the Navy SEALs of
Siebel Loyalty and Marketing

FeedXChange

User Guide

(pre-production version – please do not distribute)

©2006-2012 by Loyalty Methods Inc. All rights reserved.

This documentation is the confidential and proprietary intellectual property of Loyalty Methods, Inc.
Any unauthorized use, reproduction, preparation of derivative works, performance, or display of this document, or software represented by this document is strictly prohibited. Trademarks other than Loyalty Methods and ThreadSync are the property of their respective trademark holders.

LOYALTYmethods

Contents

Introduction 4

Who should Read this Manual 4

Product Overview 4

- Types of Processing 4
- Loyalty Specific Actions 5

Configuring Templates 5

- About Templates 5
  - Creating Input Templates 6
  - Creating Input Template Mappings 7
  - Creating Response Templates 10
  - Supported Field Formats 11
  - Activating and Revising Templates 13

Configuring Integrations 13

- About Integrations 13
- Creating Integrations 14
  - Configuring Integration Transport 16
  - Configuring User Key De-duplication 17
  - Configuring Error Translation 17
  - Configuring Alerts 17
  - Configuring Integration Parameters 18
  - Configuring Integration Events 18
  - Configuring Integration Schedules 19

Running Integrations 20

- Triggering Scheduled Integrations 20
- Triggering Integrations Manually 20
- Triggering Using the Command Line Interface 21
- Monitoring Integrations and Files 21
  - Viewing Integration Run History 21

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

Viewing File History ... 22

Applying Manual Corrections and Rerunning Files ... 23

Correcting Individual Records ... 25

Auto Fixing Multiple Records ... 25

Re-queuing Records for Reload ... 25

Marking Records Unfixable ... 25

Marking Records Resolved ... 25

Reloading a Corrected File ... 26

Migrating Templates and Integrations ... 26

Template Export/Import ... 26

Integration Export/Import ... 26

Configuring Global Parameters ... 27

Configuring Global Error Messages ... 27

Extending FeedXChange with Events ... 27

About Events ... 27

Event Types ... 28

Event Protocol ... 28

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

Introduction

FeedXChange is a product specifically designed to handle Siebel Loyalty partner operations such as transaction processing and member updates.

FeedXChange is a specialized product, and NOT a generic ETL tool. It is designed to handle Loyalty-specific tasks, and can be used for some generic purposes, but it's primary motivation is to make the life of Loyalty administrators easier first and foremost.

At the same time FeedXChange allows plenty of opportunities for pre- and pos-processing at different stages in the processing of files.

Who should Read this Manual

This manual is designed for both the developers of new partner feeds, as well as for the operations staff that monitors the feeds and resolves any issues.

Product Overview

Types of Processing

FeedXChange currently supports two types of processing which are typical in Loyalty systems:

- Transaction Processing
- Member Update

The flow for is illustrated in Figure 1 below:

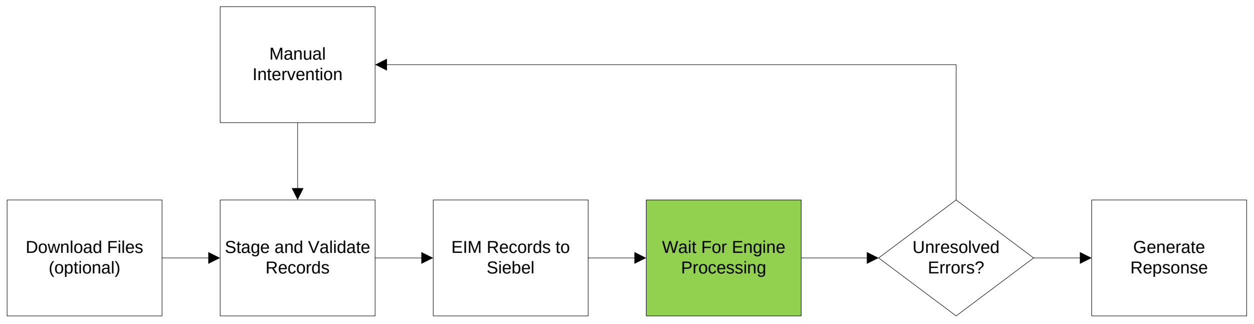

Manual Intervention
Download Files (optional)
Stage and Validate Records
EIM Records to Siebel
Wait For Engine Processing
Unresolved Errors?
Generate Repsonse

Figure 1 FeedXChange High-Level Process

FeedXChange technically recognizes 3 types of integrations, each doing slightly different things:

1. Transaction Import – this is the process illustrated in Figure 1. This is specialized and only valid for Transactions.

2. EIM Import – this is the same process as the one Figure 1 except there is no waiting for the Loyalty Engine to process transactions. It can be used for updating Member or Transaction records.
3. File Import – this is a process similar to EIM Import, but without the EIM processing. It just pre-stages and validates all fields from a source file into a pre-staging table.

Most of the FeedXChange UI is dedicated to configuring these different types of processes and monitoring their execution.

FeedXChange currently uses two staging tables for the two supported entities where text files are imported and fields validated:

- CX_FINT_MEMBER
- CX_FINT_TXN

These tables are extended in Tools, and are fully accessible to Siebel configuration as needed.

Loyalty Specific Actions

FeedXChange supports several types of actions that are specific to Loyalty:

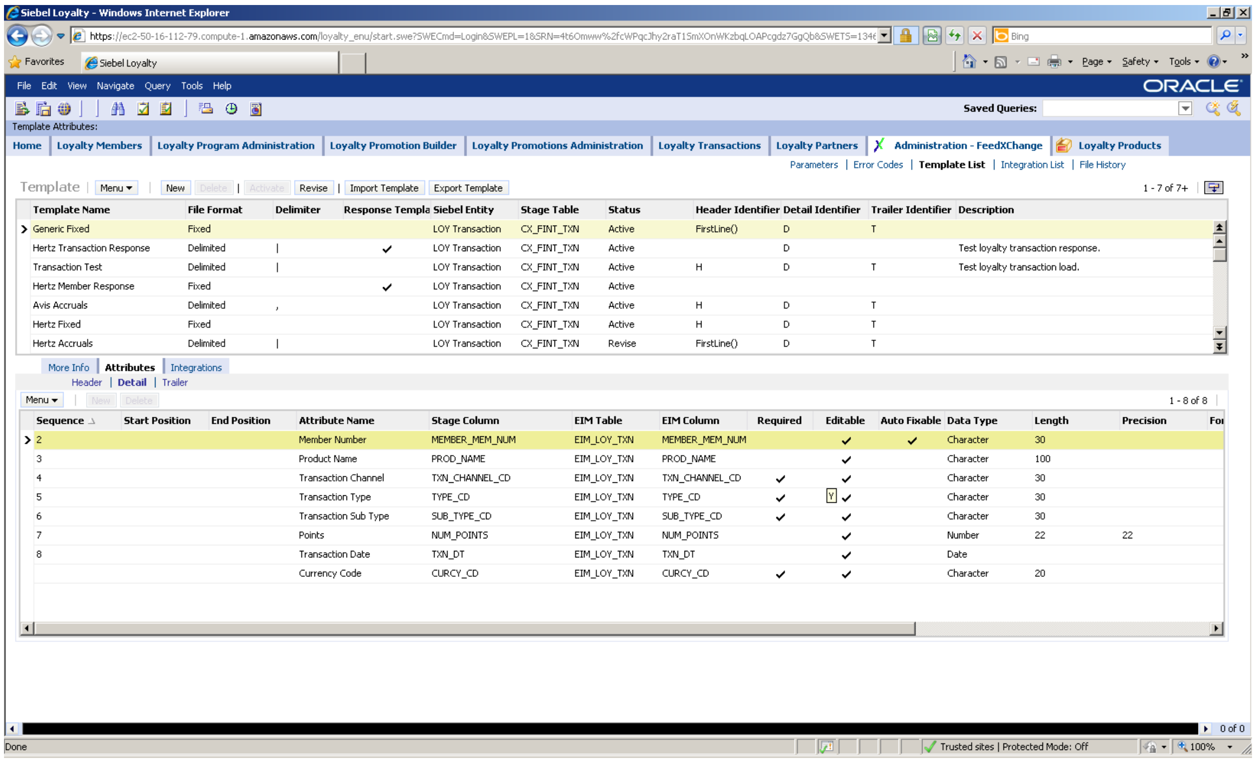
1. Member Number Validation – this ensures that member numbers exist in the program. For transaction processing FeedXChange also validates that the member is in Active status.
2. Product Validation – this ensures that the products mentioned in a transaction file are actually offered by the partner from which the file is coming, within the program for which the file is being processed.
3. Accruals and Redemption Ordering – FeedXChange ensures that if a file contains both Accruals and Redemptions, the Accruals are processed first, and then the Redemptions in order to avoid negative balances as much as possible.

Configuring Templates

About Templates

Templates are the basic building block for any integration. They are used to store information about the structure of incoming or outgoing files.

To configure templates navigate to Site Map → Administration – FeedXChange → Template List.


Template Attributes:

Home | Loyalty Members | Loyalty Program Administration | Loyalty Promotion Builder | Loyalty Promotions Administration | Loyalty Transactions | Loyalty Partners | Administration - FeedXChange | Loyalty Products

Parameters | Error Codes | Template List | Integration List | File History

Template | Menu | New | Delete | Activate | Revise | Import Template | Export Template — 1 - 7 of 7+

Template Name	File Format	Delimiter	Response Templa	Siebel Entity	Stage Table	Status	Header Identifier	Detail Identifier	Trailer Identifier	Description
Generic Fixed	Fixed			LOY Transaction	CX_FINT_TXN	Active	FirstLine()	D	T	
Hertz Transaction Response	Delimited	|	✓	LOY Transaction	CX_FINT_TXN	Active		D		Test loyalty transaction response.
Transaction Test	Delimited	|		LOY Transaction	CX_FINT_TXN	Active	H	D	T	Test loyalty transaction load.
Hertz Member Response	Fixed		✓	LOY Transaction	CX_FINT_TXN	Active				
Avis Accruals	Delimited	,		LOY Transaction	CX_FINT_TXN	Active	H	D	T	
Hertz Fixed	Fixed			LOY Transaction	CX_FINT_TXN	Active	H	D	T	
Hertz Accruals	Delimited	|		LOY Transaction	CX_FINT_TXN	Revise	FirstLine()	D	T	

More Info | Attributes | Integrations

Header | Detail | Trailer

Menu | New | Delete — 1 - 8 of 8

Sequence	Start Position	End Position	Attribute Name	Stage Column	EIM Table	EIM Column	Required	Editable	Auto Fixable	Data Type	Length	Precision	Fo
2			Member Number	MEMBER_MEM_NUM	EIM_LOY_TXN	MEMBER_MEM_NUM		✓	✓	Character	30		
3			Product Name	PROD_NAME	EIM_LOY_TXN	PROD_NAME		✓		Character	100		
4			Transaction Channel	TXN_CHANNEL_CD	EIM_LOY_TXN	TXN_CHANNEL_CD	✓	✓		Character	30		
5			Transaction Type	TYPE_CD	EIM_LOY_TXN	TYPE_CD	✓	Y ✓		Character	30		
6			Transaction Sub Type	SUB_TYPE_CD	EIM_LOY_TXN	SUB_TYPE_CD	✓	✓		Character	30		
7			Points	NUM_POINTS	EIM_LOY_TXN	NUM_POINTS		✓		Number	22	22	
8			Transaction Date	TXN_DT	EIM_LOY_TXN	TXN_DT		✓		Date			
			Currency Code	CURCY_CD	EIM_LOY_TXN	CURCY_CD	✓	✓		Character	20		

Figure 2 Template List View

Creating Input Templates

To create a new template:

1. Go to Site Map → Administration – FeedXChange → Template List.
2. In the Template List Applet, use the New button to create a new record.
3. Enter the following configuration details for the template:

Configuration Detail	**Description**
Template Name	A name for the template you are creating. For example "Generic Transaction".
File Format	Enter the input file format for the template which can be one of: - Fixed - Delimited
Delimiter	If File Format is 'Delimited', then enter the delimiter here. **Note:** For Tab delimited files, please use \t in the delimiter.
Response Template	Leave this field unchecked since this is an input template.
Siebel Entity	Which Siebel entity this template is for: - LOY Transaction - LOY Member **Note:** This will auto-populate the value of the staging table.

Header Identifier	An optional character sequence which will be used to recognize the header record. For example, "HDR" or "H". If the first record is always the header, enter FirstRecord() in this field. If there is no header, leave the field blank.
Detail Identifier	An optional character sequence that will be used to recognize detail records. For example "DTL" or "D". If the detail records do not start with a pre-determined sequence, leave this field blank.
Trailer Identifier	If a trailer needs to be recognized in the input file, this has to be set to a character sequence which will be used to recognize the trailer record. If there is no trailer record, please leave this value blank. **Note:** For efficiency reasons, there is no way to specify in FeedXChange that the last record is the trailer.
Description	A description for the template.

Creating Input Template Mappings

Each template specifies mappings between a file and its corresponding staging table for three different parts of the file, as needed – the header, detail and trailer mapping.

Creating Detail Mappings

To map fields from input file:

1. Go to Site Map → Administration – FeedXChange → Template List
2. Select the template for which you want to add detail mappings.
3. Go to the Detail view in the bottom applet.
4. Create a new record in that applet for each field from the input file you'd like to map, and enter the following details:

Configuration Detail	**Description**
Sequence	If this is a "Delimited" file format template, then this is the sequence of the field in the file. For example, the first field has a sequence of 1 and so on. **Note:** To skip fields in the file, simply omit that sequence number – for example to skip the second field in a 5 field

	record, your detail mapping records will be numbered 1, 3,4,5.
Start Position	If this is a "Fixed" file format template, this is the starting character position of the field. (1-based, so the first character in the file is at position 1).
End Position	If this is a "Fixed" file format template, this is the ending character position of the field.
Attribute Name	Select an attribute in the staging table where you want to map the field coming from your input file. **Note:** This will automatically populate the Stage Column, EIM Table and EIM Column name for this field.
Required	If this is checked, then FeedXChange will validate that there is always some value in this field.
Editable	If this is checked, FeedXChange will allow an operator to correct the value of this field if it is deemed in error. This is done through the manual intervention views described later.
Auto Fixable	If this is checked, it will be possible to use the Auto Fix feature in the error correction views to apply the same fix to multiple records that are deemed in error.
Data Type	This is the data type of the field that is coming in from the file. The field will be validated based on what it's type is. The supported data types are: - Character - Date - Number **Note:** In the CX_FINT_* staging table, all columns are character columns, so that the text files can b imported. Validation takes place on these character values so that they can be eventually pushed into the EIM staging columns which are typed.
Length	This applies to Characters and Numbers only.
Precision	This applies to Numbers only.
Format	This applies to all types and will be used to validate any specific character, numeric or date formats. Please see the section on format specifications for details on what the different types of formats are.
Lookup Type	LOV type which this value corresponds to. There are two forms of this field: - LOV_TYPE - LOV_TYPE, LOV_FIELD By default the lookup will check if the field value corresponds to the VAL column in the S_LST_OF_VAL table. However, if

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

	the LOV_FIELD is specified, it is possible to lookup any column in the LOV table. For example the following lookup type: - LOY_ACCRUAL_TYPE_CD, CODE will ensure that the field form the file corresponds to the CODE column of the S_LST_OF_VAL table where LOV_TYPE= LOY_ACCRUAL_TYPE_CD.
Default Value	This will be the default value in case the field in the file is empty. It is possible to use the "Expr:" prefix to allow defaults to be issued by a database function. For example, if you want to assign a TXN_NUM attribute with a default unique value, you can use "Expr: LM_GEN_UID()" which will effectively call the FeedXChange unique ID generator and assign the result as the default value. You can write your own database functions and call them in this way if you need to.
Comments	Any additional notes about the field.

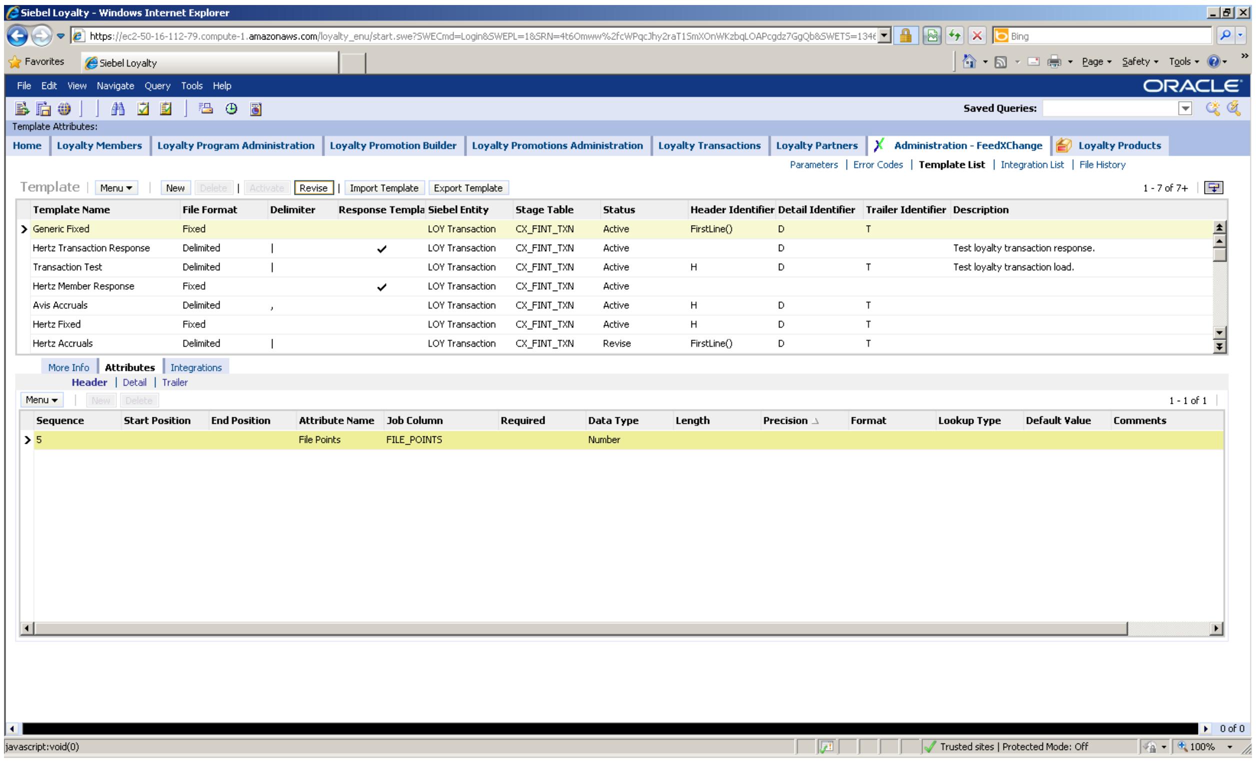
Creating Header and Trailer Mappings

When FeedXChange processes header and trailer records, the mapping information in the template tells it how to map the incoming header/trailer fields onto the FeedXChange File History record for the file which is being processed. This file history record resides in the CX_FINT_FILE table.

The idea is that when FeedXChange processes a file, it loads information from the header/trailer and store it on the file history record for the file being processed. No further processing is done for the trailer/header information, however, it is possible (as we will see later) to write event handlers that will validate this information once it is loaded onto the File History record.

To create header/trailer mappings, navigate to the Header/Trailer view under any template as shown in Figure 3 below:

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.



Template Name	File Format	Delimiter	Response Templa	Siebel Entity	Stage Table	Status	Header Identifier	Detail Identifier	Trailer Identifier	Description
Generic Fixed	Fixed			LOY Transaction	CX_FINT_TXN	Active	FirstLine()	D	T	
Hertz Transaction Response	Delimited	|	✓	LOY Transaction	CX_FINT_TXN	Active		D		Test loyalty transaction response.
Transaction Test	Delimited	|		LOY Transaction	CX_FINT_TXN	Active	H	D	T	Test loyalty transaction load.
Hertz Member Response	Fixed		✓	LOY Transaction	CX_FINT_TXN	Active				
Avis Accruals	Delimited	,		LOY Transaction	CX_FINT_TXN	Active	H	D	T	
Hertz Fixed	Fixed			LOY Transaction	CX_FINT_TXN	Active	H	D	T	
Hertz Accruals	Delimited	|		LOY Transaction	CX_FINT_TXN	Revise	FirstLine()	D	T	

Sequence	Start Position	End Position	Attribute Name	Job Column	Required	Data Type	Length	Precision	Format	Lookup Type	Default Value	Comments
5			File Points	FILE_POINTS		Number						

Figure 3 Template Header View

The validations are specified and performed similar to the detail section. Obviously there is no mapping to the pre-staging tables, as these fields are mapped to the CX_FINT_FILE table which carries each file's history.

Creating Response Templates

The response templates are used to select data from the pre-staging table and from the file history and send it back to the partner as a confirmation, as well as indication of any records that were in error and could not be fixed.

The way to specify these templates is the same as for input templates with the following differences:

1. The Response Template flag is checked to indicate that this is a response template.
2. The Header, Detail and Trailer Identifiers are placed in front of header, detail and trailer records respectively.
3. Any fields you select in the Header, Detail or Trailer mappings will be copied to the output file, at the specified sequence or start/end positions in case of fixed format.
4. If you include Record Status in the response, any records marked as Unfixable will be marked as Error in the response file.
5. If you include Record Status in the response, any records marked as Resolved will be marked as Processed in the response file.

6. The header and footer mappings can be used to copy information from the File History record for a file and send it back to the partner.

Supported Field Formats

When you specify mappings, there are special validations rule you can apply to each field, which are expressed in the Format field. These validations are different for Dates, Numbers and Characters.

Date Formats

FeedXChange recognizes the following format characters for Dates:

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	`AD`
Y	Year	Year	`1996`; `96`
M	Month in year	Month	`July`; `Jul`; `07`
W	Week in year	Number	`27`
W	Week in month	Number	`2`
D	Day in year	Number	`189`
D	Day in month	Number	`10`
F	Day of week in month	Number	`2`
E	Day in week	Text	`Tuesday`; `Tue`
A	Am/pm marker	Text	`PM`
H	Hour in day (0-23)	Number	`0`
K	Hour in day (1-24)	Number	`24`
K	Hour in am/pm (0-11)	Number	`0`
H	Hour in am/pm (1-12)	Number	`12`
M	Minute in hour	Number	`30`
S	Second in minute	Number	`55`
S	Millisecond	Number	`978`
Z	Time zone	General time zone	`Pacific Standard Time`; `PST`; `GMT-08:00`
Z	Time zone	RFC 822 time zone	`-0800`

For example, to specify a date in Month/Day/Year Hour24:Min:Seconds you can use the following format:

MM/DD/yyyy HH:mm:ss

Specifying Date Time Zone

Additionally, you can specify the time-zone in which a day is sent. This is important if your partner is collecting dates in a different time zone from yours. The time zone can be specified by appending a semicolon to the date format and using one of the values listed using the following query:

```
SELECT TZNAME FROM V$TIMEZONE_NAMES
```

For example, if your partner is sending dates in Pacific time, in Month/Day/Year Hour24:Min:Seconds you can indicate this as follows:

```
MM/DD/yyyy HH:mm:ss;America/Los_Angeles
```

Note that this is only required if your partner operates in a time zone that is different from yours.

Number Formats

Number formats can be parsed based on a specification using the following characters to construct numeric formats:

Symbol	Location	Meaning
`0`	Number	Digit
`#`	Number	Digit, zero shows as absent
`.`	Number	Decimal separator or monetary decimal separator
`-`	Number	Minus sign
`,`	Number	Grouping separator
`E`	Number	Separates mantissa and exponent in scientific notation. *Need not be quoted in prefix or suffix.*
`;`	Subpattern boundary	Separates positive and negative subpatterns
`%`	Prefix or suffix	Multiply by 100 and show as percentage
`\u2030`	Prefix or suffix	Multiply by 1000 and show as per mille value
`¤(\u00A4)`	Prefix or suffix	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
`'`	Prefix or suffix	Used to quote special characters in a prefix or suffix, for example, `"'#'#"` formats 123 to `"#123"`. To create a single quote itself, use two in a row: `"# o''clock"`.

Generally FeedXChange uses the Java DecimalFormat to parse numbers. It has come to our attention that this type of parsing in Java is fairly relaxed and will generally ensure numbers are valid, but cannot do much more strict validation.

As a workaround, if you need to do stricter validation on a number format, you can prefix the format with "RegEx:" and use a regular expression to validate the string representing the number. See the next section for more details.

Character Formats

Characters are validated using regular expression syntax which is a very powerful tool. FeedXChange uses the Java regular expression library, documented at the following URL:

http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

Activating and Revising Templates

A template needs to be activated in order to become usable in integrations. This activation makes the template available for use in Integrations. In order to make changes to an Active template, you need to put it in Revise status first. Once your changes are complete you need to Activate it again, which will automatically update any Integrations that are using this template.

To activate a New or Revised template:

1. Go to Site Map → Administration – FeedXChange → Template List
2. Select the template you want to activate
3. Click on the Activate button

To revise an Active template:

1. Go to Site Map → Administration – FeedXChange → Template List
2. Select the template you want to revise
3. Click on the Revise button

Configuring Integrations

About Integrations

An Integration is a configuration element that specifies how a specific type of file will be processed for a specific partner.

Each integration uses an input, and optionally, a response template to specify how files will be validated, loaded and optionally how responses will be generated. Additionally, each integration can specify a number of optional parameters that govern various aspects of the integration such as transport, scheduling and error handling.

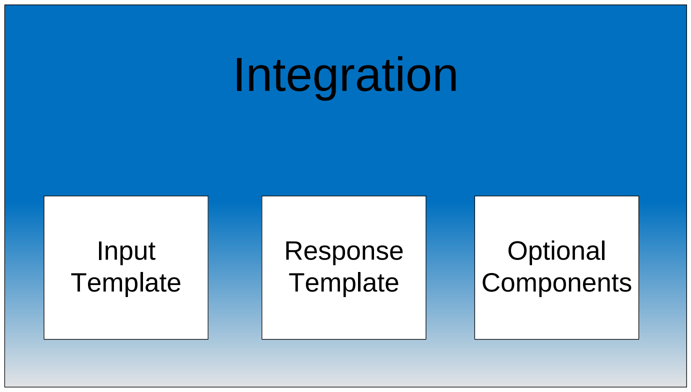
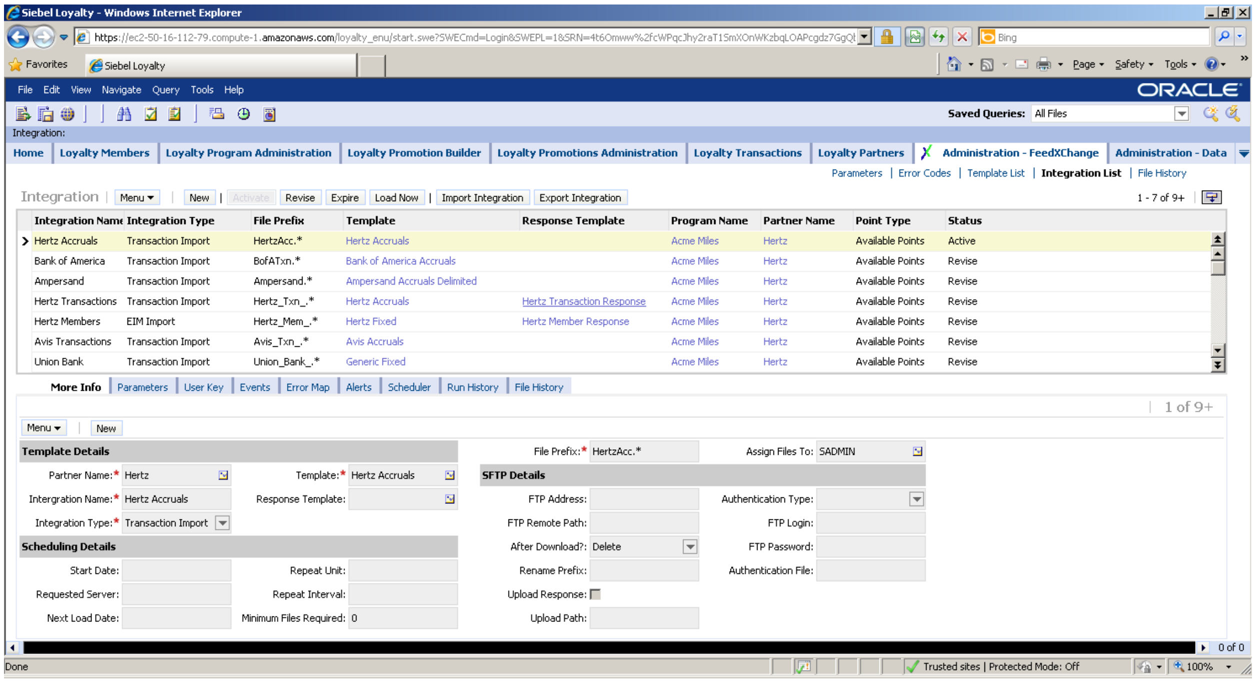


Figure 4 Integration Structure

The screen below shows the Integration List view in the Siebel UI:


Siebel Loyalty - Windows Internet Explorer
File Edit View Navigate Query Tools Help
ORACLE
Home | Loyalty Members | Loyalty Program Administration | Loyalty Promotion Builder | Loyalty Promotions Administration | Loyalty Transactions | Loyalty Partners | Administration - FeedXChange | Administration - Data
Parameters | Error Codes | Template List | Integration List | File History
Integration | Menu | New | Activate | Revise | Expire | Load Now | Import Integration | Export Integration
More Info | Parameters | User Key | Events | Error Map | Alerts | Scheduler | Run History | File History
Menu | New

Saved Queries: All Files

1 - 7 of 9+

Integration Name	Integration Type	File Prefix	Template	Response Template	Program Name	Partner Name	Point Type	Status
Hertz Accruals	Transaction Import	HertzAcc.*	Hertz Accruals		Acme Miles	Hertz	Available Points	Active
Bank of America	Transaction Import	BofATxn.*	Bank of America Accruals		Acme Miles	Hertz	Available Points	Revise
Ampersand	Transaction Import	Ampersand.*	Ampersand Accruals Delimited		Acme Miles	Hertz	Available Points	Revise
Hertz Transactions	Transaction Import	Hertz_Txn_.*	Hertz Accruals	Hertz Transaction Response	Acme Miles	Hertz	Available Points	Revise
Hertz Members	EIM Import	Hertz_Mem_.*	Hertz Fixed	Hertz Member Response	Acme Miles	Hertz	Available Points	Revise
Avis Transactions	Transaction Import	Avis_Txn_.*	Avis Accruals		Acme Miles	Hertz	Available Points	Revise
Union Bank	Transaction Import	Union_Bank_.*	Generic Fixed		Acme Miles	Hertz	Available Points	Revise

1 of 9+

Template Details

Partner Name:*	Hertz	Template:*	Hertz Accruals
Intergration Name:*	Hertz Accruals	Response Template:	
Integration Type:*	Transaction Import		

Scheduling Details

Start Date:		Repeat Unit:	
Requested Server:		Repeat Interval:	
Next Load Date:		Minimum Files Required:	0

File Prefix:*	HertzAcc.*	Assign Files To:	SADMIN

SFTP Details

FTP Address:		Authentication Type:	
FTP Remote Path:		FTP Login:	
After Download?:	Delete	FTP Password:	
Rename Prefix:		Authentication File:	
Upload Response:	☐		
Upload Path:			

Done | Trusted sites | Protected Mode: Off | 100%

Figure 5 Integrations List View

Creating Integrations

To create a new integration:

1. Go to Site Map → Administration – FeedXChange → Integration List
2. Create a new record using the New button in the Integrations list applet
3. Enter the following configuration details for the integration:

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

Configuration Detail	Description
Integration Name	Name for the integration you are creating.
Integration Type	This could be one of the following: - Transcription Import - EIM Import - File Import Please see the Product Overview section above for a description of what these processes do.
File Prefix	This is a regular expression that is used to locate the files that the integration will process. **Note:** This is not a "file mask". This is a regular expression, so for example, if you want to look for ".txt" files, you would use the following value for File Prefix: [^.]+\.txt This may seem cumbersome at first glance, but it is much more powerful than using file masks.
Template	Select the template which will be used to load the input files for this integration. **Note:** The template has to be Active to be selectable here.
Response Template	Select the template which will be used to create the response files for this integration. **Note:** The template has to be Active and marked as a Response template to be selectable here.
Program Name	Select the program for which you are using this integration.
Partner Name	Select the partner name that is sending you the files. **Note:** The partner has to be in your program in order to be selectable here.
Point Type	The default point type for which transactions are being imported. **Note:** This is a required field to import transactions correctly. However, it is really up to the program promotions to decide what points to award, etc.
Minimum Files Required	This parameter determines how the integration will behave if it does not find enough files to process. If this parameter is set 0, it is ok for the integration to find no files at all. However, if it specifies 1 or more files, then the integration will generate an error message if it finds fewer files for

	processing. In addition, it is possible to configure the scheduler to re-try later to see if the files have actually arrived. See the section on Configuring Integration Schedules for more details.
Assign Files To	Specify the user who will be responsible for this integration. This person will be given ownership to any files for this integration and will be allowed to intervene manually if necessary.

Configuring Integration Transport

Normally the integration will look for files in a specified incoming folder (see Configuring Global Parameters), and it will generate responses in a specified outgoing folder.

However, it is possible to configure the integration to use SFTP to pull files from a partner, as well as use SFTP to publish responses back to a partner.

To configure SFTP transport options:

1. Go to Site Map → Administration – FeedXChange → Integration List → More Info
2. Go to the More Info applet and enter the following configuration details:

Configuration Detail	**Description**
FTP Address	The URL of the SFTP server you are connecting to. For example: sftp.loyaltymethods.com Note that if you want to specify a port other than the default of 22, you can use the “:” notation, for example: sftp.loyaltymethods.com:7200
FTP Remote Path	This is the absolute path where FeedXChange will look for files on the remote server.
After Download?	This determines the action that FeedXChange will take after downloading the file. The possible options are: - Delete - Rename - <Blank> means don’t do anything
Rename Prefix	If you specified Rename in the above option, this will be the prefix to use, for example “downloaded”.
Authentication Type	SFTP can be configured to authenticate using two methods: - Login – Using a username and password - File – Using a private key file
FTP Login	If you are using Login authentication type, enter the

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

	username here.
FTP Password	If you are using Login authentication type, enter the password here.
Authentication File	If you are using File authentication type, then specify the absolute path on the Siebel server, where your key file is located.
Upload Response	Check this box if you want FeedXChange to publish the response file to the partner SFTP site.
Upload Path	This specifies where on the partner SFTP server the response files will be uploaded.

Configuring User Key De-duplication

Optionally, integrations can specify a user key in order to detect duplicate records in the input. The user key specifies which fields in the partner file will be considered a unique combination.

Note: FeedXChange builds an indexed de-duplication key in the CX_FINT_* staging table based on the fields selected. It automatically adds the partner name to this key in order to ensure that uniqueness is checked within the boundaries of a partner. This ensures we can detect duplicates in the file, as well as duplicates on records already in the database.

Note: If no user key is specified the de-duplication check will be skipped entirely.

To enable user key de-duplication:

1. Go to Site Map → Administration – FeedXChange → Integration List → User Key
2. Use the New and Delete buttons to select the fields which comprise the user key.

Configuring Error Translation

Optionally, an integration may configure translating record-level error codes and descriptions that FeedXChange uses to equivalent codes that a partner uses.

To enable this type of translation:

1. Go to Site Map → Administration – FeedXChange → Integration List → Error Map
2. Use the New and Delete buttons to add/remove error translations.

Configuring Alerts

Optionally, an integration can send out email alerts under certain circumstances, namely:

1. Start of integration/file
2. Success of integration/file
3. Error in integration/file

To configure these alerts:

1. Go to Site Map → Administration – FeedXChange → Integration List → Alerts
2. Use the New/Delete buttons to add alerts with the following configuration details:

Configuration Detail	Description
Inactive	Flag that inactivates the alert.
Name	Name for the alert.
Alert Type	This currently can be one of the three types: - Start - Success - Error
Email Address	This is a list of comma-separated email addresses that will receive the communication.

Configuring Integration Parameters

Optionally, an Integration can override the global parameters (see Configuring Global Parameters). This may be needed if, for example, an integration wants to use a different suffix for response files, or perhaps needs to pull data from a different directory than the rest of the integrations.

In some cases, an Integration may be scheduled to run on a separate Siebel server. In those cases, the integration will need to override the srvrmgr.param.server parameter.

To configure integration parameters:

1. Go to Site Map → Administration – FeedXChange → Integration List → Parameters
2. Use the New/Delete buttons to add parameters and corresponding override values.

Configuring Integration Events

Events are an optional mechanism to execute any command line process before or after a step in an integration. There are a number of event types that can be specified. Please review the section on Extending FeedXChange for more information on those parameters.

To configure integration events:

1. Go to Site Map → Administration – FeedXChange → Integration List → Events
2. Use the New/Delete buttons to enter the following configuration details:

Configuration Detail	Description
Sequence	The sequence determines ordering of Command Line execution for the same event.
Event	The type of event you would like to handle.

Command Line	The command line to be executed during this event. **Note:** Before configuring these events please review the section on Extending FeedXChange.

Configuring Integration Schedules

Integrations can be scheduled to run automatically using the Siebel built-in scheduler. Alternatively, integrations can be invoked via command line by a 3rd party scheduler such as cron. Please see the section on Running Integrations to find out more about command line invocation.

Just like in normal Siebel jobs, you may schedule a a repeating component request which would run an integration. Please review the Siebel Administration Guide for more information on how to configure jobs and repeating jobs. We will provide some instructions here as well, but some familiarity with the Siebel scheduler would be helpful.

To schedule an integration to run using the Siebel scheduler:

1. Go to Site Map → Administration – FeedXChange → Integration List → Scheduler
2. Enter the following configuration details:

Configuration Detail	**Description**
Scheduled Start	The scheduled start for the integration.
Expiration	Date after which the integration schedule is no longer valid.
Requested Server	Which server you want the integration to run on, or leave blank to default the server.
Request Key	Leave this field blank.
Delete Interval	Set this to 1.
Delete Unit	Set this to Never.
Retry on Error	Whether to retry on error – for an integration this only retries when the integration does not find enough files.
Sleep Time	How long to wait before re-trying.
Number of Retries	How many times to retry.
Current Num of Retries	How many times this particular job has been retried already.
Repeating	Check this box
Repeat Unit	The timeframe specified by the Repeat Interval
Repeat Interval	How long to wait before repeating.
Repeat From	How to schedule the next iteration – you can choose to schedule it from different points in time: - Scheduled Start - Actual Start - End Our advice is to use Scheduled Start to enable the integration to run at the same time every iteration.

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

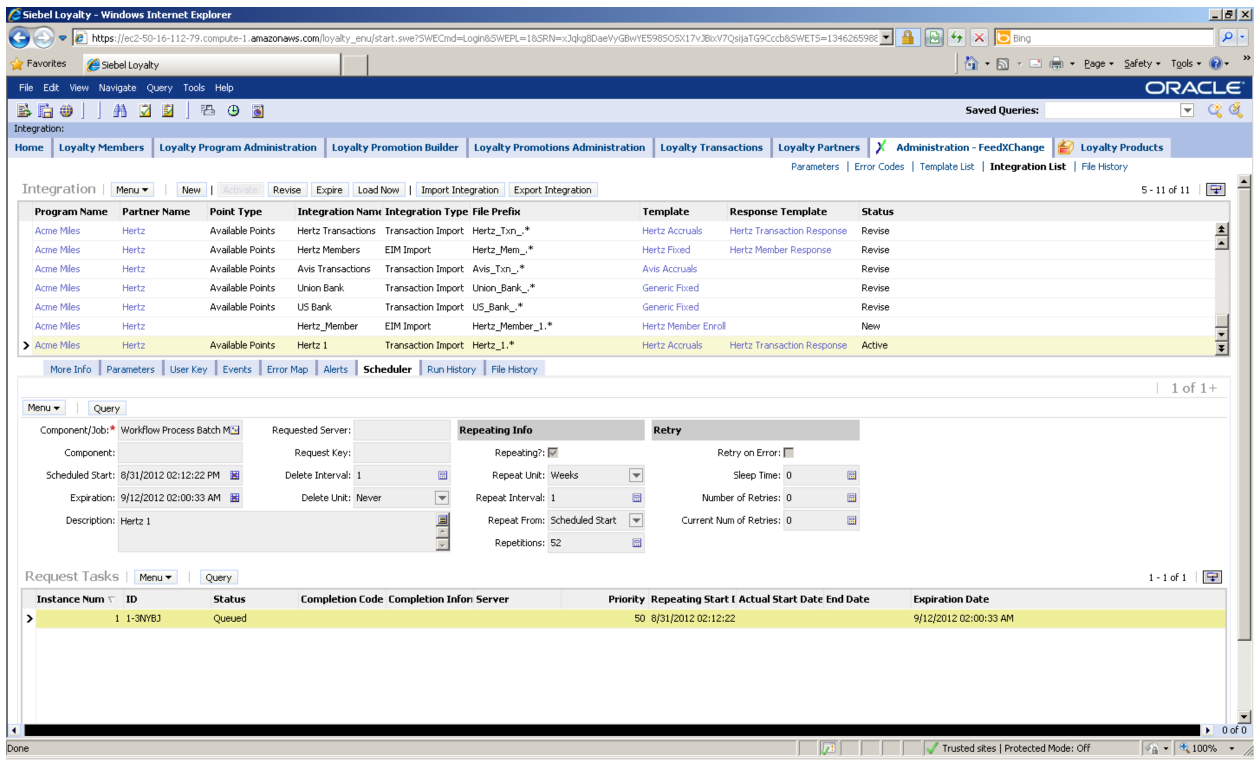
Repetitions	How many times you want to repeat this integration.

Running Integrations

There are several ways to launch an integration which we describe in the following paragraphs. In addition, this chapter will cover the monitoring and manual interventions.

Triggering Scheduled Integrations

Scheduled integrations are automatically launched when the scheduled next run occurs. This is based on the schedule you specify for each integration. Please see the section on Integration Schedules for more details on setting up schedules. The next run-time will be shown in the Scheduler view under the integration as shown in Figure 6 below:


Siebel Loyalty - Windows Internet Explorer
https://ec2-50-16-112-79.compute-1.amazonaws.com/loyalty_enu/start.swe?SWECmd=Login&SWEPL=1&SRN=xJqkg8DaeVyGBwYE598SOSX17vJBixV7QsijaTG9Cccb&SWETS=134626598
Bing
Favorites
Siebel Loyalty
Page
Safety
Tools
File Edit View Navigate Query Tools Help
ORACLE
Saved Queries:
Integration:
Home
Loyalty Members
Loyalty Program Administration
Loyalty Promotion Builder
Loyalty Promotions Administration
Loyalty Transactions
Loyalty Partners
Administration - FeedXChange
Loyalty Products
Parameters | Error Codes | Template List | Integration List | File History
Integration
Menu
New
Activate
Revise
Expire
Load Now
Import Integration
Export Integration
5 - 11 of 11
More Info
Parameters
User Key
Events
Error Map
Alerts
Scheduler
Run History
File History
1 of 1+
Menu
Query
Request Tasks
1 - 1 of 1
0 of 0
Done
Trusted sites | Protected Mode: Off
100%

Program Name	Partner Name	Point Type	Integration Name	Integration Type	File Prefix	Template	Response Template	Status
Acme Miles	Hertz	Available Points	Hertz Transactions	Transaction Import	Hertz_Txn_.*	Hertz Accruals	Hertz Transaction Response	Revise
Acme Miles	Hertz	Available Points	Hertz Members	EIM Import	Hertz_Mem_.*	Hertz Fixed	Hertz Member Response	Revise
Acme Miles	Hertz	Available Points	Avis Transactions	Transaction Import	Avis_Txn_.*	Avis Accruals		Revise
Acme Miles	Hertz	Available Points	Union Bank	Transaction Import	Union_Bank_.*	Generic Fixed		Revise
Acme Miles	Hertz	Available Points	US Bank	Transaction Import	US_Bank_.*	Generic Fixed		Revise
Acme Miles	Hertz		Hertz_Member	EIM Import	Hertz_Member_1.*	Hertz Member Enroll		New
Acme Miles	Hertz	Available Points	Hertz 1	Transaction Import	Hertz_1.*	Hertz Accruals	Hertz Transaction Response	Active

Component/Job:* Workflow Process Batch M
Component:
Scheduled Start: 8/31/2012 02:12:22 PM
Expiration: 9/12/2012 02:00:33 AM
Description: Hertz 1
Requested Server:
Request Key:
Delete Interval: 1
Delete Unit: Never

Repeating Info
Repeating?: ✓
Repeat Unit: Weeks
Repeat Interval: 1
Repeat From: Scheduled Start
Repetitions: 52

Retry
Retry on Error:
Sleep Time: 0
Number of Retries: 0
Current Num of Retries: 0

Instance Num	ID	Status	Completion Code	Completion Infor	Server	Priority	Repeating Start [	Actual Start Date	End Date	Expiration Date
1	1-3NYBJ	Queued				50	8/31/2012 02:12:22			9/12/2012 02:00:33 AM

Figure 6 Next Scheduled Integration

Triggering Integrations Manually

Integrations can be triggered manually through the Siebel UI. To trigger an integration manually:

1. Go to Site Map → Administration – FeedXChange → Integration List
2. Select the integration you want to run.
3. Click on the Load Now button.

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

Triggering Using the Command Line Interface

In some cases, you may want to run an integration through command line due to integration with a third party scheduler such as cron. When an integration runs from the command line, it will still reflect all changes and monitoring information in Siebel UI, it just won't be triggered by Siebel.

To run an integration through the command-line interface:

1. Login to a Siebel server where FeedXChange is installed.
2. Go to the fx/bin folder and source the env.sh script to setup environment variables. Type this into command line:

```
source env.sh
```

3. Now run the integration using it's name, using the following command line:

```
runbatch "Integration Name" none
```

where "Integration Name" is the name of your integration surrounded by double quotes.

If you are trying to schedule this using cron, the command line can be abbreviated in the Linux shell as:

```
source /lmva/addons/fx/bin/env.sh; /lmva/addons/fx/bin/runbatch "Integration Name"
none
```

This assumes that your FeedXChange base directory is /lmva/addons/fx. Please adjust the command line to fit your installation.

Monitoring Integrations and Files

Integrations have two types of logs that are typically of interest – the Run History, and the File History. We will examine each of these two in the next sections.

Viewing Integration Run History

Integration Run History contains records with information for each time the integration was triggered. To view integration Run History:

1. Go to Site Map → Administration – FeedXChange → Integration List → Run History

The following details are shown in the Run History view, inside the Integration Log applet:

Detail	Description
Date	The date and time when the integration was triggered.
Status	The status of the integration. Could be:

	- In Progress - Finished - Error
Run Details	This is a message that clarifies the status, indicating how many files were processed, how many failed, etc.
Log File	This is a link to the Log file for the integration. Click on this link to view the log file inside the Siebel UI.
Comments	This contains two important pieces of information: - PID – the process id of the java process which runs for this integration in case you need to kill it for some reason. - Session – the database session of this particular integration run. This is useful in case there was a problem with a database operation and the DB needs to be killed.

On the right hand side, there is a File applet, which contains one record for each file. This is the same as the File History view which we examine in the next section.

Viewing File History

The File History view shows records with details for each file that was attempted in each integration. To view the File History:

1. Go to Site Map → Administration – FeedXChange → File History

The file history view shows the following details (in the list and corresponding form applet):

Detail	**Description**
File Name	The name of the file that was attempted.
Integration Name	The parent integration that processed this file.
File Stage	This indicates at what stage the file is currently in the processing cycle.
Status	This is the file status which can be: - Complete – if the file was successfully processed. - Partially Complete – some records were in error. - Failed – no records were successfully processed. - Pending – if a Reload operation was attempted.
Error Code	A FeedXChange error code in the form of SBL-FINT-XXXX.
Error Description	An error message that indicates problems that occurred with this file when the state is not Complete.
Start Time	When the file started processing.
End Time	When the file finished processing.
Total Count	Total number of records read from the file excluding

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

	header/trailer record.
Validated Count	Total number of records which successfully passed all validation rules.
Invalid Count	Total Count – Validated Count
Loaded Count	The number of records successfully loaded through EIM.
Processed Count	The number of records successfully processed. Depending on the Integration Type, this may mean: - Successfully processed in the Engine (Transaction Import) - Successfully loaded through EIM (EIM Import) - Successfully validated (File Import)
Queued Count	Records re-queued for a Reload of the file after manual intervention.
File Date	This field can be populated from the header/footer if mapped in the template.
File Size	This field can be populated from the header/footer if mapped in the template.
Total Amount	This field can be populated from the header/footer if mapped in the template.
Total Points	This field can be populated from the header/footer if mapped in the template.
Owner	This is the Siebel user assigned to this file in the Integration configuration.
Comments	This is the Siebel user assigned to this file in the Integration configuration.
Log File	This is the processing log file for this attempt on the file. Click on this link to drill down directly into the log from the Siebel UI.

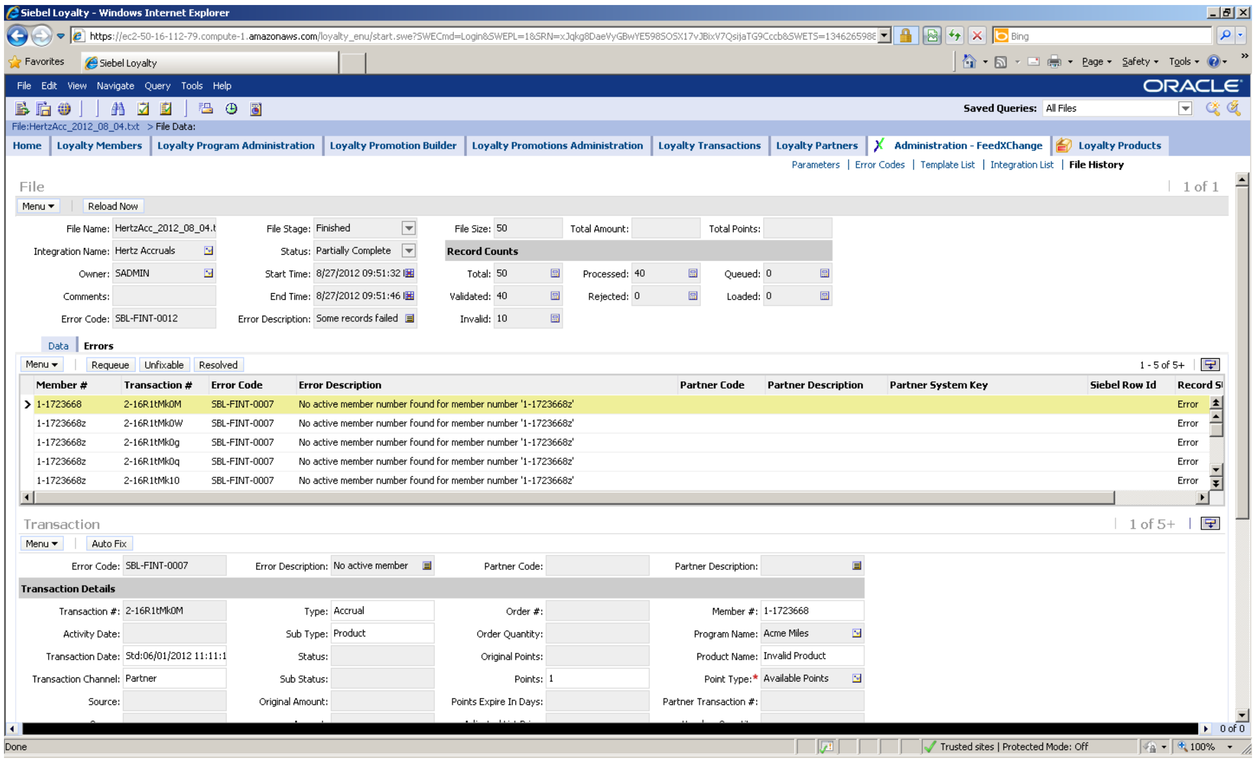
Applying Manual Corrections and Rerunning Files

When a file fails partially complete, it is possible to intervene manually and resolve any errors, then reload the same file to complete its processing without having to re-do the entire file.

To correct any problems with a file:

1. Go to Site Map → Administration – FeedXChange → File History
2. Drill down on the File Name
3. This takes you into the File Data Error view as shown below in Figure 7.

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.


File:HertzAcc_2012_08_04.txt > File Data:

File

File Name:	HertzAcc_2012_08_04.t	File Stage:	Finished	File Size:	50
Integration Name:	Hertz Accruals	Status:	Partially Complete	Total Amount:	
Owner:	SADMIN	Start Time:	8/27/2012 09:51:32	Total Points:	
Comments:		End Time:	8/27/2012 09:51:46		
Error Code:	SBL-FINT-0012	Error Description:	Some records failed		

Record Counts

Total:	50	Processed:	40	Queued:	0
Validated:	40	Rejected:	0	Loaded:	0
Invalid:	10				

Data | Errors — 1 - 5 of 5+

Member #	Transaction #	Error Code	Error Description	Partner Code	Partner Description	Partner System Key	Siebel Row Id	Record St
1-1723668	2-16R1tMk0M	SBL-FINT-0007	No active member number found for member number '1-1723668z'					Error
1-1723668z	2-16R1tMk0W	SBL-FINT-0007	No active member number found for member number '1-1723668z'					Error
1-1723668z	2-16R1tMk0g	SBL-FINT-0007	No active member number found for member number '1-1723668z'					Error
1-1723668z	2-16R1tMk0q	SBL-FINT-0007	No active member number found for member number '1-1723668z'					Error
1-1723668z	2-16R1tMk10	SBL-FINT-0007	No active member number found for member number '1-1723668z'					Error

Transaction — 1 of 5+

Error Code:	SBL-FINT-0007	Error Description:	No active member	Partner Code:		Partner Description:	

Transaction Details

Transaction #:	2-16R1tMk0M	Type:	Accrual	Order #:		Member #:	1-1723668
Activity Date:		Sub Type:	Product	Order Quantity:		Program Name:	Acme Miles
Transaction Date:	Std:06/01/2012 11:11:1	Status:		Original Points:		Product Name:	Invalid Product
Transaction Channel:	Partner	Sub Status:		Points:	1	Point Type:*	Available Points
Source:		Original Amount:		Points Expire In Days:		Partner Transaction #:	

Siebel Loyalty - Windows Internet Explorer
Favorites
Siebel Loyalty
File Edit View Navigate Query Tools Help
ORACLE
Saved Queries: All Files
Home | Loyalty Members | Loyalty Program Administration | Loyalty Promotion Builder | Loyalty Promotions Administration | Loyalty Transactions | Loyalty Partners | Administration - FeedXChange | Loyalty Products
Parameters | Error Codes | Template List | Integration List | File History
Menu | Reload Now
Menu | Requeue | Unfixable | Resolved
Menu | Auto Fix
Done
Trusted sites | Protected Mode: Off
100%

Figure 7 File Data Error View

This view contains 4 applets, which are described below from top to bottom:

Applet	Description
File Form Applet	This applet is similar to the form applet in File History view and it displays the details of the file and statistics about processing so far. This applet also has the “Reload Now” button which allows you to re-start a file processing from where it failed after you have manually intervened.
Staging List Applet	This is an applet with a list of the records from the CX_FINT_* staging table which you are processing and their status, including any errors that may have occurred. The Errors tab shows just the records which are in error. The Data tab shows all the records in the staging table for the file. If the records have been already loaded into the Siebel base tables, you can drill down to the Siebel views using the Siebel Row Id drill down.

	This applet also contains the Requeue, Unfixable and Resolved buttons.
Staging Detail Applet	This is an applet which has all the columns from the staging table shown. The columns which are marked as Editable in your template will be editable in this view. This applet also has the Auto Fix button which allows applying the same fix to multiple records.
Audit Trail Applet	This is an applet which records an audit trail of any changes made through manual intervention.

Correcting Individual Records

You can correct individual records by changing column values in the Staging Detail Applet. Once you have made your changes and saved the current record (by stepping off, pressing Ctrl+S, etc.) the audit trail records show the old and new values to any fields you changed.

Also, the Auto Fix button becomes enabled for the record which you changed.

Auto Fixing Multiple Records

If the Auto Fix button is enabled, and for the fields which you marked Auto Fixable in your template, you will be able to apply the same change (Old → New) from the audit trail to any records in the Errors view that have the same Old value for a field. To do that, just press the Auto Fix button.

Re-queuing Records for Reload

If you have corrected any records, you can mark them Queued using the Requeue button in the Staging List Applet:

1. In the Staging List Applet, select all records you want to mark Queued.
2. Click on the Requeue button.

Marking Records Unfixable

Sometimes it is impossible to fix an error and you want to send a response back to the partner that that record was not able to process. To do that:

1. In the Staging List Applet, select all records you want to mark Unfixable.
2. Click on the Unfixable button.

Marking Records Resolved

Sometimes, a record is in Error, but you are able to go directly in Siebel and apply the transaction manually. In those cases, you can mark the record as Resolved, which will send a response back to the partner that this record was successfully processed. To do that:

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

1. In the Staging List Applet, select all records you want to mark Resolved.
2. Click on the Resolved button.

Reloading a Corrected File

Once you have marked records as Queued, Unfixable or Resolved, you need to re-process the file by clicking the Reload Now button in the File Form Applet.

This will place the file in Pending status, and will re-run just that file individually, outside of any integration runs.

Note: Response will not be generated for a particular file until its processing status is Complete.

Migrating Templates and Integrations

FeedXChange is designed in such a way so as to avoid any downtime during deployment of new templates and integrations. This is done through exporting templates and integrations to XML files directly from the Siebel UI.

Template Export/Import

To export a template:

1. Go to Site Map → Administration – FeedXChange → Template List
2. Select the template you want to export.
3. Click on the Export Template button.
4. Specify a File Name on your local machine where the template will be stored.
5. Click on the OK button.

To import a template:

1. Go to Site Map → Administration – FeedXChange → Template List
2. Click on the Import Template button.
3. Click the Browse button to select an XML file with a previously exported template.
4. Select the XML file in the dialog box.
5. Click on the OK button.

Integration Export/Import

To export an integration:

1. Go to Site Map → Administration – FeedXChange → Integration List
2. Select the integration you want to export.
3. Click on the Export Integration button.
4. Specify a File Name on your local machine where the integration will be stored.
5. Click on the OK button.

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.

To import an integration:

1. Go to Site Map → Administration – FeedXChange → Integration List
2. Click on the Import Integration button.
3. Click the Browse button to select an XML file with a previously exported integration.
4. Select the XML file in the dialog box.
5. Click on the OK button.

Note: You need to ensure that any templates used by the integration either exist in the target environment, or that you import them before you attempt importing the integration that makes use of them.

Note: Integrations will be imported in Revise status, to avoid accidentally triggering them into the target environment. You will need to Activate an integration after you have imported it.

Configuring Global Parameters

FeedXChange has a number of global parameters, which can be changed for all integrations, or can be overridden in each integration's Parameters view. Please review the Comments on each parameter to determine what it is used for.

Configuring Global Error Messages

FeedXChange allows you to customize the error messages that it produces. You can change the text of each message as needed, or translate it into a different language.

To edit error messages:

1. Go to Site Map → Administration – FeedXChange → Errors
2. Edit the Error Description field as needed.

Note: Error messages are formatted with positional parameters to indicate specifics such as field names, formats, etc. This makes the messages more readable. These positional parameters are marked with %1, %2, etc. We suggest that you re-use them in your error descriptions to make your messages more concrete and understandable.

Extending FeedXChange with Events

About Events

Events are specific points of an integration which can be intercepted, in order to inject custom processing code at different stages of an integration. This is useful in several scenarios, such as custom record-level validations, file-level validations, custom member number lookups and other types of pre- and post-processing.

Event Types

The events are specified in the Events view of each integration. For each processing stage, there are Before and After events which are listed in the Event field drop-down, for example “Before Integration” event runs any code before an integration starts, the “After File Validation” runs after a file has been staged and validated, etc.

Event Protocol

Events are executed as command line strings called handlers, in the order of the Sequence specified in the Events view for the integration.

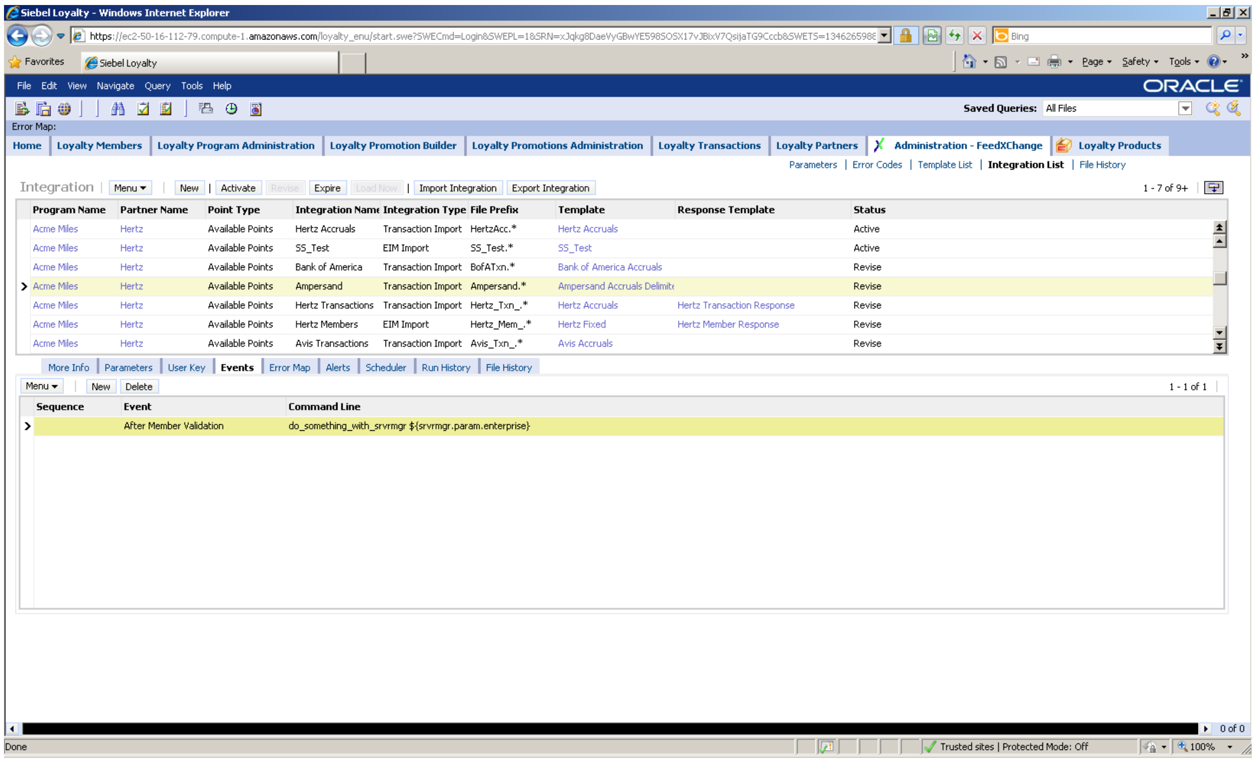
An event command line can return one of 3 values as its OS exit code, which is interpreted by FeedXChange:

Exit Code	What FeedXChange Does
0	FeedXChange assumes the handler ran successfully.
1	FeedXChange assumes the handler encountered an error which requires the current file to be marked in Error status and processing to stop.
2	FeedXChange assumes the handler failed, but processing can continue. At the end of the current file, FeedXChange will still mark the file Failed.

FeedXChange will capture both Standard Output (STDOUT) and Standard Error (STDERR) for any handler executed. This information will be stored in the file log in case of an error.

It is possible to pass any of the global or integration-level parameters to event handlers as command line parameters. For any parameter, for example srvrmgr.param.enterprise, you can use the notation ${srvrmgr.param.enterprise} to refer to it from a handler Command Line spec. See screenshot in Figure 8.

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.



Integration | Menu | New | Activate | Revise | Expire | Load Now | Import Integration | Export Integration

1 - 7 of 9+

	Program Name	Partner Name	Point Type	Integration Name	Integration Type	File Prefix	Template	Response Template	Status
	Acme Miles	Hertz	Available Points	Hertz Accruals	Transaction Import	HertzAcc.*	Hertz Accruals		Active
	Acme Miles	Hertz	Available Points	SS_Test	EIM Import	SS_Test.*	SS_Test		Active
	Acme Miles	Hertz	Available Points	Bank of America	Transaction Import	BofATxn.*	Bank of America Accruals		Revise
>	Acme Miles	Hertz	Available Points	Ampersand	Transaction Import	Ampersand.*	Ampersand Accruals Delimite		Revise
	Acme Miles	Hertz	Available Points	Hertz Transactions	Transaction Import	Hertz_Txn_.*	Hertz Accruals	Hertz Transaction Response	Revise
	Acme Miles	Hertz	Available Points	Hertz Members	EIM Import	Hertz_Mem_.*	Hertz Fixed	Hertz Member Response	Revise
	Acme Miles	Hertz	Available Points	Avis Transactions	Transaction Import	Avis_Txn_.*	Avis Accruals		Revise

More Info | Parameters | User Key | **Events** | Error Map | Alerts | Scheduler | Run History | File History

Menu | New | Delete

1 - 1 of 1

	Sequence	Event	Command Line
>		After Member Validation	do_something_with_srvrmgr ${srvrmgr.param.enterprise}

Figure 8 Passing a Parameter to a Handler

In addition to all the global parameters, there are some dynamic run-time parameters based on the file you are currently processing:

Dynamic Parameter	**Description**
${ctx.fileId}	The ROW_ID of the File History record for the file which we are running currently.
${ctx.fileName}	The filename which we are currently processing.
${ctx.integrationId}	The ROW_ID of the Integration which we are currently running.
${ctx.stepName}	The name of the step we are currently executing.
${ctx.stepStatus}	The status of the last executed step. Can be either “Error” or “Complete”.
${ctx.filePattern}	The regular expression we are using to locate files for this integration.

Copyright ©2010-2012 by Loyalty Methods Inc. All rights reserved.