                                <BranchCode>SFPM</BranchCode>
                                <AccountName>SPIDERROCK CAPITAL MANAGEMENT, LLC</AccountName>
                                <AccountType>2</AccountType>
                                <LongPosSettle>7</LongPosSettle>
                                <ShortPosSettle>0</ShortPosSettle>
                                <LongMKSettle>192.5</LongMKSettle>
                                <ShortMKSettle>0</ShortMKSettle>
                                <LongPosTrade>7</LongPosTrade>
                                <ShortPosTrade>0</ShortPosTrade>
                                <LongMKTrade>192.5</LongMKTrade>
                                <ShortMKTrade>0</ShortMKTrade>
                                <SegRequiredByAccount>0</SegRequiredByAccount>
                                <LastActivitySettleDate>2011-05-02</LastActivitySettleDate>
                                <LastActivityTradeDate>2011-05-02</LastActivityTradeDate>
                                <AllocationCode>98</AllocationCode>
                                <COACode>008</COACode>
                        </Account>
                         <Account>
                                <AccountNumber>89470000</AccountNumber>
                                .                                .                .
                                (more data for another account containing this security)
                                .                .
                        </Account>
                </Accounts>
        </Security>
        < Security>
                .
                (data for another Security, including Accounts information)
                .
```

[5] Note that the whitespace (returns, tabs) *between* the elements is optional – an XML file *could* contain every single element all on the same line (albeit a very very long line) with no returns or tabs. XML editors offer a "Pretty Print" feature that will provide the missing returns and spacing, making the file much more readable.

```
        </Security>
        .
        (more Securities)
        .
</StockRecord>
```

You do not have to be a computer to understand what is being represented in this file. It is easy to see that there is a base (or "root") element called *StockRecord* that contains the rest of the data. Note that every piece of information in the file is represented as follows:

<Tag>data value</Tag>

<Tag> and </Tag> are referred to as the *open tag* and the *close tag*, respectively; those two tags surround the data value itself. So, the following line from our example lets a reader know that the Account Number for that Account is 89165000.

<AccountNumber>89165000</AccountNumber>

We can also gather from this file the hierarchy of the data:

Root element: ***StockRecord***
Contains: ***Security*** elements (a StockRecord may contain 1 or more Securities)
Which, in turn, contain: ***Account*** elements (a Security may contain 1 or more Accounts)

Appendix D – XML Schema (XSD) and Firm-Side Validation

XSDs are a sort of a contract between a data provider and a data consumer, where the data being exchanged is formatted in XML. XSDs provide an explicit format definition that allows the data provider (in the case of eAnalytics, the firm) to have a clear understanding of how the data must be formatted for the data to be considered acceptable. The eAnalytics application allows firms to submit a variety of different XML files (SR, SADT, ACAT, APOH, etc.); each of those XML files is governed by its own schema. The latest and greatest schema for each file type is always available at *http://www.finra.org/finops/stockrecord*.

When data is submitted to FINRA, it undergoes several kinds of validation. The first and most basic validation is: *is the submitted data valid XML and does the submitted data adhere to the XSD?*

This level of validation can be performed by the firm before submitting the data to FINRA, and FINRA recommends that firms take advantage of this feature of XML and perform the XML validation prior to submitting their data.

At a minimum, this kind of "local" validation is *strongly* recommended during the onboard and testing process; ideally, the local validation would be performed even as part of the production data submission process.[6] XSD validation will guarantee that:

- The XML is "well-formed" (e.g., all open tags have close tags, there are no invalid characters in the data. For more details see http://www.w3schools.com/xml/xml_syntax.asp), and
- The XML is "valid" with respect to the XSD (e.g., elements that the XSD says are required are provided, elements are in the correct order, etc.).

The whole reason for having the XSD is to allow the data provider (the firm) to do a sanity check of the data to make sure it passes those basic well-formed and valid checks *before* submitting that data to the data consumer (FINRA). There are many free and for-fee validation tools out there that will perform the validations, including libraries that the firm would use as part of its automated application and full standalone products (such as XML Spy, although XML Spy may not be appropriate for working with very large files). The alternative – expecting the consumer (FINRA in this case) to give the submitter (the firm) feedback on its XML validation failures – is a long and frustrating process because often the XML validation will hit an error in the file…and stop processing. The XML validation functionality will report on that first error, and because the data consumer (FINRA in this case) expects the submitter to have performed the XML validation prior to submission, the consumer has not invested a tremendous amount of effort in detailed, understandable error messaging here (the consumer really should not see any of these errors!). So the firm has to fix the error with its data, re-upload the files, and trigger the FINRA-side processing again. This time, the XML validation goes to the *next* element and finds another error…and stops. It reports that error. This cycle can repeat over and over. It's much more efficient to run the XSD validation locally, as part of the firm's process and work these issues out before even submitting the data to FINRA.

[6] Assuming the firm's testing process is extensive and addresses all foreseen permutations of data, it may make sense to perform the in-house/local validation during the testing cycle only, and then turn off that local validation for production runs.

Appendix E – XSD/XML Versioning

Data requirements change, they evolve over time. As a data consumer (in the case of the Stock Record application, FINRA is the data consumer) identifies additional data elements that it wishes to receive, it will add those new elements to the XSD. Typically, the new element will be *optional* for some period of time, and then, assuming the information is necessary for business purposes, the element will become *required*. It goes without saying that strong data exchange partners work hard to minimize the changes to their interface. FINRA is committed to minimizing the change work required by data submitters, while at the same time gathering the information it needs to execute on its mission.

As a general rule, FINRA will support two schemas – whatever is the current version, and one previous version. As of Q3 2011, FINRA will accept only those submissions that adhere to the following XSDs (see table, below); data that does not agree with these XSDs will fail validation and have to be recreated and resubmitted. Again, the latest schemas are available at *http://www.finra.org/finops/stockrecord***.**

File Type	Latest Schema			Previous Schema (still valid)		
	Vrsn	Schema File Name	Date In Service	Vrsn	Schema File Name	Date In Service
ACAT	6	AEP_FinOPs_ACAT_V6_annotated.xsd or AEP_FinOPs_ACAT_V6.1_annotated.xsd	08/15/2011 11/1/2011	5	AEP_FinOPs_pilot_ACAT_V5.xsd	09/13/2008
ACSM	7	AEP_FinOPs_ACSM_V7_annotated.xsd	08/15/2011	6	AEP_FinOPs_pilot_ACSM_V6.xsd	09/13/2008
APOH	6	AEP_FinOPs_APOH_V6_annotated.xsd	08/15/2011	5	AEP_FinOPs_pilot_APOH_V5.xsd	09/13/2008
APOS	8	AEP_FinOPs_APOS_V8_annotated.xsd	08/15/2011	7	AEP_FinOPs_APOS_V7.xsd	05/21/2009
COA	6	AEP_FinOPs_COA_V6_annotated.xsd	08/15/2011	5	AEP_FinOPs_pilot_COA_V5.xsd	09/13/2008
SADT	9	AEP_FinOPs_SADT_V9_annotated.xsd	08/15/2011	8	AEP_FinOPs_SADT_V8.xsd	07/10/2010
SR	9.2	AEP_FinOPs_SR_V9.2_annotated.xsd	07/01/2013	9	AEP_FinOPs_SR_V9_annotated.xsd Or AEP_FinOPs_SR_V9.1_annotated.xsd	08/15/2011 11/1/2011

Appendix F – XSD Version History

The table below provides an indication of the changes made to each schema (XSD) file as that file was updated to the subsequent version.

FINRA's plan, going forward, is to always support the most recent version of a given XSD file *and one prior version of that XSD*. Versions older than that one prior version will not be supported.

Move From…	To…	Changes
ACAT		

AEP_FinOPs_pilot_ACAT_Draft_V1.xsd	AEP_FinOPs_pilot_ACAT_V5.xsd	• Removed *use="required"* from *FirmFocusNumber* attribute to */AllocationCategories*
		  o Makes that root element attribute optional
		• Added optional attribute *FirmCrdNumber* to */AllocationCategories*
		  o Supports Firm CRD Number as attribute on that root element

AEP_FinOPs_pilot_ACAT_V5.xsd	AEP_FinOPs_ACAT_V6.xsd *[sic]*	• Added Annotations/Documentation to XSD
		• Changed type for */AllocationCategories/AllocationCategory/CategoryCode* and */AllocationCategories/AllocationCategory/CategoryDescription* elements from custom type *RequiredString* to custom type *DQS.NonEmptyString*.
		  o Only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space. Should not affect how firms create/provide this element value.
		• Changed type for */AllocationCategories/AllocationCategory/LongShortIndicator* element from custom type *RequiredString* to custom type *DQS.LongShortIndicator*.
		  o Whereas *RequiredString* allowed any value of at least one character, *DQS.LongShortIndicator* forces the *LongShortIndicator* value to be one of the following. Note that although the V5 XSD allowed any character, the application has always expected to see an "L" or an "S".
		  ▪ **L** Long security position – e.g., Stock Borrow would be flagged with an "L" since it is a Long allocation code
		  ▪ **S** Short security position – e.g., Stock Loan would be flagged with an "S" since it is a Short allocation code
		  ▪ **B** Both – code may be used as Long and Short
		  ▪ **U** Unallocated – special code for use by the very few firms that explicitly identify unallocated positions, often using 99*, ZZ*, or XX* Allocation Category Codes. Those "unallocated" Allocation Category Codes should be identified by setting the LongShortIndicator to "U".
		• Removed */AllocationCategories/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.)
		  o Firms must not provide that *@FirmFocusNumber* attribute at all.
		• Changed type for */AllocationCategories/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required.

		○ CRD number is always a number and it should always be provided, so this change should not affect how firms create/provide this element value.
AEP_FinOPs_ACAT_V6.xsd	AEP_FinOPs_ACAT_V6.1.xsd	• Added "X" to allowed values for */AllocationCategories/AllocationCategory/LongShortIndicator* element: ○ **X** Exclude from Allocation Process – special code for use by those firms that identify on the SR positions that should not be included in the Allocation Pairoff process.
ACSM		
AEP_FinOPs_pilot_ACSM_Draft_V1.xsd	AEP_FinOPs_pilot_ACSM_V5.xsd	• Removed *use="required"* from *FirmFocusNumber* attribute to */AllocationCategorySummaries* ○ Makes that root element attribute optional • Added optional attribute *FirmCrdNumber* to */AllocationCategorySummaries* ○ Supports Firm CRD Number as attribute on that root element • Changed type for */AllocationCategorySummaries/AllocationCategorySummary/TotalLongMarketValue* and *TotalShortMarketValue* from custom type *RequiredFloat* to XML standard type *xs:float* ○ Unclear why they used a custom type in the first place. Here is the deprecated custom type: <xs:simpleType name="RequiredFloat"> <xs:restriction> <xs:simpleType> <xs:list itemType="xs:float"/> </xs:simpleType> <xs:minLength value="1"/> <xs:maxLength value="1"/> </xs:restriction> </xs:simpleType>
AEP_FinOPs_pilot_ACSM_V5.xsd	AEP_FinOPs_pilot_ACSM_V6.xsd	• Changed */AllocationCategorySummaries/AllocationCategory/TotalLongQuantity* and *TotalShortQuantity* from *xs:long* to *xs:float*
AEP_FinOPs_pilot_ACSM_V6.xsd	AEP_FinOPs_ACSM_V7.xsd *[sic]*	• Added Annotations/Documentation to XSD • Changed type for */AllocationCategorySummaries/AllocationCategorySummary/CategoryCode* element from custom type *RequiredString* to custom type *DQS.NonEmptyString*. ○ Only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space. Should not affect how firms create/provide this element value. • Changed type for */ AllocationCategorySummaries/AllocationCategorySummary/CategoryDescription* from custom type *RequiredString* to XML standard type *xs:normalizedString*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value. • Changed type for */AllocationCategorySummaries/AllocationCategorySummary/TotalLongQuantity, TotalShortQuantity, TotalLongMarketValue,* and *TotalShortMarketValue* from XML standard type *xs:float* to custom type *DQS.NonNegativeDecimal*. ○ The latter type is, at core, a decimal value that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers or scientific notation values (e.g., "1.03E5") for these elements, this aspect should have no effect on firm data.

		• Removed */AllocationCategorySummaries/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.)
		  ○ Firms must not provide that *@FirmFocusNumber* attribute at all.
		• Changed type for */AllocationCategorySummaries/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required.
		  ○ CRD number is always a number and it should always be provided, so this change should not affect how firms create/provide this element value.

APOH		

AEP_FinOPs_pilot_APOH_Draft_V1.xsd	AEP_FinOPs_pilot_APOH_V5.xsd	• Removed *use="required"* from *FirmFocusNumber* attribute to */AllocationPairoffHierarchy*
		  ○ Makes that root element attribute optional
		• Added optional attribute *FirmCrdNumber* to */AllocationPairoffHierarchy*
		  ○ Supports Firm CRD Number as attribute on that root element

AEP_FinOPs_pilot_APOH_V5.xsd	AEP_FinOPs_APOH_V6.xsd *[sic]*	• Added Annotations/Documentation to XSD
		• Changed type for */AllocationPairoffHierarchy/AllocationPairoffs/Sequence* from XML standard type *xs:int* to custom type *DQS.NonNegativeInteger*.
		  ○ The latter type is, at core, an integer that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers for this element, this aspect should have no effect on firm data.
		• Changed type for */AllocationPairoffHierarchy/AllocationPairoffs/CategoryCodeLong* and *CategoryCodeShort* elements from custom type *RequiredString* to custom type *DQS.NonEmptyString*.
		  ○ Only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space. Should not affect how firms create/provide this element value.
		• Removed */AllocationPairoffHierarchy/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.)
		  ○ Firms must not provide that *@FirmFocusNumber* attribute at all.
		• Changed type for */AllocationPairoffHierarchy/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required.
		  ○ CRD number is always a number and it should always be provided, so this change should not affect how firms create/provide this element value.

APOS		

AEP_FinOPs_pilot_APOS_Draft_V1.xsd	AEP_FinOPs_pilot_APOS_V5.xsd	• Removed *use="required"* from *FirmFocusNumber* attribute to */AllocationPairoffSummaries*
		  ○ Makes that root element attribute optional
		• Added optional attribute *FirmCrdNumber* to */AllocationPairoffSummaries*
		  ○ Supports Firm CRD Number as attribute on that root element
		• Removed *minOccurs="0"* from */AllocationPairoffSummaries/AllocationPairoffSummary/AllocatedValue*
		  ○ Makes that element a required element

AEP_FinOPs_pilot_APOS_V5.xsd	AEP_FinOPs_pilot_APOS_V6.xsd	• Changed */AllocationPairoffSummaries/AllocationPairoffSummary/AllocatedQuantity* from *xs:long* to *xs:float*
AEP_FinOPs_pilot_APOS_V6.xsd	AEP_FinOPs_ APOS_V7.xsd *[sic]*	• Changed */AllocationPairoffSummaries/AllocationPairoffSummary/AllocatedValue* from *xs:float* to following *xs:choice*: <xs:choice> <xs:element name=”AllocatedValue” type=”xs:float” /> <xs:sequence> <xs:element name=”LongAllocatedValue” type=”xs:float” /> <xs:element name=”LongAllocatedValueType” type=”AllocatedValueType” /> <xs:element name=”ShortAllocatedValue” type=”xs:float” /> <xs:element name=”ShortAllocatedValueType” type=”AllocatedValueType” /> </xs:sequence> </xs:choice> ○ Requires *either* a single float value for *AllocatedValue* (which is same as previous version) *or* both a *LongAllocatedValue* (and *LongAllocatedValueType* [“C” or “M” or blank or empty string]) and a *ShortAllocatedValue* (and *ShortAllocatedValueType* [“C” or “M” or blank or empty string]).
AEP_FinOPs_ APOS_V7.xsd	AEP_FinOPs_ APOS_V8.xsd	• Added Annotations/Documentation to XSD • Changed type for */AllocationPairoffSummaries/AllocationPairoffSummary/Sequence* from XML standard type *xs:int* to custom type *DQS.NonNegativeInteger*. ○ The latter type is, at core, an integer that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers for this element, this aspect should have no effect on firm data. • Changed type for */ AllocationPairoffSummaries/AllocationPairoffSummary/CategoryCodeLong* and *CategoryCodeShort* elements from custom type *RequiredString* to custom type *DQS.NonEmptyString*. ○ Only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space. Should not affect how firms create/provide this element value. • Changed type for */ AllocationPairoffSummaries/AllocationPairoffSummary/AllocatedQuantity, AllocatedValue, LongAllocatedValue* and *ShortAllocatedValue* elements from XML standard type *xs:float* to custom type *DQS.NonNegativeDecimal*. ○ The latter type is, at core, a decimal that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers or scientific notation values (e.g., “1.03E5”) for this element, this aspect should have no effect on firm data. • Removed */AllocationPairoffSummaries/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.) ○ Firms must not provide that *@FirmFocusNumber* attribute at all. • Changed type for */AllocationPairoffSummaries/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required. ○ CRD number is always a number and it should always be provided, so this change should not affect how firms create/provide this element value. • Made a change to *AllocatedValueType* definition – based it upon *xs:normalizedString* instead of *xs:string*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide these values.

COA		

AEP_FinOPs_pilot_COA_Draft_V1.xsd	AEP_FinOPs_pilot_COA_V5.xsd	• Removed *use="required"* from *FirmFocusNumber* attribute to */ChartOfAccountCategories*
		  ○ Makes that root element attribute optional
		• Added optional attribute *FirmCrdNumber* to */ChartOfAccountCategories*
		  ○ Supports Firm CRD Number as attribute on that root element

AEP_FinOPs_pilot_COA_V5.xsd	AEP_FinOPs_COA_V6.xsd *[sic]*	• Added Annotations/Documentation to XSD
		• Changed type for */ChartOfAccountCategories/ChartOfAccountCategory/CategoryDescription* from custom type *RequiredString* to XML standard type *xs:normalizedString*.
		  ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value.
		• Changed type for */ChartOfAccountCategories/ChartOfAccountCategory/FromAccountNumber, ToAccountNumber,* and *SpecificAccountNumber* from XML standard type *xs:string* to XML standard type *xs:normalizedString*.
		  ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value.
		• Removed */ChartOfAccountCategories/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.)
		  ○ Firms must not provide that *@FirmFocusNumber* attribute at all.
		• Changed type for */ChartOfAccountCategories/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required.
		  ○ CRD number is always a number and it should always be provided, so this change should not affect how firms create/provide this element value.

SADT		

AEP_FinOPs_pilot_SADT_Draft_V1.xsd	AEP_FinOPs_pilot_SADT_V5.xsd	• Removed *minOccurs="0"* from */SecurityAllocationDetails/SecurityAllocationDetail/Cusip*
		  ○ Makes that element a required element
		• Removed *minOccurs="0"* from */SecurityAllocationDetails/SecurityAllocationDetail/SecurityId*
		  ○ Makes that element a required element
		• Removed *minOccurs="0"* from */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValue*
		  ○ Makes that element a required element
		• Removed *minOccurs="0"* from */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValueType*
		  ○ Makes that element a required element
		• Changed type on */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValueType* from *xs:string* to custom type *AllocValueType*:
		`<xs:simpleType name="AllocValueType">`
		`<xs:restriction base="xs:string">`
		`<xs:enumeration value=""/>`
		`<xs:enumeration value="C"/>`
		`</xs:restriction>`
		`</xs:simpleType>`

		• Removed *use="required"* from *FirmFocusNumber* attribute to */SecurityAllocationDetails*
		  ○ Makes that root element attribute optional
		• Added optional attribute *FirmCrdNumber* to */SecurityAllocationDetails*
		  ○ Supports Firm CRD Number as attribute on that root element

AEP_FinOPs_pilot_SADT_V5.xsd	AEP_FinOPs_pilot_SADT_V6.xsd	• Reinstated *minOccurs="0"* from */SecurityAllocationDetails/SecurityAllocationDetail/Cusip*
		  ○ Makes that element optional
		• Reinstated *minOccurs="0"* from */SecurityAllocationDetails/SecurityAllocationDetail/SecurityId*
		  ○ Makes that element optional
		• Changed */SecurityAllocationDetails/SecurityAllocationDetail/LongQuantity* from *xs:long* to *xs:float*
		• Changed */SecurityAllocationDetails/SecurityAllocationDetail/ShortQuantity* from *xs:long* to *xs:float*
		• Changed */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedQuantity* from *xs:long* to *xs:double*
		• Added *minOccurs="0"* to */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValue*
		  ○ Makes that element optional
		• Added *minOccurs="0"* to */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValueType*
		  ○ Makes that element optional

AEP_FinOPs_pilot_SADT_V6.xsd	AEP_FinOPs_SADT_V7.xsd *[sic]*	• Changed */SecurityAllocationDetails/SecurityAllocationDetail/AllocatedValue* and *AllocatedValueType* into an *xs:choice* as follows: FROM: <xs:element name="AllocatedValue" type="xs:float" minOccurs="0"/> <xs:element name="AllocatedValueType" type="AllocValueType" minOccurs="0"/> TO: <xs:choice> <xs:sequence> <xs:element name="AllocatedValue" type="xs:float" /> <xs:element name="AllocatedValueType" type="AllocValueType" /> </xs:sequence> <xs:sequence> <xs:element name="LongAllocatedValue" type="xs:float" /> <xs:element name="LongAllocatedValueType" type="AllocValueType" /> <xs:element name="ShortAllocatedValue" type="xs:float" /> <xs:element name="ShortAllocatedValueType" type="AllocValueType" /> </xs:sequence> </xs:choice>

		○ Requires *either* a single float value for *AllocatedValue* and an *AllocatedValueType* (which is same data structure as previous version; however, previous version had the *minOccurs"0"* on both those elements which made them optional) ***or*** both a *LongAllocatedValue* (and *LongAllocatedValueType* [see below]) and a *ShortAllocatedValue* (and *ShortAllocatedValueType* [see below]). • Changed custom type *AllocValueType*: *AllocValueType* defined as "C" or empty string above in AEP_FinOPs_pilot_SADT_V5.xsd; here *AllocValueType* is expanded to allow "M" and blank, too
AEP_FinOPs_SADT_V7.xsd *[sic]*	AEP_FinOPs_SADT_V8.xsd *[sic]*	• Added *minOccurs="1"* to */SecurityAllocationDetails/SecurityAllocationDetail/LongQuantity*, *ShortQuantity*, and *AllocatedQuantity*. That is the same as the default behavior (i.e., if there is no *minOccurs* or *maxOccurs* attribute, XSD assumes a value of "1"), so these changes really were unnecessary and had no effect.
AEP_FinOPs_SADT_V8.xsd	AEP_FinOPs_SADT_V9.xsd	• Added Annotations/Documentation to XSD • Changed type for / *SecurityAllocationDetails/FileControl/@Version* from XML standard type *xs:string* to XML standard type *xs:normalizedString*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value. • Changed type for / *SecurityAllocationDetails/SecurityAllocationDetail/Cusip* and *SecurityId* from XML standard type *xs:string* to XML standard type *xs:normalizedString*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value. • Changed type for / *SecurityAllocationDetails/SecurityAllocationDetail/SecurityDescription* from custom type *RequiredString* to XML standard type *xs:normalizedString*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value. • Changed type for / *SecurityAllocationDetails/SecurityAllocationDetail/CategoryCodeLong* and *CategoryCodeShort* elements from custom type *RequiredString* to custom type *DQS.NonEmptyString*. ○ Only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space. Should not affect how firms create/provide this element value. • Changed type for / *SecurityAllocationDetails/SecurityAllocationDetail/LongQuantity, ShortQuantity, AllocatedQuantity, AllocatedValue, LongAllocatedValue,* and *ShortAllocatedValue* elements from XML standard type *xs:float* to custom type *DQS.NonNegativeDecimal* (*AllocatedQuantity* was *xs:double*, not *xs:float*) ○ The latter type is, at core, a decimal that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers or scientific notation values (e.g., "1.03E5") for this element, this aspect should have no effect on firm data. • Removed */SecurityAllocationDetails/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.) ○ Firms must not provide that *@FirmFocusNumber* attribute at all. • Changed type for */SecurityAllocationDetails/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required. ○ CRD number is always a number and it should always be provided, so this change should not affect how firms create/provide this element value.

		• Made a change to *AllocValueType* definition – based it upon *xs:normalizedString* instead of *xs:string*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide these values.
SR		
AEP_FinOPs_pilot_SR_Draft_V2.xsd	AEP_FinOPs_pilot_SR_V5.xsd	• Added optional attribute *FirmCrdNumber* to */StockRecord* ○ Supports Firm CRD Number as attribute on that root element
AEP_FinOPs_pilot_SR_V5.xsd	AEP_FinOPs_pilot_SR_V6.xsd	• Removed *use="required"* from *FirmFocusNumber* attribute to */StockRecord* ○ Makes that root element attribute optional • Changed all of the following elements found at */StockRecord/Security* from *xs:long* to *xs:float*: *TotalLongSD* *TotalShortSD* *TotalLongTD* *TotalShortTD* • Changed all of the following elements found at */StockRecord/Security/Accounts/Account* from *xs:long* to *xs:float*: *LongPosSettle* *ShortPosSettle* *LongPosTrade* *ShortPosTrade*
AEP_FinOPs_pilot_SR_V6.xsd	AEP_FinOPs_SR_V7.xsd *[sic]*	• Added *AllocationCode* and *COACode* elements to */StockRecord/Security/Accounts/Account*.
AEP_FinOPs_SR_V7.xsd *[sic]*	AEP_FinOPs_SR_V8.xsd *[sic]*	• Changed */StockRecord/Security/SegRequiredBySecurity* from *xs:long* to *xs:float*.

AEP_FinOPs_SR_V8.xsd	AEP_FinOPs_SR_V9.xsd	• Added Annotations/Documentation to XSD
		• Changed type for / *StockRecord/FileControl/@Version*, */StockRecord/Security/Cusip, /StockRecord/Security/SecurityId, /StockRecord/Security/SecuritySymbol, /StockRecord/Security/Factor, /StockRecord/Security/Currency, /StockRecord/Security/ExchangeRate, /StockRecord/Security/Accounts/Account/BranchCode, /StockRecord/Security/Accounts/Account/AllocationCode, /StockRecord/Security/Accounts/Account/MemoField/MemoFieldType* and *MemoFieldQuantity* from XML standard type *xs:string* to XML standard type *xs:normalizedString*.
		  ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value.
		• Changed type for / *StockRecord/Security/SecurityDescription* element from XML standard type *xs:string* to custom type *DQS.NonEmptyString*.
		  ○ As its name implies, *DQS.NonEmptyString* requires at least one character to be present – no empty strings. Other than that, the only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space.
		• Changed */StockRecord/Security/Price* from an in-line type definition to a type reference definition. Changed base type on Price from XML standard type *xs:float* to custom type *DQS.NonNegativeDecimal.* Changed the *@PriceIndicator* attribute to *use="required"*.

		○ Firms that have submitted *Price* information using scientific notation (i.e., "1.04E2" for 104) must submit using regular decimal values; scientific notation is invalid going forward. ○ All *Price* data must have the *@PriceIndicator* attribute provided (this value has always been documented as Required). • Changed type for */StockRecord/Security/TotalLongSD, TotalShortSD, TotalLongTD, TotalShortTD, SegRequiredBySecurity,* elements and */StockRecord/Security/Accounts/Account/LongPosSettle, ShortPosSettle, LongPosTrade, ShortPosTrade* from XML standard type *xs:float* to custom type *DQS.NonNegativeDecimal* ○ Firms that have submitted values for these elements using scientific notation (i.e., "1.04E2" for 104) must submit using regular decimal values; scientific notation is invalid going forward. ○ *DQS.NonNegativeDecimal* type is, at core, a decimal value that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers for this element, this aspect should have no effect on firm data. • Changed type for */StockRecord/Security/AllocatedPrice, TotalLongMKSD, TotalShortMKSD, TotalLongMKTD, TotalShortMKTD* and */StockRecord/Security/Accounts/Account/LongMKSettle, ShortMKSettle, LongMKTrade, ShortMKTrade, SegRequiredByAccount* elements from custom type *RequiredFloat* to custom type *DQS.NonNegativeDecimal* ○ Firms that have submitted values for these elements using scientific notation (i.e., "1.04E2" for 104) must submit using regular decimal values; scientific notation is invalid going forward. ○ *DQS.NonNegativeDecimal* is, at core, a decimal that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers for this element, this aspect should have no effect on firm data. • Changed type for */StockRecord/Security/GrandTotalLongSD, GrandTotalShortSD, GrandTotalLongTD, GrandTotalShortTD, TotalNumberOfSecurities* elements from XML standard type *xs:decimal* to custom type *DQS.NonNegativeDecimal.* ○ The latter type is, at core, a decimal that must be greater than or equal to 0. (The older XSD allowed negative numbers to be provided.) Assuming firms never provided negative numbers for this element, this aspect should have no effect on firm data. • Changed type for / *StockRecord/Security/Accounts/Account/AccountNumber*, */StockRecord/Security/Accounts/Account/AccountType* elements from custom type *RequiredString* to custom type *DQS.NonEmptyString*. ○ Only difference is the use of *xs:whitespace collapse* setting – main effect is that multiple contiguous whitespace characters in a row (e.g., multiple space characters) will be treated as one space. Should not affect how firms create/provide this element value. • Changed type for / *StockRecord/Security/Accounts/Account/AccountName* elements from custom type *RequiredString* to XML standard type *xs:normalizedString*. ○ With *xs:normalizedString*, all whitespace characters within the attribute value are interpreted as spaces. Should not affect how firms create/provide this element value. • Changed custom type definition *SecurityTypes* from an *xs:int* with *xs:minInclusive* and *xs:maxInclusive* values of 1 and 18, respectively, to an enumeration that lists those 18 values out as the allowed enum values. ○ Should not affect how firms provide this element value. • Removed */StockRecord/@FirmFocusNumber* attribute. (Note that the use of *@FirmFocusNumber* has been deprecated for some time – firms should be using *@FirmCrdNumber* instead.) ○ Firms must not provide that *@FirmFocusNumber* attribute at all. • Changed type for / *StockRecord/@FirmCrdNumber* from XML standard type *xs:string* to XML standard type *xs:int*, and made the attribute required. ○ CRD number is always a number and it should always be provided, so this change should not affect how firms

		create/provide this element value.
AEP_FinOPs_SR_V9.xsd	AEP_FinOPs_SR_V9.1.xsd	• Changed type for */StockRecord/Security/Accounts/Account/AllocationCode* from XML standard type *xs:normalizedString* to custom type *CollapsingString*. ○ With *CollapsingString*, multiple contiguous spaces are treated as one space, which agrees with the approach we are using on other files. Should not affect how firms create/provide this element value.
AEP_FinOPs_SR_V9.1.xsd	AEP_FinOPs_SR_V9.2.xsd	• Removed Threshold Indicator field. • Changed type for *StockRecord/Security/Factor* from string to DQS.NonNegativeDecimal. • Updated annotations for text fields to include maximum length.

Appendix G – Data Guide Version History

The table below provides an indication of the changes made with each version of this document.

Move From…	To…	Effective Date	Changes
Version 5	Version 5.1		• Section 2 Glossary – Added definitions for Unallocated and Unmapped positions. • Section 6.2 - New Validations and the New Data Quality Scorecard (DQS) – grammar changes and added paragraph noting that the process performs only those validations related to the specific set of files provided with the submission. • Section 4 Overarching Guidelines – added entry regarding the problems with using *float* or *double* data types. • Section 5.3 ACAT – updated ACAT information to include the availability of "X" as an allowed value for the LongShortIndicator field. Updated the table of ACAT field descriptions and the diagram of the ACAT schema. • Appendix F XSD Versioning History – updated to reflect the changes in the ACAT schema from V6 to V6.1, which allows the "X" value for the LongShortIndicator field.
Version 5.1	Version 5.2		• Section 6.2 - Updated Validations. • Removed references of MAPP, SCOA and DALLOC Files throughout, since these are no longer required as a separate file. These codes shall be provided in the Stock Record file going forward.
Version 5.2	Version 5.3		• Section 6.2 - Updated Validations.
Version 5.3	Version 5.4		• Section 6.2 - Updated Validations for Release 2012.01.02 (March).
Version 5.4	Version 5.5		• Section 6.2 - Updated Validations for Release 2012.02.01 (May).
Version 5.5	Version 5.6		• Section 6.2 - Updated FINRA Team and Contact information • Section 6.2 - Updated Validations for Release 2012.04.01 (December).
Version 5.6	Version 5.7		• Section 5.1 - Removed element 'Threshold Indicator' from the SR File. • Section 6.2 - Updated Validations for Release 2013.01.01 (April).
Version 5.7	Version 5.8		• Section 6.2 - Updated Validations for Release 2013.02.01 (June). • Section 5.1 - Updated SR XSD to 9.2. • Section 5 - Updated text field annotations to include maximum length.
Version 5.8	Version 5.9	12/13/2014	• Section 6.2 - Updated Validations for Release 2013.04.01 (Dec). • Section 8 - Updated to add more details regarding file submission, compression and naming. • Section 9 - Added NEW section for FINRA file submission connections and accounts.

Version 5.9	Version 5.10	2/14/2014	• Section 9.2 and 9.3 - Updated to use the same account in PROD and QC/UAT environments.
Version 5.10	Version 5.11	3/31/2014	• Appendix G – Data Guide Versioning History Added effective date of change in PROD. • Section 5.1 - Updated SR XSD annotation for Account Number to remove this sentence: "The combination of Account Number and Account Type must be unique within a given security container." Note: XSD structure was not changed and this rule has not been in effect for some time. This was a correction to the description only.
Version 5.11	Version 5.12	6/30/2014	• Section 6.2 – Updated Validations for Release 2014.02.01 (June).