TI 301D: XRF Data Processing

Table of Contents

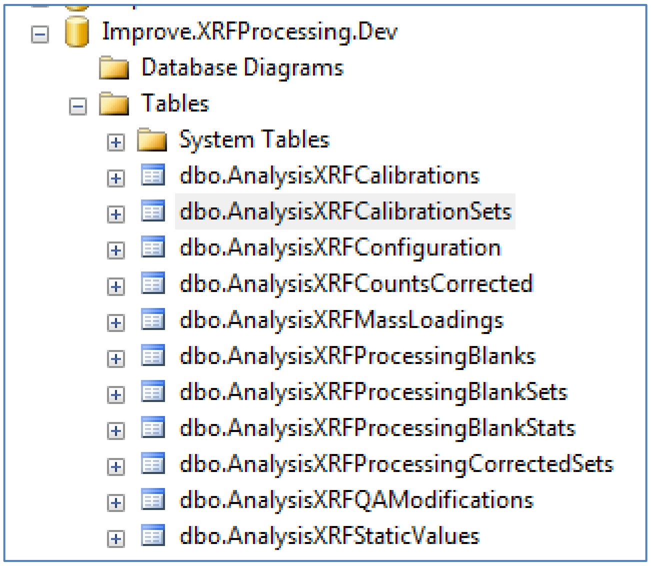
1. PURPOSE AND APPLICABILITY

The purpose of this document is to explain the stored procedures/tables/workflows involved in the blank correction and conversion of PANalytical analysis data into mass loadings, as well as their respective error and detection limits. This document is intended for users of the stored procedures and those involved in the IMPROVE data processing in total. The general audience is required to have a fundamental knowledge of SQL. The audience who wants to use the stored procedures must understand how to query data and execute stored procedures with parameters.

2. DATABASE INFORMATION

The stored procedures will be located in the CL-SQL (169.237.146.117), specifically in the database Improve.XRFProcessing. Users must have Execute/Insert/Select permissions on the Improve.XRFProcessing database, and Select in the Improve database.

3. TABLES

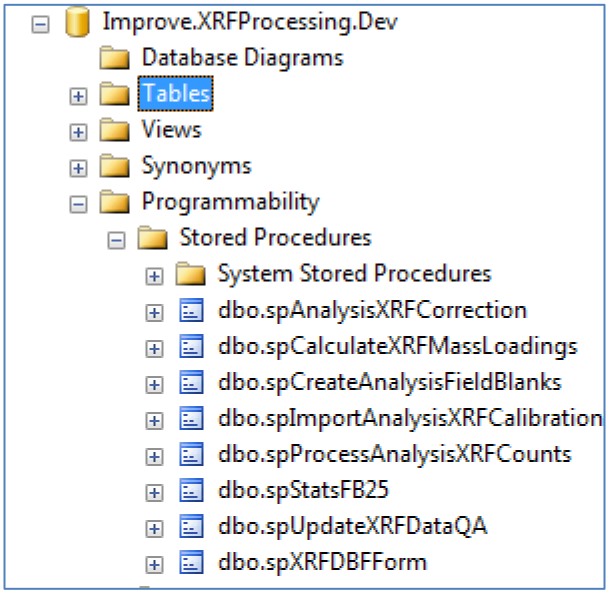


Improve.XRFProcessing.Dev
- Database Diagrams
- Tables
  - System Tables
  - dbo.AnalysisXRFCalibrations
  - dbo.AnalysisXRFCalibrationSets
  - dbo.AnalysisXRFConfiguration
  - dbo.AnalysisXRFCountsCorrected
  - dbo.AnalysisXRFMassLoadings
  - dbo.AnalysisXRFProcessingBlanks
  - dbo.AnalysisXRFProcessingBlankSets
  - dbo.AnalysisXRFProcessingBlankStats
  - dbo.AnalysisXRFProcessingCorrectedSets
  - dbo.AnalysisXRFQAModifications
  - dbo.AnalysisXRFStaticValues

Name	Description
dbo.AnalysisXRFCalibrations	Stores the calibration E values for each PANalytical XRF Machine. Many to one AnalysisXRFCalibrationSets on [CalibrationSetId].
dbo.AnalysisXRFCalibrationSets	Set information for the calibrations. Which machine and dates the calibrations are pertinant. One to many AnalysisXRFCalibrations
dbo.AnalysisXRFConfiguration	Contains the different configurations for the XRF Machines and their effective dates.
dbo.AnalysisXRFCountsCorrected	Stores the analysis counts after blank subtraction using 25 field blanks and their median.
dbo.AnalysisXRFMassLoadings	Stores the mass loadings for each result after processing the corrected counts using the Calibrations.
dbo.AnalysisXRFProcessingBlanks	Stores the actual blanks used for blank correction. Many to one AnalysisXRFProcessingBlankSets on [BlankSetId].

dbo.AnalysisXRFProcessingBlankSets	Stores the set information for field blanks: the configuration, determination date, processed date. One to many AnalysisXRFBlanks and AnalysisXRFBlankStats on [BlankSetId]
dbo.AnalysisXRFProcessingBlankStats	Contains the statistical calculations (median/95th) for each element in each field blank set. Many to one AnalysisXRFProcessingBlankSets on [BlankSetId]
dbo.AnalysisXRFProcessingCorrectedSets	Contains the information about each set of counts processed: the minimum/maximum date and processed date. One to many AnalysisXRFCountsCorrected on [ProcessedSetId].
dbo.AnalysisXRFQAModifications	The QA Modification that have been made to the XRF data on validity, sampleident, xrfdate, and source. Used in tandem with spUpdateXRFDataQA to keep the migrated data up to date if it is ever reimported.
dbo.AnalysisXRFStaticValues	The values chose by Warren to be used in the ERR and MDL calculations for each machine when they are converted into their mass loadings.

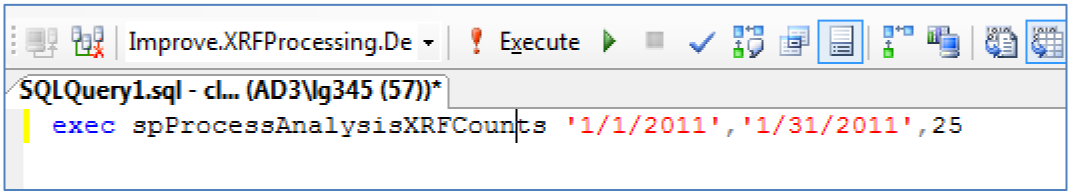
4. THE STORED PROCEDURES



```
⊟ Improve.XRFProcessing.Dev
     Database Diagrams
  ⊞ Tables
  ⊞ Views
  ⊞ Synonyms
  ⊟ Programmability
     ⊟ Stored Procedures
        ⊞ System Stored Procedures
        ⊞ dbo.spAnalysisXRFCorrection
        ⊞ dbo.spCalculateXRFMassLoadings
        ⊞ dbo.spCreateAnalysisFieldBlanks
        ⊞ dbo.spImportAnalysisXRFCalibration
        ⊞ dbo.spProcessAnalysisXRFCounts
        ⊞ dbo.spStatsFB25
        ⊞ dbo.spUpdateXRFDataQA
        ⊞ dbo.spXRFDBFForm
```

1) dbo.spProcessAnalysisXRFCounts:

This stored procedure is the start for the entire XRF Data Process. It accepts three parameters: startdate(datetime), enddate (datetime), and number of field blanks (int). Dates are used so sets can be processed outside of the normal month datespan. The number of field blanks will be 25, but the parameter is there to allow future expansion.

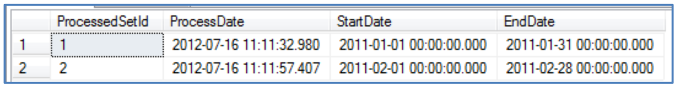


```
exec spProcessAnalysisXRFCounts '1/1/2011','1/31/2011',25
```

Sections of spProcessAnalysisXRfCounts

Generate the Correction Set:

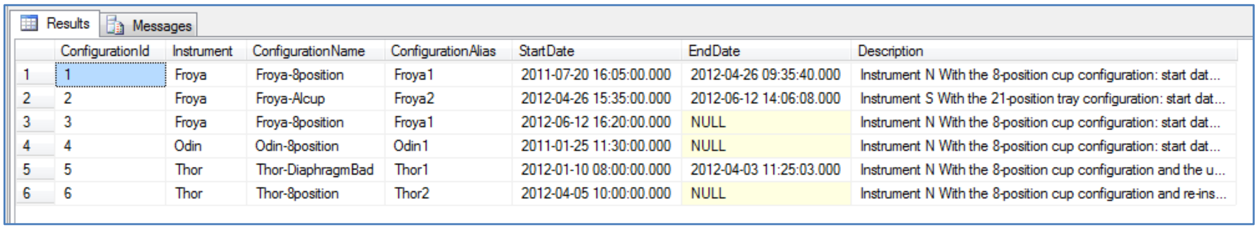
Inserts the stored procedure parameters into AnalysisXRFProcessingCorrectedSets for historical purposes and binding the corrected values for this run of the stored procedures into a single set which will be manipulated later.



	ProcessedSetId	ProcessDate	StartDate	EndDate
1	1	2012-07-16 11:11:32.980	2011-01-01 00:00:00.000	2011-01-31 00:00:00.000
2	2	2012-07-16 11:11:57.407	2011-02-01 00:00:00.000	2011-02-28 00:00:00.000

Configuration Cursor:

The cursor is used to find all of the distinct active configurations that were used during the sampling dates specified through the parameters. It uses each configuration in a loop to execute the rest of the stored procedures.



	ConfigurationId	Instrument	ConfigurationName	ConfigurationAlias	StartDate	EndDate	Description
1	1	Froya	Froya-8position	Froya1	2011-07-20 16:05:00.000	2012-04-26 09:35:40.000	Instrument N With the 8-position cup configuration: start dat...
2	2	Froya	Froya-Alcup	Froya2	2012-04-26 15:35:00.000	2012-06-12 14:06:08.000	Instrument S With the 21-position tray configuration: start dat...
3	3	Froya	Froya-8position	Froya1	2012-06-12 16:20:00.000	NULL	Instrument N With the 8-position cup configuration: start dat...
4	4	Odin	Odin-8position	Odin1	2011-01-25 11:30:00.000	NULL	Instrument N With the 8-position cup configuration: start dat...
5	5	Thor	Thor-DiaphragmBad	Thor1	2012-01-10 08:00:00.000	2012-04-03 11:25:03.000	Instrument N With the 8-position cup configuration and the u...
6	6	Thor	Thor-8position	Thor2	2012-04-05 10:00:00.000	NULL	Instrument N With the 8-position cup configuration and re-ins...

Cursor Loop:

The cursor loop section opens the cursor and each distinct configuration is passed as parameters into other stored procedures.

Loop Member – Choosing Fieldblanks:

Executes the stored procedure spCreateAnalysisFieldBlanks passing in the startdate, enddate, configuration, and number of fieldblanks as parameters. This generates the 25 field blanks per configuration and their statistics

Loop Member – Correction:

Executes the stored procedure spAnalysisXRFCorrection passing in the startdate, enddate, and configuration as parameters. This performs the blank subtraction using the statistics generated in "Loop Member – Choosing Fieldblanks".

Calculate Mass Loadings:

This data is the converted to mass loadings values by applying the PANalytical machine calibration values to the corrected energies calculated in section "Loop Member –Correction".

2) dbo.spCreateAnalysisFieldBlanks

This stored procedure finds the 25 field blanks and their statistics for the configuration specified in the parameters of this stored procedure. This procedure also needs a startdate and enddate as parameters to select the correct field blanks.

Sections of spCreateAnalysisFieldBlanks

Determination Date:

This section finds the date that determines which field blanks to use. The date is determined by the greatest analysis date of the sampling dates specified by the parameters for the configuration.

Create Blank Set:

This section generates the blank set value by inserting into dbo.analysisXRFProcessingBlankSets. It inserts the run date, determination date, and XRF Machine Configuration.

Choose Blanks:

This section selects the 25 field blanks that will be members of the set. It selects 25 field blanks that were analyzed from the determination date backwards. If it cannot fill 25, it will continue filling going forwards from the determination date until it reaches 25.

Generate Field Blank Statistics:

This section generates the median and 95th percentile for each element of the 24 we currently use. It runs the stored procedure 24 times (one for each element). This was not hardcoded into a single stored procedure to allow the expansion later on.

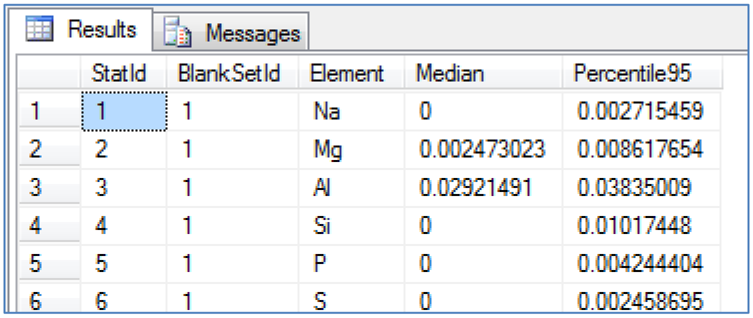
3) dbo.spStatsFB25

This stored procedure generates the statistics (median/95th) for a given set of field blanks. It accepts the parameters element and the blank set ID.

Sections of spStatsFB25

Generate Field Blank Statistics:

This section inserts the statistics for a given field blank set and given element as defined by the parameters. It does this by ordering the values for the given element from least to greatest and selecting the 13th (median) and 24th (95th percentile).



	StatId	BlankSetId	Element	Median	Percentile95
1	1	1	Na	0	0.002715459
2	2	1	Mg	0.002473023	0.008617654
3	3	1	Al	0.02921491	0.03835009
4	4	1	Si	0	0.01017448
5	5	1	P	0	0.004244404
6	6	1	S	0	0.002458695

4) dbo.spAnalysisXRFCorrection

This stored procedure corrects the XRF values for the datespan (it accepts as a parameter) using the field blank statistics generated in the former stored procedure. It also accepts the configuration.

Sections of spAnalysisXRFCorrection

Set Identity Columns:

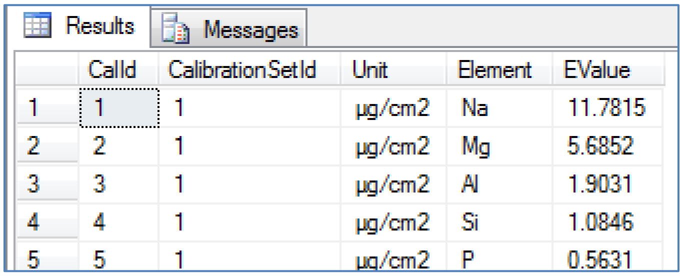
This section selects the currently incremented identity values that were established in the previous stored procedures: Blank set identity and corrected set. These are inserted with the corrected values to establish relationships.

Blank Correction:

This section performs the actual blank correction of the date span; based on validity of the sample, the configuration, and the field blank statistics. It inserts these values into dbo.AnalysisXRFCountsCorrected.

5) dbo.spCalculateXRFMassLoadings

This stored procedure converts the corrected count values into their mass loading values by using the calibrations from each machine. It accepts only the Processed Set Id. This stored procedure is called last and outside of the main loop of spProcessAnalysisXRFCounts.



	CalId	CalibrationSetId	Unit	Element	EValue
1	1	1	µg/cm2	Na	11.7815
2	2	1	µg/cm2	Mg	5.6852
3	3	1	µg/cm2	Al	1.9031
4	4	1	µg/cm2	Si	1.0846
5	5	1	µg/cm2	P	0.5631

Sections of spCalculateXRFMassLoadings

Insert Corrected Mass Loadings:

This section takes the corrected counts, applies the calibration factor, and inserts them into dbo.AnalysisXRFMassLoadings. Likewise, it inserts the error and detection limit values. The detection limits are static and in dbo.AnalysisXRFStaticValues. The errors are calculated using an equation applied to the corrected intensities, proportional uncertainties, and detection limits.

6) dbo.spXRFDBFForm

This stored procedure accepts a processed set Id. The procedure converts the XRF Data into a form readable for the SWAP visual foxpro application. This procedure is called separately from spProcessAnalysisXRFCounts.

Sections of spXRFDBFForm

Column Matching to DBF:

This section selects from the next section subquery. This section selects dummy values into unused columns, and selects the elements, errors, and detection limits with the appropriate names and prefixes.

Max Values for the Pivot Query:

This section holds the elemental select subquery which selects the maximum values for all columns for each split record. Because it selects from a triple pivoted subquery, three records are generated for each one record being pivoted. This query forces them back into one record by selecting the non-null values for each column per triplet record.

The Pivoted Query:

This section contains the initial query from dbo.AnalysisXRFMassLoadings. It casts the sampling dates, and has the three pivots (and titles the pivoted columns). It pivots the data on the concentration, errors and detection limits for each record.

7) dbo.spUpdateXRFDataQA

This stored procedure is separate and is used to update dbo.AnalysisXRFCountsMigrate by joining with AnalysisXRFQAModifications. It also stores comments and the date performed.

Remarks:

Tips:

- **[**Columns**]**, **dbo.**TableName, **vw**ViewName, **sp**StoredProcedureName, **fx**FunctionName are the naming conventions both in this document and in the code.
- cps/mA is referred to as counts or energies. ng/cm^2 is usually referred to as mass loadings or two dimensional density. These units are interchangeable by using the elemental calibrations of each PANalytical XRF machine.
- CamelCasing is keeping multiple words together but making the first letter of each word capital. This is useful for when only one word is allowed or if it makes it easier to deal with (column names).

- SAMDAT is always explicitly called [SAMDAT]. [XRFDate] and [DateTime] correspond to the analysis date/time.
- All properties of a filter (site,samdat,status) are derived from the [SAMPLEIDENT]. This is the only way that properties of filter can be persisted through XRF analysis.

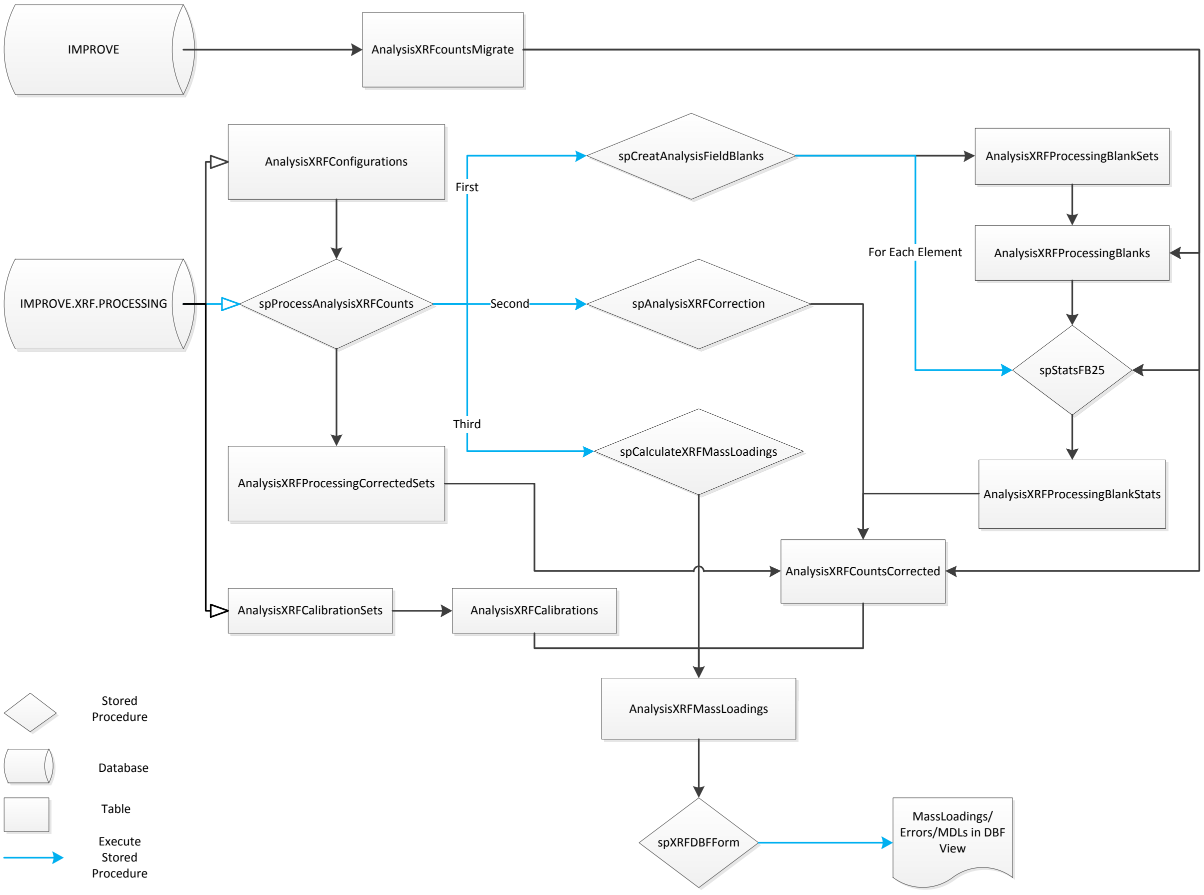
ChangeLog:

Version	Completion Date	Changes
0.900	6/25/2012	Initial Creation
0.905	6/27/2012	Fixed error with Thor checking for Odin data to determine BlankSet existence. Added the Processed set to identify processed data.
0.907	6/28/2012	Converted some Functions into correlated subqueries for performance. This translates to higher performance when there is more data. However, there is less performance when there is no data.
0.909	7/2/2012	Optimized queries (Median/95th, max/min analysis date for configuration determination). Made the calibration import SP import the calibrations correctly assigning them to sets. Even if a calibration import contains data from multiple or one machine, it will assign sets correctly.
0.914	7/3/2012	Queries further optimized using multiple joins for the configuration aliases (incomplete). Created spCalculateXRFMassLoadings as a primitive form of the mass loading conversion.
0.925	7/5/2012	Changed the statistical queries to avoid joining to views. The performance increases are very good. The sp for creating the dbf view will now incorporate the static statistical values warren calcluated. Seeding for the static statistics The dbf view sp is not complete still.
0.965	7/6/2012	Massive overhaul of query structure. Especially for the blank correction. The table join was not effective given the possibility that the same set of data could be processed again. Instead the sproc was changed to accept a configuration paramter and run per configuration as part of the cursor in the calling procedure. Added indexing and some table statistics. The processing takes 2-4 seconds now. The competing factors are table size and SQL server query optimization. DBF view proc is tentatively complete. Columns must be verified. Values must also be verified. Function is complete. At first glance of the march data it seems to be pretty close to the

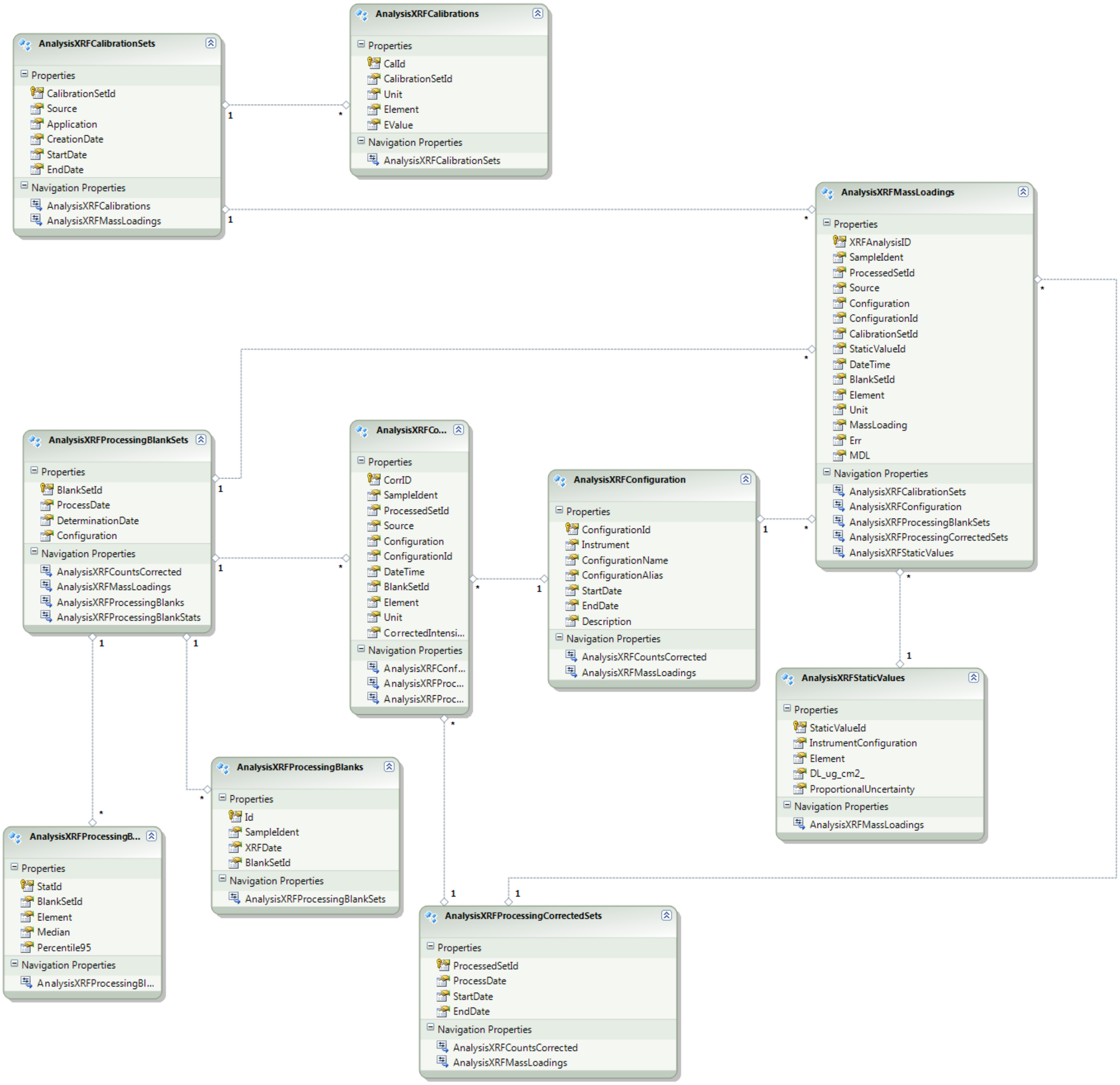
		Panalytical corrected values.
0.980	7/12/2012	Added the mass loading conversion as a hard coded table. This has increased processing time to 10-13 seconds. Testing has thus so far confirmed correct values at all levels of processing.

Appendices

Appendix 1: Stored Procedure Process Flow Diagram.


IMPROVE
AnalysisXRFcountsMigrate
AnalysisXRFConfigurations
First
spCreatAnalysisFieldBlanks
AnalysisXRFProcessingBlankSets
For Each Element
AnalysisXRFProcessingBlanks
IMPROVE.XRF.PROCESSING
spProcessAnalysisXRFCounts
Second
spAnalysisXRFCorrection
spStatsFB25
Third
spCalculateXRFMassLoadings
AnalysisXRFProcessingCorrectedSets
AnalysisXRFProcessingBlankStats
AnalysisXRFCountsCorrected
AnalysisXRFCalibrationSets
AnalysisXRFCalibrations
AnalysisXRFMassLoadings
spXRFDBFForm
MassLoadings/ Errors/MDLs in DBF View
Stored Procedure
Database
Table
Execute Stored Procedure

Appendix 2: Table Design and Relationships.



Appendix 3: Database Technical Description.

Improve.XRFProcessing.Dev Database

CL-DEV-SQL > Improve.XRFProcessing.Dev

Project Information

Author	Leland Gee
Created	25 June 2012 11:16

Tables

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables

Name
dbo.AnalysisXRFCalibrations
dbo.AnalysisXRFCalibrationSets
dbo.AnalysisXRFConfiguration
dbo.AnalysisXRFCountsCorrected
dbo.AnalysisXRFMassLoadings
dbo.AnalysisXRFProcessingBlanks
dbo.AnalysisXRFProcessingBlankSets
dbo.AnalysisXRFProcessingBlankStats
dbo.AnalysisXRFProcessingCorrectedSets
dbo.AnalysisXRFQAModifications
dbo.AnalysisXRFStaticValues

[dbo].[AnalysisXRFCalibrations]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFCalibrations

Columns

	Name	**Data Type**	**Max Length (Bytes)**	**Allow Nulls**	**Identity**
🔑	CalId	bigint	8	✗	1 – 1
🔑	CalibrationSetId	bigint	8	✗	
	Unit	nchar(10)	20	✗	
	Element	nvarchar(4)	8	✗	
	EValue	float	8	✗	

Indexes

	Name	**Columns**	**Unique**
🔑	PK_CalId	CalId	✓

Foreign Keys

Name	**Columns**
FK_AnalysisXRFCalibrations_AnalysisXRFCalibrationSets	CalibrationSetId->[dbo].[AnalysisXRFCalibration-Sets].[CalibrationSetId]

Uses

[dbo].[AnalysisXRFCalibrationSets]

dbo

[dbo].[AnalysisXRFCalibrationSets]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFCalibrationSets

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	CalibrationSetId	bigint	8	✗	1 – 1
	Source	nvarchar(255)	510	✗	
	Application	nvarchar(255)	510	✗	
	CreationDate	datetime	8	✗	
	StartDate	datetime	8	✗	
	EndDate	datetime	8	✓	

Indexes

	Name	Columns	Unique
🔑	PK_CalibrationSetId	CalibrationSetId	✓

Uses

dbo

Used By

[dbo].[AnalysisXRFCalibrations]

[dbo].[AnalysisXRFMassLoadings]

[dbo].[AnalysisXRFConfiguration]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFConfiguration

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	ConfigurationId	int	4	✗	1 – 1
	Instrument	nvarchar(100)	200	✗	
	ConfigurationName	nvarchar(100)	200	✗	
	ConfigurationAlias	nvarchar(100)	200	✗	
	StartDate	datetime	8	✗	
	EndDate	datetime	8	✓	
	Description	nvarchar(max)	max	✓	

Indexes

	Name	Columns	Unique
🔑	PK_ConfigurationId	ConfigurationId	✓

Uses

dbo

Used By

[dbo].[AnalysisXRFCountsCorrected]

[dbo].[AnalysisXRFMassLoadings]

[dbo].[spCalculateXRFMassLoadings]

[dbo].[spImportAnalysisXRFCalibration]

[dbo].[AnalysisXRFCountsCorrected]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFCountsCorrected

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	CorrID	bigint	8	✗	1 – 1
	SampleIdent	nvarchar(128)	256	✗	
🔑	ProcessedSetId	bigint	8	✗	
	Source	nvarchar(100)	200	✗	
	Configuration	nvarchar(100)	200	✗	
🔑	ConfigurationId	int	4	✗	
	DateTime	datetime	8	✗	
🔑	BlankSetId	bigint	8	✗	
	Element	nvarchar(4)	8	✗	
	Unit	nvarchar(10)	20	✗	
	CorrectedIntensity	real	4	✗	

Indexes

	Name	Columns	Unique
🔑	PK_CorrID	CorrID	✓
	_dta_index_AnalysisXRFCountsCorrected_10_85575343__K3_K4_K6_K8_2_110	SampleIdent, CorrectedIntensity, ProcessedSetId, Source, DateTime, Element	

Statistics

Name	Columns
_dta_stat_85575343_3	ProcessedSetId

Foreign Keys

Name	Columns
FK_AnalysisXRFCountsCorrected_AnalysisXRFProcessingBlankSets	BlankSetId->[dbo].[AnalysisXRFProcessingBlankSets].[BlankSetId]
FK_AnalysisXRFCountsCorrected_AnalysisXRFProcessingConfiguration	ConfigurationId->[dbo].[AnalysisXRFConfiguration].[ConfigurationId]
FK_AnalysisXRFCountsCorrected_AnalysisXRFProcessingCorrectedSets	ProcessedSetId->[dbo].[AnalysisXRFProcessingCorrectedSets].[ProcessedSetId]

Uses

[dbo].[AnalysisXRFConfiguration]

[dbo].[AnalysisXRFProcessingBlankSets]

[dbo].[AnalysisXRFProcessingCorrectedSets]

dbo

[dbo].[AnalysisXRFMassLoadings]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFMassLoadings

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	XRFAnalysisID	bigint	8	✗	1 – 1
🖧	SampleIdent	nvarchar(128)	256	✗	
🗝 🖧	ProcessedSetId	bigint	8	✗	
🖧	Source	nvarchar(100)	200	✗	
	Configuration	nvarchar(100)	200	✗	
🗝	ConfigurationId	int	4	✗	
🗝	CalibrationSetId	bigint	8	✗	
🗝	StaticValueId	int	4	✗	
🖧	DateTime	datetime	8	✗	
🗝	BlankSetId	bigint	8	✗	
🖧	Element	nvarchar(4)	8	✗	
	Unit	nvarchar(10)	20	✗	
🖧	MassLoading	real	4	✗	
	Err	real	4	✗	
	MDL	real	4	✗	

Indexes 🖧

	Name	Columns	Unique
🔑	PK_XRFAnalysisId	XRFAnalysisID	✓
	_dta_index_AnalysisXRFMassLoadings_10_85575343__K3_K4_K6_K8_2_101	SampleIdent, MassLoading, ProcessedSetId,	

		Source, DateTime, Element	

Foreign Keys

Name	Columns
FK_AnalysisXRFMassLoadings_AnalysisXRFProcessingBlankSets	BlankSetId->[dbo].[AnalysisXRFProcessingBlankSets].[BlankSetId]
FK_AnalysisXRFMassLoadings_AnalysisXRFCalibrationSets	CalibrationSetId->[dbo].[AnalysisXRFCalibrationSets].[CalibrationSetId]
FK_AnalysisXRFMassLoadings_AnalysisXRFConfiguration	ConfigurationId->[dbo].[AnalysisXRFConfiguration].[ConfigurationId]
FK_AnalysisXRFMassLoadings_AnalysisXRFProcessingCorrectedSets	ProcessedSetId->[dbo].[AnalysisXRFProcessingCorrectedSets].[ProcessedSetId]
FK_AnalysisXRFMassLoadings_AnalysisXRFStaticValues	StaticValueId->[dbo].[AnalysisXRFStaticValues].[StaticValueId]

Uses

[dbo].[AnalysisXRFCalibrationSets]

[dbo].[AnalysisXRFConfiguration]

[dbo].[AnalysisXRFProcessingBlankSets]

[dbo].[AnalysisXRFProcessingCorrectedSets]

[dbo].[AnalysisXRFStaticValues]

dbo

Used By

[dbo].[spAnalysisXRFCorrection]

[dbo].[AnalysisXRFProcessingBlanks]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFProcessingBlanks

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑🖧	Id	bigint	8	✗	1 - 1
🖧🖧	SampleIdent	nvarchar(128)	256	✗	
🖧🖧	XRFDate	datetime	8	✗	
🔑🖧🖧	BlankSetId	bigint	8	✗	

Indexes

	Name	Columns	Unique
🔑	PK_Id	Id	✓
	_dta_index_AnalysisXRFProcessingBlanks_10_117575457__K4_K2_K3	Id, BlankSetId, SampleIdent, XRFDate	
	_dta_index_AnalysisXRFProcessingBlanks_10_117575457__K4_K2_K6	BlankSetId, SampleIdent, XRFDate	

Foreign Keys

Name	Columns
FK_AnalysisXRFProcessingBlanks_AnalysisXRFProcessingBlank-Sets	BlankSetId->[dbo].[AnalysisXRFProcessingBlankSets].[BlankSet-Id]

Uses

[dbo].[AnalysisXRFProcessingBlankSets]

dbo

Used By

[dbo].[spCalculateXRFMassLoadings]

[dbo].[spStatsFB25]

[dbo].[AnalysisXRFProcessingBlankSets]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFProcessingBlankSets

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	BlankSetId	bigint	8	✘	1 - 1
	ProcessDate	datetime	8	✘	
	DeterminationDate	datetime	8	✘	
	Configuration	nvarchar(100)	200	✘	

Indexes

	Name	Columns	Unique
🔑	PK_BlankSetId	BlankSetId	✓

Uses

dbo

Used By

[dbo].[AnalysisXRFCountsCorrected]

[dbo].[AnalysisXRFMassLoadings]

[dbo].[AnalysisXRFProcessingBlanks]

[dbo].[AnalysisXRFProcessingBlankStats]

[dbo].[spCalculateXRFMassLoadings]

[dbo].[AnalysisXRFProcessingBlankStats]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFProcessingBlankStats

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	StatId	bigint	8	✗	1 - 1
🔑	BlankSetId	bigint	8	✗	
	Element	nchar(4)	8	✗	
	Median	real	4	✗	
	Percentile95	real	4	✓	

Indexes

	Name	Columns	Unique
🔑	PK_StatId	StatId	✓

Foreign Keys

Name	Columns
FK_AnalysisXRFBlankStats_AnalysisXRFProcessingBlankSets	BlankSetId->[dbo].[AnalysisXRFProcessingBlankSets].[BlankSet-Id]

Uses

[dbo].[AnalysisXRFProcessingBlankSets]

dbo

Used By

[dbo].[spStatsFB25]

[dbo].[AnalysisXRFProcessingCorrectedSets]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFProcessingCorrectedSets

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	ProcessedSetId	bigint	8	✗	1 - 1
	ProcessDate	datetime	8	✗	
	StartDate	datetime	8	✗	
	EndDate	datetime	8	✗	

Indexes

	Name	Columns	Unique
🔑	PK_ProcessedSetId	ProcessedSetId	✓

Uses

dbo

Used By

[dbo].[AnalysisXRFCountsCorrected]

[dbo].[AnalysisXRFMassLoadings]

[dbo].[spImportAnalysisXRFCalibration]

[dbo].[AnalysisXRFQAModifications]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFQAModifications

Columns

	Name	Data Type	Max Length (Bytes)	Allow Nulls	Identity
🔑	ModificationId	bigint	8	✘	1 - 1
	SampleIdentOld	nvarchar(100)	200	✘	
	XRFDateOld	datetime	8	✘	
	SourceOld	nvarchar(100)	200	✘	
	ValidityOld	int	4	✘	
	SampleIdentNew	nvarchar(100)	200	✘	
	XRFDateNew	datetime	8	✘	
	SourceNew	nvarchar(100)	200	✘	
	ValidityNew	int	4	✘	
	ChangeDate	datetime	8	✘	
	Comment	nvarchar(4000)	8000	✓	

Indexes

	Name	Columns	Unique
🔑	PK_ModificationId	ModificationId	✓

Uses

dbo

Used By

[dbo].[spUpdateXRFDataQA]

[dbo].[AnalysisXRFStaticValues]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Tables > dbo.AnalysisXRFStaticValues

Columns

	Name	**Data Type**	**Max Length (Bytes)**	**Allow Nulls**	**Identity**
🔑	StaticValueId	int	4	✗	1 - 1
	InstrumentConfiguration	nvarchar(255)	510	✗	
	Element	nvarchar(255)	510	✗	
	DL(ug/cm2)	float	8	✗	
	ProportionalUncertainty	float	8	✗	

Indexes

	Name	**Columns**	**Unique**
🔑	PK_ValueId	StaticValueId	✓

Uses

dbo

Used By

[dbo].[AnalysisXRFMassLoadings]

Stored Procedures

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures

Name
dbo.spAnalysisXRFCorrection
dbo.spCalculateXRFMassLoadings
dbo.spCreateAnalysisFieldBlanks
dbo.spImportAnalysisXRFCalibration
dbo.spProcessAnalysisXRFCounts
dbo.spStatsFB25
dbo.spUpdateXRFDataQA
dbo.spXRFDBFForm

[dbo].[spAnalysisXRFCorrection]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spAnalysisXRFCorrection

Parameters

Name	Data Type	Max Length (Bytes)
@startdate	datetime	8
@enddate	datetime	8
@configuration	nvarchar(100)	200

Uses

[dbo].[AnalysisXRFMassLoadings]

dbo

Used By

[dbo].[spImportAnalysisXRFCalibration]

[dbo].[spCalculateXRFMassLoadings]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spCalculateXRFMassLoadings

Properties

Property	Value
ANSI Nulls On	✓
Quoted Identifier On	✓

Parameters

Name	Data Type	Max Length (Bytes)
@ProcessedSetId	bigint	8

Uses

[dbo].[AnalysisXRFConfiguration]

[dbo].[AnalysisXRFProcessingBlanks]

[dbo].[AnalysisXRFProcessingBlankSets]

[dbo].[spStatsFB25]

dbo

[Improve].[dbo].[AnalysisXRFCountsMigrate]

Used By

[dbo].[spImportAnalysisXRFCalibration]

[dbo].[spCreateAnalysisFieldBlanks]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spCreateAnalysisFieldBlanks

Parameters

Name	**Data Type**	**Max Length (Bytes)**
@startdate	datetime	8
@enddate	datetime	8
@Configuration	nvarchar(100)	200
@NumberOfFieldBlanks	int	4

Uses

dbo

Used By

[dbo].[spImportAnalysisXRFCalibration]

[dbo].[spImportAnalysisXRFCalibration]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spImportAnalysisXRFCalibration

Parameters

Name	**Data Type**	**Max Length (Bytes)**
@SourceTable	varchar(100)	100
@EffectiveStart	varchar(10)	10

Uses

[dbo].[AnalysisXRFConfiguration]

[dbo].[AnalysisXRFProcessingCorrectedSets]

[dbo].[spAnalysisXRFCorrection]

[dbo].[spCalculateXRFMassLoadings]

[dbo].[spCreateAnalysisFieldBlanks]

dbo

[Improve].[dbo].[AnalysisXRFCountsMigrate]

[dbo].[spProcessAnalysisXRFCounts]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spProcessAnalysisXRFCounts

Parameters

Name	Data Type	Max Length (Bytes)
@StartDate	datetime	8
@EndDate	datetime	8
@NumberOfFieldBlanks	int	4

Uses

Dbo

[dbo].[spStatsFB25]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spStatsFB25

Parameters

Name	Data Type	Max Length (Bytes)
@Element	nvarchar(4)	8
@BlankSetId	bigint	8

Uses

[dbo].[AnalysisXRFProcessingBlanks]

[dbo].[AnalysisXRFProcessingBlankStats]

dbo

[Improve].[dbo].[AnalysisXRFCountsMigrate]

Used By

[dbo].[spCalculateXRFMassLoadings]

[dbo].[spUpdateXRFDataQA]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spUpdateXRFDataQA

Uses

[dbo].[AnalysisXRFQAModifications]

dbo

[Improve].[dbo].[AnalysisXRFCountsMigrate]

[dbo].[spXRFDBFForm]

CL-DEV-SQL > Improve.XRFProcessing.Dev > Stored Procedures > dbo.spXRFDBFForm

Properties

Property	Value
ANSI Nulls On	✓
Quoted Identifier On	✓

Parameters

Name	Data Type	Max Length (Bytes)
@ProcessedSetId	bigint	8

Uses

Dbo

Appendix 4: Source Code.

Note: Readability is decreased in .doc format. For better readability see the solution code itself in visual studio.

CREATE PROCEDURE [dbo].[spProcessAnalysisXRFCounts]

```
-- =============================================
-- Author: Leland Gee (LBGEE@ucdavis.edu)
-- Create date: 7/12/2012
-- Description: Begins the Processing of XRF Data.
-- =============================================
        @StartDate datetime,
        @EndDate datetime,
        @NumberOfFieldBlanks integer
AS
set nocount on
declare
--
@configuration nvarchar(100),
@processedsetId bigint,
@today datetime = getdate()
begin
BEGIN TRAN T1
-- =============================================
-- Section:     Generate the Correction Set
-- Description: Inserts the SP parameters
--      into AnalysisXRFProcessingCorrectedSets for historical purposes
--      and binding the corrected values for this run of the SP into a single set
-- =============================================

insert into AnalysisXRFProcessingCorrectedSets(ProcessDate,StartDate,EndDate) values
(@today,@startdate,@enddate)

-- =============================================
-- Section:     Configuration Cursor
-- Description: The cursor is used to find all of
--      the distinct active configurations that were occuring
--      during the sampling dates specified in the parameters
--      and is also used to loop through other SPs.
-- =============================================
declare configuration_cursor CURSOR for
select distinct(f.ConfigurationAlias)
from improve.dbo.AnalysisXRFCountsMigrate c join dbo.AnalysisXRFConfiguration f
on c.[Source] = f.[Instrument] and c.[DateTime] between f.StartDate and isnull(f.EndDate,@today)
where c.Validity = 1  and   LEN(c.SampleIdent) = 15
and RIGHT(c.SampleIdent,2) in ('NM','QD','FB')
and Cast(SUBSTRING(c.[SampleIdent],10,2)+ CHAR(47)+
                                 SUBSTRING(c.[SampleIdent],12,2)+ CHAR(47)+
                                 SUBSTRING(c.[SampleIdent],6,4)as DateTime)
 between @startdate and @enddate

open configuration_cursor
-- =============================================
-- Section:     Cursor Loop
-- Description: The Cursor is opened and each distinct
--      configuration is processed through other SPs to generate the fieldblanks/statistics then
correction.
-- =============================================
Fetch Next from configuration_cursor
```

```
into @configuration
while @@FETCH_STATUS = 0
Begin
-- =============================================
-- Section:     Loop Member - Choosing FieldBlanks
-- Description: Create Analysis Field Blank will
--      generate the 25 fieldblank and statistics necessary for Blank Subtraction.
-- =============================================
        exec spCreateAnalysisFieldBlanks @StartDate, @EndDate, @configuration,@NumberOfFieldBlanks
-- =============================================
-- Section:     Loop Member - Correction
-- Description: Performs actual blank correction for the set of data.
--      Using the statistics generated in the previous section.
-- =============================================
        exec spAnalysisXRFCorrection @StartDate, @EndDate, @configuration
        Fetch Next from configuration_cursor into @configuration
End
Close configuration_cursor
DEALLOCATE configuration_cursor
--Duplicity of data requested.
-- =============================================
-- Section:     Calculate Mass Loadings
-- Description: This is an intentional duplicity of the counts data as mass loadings.
--      In the future if everything goes as planned we should not have to store actual counts
--      values that were generated, but merely the parameters/statistics that were part of their
generation.
-- =============================================
select @processedSetId = ident_Current ('AnalysisXRFProcessingCorrectedSets')
exec spCalculateXRFMassLoadings @processedSetId
if @@ERROR <>0
        begin
        Rollback TRAN T1
        End
Else
        begin
        Commit TRAN T1
        end

end
```

CREATE PROCEDURE [dbo].[spCreateAnalysisFieldBlanks]

```
-- =============================================
-- Author: Leland Gee (LBGEE@ucdavis.edu)
-- Create date: 7/12/2012
-- Description: Finds the field blanks to use for processing. Creates Statistics.
-- =============================================
(@startdate datetime,
@enddate datetime,@Configuration NVARCHAR(100), @NumberOfFieldBlanks int)
as
begin
declare
@BlankSetId bigint,
@DetDate datetime,
@today datetime =getdate()

-- =============================================
-- Section: Determination Date
-- Description: Find the date that determines which field blanks to use.
--      In this case that date will be the latest analysis date of
--      the datespan of sampling dates specified in the parameters.
-- =============================================
select  @DetDate = MAX(t.[DateTime])
                                                 FROM improve.dbo.AnalysisXRFCountsMigrate t join
AnalysisXRFConfiguration y on t.[Source] =y.Instrument and t.[DateTime] between y.StartDate and
ISNULL(y.EndDate,@today)
                                                 where Validity=1 and LEN(t.SampleIdent) = 15
                                                         and RIGHT(t.SampleIdent,2)in ('FB','NM','QD')
                                                         and Cast(SUBSTRING(t.[SampleIdent],10,2)+
CHAR(47)+
                                                                 SUBSTRING(t.[SampleIdent],12,2)+
CHAR(47)+
                                                                 SUBSTRING(t.[SampleIdent],6,4)as
DateTime) <= @enddate
                                                                         and y.ConfigurationAlias =
@Configuration
-- =============================================
-- Section: Create Blank Set
-- Description: This section generates the blank set
--      which includes the SP run date, determination date,
--      and XRF Machine Configuration
-- =============================================
Insert into AnalysisXRFProcessingBlankSets (ProcessDate, DeterminationDate, Configuration)
                                                                         values  (getdate(),@DetDate,
@Configuration )

-- =============================================
-- Aside: @BlankSetId is set to whatever the identity column in AnalysisXRFProcessingBlankSets
--      was incremented to on the insert in Section: Create Blank Set. This will allow the next insertion
to associate the blanks correctly with their set.
--      This is a safety to match the fieldblank to their set
--      and circumvent issues involved with MAX(identKey). More reliability and faster.
-- =============================================
SET @BlankSetId = IDENT_CURRENT('AnalysisXRFProcessingBlankSets')
  IF @NumberOfFieldBlanks = 25
  Begin
-- =============================================
-- Section: Choose Blanks
-- Description: This section selects the 25 field blanks
--      that will be the members of the set.  It selects 25 field
--      blanks that were analyzed from the determination date backwards.
--      If it cannot fill 25, it will add going from the determination date
--      forward until 25 field blanks are found.
-- =============================================
        Insert into AnalysisXRFProcessingBlanks (SampleIdent,XRFDate,BlankSetId)

Select  top 25 SampleIdent,[DateTime]as XRFDate,@BlankSetId as BlankSetId
                FROM
                (select * from (
                        select top 25 n.SampleIdent, n.[DateTime]
                                FROM
                                improve.dbo.AnalysisXRFCountsMigrate n
                                        join AnalysisXRFConfiguration z
                                        on n.Source=z.Instrument and n.[DateTime] between z.StartDate and
isnull(z.EndDate,@today)
                                                        where Validity=1 and n.[DateTime] <= @DetDate and
                                                        z.ConfigurationAlias= @Configuration and
RIGHT(n.SampleIdent,2)='FB'and LEN(sampleident) =15 group by n.SampleIdent, n.[DateTime] order by DateTime
desc)a
                        union all
                        select * from (select top 25 n.SampleIdent, n.[DateTime]
                                FROM
                                improve.dbo.AnalysisXRFCountsMigrate n
                                        join AnalysisXRFConfiguration z
                                        on n.Source=z.Instrument and n.[DateTime] between z.StartDate and
isnull(z.EndDate,@today)
                                                where Validity=1 and n.[DateTime] > @DetDate and
                                                z.ConfigurationAlias= @Configuration and
RIGHT(n.SampleIdent,2)='FB' and LEN(sampleident) =15 group by n.SampleIdent, n.[DateTime] order by
DateTime asc)b)c
-- =============================================
-- Section: Generate Field Blank Statistics
-- Description: This section generates the median and
--      95th percentile for each element of the 24 we currently use.
--      This has not been hardcoded into one stored procedure to allow expansion
--      for more elements if necessary. The SPs take the @BlankSetId and generate the Median/95th
--      using the fieldblanks associated with the set.
-- =============================================

  exec spStatsFB25 'Na', @BlankSetId
  exec spStatsFB25 'Mg', @BlankSetId
  exec spStatsFB25 'Al', @BlankSetId
  exec spStatsFB25 'Si', @BlankSetId
  exec spStatsFB25 'P', @BlankSetId
  exec spStatsFB25 'S', @BlankSetId
  exec spStatsFB25 'Cl', @BlankSetId
  exec spStatsFB25 'K', @BlankSetId

  exec spStatsFB25 'Ca', @BlankSetId
  exec spStatsFB25 'Ti', @BlankSetId
  exec spStatsFB25 'V', @BlankSetId
  exec spStatsFB25 'Cr', @BlankSetId
  exec spStatsFB25 'Mn', @BlankSetId
  exec spStatsFB25 'Fe', @BlankSetId
  exec spStatsFB25 'Ni', @BlankSetId
  exec spStatsFB25 'Cu', @BlankSetId

  exec spStatsFB25 'Zn', @BlankSetId
  exec spStatsFB25 N'As', @BlankSetId
  exec spStatsFB25 'Se', @BlankSetId
  exec spStatsFB25 'Br', @BlankSetId
  exec spStatsFB25 'Rb', @BlankSetId
  exec spStatsFB25 'Sr', @BlankSetId
  exec spStatsFB25 'Zr', @BlankSetId
  exec spStatsFB25 'Pb', @BlankSetId
  END

End
go
```

CREATE PROCEDURE [dbo].[spStatsFB25]

```
-- =============================================
-- Author: Leland Gee (LBGEE@ucdavis.edu)
-- Create date: 7/12/2012
-- Description: Generates the Median and 95th Percentile for 25
--      Field blanks that are associated with a blanksetId
-- =============================================
(@Element nvarchar(4),
@BlankSetId bigint)

as
begin
-- =============================================
-- Section: Generate FieldBlank Statistics
-- Description: This section inserts the statistics
--      for a given Field Blank Set and a given Element as
--      defined by the parameters. It does this by ordering
--      the values for the given element from least to greatest
--      and selecting the 13th (median) and 24th(95th)
-- =============================================
Insert into AnalysisXRFProcessingBlankStats (BlankSetId, Element, Median, Percentile95)
Select @BlankSetId as BlankSetId, @Element as Element,   Median, Percentile95
from
        (select Median as Median, blanksetId
        from
                (select ROW_NUMBER() over (order by v.[RawIntensity] asc) R,BlankSetId , v.RawIntensity as
Median
                from
                        improve.dbo.AnalysisXRFCountsMigrate v
                                join dbo.AnalysisXRfProcessingBlanks c
                                on v.sampleident = c.sampleident and v.[DateTime] = c.XRFDate
                where v.Validity=1 and c.BlankSetId = @BlankSetId and v.element= @Element  )t where R =
13)p

 join
        (select Percentile95,blanksetid
        from
                (select ROW_NUMBER() over (order by v.[RawIntensity] asc) R,BlankSetId, v.RawIntensity as
Percentile95
                from
                        improve.dbo.AnalysisXRFCountsMigrate v
                                join dbo.AnalysisXRfProcessingBlanks c
                                on v.sampleident = c.sampleident and v.[DateTime] = c.XRFDate
                where v.Validity=1 and   c.BlankSetId = @BlankSetId  and v.element = @Element)i
where R = 24)y
on p.BlankSetId=y.BlankSetId

End
GO
```

CREATE PROCEDURE [dbo].[spAnalysisXRFCorrection]

```
-- =============================================
-- Author: Leland Gee (LBGEE@ucdavis.edu)
-- Create date: 7/12/2012
-- Description: Corrects the XRF values using the
--      previously generated statistics.
-- =============================================
(@startdate datetime,
@enddate datetime,
@configuration nvarchar(100))
as
begin
declare @AnalysisXRFCorrectionSet as bigint,
@today as datetime = getdate(),
@AnalysisBlankSet as bigint
-- =============================================
-- Section: Set Identity Columns
-- Description: The Currently incremented value of the blankset
--      and corrected set is assigned to their respective
--      variables so they can be associated with the corrected values.
-- =============================================
select @AnalysisBlankSet  = IDENT_CURRENT('AnalysisXRFProcessingBlankSets')
select @AnalysisXRFCorrectionSet = IDENT_Current('AnalysisXRFProcessingCorrectedSets')


-- =============================================
-- Section: Blank Correction
-- Description: Performs the configuration specific blank
--      correction by applying the statistic previously
--      generated.
-- =============================================
Insert into AnalysisXRFCountsCorrected
(SampleIdent,ProcessedSetId,[Source],[Configuration],[ConfigurationId],[DateTime],BlankSetId,Element,Unit,
CorrectedIntensity)
select  c.SampleIdent, @AnalysisXRFCorrectionSet as ProcessedSetId, c.[Source],
                        x.ConfigurationAlias as [Configuration], x.ConfigurationId,
                        c.[DateTime],
                        @AnalysisBlankSet as BlankSetId,
                        c.Element, c.Unit,
                        (c.RawIntensity - u.Median) as CorrIntensity
        from
                improve.dbo.AnalysisXRFCountsMigrate c
                        join AnalysisXRFConfiguration x
                        on c.[DateTime] between x.StartDate and isnull(x.EndDate,@today) and c.[Source] =
x.Instrument
                        join AnalysisXRFProcessingBlankStats u
                        on u.BlankSetId = @analysisblankset and c.Element = u.Element
        where c.Validity=1 and x.ConfigurationAlias =@configuration and c.Element
        in ('Na','Mg','Al','Si','P','S','Cl','K','Ca','Ti','V','Cr','Mn','Fe','Ni','Cu','Zn'
      ,'As','Se','Br','Rb','Sr','Zr','Pb')
                and   LEN(c.SampleIdent) = 15
                and RIGHT(c.SampleIdent,2) in ('NM','QD')
                and Cast(SUBSTRING(c.[SampleIdent],10,2)+ CHAR(47)+
                                SUBSTRING(c.[SampleIdent],12,2)+ CHAR(47)+
                                SUBSTRING(c.[SampleIdent],6,4)as DateTime)
                        between @startdate and @enddate
END

GO
```

CREATE PROCEDURE **[dbo].[spCalculateXRFMassLoadings]**

```
-- =============================================
-- Author: Leland Gee (LBGEE@ucdavis.edu)
-- Create date: 7/12/2012
-- Description: Converts the corrected count values into
--      their mass loading values by using the calibrations
--      from each machine.
-- =============================================
        @ProcessedSetId bigint
AS
declare @today as datetime
set @today = GETDATE()

-- =============================================
-- Section: Insert Corrected Mass Loadings
-- Description: This section takes the corrected counts.
--      applies the calibration factor, and inserts them into
--      AnalysisXRFMassLoadings. It also calculates the uncertainties and inserts the MDLs
-- =============================================

insert into AnalysisXRFMassLoadings (Sampleident, ProcessedSetId, Source, Configuration, CalibrationSetId,
[DateTime],
BlanksetId, Element, Unit, MassLoading, Err, MDL)

        SELECT c.SampleIdent, @ProcessedSetId as ProcessedSetId,
c.Source,c.Configuration,q.CalibrationSetId,c.[DateTime],
                c.BlankSetId,c.Element ,'ng/cm2' as Unit,
                (c.CorrectedIntensity *p.EValue*1000) as MassLoading,
                        SQRT(square(0.608*y.[DL(ug/cm2)]*1000)+square(y.ProportionalUncertainty*(case
when c.CorrectedIntensity< 0 then 0 else c.CorrectedIntensity end)*1000*p.EValue)) as 'err',
                                y.[DL(ug/cm2)]*1000 as 'mdl'
        from AnalysisXRFCountsCorrected c
                join AnalysisXRFCalibrationSets q
                        on c.Source = q.Source and c.[DateTime] between q.StartDate and
isnull(q.EndDate,@today)
                join AnalysisXRFCalibrations p
                        on p.CalibrationSetId = q.CalibrationSetId and c.Element = p.Element
                join AnalysisXRFConfiguration t
                        on c.Source = t.Instrument and c.[DateTime] between t.StartDate and
ISNULL(t.EndDate,@today)
                join AnalysisXRFStaticValues y
                        on t.ConfigurationAlias = y.InstrumentConfiguration and c.Element = y.Element
        where c.ProcessedSetId = @ProcessedSetId
```

```
CREATE PROCEDURE [dbo].[spXRFDBFForm]
        @ProcessedSetId bigint
AS
declare @today as datetime
set @today = GETDATE()
select [Site], Samdat,
 0000 Strtim,
 'NM' as Pesstat,
 CHAR(32)as Pixstat,
 [StatusCode] as Xrmstat,
 [StatusCode] as Xrcstat,
 0.00 H,
 0.00 H_err,
 0.00 H_mdl,
 S,
 S_err,
 S_mdl,
 Cl,
 Cl_err,
 Cl_mdl,
 K,
 K_err,
 K_mdl,
 Ca,
 Ca_err,
 Ca_mdl,
 Char(32) Ba,
 Char(32) Ba_err,
 Char(32) Ba_mdl,
 Ti,
 Ti_err,
 Ti_mdl,
 V,
 V_err,
 V_mdl,
 Cr,
 Cr_err,
 Cr_mdl,
 Mn,
 Mn_err,
 Mn_mdl,
 Fe,
 Fe_err,
 Fe_mdl,
 Char(32) Co,
 Char(32) Co_err,
 Char(32) Co_mdl,
 Ni,
 Ni_err,
 Ni_mdl,
 Cu,
 Cu_err,
 Cu_mdl,
 Zn,
 Zn_err,
 Zn_mdl,
 Char(32) Ga,
 Char(32) Ga_err,
 Char(32) Ga_mdl,
 Char(32) Au,
 Char(32) Au_err,
 Char(32) Au_mdl,
 Char(32) Hg,
 Char(32) Hg_err,
 Char(32) Hg_mdl,
 [As],
 As_err,
 As_mdl,
 Pb,
 Pb_err,
 Pb_mdl,
 Se,
 Se_err,
 Se_mdl,
 Br,
 Br_err,
 Br_mdl,
 Rb,
 Rb_err,
 Rb_mdl,
 Sr,
 Sr_err,
 Sr_mdl,
 Char(32) Y,
 Char(32) Y_err,
 Char(32) Y_mdl,
 Zr,
 Zr_err,
 Zr_mdl,
 Char(32) Cd,
 Char(32) Cd_err,
 Char(32) Cd_mdl,
 Char(32) Ag,
 Char(32) Ag_err,
 Char(32) Ag_mdl,
 na Yna,
 na_err Yna_err,
 na_mdl Yna_mdl,
 mg Ymg,
 mg_err Ymg_err,
 mg_mdl Ymg_mdl,
 al Yal,
 al_err Yal_err,
 al_mdl Yal_mdl,
 si Ysi,
 si_err Ysi_err,
 si_mdl Ysi_mdl,
 p Yp,
 p_err Yp_err,
 p_mdl Yp_mdl,
 s Ys,
 s_err Ys_err,
 s_mdl Ys_mdl,
 cl Ycl,
 cl_err Ycl_err,
 cl_mdl Ycl_mdl,
 k Yk,
 k_err Yk_err,
 k_mdl Yk_mdl,
 ca Yca,
 ca_err Yca_err,
 ca_mdl Yca_mdl,
 Char(32) Yba,
 Char(32) Yba_err,
 Char(32) Yba_mdl,
 ti Yti,
 ti_err Yti_err,
 ti_mdl Yti_mdl,
v Yv,
v_err Yv_err,
v_mdl Yv_mdl,
cr Ycr,
cr_err Ycr_error,
cr_mdl Ycr_mdl,
mn Ymn,
mn_err Ymn_err,
mn_mdl Ymn_mdl,
fe Yfe,
fe_err Yfe_err,
fe_mdl Yfe_mdl,
Char(32) Yco,
Char(32) Yco_err,
Char(32) Yco_mdl,
ni Yni,
ni_err Yni_err,
ni_mdl Yni_mdl,
0.00 Pesalt,
Char(32) Pixelt,
0.99 Xrmlt,
0.99 Xrclt,
configuration Cuxrfid



from(
select Site,StatusCode,Samdat,Unit,Configuration,max(Na) as Na,max(Na_err) as Na_err,     max(Na_mdl) as 
Na_mdl,
max(Mg) as Mg,  max(Mg_err) as Mg_err,   max(Mg_mdl) as Mg_mdl,
max(Al) as Al,  max(Al_err) as Al_err,   max(Al_mdl) as Al_mdl,
max(Si) as Si,  max(Si_err) as Si_err,   max(Si_mdl) as Si_mdl,
max(P) as P,    max(P_err) as P_err,     max(P_mdl) as P_mdl,
max(S) as S,    max(S_err) as S_err,     max(S_mdl) as S_mdl,
max(Cl) as Cl,  max(Cl_err) as Cl_err,   max(Cl_mdl) as Cl_mdl,
max(K) as K,    max(K_err) as K_err,     max(K_mdl) as K_mdl,
max(Ca) as Ca,  max(Ca_err) as Ca_err,   max(Ca_mdl) as Ca_mdl,
max(Ti) as Ti,  max(Ti_err) as Ti_err,   max(Ti_mdl) as Ti_mdl,
max(V) as V,    max(V_err) as V_err,     max(V_mdl) as V_mdl,
max(Cr) as Cr,  max(Cr_err) as Cr_err,   max(Cr_mdl) as Cr_mdl,
max(Mn) as Mn,  max(Mn_err) as Mn_err,   max(Mn_mdl) as Mn_mdl,
max(Fe) as Fe,  max(Fe_err) as Fe_err,   max(Fe_mdl) as Fe_mdl,
max(Ni) as Ni,  max(Ni_err) as Ni_err,   max(Ni_mdl) as Ni_mdl,
max(Cu) as Cu,  max(Cu_err) as Cu_err,   max(Cu_mdl) as Cu_mdl,
max(Zn) as Zn,  max(Zn_err) as Zn_err,   max(Zn_mdl) as Zn_mdl,
max([As]) as [As],      max([As_err]) as [As_err],      max([As_mdl]) as [As_mdl],
max(Se) as Se,  max(Se_err) as Se_err,   max(Se_mdl) as Se_mdl,
max(Br) as Br,  max(Br_err) as Br_err,   max(Br_mdl) as Br_mdl,
max(Rb) as Rb,  max(Rb_err) as Rb_err,   max(Rb_mdl) as Rb_mdl,
max(Sr) as Sr,  max(Sr_err) as Sr_err,   max(Sr_mdl) as Sr_mdl,
max(Zr) as Zr,  max(Zr_err) as Zr_err,   max(Zr_mdl) as Zr_mdl,
max(Pb) as Pb,  max(Pb_err) as Pb_err,   max(Pb_mdl) as Pb_mdl
from

        (SELECT LEFT(c.[SampleIdent],5) AS 'Site',
                         RIGHT(c.[SampleIdent],2) AS 'StatusCode',
                                 Cast(SUBSTRING(c.[SampleIdent],10,2)+ CHAR(47)+
                                 SUBSTRING(c.[SampleIdent],12,2)+ CHAR(47)+
                                 SUBSTRING(c.[SampleIdent],6,4)as DateTime) AS 'Samdat',  c.Unit,
c.element,
         c.MassLoading,
 c.Err,
        c.Element+'_err' as Element_err,
        c.Element+'_mdl' as Element_mdl,
         mdl, c.Configuration from AnalysisXRFMassLoadings c

        where c.ProcessedSetId = @ProcessedSetId) as query
        pivot (max(MassLoading)
        for Element in (Na,Mg,Al,Si,P,
        S,Cl,K,Ca,Ti,V,Cr,Mn,Fe,Ni,Cu,Zn,
        [As],Se,Br,Rb,Sr,Zr,Pb)) as pivot1

        pivot (max(err)
        for Element_err in (Na_err,      Mg_err, Al_err, Si_err, P_err,
        S_err,  Cl_err, K_err,  Ca_err, Ti_err, V_err,  Cr_err, Mn_err,
        Fe_err, Ni_err, Cu_err, Zn_err,
        [As_err],Se_err,Br_err,  Rb_err, Sr_err, Zr_err, Pb_err
)) as pivot2

        pivot (max(mdl)
        for Element_mdl in (Na_mdl,      Mg_mdl, Al_mdl, Si_mdl, P_mdl,
        S_mdl,  Cl_mdl, K_mdl,  Ca_mdl, Ti_mdl, V_mdl,  Cr_mdl,
        Mn_mdl, Fe_mdl, Ni_mdl, Cu_mdl, Zn_mdl,
        [As_mdl],Se_mdl,Br_mdl,  Rb_mdl, Sr_mdl, Zr_mdl, Pb_mdl)) pivot3 group by
Site,StatusCode,Samdat,Unit,Configuration)outside
```