

IBM Unica Detect
Version 8.5.0 Publication Date: June 7, 2011

Library Guide



Copyright

© Copyright IBM 2011
IBM Corporation
Reservoir Place North
170 Tracer Lane
Waltham, MA 02451-1379

All software and related documentation is subject to restrictions on use and disclosure as set forth in the IBM International Program License Agreement, with restricted rights for U.S. government users and applicable export regulations.

Companies, names, and data used in examples herein are fictitious unless otherwise noted.

IBM, the IBM logo, Unica and the Unica logo, NetInsight, Affinium and MarketingCentral are trademarks or registered trademarks of the IBM Corporation in the United States, other countries or both. Other product and service names might be trademarks of IBM or other companies. © Copyright IBM Corporation 2011. All rights reserved.

Table of Contents

1 About this Guide	8
About this guide	8
2 Overview of Detect and the Trigger Libraries	10
Overview of Detect and the trigger libraries	10
About Detect	10
About the input data that Detect uses	10
About the output data that Detect produces	11
About behaviors that Detect can monitor.....	11
About trigger systems	11
How trigger systems monitor behavior	12
Example of a trigger system	12
About the Detect trigger libraries	12
Purpose of the trigger libraries.....	13
List of trigger libraries.....	13
About the content of each trigger library.....	14
About the category groupings used in libraries.....	14
About the structure of the library directory	14
About the documentation for the trigger libraries.....	15
About other documentation you should use	16
About the individual trigger system guides	16
About the Visio diagrams that come with each library.....	16
About the high-level Visio diagram	17
About the detailed Visio diagram	17
3 Working with a Trigger Library	18
About working with a trigger library	18
Obtaining the trigger library from IBM Unica	18
Installing a library	18
Moving trigger systems to another environment.....	21
About the environments you may choose to use.....	21

About suggested environments	21
4 Understanding the Library Trigger Systems	24
To understand a trigger system in a library	24
Importing a selected trigger system.....	25
To extract a trigger from the library.....	25
Creating the lookup tables required for a trigger system.....	26
To determine whether you must create lookup tables for your library.....	26
To create the lookup tables	26
To locate the documentation and for the trigger system	26
To obtain a high-level understanding of a trigger system's logic.....	27
About the Visio diagrams for trigger systems.....	27
About the test cases for a trigger system	28
To test the trigger system against the provided feed files	29
About the outcome and the additional information	30
To obtain a deeper understanding of the logic.....	30
Using the Library Management tools to remove unnecessary fields.....	31
To limit dependencies	31
Locating the trigger system in Detect	32
About workspaces.....	32
Using runsets to group triggers	32
Refining a trigger system	32
To document trigger system logic.....	33
5 Example of How to Understand a Trigger System	34
About this example	34
About the Retail Banking On-boarding library	35
About the trigger system used in this example.....	35
To locate the trigger system in this guide	36
About the descriptive documentation and Visio diagrams for the trigger system.....	36
About the component logic	37
Understanding the logic in greater detail	40
About the test cases for the example trigger system	41
About the data sources for this trigger system	41
About the location of the test case files	42
About entity types in data sources.....	42

About the transaction feed file	43
About the account profile feed file for the sample trigger system.....	43
About the lookup tables	45
About how you would run the test cases	45
A Overview of the Implementation Process	46
About the implementation process	46
Building an operational environment	46
Transitioning the trigger systems to run against your own data	47
Using the results	48
B About Data Sources	50
About data sources	50
About transaction data files	51
About the account profile file	51
About lookup tables	51
About rule parameter lookup tables.....	52
About the Merchant Category Code lookup table.....	52
About how trigger systems use lookup tables	53
About the transaction code table	53
Example of using a trancode lookup table.....	53
Example of a trancode lookup table	53
About data sources and fields used in libraries.....	54
Additional resources	54
C OA1 - Sample Trigger Documentation.....	55
OA1 - Sample trigger documentation	55
D Credit Card Event Based Marketing (CCEBM) Library	59
About the Credit Card Event Based Marketing (CCEBM) library	59
List of Credit Card Event Based Marketing (EBM) trigger systems.....	59
E Retail Banking Alerts Library	61
Retail banking alerts library	61
List of triggers in the Retail Banking Alerts library.....	61
F Additional Information About Using the Alerts Library for Retail Banking	66
About alerts and outcome values	66
Subscription information in the account profile file	67
Example of a subscription file format.....	67

About transaction data files for alerts	68
Outcome message XML	68
Identifying and using transaction variables.....	68
XML for transaction variables	69
Deactivating Subscribers	70
Alerts Transaction Code Categories	70
List of alerts and alert-specific variables.....	71
Deposit Account Alerts.....	72
Approaching Balance or Overdraft Threshold Alerts	76
Overdraft Occurred Alert.....	77
Budget Alerts	77
Unusual Account Activity Alerts	84
Scheduled Deposit Alerts.....	85
Scheduled Payment Alerts.....	86
G Retail Banking Event Based Marketing (RBEBM) Library	88
About the Retail Banking Event Based Marketing (RBEBM) library.....	88
List of Retail Banking Event Based Marketing (RBEBM) trigger systems.....	88
H Additional Information About Using the Retail Banking Event Based Marketing Library.....	97
More About Event Based Marketing Libraries	97
Account profile file for RBEBM triggers systems	97
Transaction code categories for RBEBM triggers	98
I Retail Banking On-boarding Library	101
About the Retail Banking On-boarding library	101
Retail Banking On-boarding Trigger System Library	101
J Telco Event Based Marketing, Account Based (TELA) Library.....	105
Telco Event Based Marketing, Account Based (TELA) library	105
About data requirements for the TELA library	105
List of Telco EBM, Account Based (TELA) trigger systems	106
K Telco Event Based Marketing, Phone Number Based (TELP) Library	108
Telco Event Based Marketing, Phone Number Based (TELP) library	108
About data requirements for the TELP library	108
List of Telco EBM, Phone Number Based (TELP) trigger systems	109
L Terminology Used in this Guide.....	111

Terminology used in this guide.....	111
-------------------------------------	-----

1 About this Guide


■ About this guide

About this guide

This guide is a general introduction to the packaged trigger libraries that can be purchased with IBM Unica Detect. It is also a companion to the documentation that comes with each trigger library. This guide provide:

- overview information about what is included in each of the trigger libraries
- information about how to use the trigger library documentation
- descriptions of some of the general mechanisms such as look up tables and other topics that the documentation for the specific triggers will assume that you know
- general information to help you understand how trigger systems work
- general information about how to extract the triggers you need and deploy them in your environment

The primary goal of this guide is to help you to understand and use your first trigger from the library. This document explains how to find a trigger, understand it, test it, and modify it for your business purposes.

 This document assumes that you have a compatible version of Detect installed in the environment in which you will be running the trigger systems.

Terminology

Within this guide the term trigger is generically used to mean a rule that causes an event to be inserted in the outcome table. However, the libraries are differentiated into one of two categories of trigger systems: triggers and alerts. In that sense, triggers are events sent to internal institution personnel, alerts are sent to the institution's customers. Additional terminology is described in "[Terminology used in this guide](#)".

About the scope of this guide

This guide provides an overview of the Detect trigger and alert libraries, the data required by Detect, and the output produced by Detect. This guide also provides an overview of the planning process to follow when you implement a trigger system in your organization. It is beyond the scope of this guide to describe subscriber support or internal processing. For additional details about working with Detect, refer to the *Detect Administration Guide* and the *Detect User Guide*.

About how to use this guide

Use this guide in conjunction with trigger-specific documentation included in the library, which explains the behavior of individual triggers. When you begin working with trigger libraries, use this guide to become familiar with the general information. After you are familiar with all the general information, you will be able to use the documentation for specific triggers.

If you are new to the libraries, familiarize yourself with the overall process by reading this guide. After you have acquired a general understanding, review the catalogs of triggers listed at the end of this guide in order to understand what triggers are available. If you have a trigger that you would like to deploy, locate the documentation for that trigger, understand what it does, and run the test cases. Refer to the general material in this guide for background information if needed.

2 Overview of Detect and the Trigger Libraries

- Overview of Detect and the trigger libraries
- About Detect
- About trigger systems
- About the Detect trigger libraries
- About the documentation for the trigger libraries
- About the Visio diagrams that come with each library

Overview of Detect and the trigger libraries

This section provides an overview of Detect and of the trigger libraries.

About Detect

Detect is an application that detects patterns of behavior in low-level customer transactions. Detect accepts high-volume feeds from a variety of transaction processing systems, such as Demand Deposit Account (DDA) transactions in banking or Call Detail Records (CDR) in telecommunications. Detect watches for patterns that indicate signs of attrition, readiness to be cross-sold, or opportunities to nurture your customer relationship. The output of Detect can be used to enhance the effectiveness of marketing campaigns or customer interactions.


About the input data that Detect uses

Detect can process data from multiple external sources, which are referred to as data sources. The inputs to Detect can include profile and transaction feed files, as well as lookup tables. Typical data sources are:

- **Profile feed files:** flat files that contain customer (entity) profile data such as account information, marketing segments, profitability scores, or demographic information not customarily included on the transactions themselves. This information can be alternatively provided in a profile table held within an external database. The information is available for use within Detect when processing transactions from any of the transaction feed files associated with that entity. Only one profile is allowed per entity type.

- **Transaction feed files:** flat files that contain raw transactions. At least one transaction feed file data source is required to run the Detect engine, although many may be used for the same entity type.
- **Lookup tables:** internal or external database tables that contains data that enhances the transactional information.

The format requirements for transaction feed files and profile feed files are standard, but the fields they must contain depend on what fields are required by the trigger systems that are implemented.

 Each of the trigger libraries includes sample input data required to test the trigger systems included in them.

About the output data that Detect produces

Detect writes its output to an outcome table. The output table contains a row for every event (trigger) detected, and contains information about what trigger fired, the time that it fired, who it fired for, and other relevant information supporting why the trigger fired. You can extract the information in the outcome table and use it in whatever manner suits your business purposes. It is natural to send this data through Campaign in order to attach additional information to the trigger, direct the outcome to the appropriate process or channel, and control the frequency of contact with your customers.

About behaviors that Detect can monitor

Detect functions by recognizing transaction events and patterns of those events, and then notifying (triggering) that a pattern was detected. It can recognize complex patterns of activity in incoming streams of transactions. For example, Detect can be configured to watch for behaviors that indicate:

- signs of attrition
- opportunities to cross sell
- opportunities to nurture the customer relationship

The types of patterns it uncovers depend on the types of event-detection algorithms for which it is configured to watch. A detection algorithm resulting in a triggered action is referred to as a trigger system.

About trigger systems

Detect recognizes transactions and patterns in customer data based on the trigger systems defined within it. Each trigger system is a collection of Detect components that define an event detection algorithm. The purpose of each trigger system is to recognize specific customer behavior. When a behavior is detected by a trigger system, Detect triggers an outcome.

Each trigger system defines how to pick out important events within the input transactions, it defines the patterns that you want to capture within the input events, and it defines how to package up gathered information and attach it to the triggered outcome.

You can:

- define your own trigger systems
- use trigger system libraries defined by IBM Unica Marketing. The trigger systems in these libraries can be used as delivered from IBM Unica or they can be further customized

How trigger systems monitor behavior

A trigger looks for patterns of activity in low level transactions. These transactions are fed into Detect as they become available. Generally transaction records are provided to Detect once a day in batch mode. Each trigger system is a consumer of these transactions.

Within Detect is an engine that stores key pieces of transaction information for each customer in its state history database table. Using sophisticated techniques for gauging normal activity and highly efficient strategies for utilizing historic information, Detect triggers an action as soon as it encounters a transaction that completes the detection criteria.

The outcome of Detect is more than an observation about customer behavior, and is more accurately treated as a call to action. For example, consider a trigger system that is watching for signs of attrition. If Detect indicates that a customer has a severe drop or complete stoppage in the frequency of interactions (deposits or purchases, for example) with your company, then this is most likely call to action that you would like to respond to immediately.

Example of a trigger system

An example of a detection scenario for customer attrition in retail banking focuses on a pattern of behavior involving a credit card and an ATM card for a checking account. The trigger system detects customers who have an average card balance that declines by 20% over two months, in conjunction with out of network ATM transaction volumes that increase by 20% over the same period of time. In this example, Detect evaluates the nightly transaction feeds for every customer, noting changes in balances and out-of-network ATM transactions volumes. If the 20% thresholds for balance decline and out of network ATM transaction volumes occur simultaneously, within a configurable overlap period, the trigger fires and sends a record to the outcome table.

About the Detect trigger libraries

IBM Unica Marketing has developed extensive libraries of triggers and alerts, each with its own business purpose. They are described in the library appendixes in this guide.

Purpose of the trigger libraries

Each Detect trigger library consists of a collection of detection triggers that serve a common business purpose. The trigger libraries have been built by Detect experts in partnership with domain experts and can be used to simplify your usage of Detect. You may choose to deploy some triggers from the library or simply use the library as a source of ideas for building your own trigger systems.

List of trigger libraries

There are a number of trigger libraries available for Detect. In some cases there are multiple libraries for an industry. The trigger libraries include:

- **Credit Card Event Based Marketing (CCEBM):** This library contains trigger systems that are used by credit card providers to watch accounts and detect behaviors that are indicative of opportunities to cross-sell, retain, or nurture the customer.
- **Retail Banking Alerts (RBA):** This library contains trigger systems that provide a watchful eye on the account on behalf of the retail banking customer. Through a bank-provided portal, banking customers can both subscribe to alerts and specify trigger thresholds and sensitivity. For the customers, the outcome of an alert could be an email or SMS message that notifies them of an account behavior that they are concerned about. The Alerts trigger library uses subscriber settings and detects account behaviors. The mechanism for gathering subscriber settings through a portal and the sending of emails is done outside of Detect.
- **Retail Banking On-boarding (RBONB):** This library contains trigger systems that provide a mechanism for monitoring the behavior of newly-acquired retail banking accounts with the intent of following up with individual customers who are not using banking services as expected. These trigger systems are intended to help identify a new account that has not fully come onboard with the bank, possibly because a back office process may have gone wrong when making a new customer into a fully functioning account, or possibly because the customer has not yet decided to fully stop using their older account with their previous bank and move over to this new account.
- **Retail Banking Event Based Marketing (RBEBM):** This library contains trigger systems that monitor the behavior of mature retail banking customers. The triggers identify behaviors that indicate opportunities for cross-selling, retention, and nurturing. You can distribute the outcome of these triggers, for example through Campaign, to a member of your organization who will use the content of the triggers as a basis for contact; or you can feed the triggers directly to Campaign to be used as content for a marketing campaign.
- **Telco Event Based Marketing, Account based (TELA):** This library contains trigger systems used by telephone service providers to monitor call usage and recharge records for behaviors that indicate opportunities to cross-sell, retain, or nurture the owner of the account. The trigger systems in this library use the account number of the telephone company (telco) customer as the focus of detection and can monitor either outgoing or incoming call behavior.

- **Telco Event Based Marketing, Phone number based (TELP):** This library contains trigger systems used by telephone service providers to monitor call usage and recharge records for behaviors of individual phone numbers that indicate opportunities to cross-sell, retain, or nurture the owner of the phone. The trigger systems in this library use the phone number of the calling party as the focus of detection and monitor outgoing call behavior.

About the content of each trigger library

Each library contains multiple pieces of information, arranged within a directory. The majority of the content is contained in two folders.

- **Exports folder:** This folder contains the trigger system files that you can import into Detect using the Library Management tools included with Detect. It also contains scripts to create and populate database lookup tables, if any are needed by the trigger systems in the library.
- **Documentation folder:** This folder contains a folder for each category of trigger in the library. Within each category is a document that describes the trigger category as well as any special information about the individual trigger systems within the category. Test case data is provided for each individual trigger system so you can test the trigger system and examine the results.

About the category groupings used in libraries

Within a library, trigger systems with similar structure are grouped together into a conceptual collection called a category. The trigger system documentation that is shipped within each library reflects this same category structure.

About the structure of the library directory

This illustration shows the structure of the library directory. In this example, there is only one category within the library, but libraries typically have multiple categories.

```
| Library-1
|
|----Library-1 Spreadsheet (Library Overview and Data Sources, shown on separate worksheets)
|
|----Documentation folder
|
|-----Category-1 folder
|
|-----Category-1 Visio design document (.vsd)
|
|-----Category-1 documentation for the trigger or category (.doc)
|
|-----Feed file folder for individual trigger system (may be named for the workspace)
|
|-----Profile feed file(s) for testing the trigger system
|
|-----Transaction feed file(s) for testing the trigger system
|
|----Exports Folder
|
|-----The .xml file for the trigger systems, plus scripts for creating any required lookup tables
```

About the documentation for the trigger libraries

Each library comes with documentation for the trigger systems that it contains. This list describes the documentation for the trigger libraries:

- **Detect Library Guide:** This is the guide you are reading and is the document with which to start. This guide contains an overview of how to use the libraries. This guide also lists all of the libraries and describes the trigger systems that are in each library.
- **Individual trigger system guides:** There is a descriptive document for every trigger system, or category of trigger systems, in each library. Each document provides a detailed description of the trigger category as well as any special information about the individual trigger systems within the category. The document provides details about the test cases and expected results as well as design decisions and modification options for the triggers in the category. The guides are included in the directory with the other library files.
- **Visio diagrams:** Each group of trigger systems in a library has at least one Visio diagram, which represents the logic used in that category of trigger systems.

- **Spreadsheets:** Within each library directory is a spreadsheet with two worksheets. One worksheet contains an overview of the trigger systems in the library along with various high-level trigger and resource information, and the other identifies all of the fields for each data source used by the library.

These documents provide the information you need in order to understand and run the trigger systems against test data, including information about the expected results.

About other documentation you should use

In addition to the documentation for the trigger libraries, you may choose to refer to these documents for more detail about Detect and how to build and run trigger systems using it:

- *IBM Unica Detect User Guide*
- *IBM Unica Detect Administration Guide*

About the individual trigger system guides

Every trigger system in the library is documented, either individually or as part of a collection of similar trigger systems. The document may provide the following information, if it applies to the trigger system(s) it describes:

1. High level and detailed descriptions of the trigger systems in the trigger or category
2. Information about the feed files and fields required for the trigger system
3. Information about the trigger's outcome and the additional information included in the outcome
4. Details about any lookup tables used, such as transaction code tables and rule parameter tables, and how they affect the triggers
5. Design decisions made when creating the triggers, including algorithms and limitations
6. Modification decisions and suggestions on how to tailor the trigger systems for other uses
7. Test information including any required run instructions and high-level test objectives
8. Details about the expected results when running the test cases against the trigger systems

About the Visio diagrams that come with each library

Visio diagrams are the primary means of obtaining an overview of the trigger system logic. If the trigger systems are relatively complicated, their Visio document may include simplified high-level diagrams as well as detailed diagrams.

- **high-level Visio diagram:** conveys the essence of the detection algorithm
- **detailed Visio diagram:** shows every detail of the logic

The Visio diagrams included with the libraries are based on the diagram created when the Export Trigger System to Visio report is run on the trigger system within Detect.

✧ This document provides overview information about the diagrams. For more detail on how to read and work with the Visio diagrams, refer to the *IBM Unica Detect Administration Guide*.

About the high-level Visio diagram

The high-level Visio diagram is intended to be your primary resource for understanding the event flow and for conveying the trigger's logic to anyone who is unfamiliar with it. The high-level diagram is a simplified depiction of the trigger system's logic. It primarily shows the essential data flow, and it may reduce complexity of the event flow for the sake of clarity.

About the detailed Visio diagram

Each detailed Visio diagram originated from an Export Trigger System to Visio report that was produced using Detect. The diagram shows a node in the graph for every component in the trigger system and shows a connection in the graph for every relationship between components. The nodes are labeled with the name of the component. The report's original arrangement of event flow and data components gives visual coherence to the event flow but may not show the relevance of the data components. Therefore, many of the original reports have been rearranged by putting the data components in close proximity to the components that use them.

3 Working with a Trigger Library

- About working with a trigger library
- Obtaining the trigger library from IBM Unica
- Installing a library
- Moving trigger systems to another environment
- About the environments you may choose to use
- About suggested environments

About working with a trigger library

This section provides information you may need when obtaining and installing a trigger library.

Obtaining the trigger library from IBM Unica

The trigger libraries you purchase are available the same location as the Detect executables. The files include the XML files and documentation for the triggers as well as test feed files. Contact IBM Unica for information about how to obtain the library you require.

Installing a library

Detect libraries are distributed as Library Management export files. To install a library, use the Library Management tools to import it. The library is delivered as an indivisible package and initially you must import all the triggers in the library. Depending on how many triggers you want to deploy, you may re-employ the Library Manager in order to extract a subset of its triggers.

Special considerations for importing the Retail Banking EBM library

In the Retail Banking EBM library, each workspace is in its own file. You must import these individually, in the order shown below.

- COM
- EA1
- EA2
- EA3
- EA4
- EA5
- EA6
- EA7
- EA8
- EA9
- EAa
- EB1
- EB2
- EB3
- EB4
- EB5
- EB6
- EB7
- EB8
- EB9
- EBa
- Ebb
- ED1
- ED2
- ED3
- ED4
- EE1
- EE2
- EE3
- EE4
- EE5
- EF1
- EF2

- EF3
- EF4
- EG1
- EH1
- EH2
- EH3
- EH4
- EH5
- EH6
- EH7
- EH8
- EH9
- EI1
- EK1
- EL1
- EL2
- EM1
- EN1
- EO1
- EP1
- EQ1
- EQ2
- ER1
- ER2
- ER3
- ES1
- ET1

Mmoving trigger systems to another environment

The Library Management tools are intended for moving triggers from one Detect environment to another. The process of capturing a trigger from one environment is called exporting a trigger, and the process of bringing an exported trigger into another environment is called importing a trigger. Because an earlier version of a trigger may already exist in the target environment, the Library Manager also provides a mechanism to show the differences between the trigger-to-be-imported and the trigger that is already there. A trigger export is an XML file and can be easily moved from one environment to another environment.

For suggestions and information about the environments you can use while working with a library and trigger systems in it, refer to [“About the environments you may choose to use”](#). For more information about the Library Management tools, refer to the *IBM Unica Detect Administration Guide*.

About the environments you may choose to use

The environments in which you test, develop, and implement Detect are largely determined by your corporate processes and your knowledge of Detect. Most implementations will use separate test, development, and production environments. You may also choose to use additional environments. For example, when you choose a library with which to work, you must import all of the trigger systems that it contains into a Detect environment. As you narrow the selection of trigger systems that you wish to implement, you may create another Detect environment that will hold a subset of triggers in your selected library.

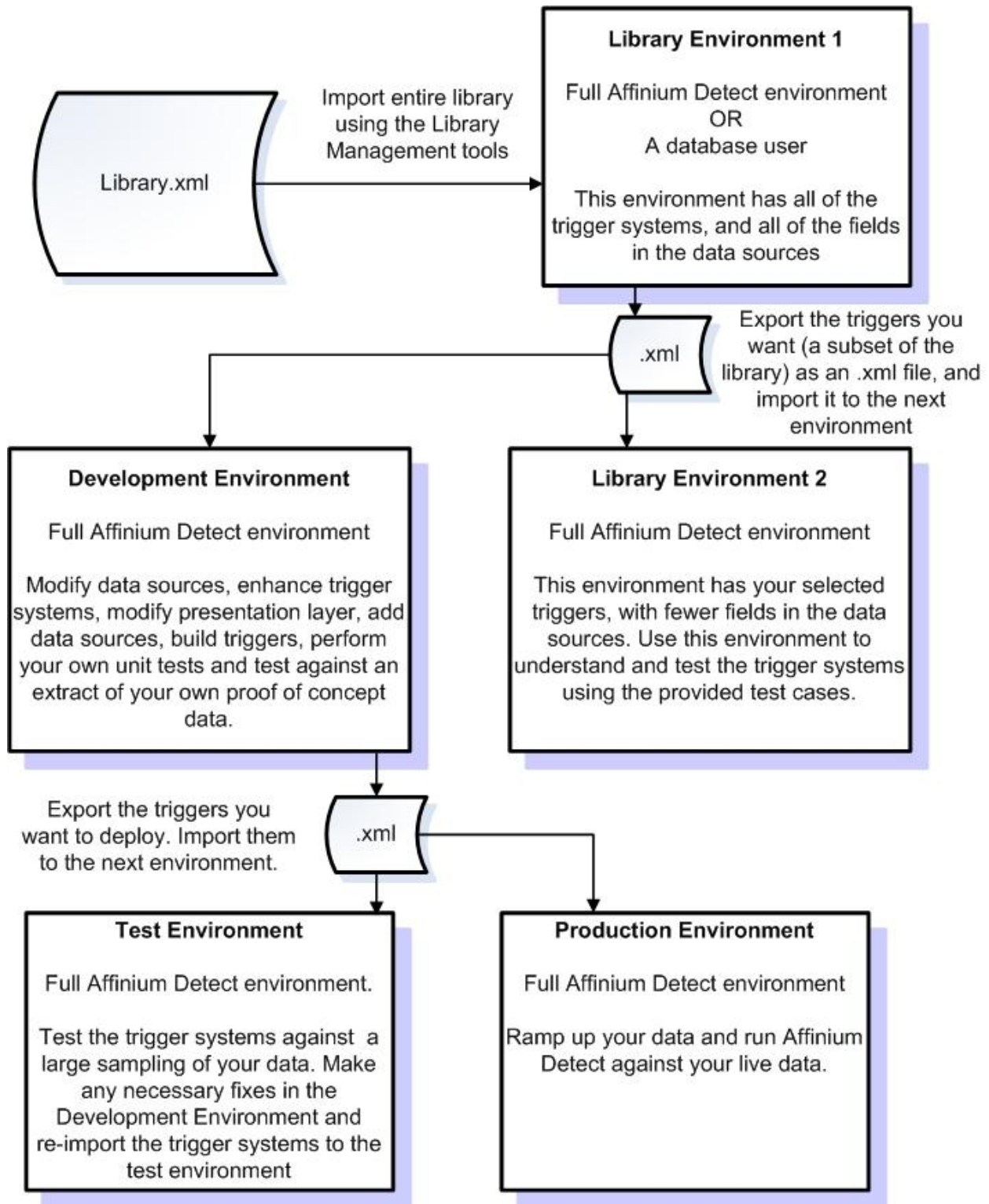
About suggested environments

This section describes one set of environments that you may choose to use, as well as the benefits and limitations of using them. This guide focuses primarily on the work you would do within the library environments.

Because a Detect library comes with its own set of data source definitions and a fixed set of trigger definitions, it is a good practice to maintain an unaltered library environment in order to view trigger systems that correspond exactly to the documentation and in order to work with the test data. Your development, test, and production environments will be the environments in which you modify trigger configurations and data source definitions and build new triggers.

- **Library Environment 1:** This environment will hold your initial import of the library. This environment could be a complete Detect environment, or it could be simply a database user. (If it is a database user, it must not be one of the Detect database users that are used to support the Detect application.) There are some limitations when this environment is only a database user, such as the inability to open Detect to examine the trigger systems, presentation layer, and data sources. If you choose to use a database user as the environment, you must know exactly which trigger systems you want.
- **Library Environment 2:** This environment should be a complete Detect environment. Use the library manager tools to export selected trigger systems from Library Environment 1 to this environment. When you import this smaller set of trigger systems, the data source definitions within Detect are limited to the ones that are associated with your selected trigger systems. Reducing the set of definitions will help you focus your attention to what you need. Use this environment to run the provided test cases, and to investigate and understand the trigger systems. Note that Library Environment 2 is optional. A second library environment provides a way for you to study your target triggers in the context of data source definitions that are limited to just those fields used by your target triggers. If you run your triggers in Library Environment 1, you will have to ignore the data source fields that are not used by your triggers.
- **Development Environment:** This environment must be a complete Detect environment. Use the library manager tools to import the .xml file that you imported to Library Environment 2 so that you begin with the same trigger systems. Use this environment to enhance the trigger systems, modify the presentation layer, add data sources, build more triggers to include in the trigger systems, and develop your own unit tests. Note: When you make modifications in this environment, the provided test cases will not work. You must develop your own unit tests.
- **Test Environment:** This environment must be a complete Detect environment. Use this environment to test the trigger systems against a large and statistically significant sampling of your data. If the trigger systems require modifications, make the changes in the Development Environment and re-import them to this test environment.
- **Production Environment:** This environment must be a complete Detect environment. Before you run with your live data, ramp up the data if necessary. The outcome data from this environment can be used in downstream systems.

Diagram of suggested environments



4 Understanding the Library Trigger Systems

- To understand a trigger system in a library
- Importing a selected trigger system
- Creating the lookup tables required for a trigger system
- To locate the documentation and for the trigger system
- To obtain a high-level understanding of a trigger system's logic
- About the Visio diagrams for trigger systems
- About the test cases for a trigger system
- To obtain a deeper understanding of the logic
- Using the Library Management tools to remove unnecessary fields
- Locating the trigger system in Detect
- Refining a trigger system
- To document trigger system logic

To understand a trigger system in a library

Before you begin, you must determine which trigger systems you want, and obtain the library from IBM Unica as described in ["About working with a trigger library"](#).

Use the following steps as a guideline to help you understand a trigger system in a library. When you are first analyzing a trigger system, it is a best practice to do so in a Detect environment outside of your normal development environment, such as one dedicated to working with a library. The tasks listed here are described in greater detail in other sections.

1. Import the trigger system you want to your library environment.
2. Create the lookup tables required for the trigger system, using the provided scripts.
3. Locate the documentation for the trigger system within the library tree.
4. Understand, at a high level, what the trigger system does by examining the high level Visio diagram and the descriptive document associated with the trigger system.
5. Make a copy of the Visio diagram, and annotate it with your observations.

6. Run the test cases for the trigger system, using the provided feed files, and examine the outcome.
7. Acquire a deeper understanding of the trigger system by examining the results and looking at the more detailed description and the detailed Visio diagram associated with the trigger system.
8. Optionally, open Detect and assess the components that form the trigger system.
9. Optionally, make modifications to the trigger system.
10. Optionally, document observations and changes you make to the trigger system.

Refer to ["Example of how to understand a trigger system"](#) for an example of how to understand a specific trigger system.

Importing a selected trigger system

Because the library with all of its trigger systems exists as a single export file, you will have to follow a procedure in order to extract a single trigger from it.


To extract a trigger from the library

Because the library with all of its trigger systems exists as a single export file, you will have to follow a procedure in order to extract a single trigger from it. Use this procedure to extract an individual trigger system from a library. This process assumes that you are using the library environments described in ["About the environments you may choose to use"](#). Use the Library Management tools (which are described in the *IBM Unica Detect Administration Guide*).

1. Import the library into a Detect environment. For clarity, this environment is referred to as Library Environment 1 (LE1). When you want to begin the process of understanding and deploying another trigger, LE1 will be your source for triggers from the library. Note that LE1 need only be a database user and need not be an entire instance of Detect including the user interface. It can be a database user because you will only be using LE1 as a source of triggers and will not need to work with the entire library in the Detect user interface.
2. Using the Library Manager, export your selected trigger. Identify your selected trigger by using its workspace code. The workspace code is listed in the library's appendix in this guide. You should configure the export process so that it limits dependencies to only those data sources used by your triggers. This method eliminates data source fields and entire data sources that are not relevant to your selected trigger.
3. Import your export into a second Detect environment. For clarity, this environment is referred to as Library Environment 2 (LE2). LE2 is the place where you learn how your trigger behaves and it must be a fully configured Detect environment. It is here that you will open the trigger system and understand its behavior and it is here that you will run the trigger system against the test cases provided in the library.

Creating the lookup tables required for a trigger system

If any of the trigger systems in the library you choose requires a lookup table, the library includes the scripts to create and populate them.

 Even if the library has scripts for lookup tables, an individual trigger system within the library may not use the tables.

To determine whether you must create lookup tables for your library

Use this procedure to determine whether the library you have chosen requires lookup tables.

1. Navigate to the Exports folder in your library tree.
2. Look for database scripts within the folder. If there are no scripts, then you do not need to create the lookup tables.
3. Refer to the Library Overview Spreadsheet associated with your library to see if your trigger uses a lookup table. If so, run the script to create and populate every lookup table that your trigger system uses.

To create the lookup tables

Use this procedure to create and populate the lookup tables required in order to test the trigger systems in the library.

1. Navigate to the Exports folder in your library tree.
2. Look for database scripts within the folder.
3. Run the script that corresponds with the database type you are using. The script creates and populates the lookup tables.

To locate the documentation and for the trigger system

Trigger systems are briefly described in the appendices of this guide. The appendices also identify which workspace holds each trigger system. Each trigger system (or category) is more fully described in documentation included in the library. Review that documentation for more detailed information to help you confirm which trigger system you want to use.

1. Locate the appendix in this guide that describes your library.

2. Use the appendix to identify the category in which the trigger system can be found.
3. Locate the trigger documentation tree for the library. Refer to the section "[About the structure of the library directory](#)" if you need more information about the structure of the trigger library directory.
4. Find the descriptive document and Visio diagram that support your trigger system.
5. Find the spreadsheets at the top of the library directory. They provide some reference information about the trigger systems in the library and about data sources.

To obtain a high-level understanding of a trigger system's logic

1. Read the high-level description for the trigger system.
2. Review the high-level Visio diagram. Understand the event flow in conjunction with the high-level description.
3. Understand the data sources that are used.
4. Make a copy of the Visio diagram and annotate it with any observations that help you to understand the logic.

About the Visio diagrams for trigger systems

The libraries contain detailed and, if required, high-level Visio diagrams for each trigger system or category of trigger systems within the library.

About component types in the Visio diagrams

The Visio reports distinguish between two types of components:

- the event-flow components
- the auxiliary components

About the event flow in the Visio diagrams

The event-flow begins with the simple components, flows through a network of patterns and timers, and terminates in an action component. On the original report, the event flow takes up the bottom half of the Visio and flows from left to right.

About the data components in the Visio diagrams

The data components consist of:

- Containers

- Qualifiers
- Select Functions
- Math Expressions

About the containers in the Visio diagrams

Containers are the primary data component and typically support data transfer from one part of the event flow to another. The other data components are used to qualify and transform the data in Containers to suit the requirements of the logic. On the original report, the data components take up the top part of the Visio and are grouped by components of the same type.

To follow the logic in the Visio diagrams

In order to understand the logic, first focus on the event flow. Approach the data flow by first understanding the usage of Containers. Use the layered properties of the diagram to reduce the number of connections shown and selectively reduce the detail of the diagram created by multiple data connections. To deepen your understanding, follow the diagrams in conjunction with the descriptions in the trigger documentation. Refer to the *IBM Unica Detect Administration Guide* for more information about the Visio diagrams.

About the test cases for a trigger system

Each trigger library contains the test data that you need in order to run and understand the trigger systems in it. When you run the test cases and generate the outcome table, you can then correlate the outcome of the trigger with the transactional content contained in the test source files. This method provides an effective approach for developing an understanding of what the trigger can detect.

Trigger systems can be run from either the Engine Manager in the user interface or from a command line interface. The command line interface is generally used during production runs, often using a scheduler program, where the running of Detect can be orchestrated with the processes that prepare the input data and with the processes that use the outcome of Detect. For testing a single trigger, the Engine Manager is typically the preferred method.

About the provided data sources required to test the trigger system

Each library contains the data and documentation that you need to test and understand the trigger system. Depending on what the trigger systems in the library require, the data may include:

- **Lookup tables:** If lookup tables are required by the trigger system in the library, they are created and populated by running a provided database script
- **Feed files:** The feed files may include profile feed files and transaction feed files

- 💡 The process of importing the library creates the corresponding data source definitions within Detect.
-

The test data files enable you to understand what a trigger is doing by examining and manipulating the data that causes the trigger to fire, or not fire. Each library contains a profile feed file that contains a collection of all of the fields needed by all of the trigger systems within it. Ultimately, when you implement Detect in your own environment, you will want to have a single definition of your own profile, one that contains all the profile fields (as opposed to transactional fields) used across all the triggers that you deploy.

The nature of the test data is described in the documentation for each trigger category.

About the structure of the test case feed files

Most test cases are packed into a single file. Separate test cases are distinguished by separate entityIDs. The entityID is structured to indicate the test number. It has the following format:

[Workspace_code][Sequence_Number]

- [Workspace_code]– indicates the workspace that holds the library trigger system.
- [Sequence_Number]– a number of the form “000x”. This is an index into the test case number.

To test the trigger system against the provided feed files

1. In Detect, select **Engine**.
2. In the **Start New Run** tab, select the trigger’s workspace. The workspace code for the trigger system is listed in the appendix as well as in the description documentation for the trigger system.
3. Enter the location of the feed directory for the test data feed files. The feed files are in a folder under in the documentation tree for your trigger system. Typically, the feed file folder is named for the workspace.
4. Select the entity data type. If you are unsure about the entity type, refer the high-level spreadsheet included in your library directory. Also, the entity type code is indicated by the second element of the feed file name. For example, in a file “unica.a.acct_profile.20080809” the ‘a’ indicates the entity type code for the account entity. Refer to the Configuration Utility to see the actual mapping of entity type codes to entity types.
5. Select the inactivity end date. If the trigger system(s) in the workspace track inactivity, then select the inactivity date. Typically the inactivity date is identified in the trigger system documentation. If the trigger system(s) in the workspace do require an inactivity date, accept the default date in the user interface.
6. Select the run parameters. Generally you should select the option to clear state history before the first run of a test set. If the trigger has an artificial transaction (ATX) component in its logic, set the **End Event** option to **End of Run Event**.

7. Click **Start Run**.
8. Look for your results in the Outcome User of the Detect database. You will find your results in the table named `[Vendor]_[WorkspaceID]_OUTCOME`.
9. Verify your results against the documented test cases in the trigger's documentation.
10. Examine the transaction feed file records in order to get a sense for how the data corresponds to the outcome.

About the outcome and the additional information

At the end of the processing the test run, triggers will fire or not. Triggers that fire are written to the outcome table. The outcome of each trigger expresses "This trigger (RULEID) fired for this customer (ENTITYID) on this date (FIRINGTIME), and here is supporting information (MESSAGE) that was accumulated during the detection process". An example of supporting information is exact details from relevant transactions or sums that were accumulated during the detection process. This supporting information is derived from the "Additional Information" section of the trigger's action component. To understand this outcome, examine an outcome record generated by a test run and open the action for that trigger. Note that the message field contains an entry for every entry in the Additional Outcome section.

The Message field of the outcome record is written in an XML format. It is written in this format so that it can flexibly hold different pieces of data for each trigger and so that it can be easily processed by a post-processing application. If you copy and paste the Message field into a blank file and give it an .xml extension, you can view its contents in Internet Explorer or other XML viewer. Alternatively, you can view the content of the Message field in an unpacked format by using the Outcome Management Tool. The Outcome Management Tool (OMT) is a Detect utility that processes the outcome message and generates a database table for each trigger in the outcome table. Within each of these trigger tables, the OMT creates and populates database fields for each of the XML fields in the Message. Although the primary purpose of the OMT is to assist in the integration of this additional information into downstream systems, it also a useful way to see the Message fields of the test cases extracted from the XML formatting. Refer to the *IBM Unica Detect Administration Guide* for more information about the OMT.

To obtain a deeper understanding of the logic

You can gain a deeper understanding of the logic used in a trigger system by following these steps.

1. Study the detailed Visio diagram to understand all the components that make up the trigger's logic.
2. Read through the Design Decisions section in the trigger's documentation to understand the trigger's approach and limitations.

3. If you have imported the trigger system, you can look at its components within Detect. Open and review key components with the intent of understanding their logic.
4. If you have run the trigger system against test data, you can examine the outcome. Navigate to the workspace that contains the trigger within the Detect user interface.

Using the Library Management tools to remove unnecessary fields

You should limit dependencies when you export a trigger system from your initial library environment to the environment in which you will examine the trigger system. This eliminates all data sources, data source fields, presentation layer lists, type descriptors, and other dependencies that are not used by the trigger system you are examining. Clearing out unnecessary information allows you to focus on the trigger system with which you are working, with less distraction.

For example, when you initially import a library into Detect, it contains the data source definitions and fields for all of the trigger systems in the library. The data source definitions that you see in the Detect user interface are therefore cluttered with far more information than you need for a single trigger system.

✧ This process allows for a natural expansion of your import environment. That is, if you follow this process with every trigger you test, your import environment will keep growing with new triggers and new required data sources and new required fields in the already existing data sources. Your import environment will always contain the data source definitions required to run all the triggers that have been imported into it.

To limit dependencies

You must have another Detect environment for this operation.

Use this procedure to limit the dependencies to include only those data source and other definitions that are needed by the trigger system being exported. The process involves using the Library Manager tools to export the workspace that contains the trigger system and then importing it into a new, clean Detect environment.

1. Export the workspace, using the Library Manager, and configure it to limit dependencies to those used by the exported trigger system. (Refer to the *IBM Unica Detect Administration Guide* for details on limiting dependencies for an exported trigger.) This process eliminates all data sources definitions and fields that are not used by your exported trigger.
2. Import the export file into a clean Detect environment. Your clean environment will now contain the definitions for the data sources and fields upon which the trigger system depends.

3. If you wish to import another trigger system, you may repeat the process. You can import to the same Detect environment used in step 2. Note that as you follow this process with every trigger you test, your import environment will keep growing with new required data sources and new required fields in the already existing data sources.

Locating the trigger system in Detect

A trigger system is built and stored within a workspace and, by convention, the trigger system is sometimes referred to by the name of the workspace. When you install a library you will see a number of new workspaces in your installation. Once imported, these new workspaces look like any other workspaces that you already had in your installation. You can find the trigger in the workspace editor by locating the workspace that contains the trigger system. (The workspace for each trigger is listed in the library appendices in this guide.)

About workspaces

Workspaces play an important role in using Detect.

- Trigger systems are built within workspaces: When you view a library trigger in Detect, you will do so by finding its workspace in the application. When you make major modifications to a trigger, you will copy its workspace and then make modifications in the copy.
- Reports are based on workspaces: For example, in order to run the “Export to Visio” report, you first locate the workspace in which that the trigger resides.
- One of the ways to run a trigger system involves running the workspace that the trigger resides in : For example, you run the test cases for a library trigger by referring to the workspace in the Engine Manager.

Using runsets to group triggers

In production, you will invariably run a set of triggers against your incoming transactions and a runset is the application’s mechanism for assembling your set of triggers. A runset provides you the convenience of building each trigger system in its own workspace and afterwards the ability of gathering them together into a runtime collection. When you run Detect on a runset, it executes all the trigger systems pointed to by the runset. Runsets also provide a mechanism to collect trigger systems together so that you can refer to them as a single group.

Refining a trigger system

You can make a number of changes to refine a trigger and the resources it depends on, as described below.

- More fields can be introduced into the profile and these fields could be used to further qualify the trigger's decision points
- Field names in data sources can be added and changed
- The thresholds in the conditions can be modified
- Thresholds can be parameterized
- Changes can be made to the logic
- Additional information can be added in the "Additional Information" section of the Action component

Read through the "Modification Suggestions" section in the trigger documentation and consider if any of the suggestions make sense for your business needs.

To document trigger system logic

As a best practice, you should complete this procedure after you complete any trigger system changes.

1. In Detect, run the **Export Trigger System to Visio** for the trigger system.
2. Save the raw report as a snapshot of the logic.
3. Optionally, modify a copy of the report to simplify it and include your notes.
4. Optionally, create a trigger system description document that describes your parameter settings and design decisions.
5. Run and save a copy of the Trigger System Detail report for the action associated with this trigger under the Reports section of the Detect user interface. The report lists the details of each of the components used in the workspace in a hierarchical listing similar to that shown in the Component Manager hierarchy view.

5 Example of How to Understand a Trigger System

- [About this example](#)
- [About the Retail Banking On-boarding library](#)
- [About the trigger system used in this example](#)
- [To locate the trigger system in this guide](#)
- [About the descriptive documentation and Visio diagrams for the trigger system](#)
- [About the component logic](#)
- [Understanding the logic in greater detail](#)
- [About the test cases for the example trigger system](#)

About this example

This section uses one trigger from a library to illustrate the steps used to understand a trigger system in a library. The example trigger system is from the Retail Banking On-boarding Library. Although the structure of the trigger is simple, it illustrates some key aspects of trigger construction.

This example skips the process of choosing and importing the library and the trigger system with which you want to work. If it were not already chosen for you, you would examine the appendixes in this guide to determine which libraries are most useful for your business, and then choose trigger systems within them. It also skips the process of running the scripts to create the lookup tables. However, it does describe the lookup tables and how the trigger system uses the information in them.

About the Retail Banking On-boarding library

Banks often go through great expense to acquire new customers by spending money on advertising and promotions, and the costs of acquiring a new account only pay off if new account holders use their account regularly. However, things could go wrong during the early period of the account's opening and the account holder could become frustrated and close down the account or choose not to make it their primary account before it becomes profitable to the bank. For example, it might be that the account holder wants to primarily use the account's debit card to withdraw money through an automatic teller machine. If through some glitch in the back office processing the debit card is never issued, then the new account holder might lose faith in the bank and soon after shut down the account or even keep the account open but continue using their debit card from their previous bank instead.

The Detect On-boarding Library is intended to assist a bank in carrying out a proper on-boarding campaign. On-boarding is a marketing practice many banks use to ensure that new customers are carefully watched over during their early days with the bank, to ensure that they have all the implements they need to use their account properly. A proper on-boarding campaign stays out of the customer's way if everything is going smoothly, proactively corrects any faults with back office procedures, and will discretely follow-up after some days when things went wrong.

About the trigger system used in this example

The trigger system used in this example was designed to notify the bank when an account is opened but no direct deposit is made within a certain number of days. In the Retail Banking On-boarding library appendix it is listed as **No Direct Deposit after an opening deposit**.

Background of the trigger system

If you analyze the behavior of a person who has an established and well-used checking account, you will notice that the customer regularly puts money into the account and regularly takes money out of it. When everything is working correctly the whole process looks perfectly mundane: the account holder deposits money into the account by making an over the counter deposit at a bank, a direct payroll deposit, or any of a variety of other ways; and then uses that money on demand by withdrawing it through writing checks, using a debit card, withdrawing from an automatic teller machine, or any of a variety of other ways.

One common behavior seen in a checking account is the automatic deposit of the paycheck of the account holder into the account. When a new account is opened, it is natural for the bank to look for a payroll check to be automatically deposited within a short period of time. Because the setup of direct deposit is done entirely without the bank's knowledge, the bank only knows that the payroll check is deposited when an ACH deposit is made to the bank. This trigger looks for a pattern of an account opening followed by a direct deposit within a set period of time. The trigger fires when a direct deposit does not occur or, in Detect terminology, it fires on an inactivity.

To locate the trigger system in this guide

Locate the **OA1, No Direct Deposit after an opening deposit** trigger system in the Retail Banking On-Boarding appendix in this guide. The appendix identifies which category holds the trigger system. It also identifies in which workspace the trigger system will be when the trigger system is imported into Detect.

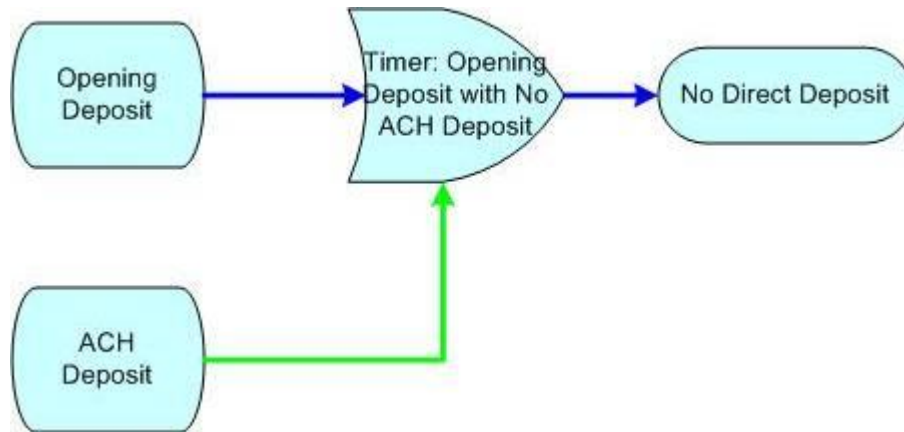
- Category : Category_A_Lack of activity after opening deposit
- Trigger name : No Direct Deposit after an opening
- Workspace: OA1

About the descriptive documentation and Visio diagrams for the trigger system

The Retail Banking On-Boarding library includes Visio diagram and a document that describes the trigger category. Portions of the Visio diagrams are included in this section. A sample copy of the document that describes the OA1 trigger system is included as an appendix to this guide. Refer to "[OA1 - Sample trigger documentation](#)".

About the high-level Visio diagram of this trigger system

The high-level Visio diagram of the trigger system is streamlined and restructured to show the essential elements of the event-flow logic.



The Visio shows:

- An Opening Deposit simple event activates a Timer
- An ACH Deposit (Direct Deposit) deactivates the Timer
- The Timer fires when an Opening Deposit is made with no follow-up ACH Deposit within a set amount of time
- When the Timer fires, it triggers “No Direct Deposit”

The following questions are suggested by the diagram:

- What constitutes an Opening Deposit and an ACH Deposit?
- What is the duration of the Timer?
- What information is attached to the triggered event?

About the component logic

This section provides details about the logic used in the components that comprise the trigger system.

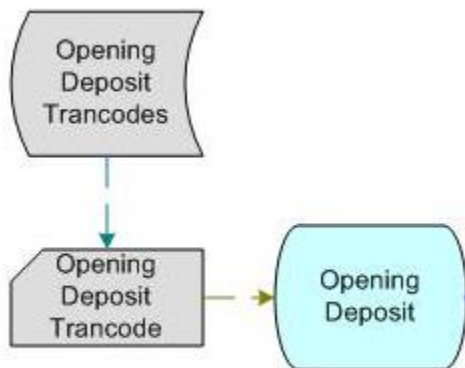
The opening deposit

An opening deposit is a transaction whose transaction code (or trancode) indicates an opening deposit. The trigger system is constructed so that the “Direct Deposit” simple component fires when it encounters a transaction with an opening deposit trancode.

The Simple component could have been constructed in such a way that it simply checked the trancode of the transaction to be equal to any of your bank’s opening deposit trancodes. But because these trancodes are different for different banks, hard-coding the comparison values in the simple component itself would mean that when your bank implements this on-boarding trigger you would have to open the simple component and change the value of the opening deposit trancode. Because that process is error prone, the Opening Deposit component was constructed to use the trancode lookup table to determine the comparison value.

The trancode lookup table is structured to hold every possible trancode value and every possible high-level activity that is important to the Retail Banking trigger libraries. Some of these high-level activities include a generic deposit, a generic withdrawal, a debit card purchase, a wire transfer, and so on. Because a generic deposit is comprised of many types of particular deposits (such as ATM deposit, over the counter deposit, and direct deposit), the generic Deposit column in the trancode lookup table has an entry of one (1) for every trancode which is a deposit. In the case of the opening deposit category, there is a one for every type of transaction that equates to an opening deposit. Note that the test cases supplied with the library all use the value of the opening deposit trancode that is provided in the test trancode table. However, when the library is deployed at your bank with your different trancode values, you will need to modify the trancode lookup table to make the trigger work against your transactions. All you need to do to make this Opening Deposit detection work for your enterprise, and every other simple detection within the banking libraries, is to recreate the trancode table to reflect the actual mappings of transcodes to general categories at your bank.

Detect uses the following cluster of components to detect an opening deposit using the trancode lookup table:



- **Opening Deposit Trancodes:** This is a select function that extracts all transcodes with a value of '1' in the Opening Deposit category
- **Opening Deposit Trancode:** This is a qualifier that tests the trancode of the transaction to the Opening Deposit transcodes. It performs the test by using an "Is Member Of" operator. This simply means, that it checks to see whether the incoming trancode is one of the transcodes collected in the select function
- **Opening Deposit:** This is a simple component that checks that the "Opening Deposit Trancode" qualifier is true whenever it sees an incoming DDA transaction

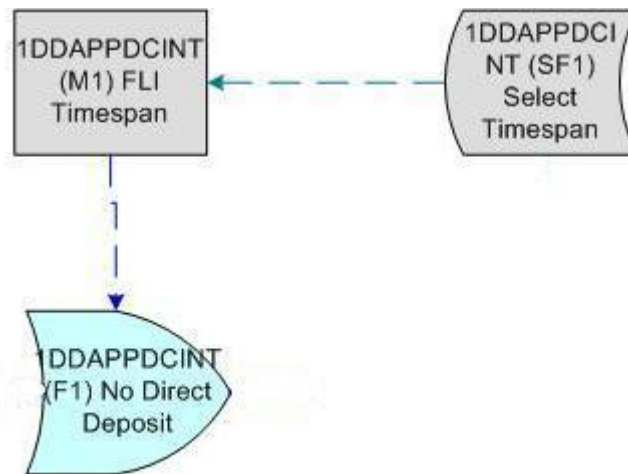
Although this cluster of logic is more complex than simply testing for each of the opening deposit transcodes explicitly as conditions in the Opening Deposit simple component itself, it allows one to change the value of the opening deposit transcodes by simply changing the values in a table.

💡 When the number of transcodes that comprise the category is large, it is impractical and error prone to list them explicitly in the simple component itself.

About the Timer

The timer in this trigger system is set using a rule parameter. Given that you may want this value to change based on business conditions or in order to control the volume of firings, it is worth setting up the timer duration as a rule parameter. By using a rule parameter, you can change the value by simply modifying an entry in a table.

The logic to set the timer duration through the rule parameter lookup table is given by the following configuration of components.



The value of the time span is kept in the RULE_PARAMETERS_EBM lookup table and indexed by the trigger name and the parameter name. It works like this:

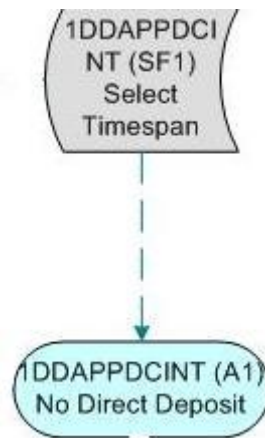
- SF1 Select Timespan is a select function that retrieves the parameter value from the rule parameters table.
- M1 FLI Timespan is a math expression (component) that simply returns the value fetched by SF1, the select function. The reason the math expression is needed is simply a limitation of the type of argument required by the time span in the FLI itself. It requires a scalar number like that coming from a Math Expression. The Math expression is capable of converting a potential list of values, which a Select function can contain, into a scalar by selecting a particular value from the list. In this case there is only one value in the list returned by the select if the Rule Parameter is set up properly without duplicates.
- F1 No Direct Deposit is the Forward Looking Inactivity (FLI) component that uses the value of the timespan parameter.

💡 Other methods to set the timer duration include hard coding the timer or setting it according to some historic usage of the user, such as the average deposit interval. Logic would have to be added to the trigger system to determine the average deposit interval.

About the information attached to the trigger system

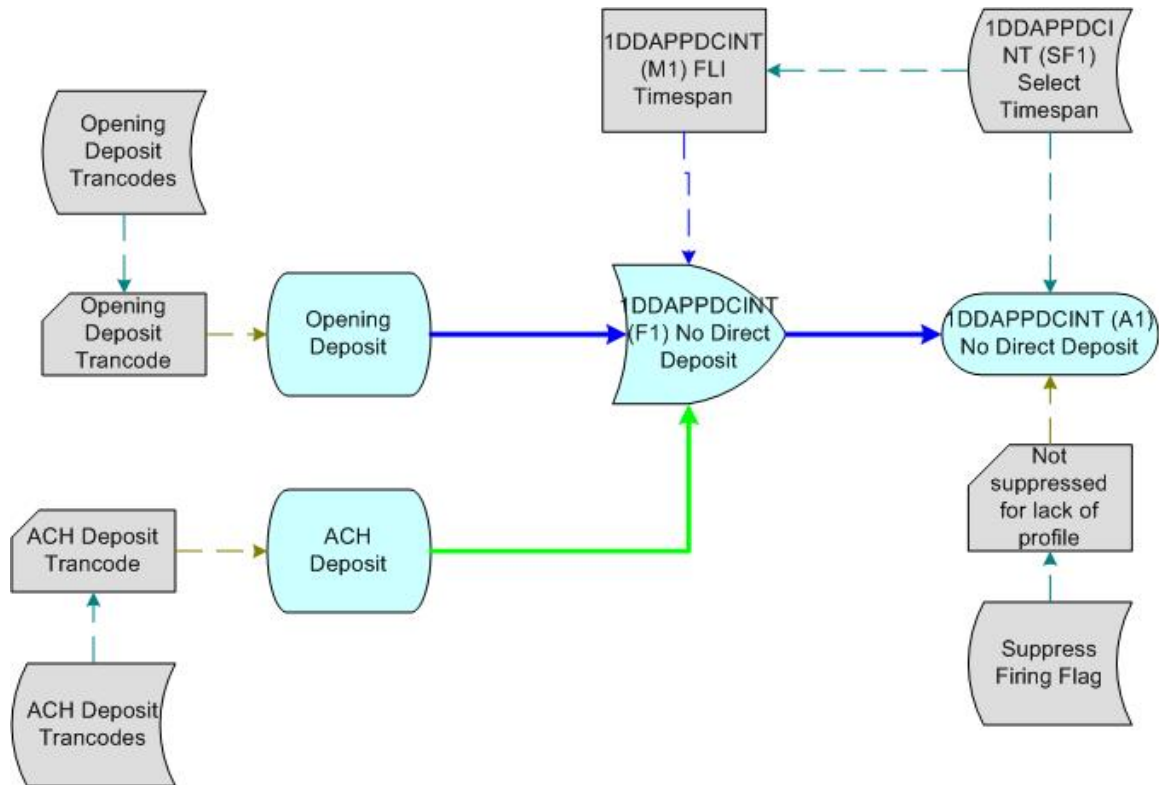
The additional information attached to the trigger should provide the downstream process or person with relevant information that will help to successfully complete the trigger's intent. In this case, the bank is expecting that a person who opens a new account will generally automatically deposit a family paycheck to the account. One piece of critical information that should accompany the outcome is the duration of the timer, especially if the timer is of variable length. Providing the timer duration enables the person in the branch or call center to have a fact-based conversation with the account holder about the missing direct deposit.

This trigger simply uses the value of the timespan used by the Forward Looking Inactivity (FLI) as the value that it attaches as additional information. The cluster of logic for achieving this is shown in the following figure.



Understanding the logic in greater detail

In general, the answers to many questions about a trigger can be found by reading the library documentation and understanding the complete Visio diagram. This image shows the detailed Visio diagram for the OA1 trigger system. For a copy of the library documentation for OA1, refer to ["OA1 - Sample trigger documentation"](#).



About the test cases for the example trigger system

As with all triggers in the library, the OA1 trigger comes with a set of test case data files as well as scripts that create and populate the lookup tables that the trigger system needs. These sample data sources enable you to run the trigger, look at the results, and correlate the results to the test cases. These test cases provide a convenient and concrete way to get an intuitive grasp of the trigger's behavior.

About the data sources for this trigger system

As indicated in the descriptive document for the trigger system, the trigger system requires the use of lookup tables for transaction codes and rule parameters. Database scripts that are provided for the library create and populate the lookup tables for testing the trigger system. Test case feed files for testing the trigger system are included as well. The example trigger system uses these data sources:

- Transaction feed file
- Profile feed file
- Lookup tables

About the location of the test case files

The test cases are located in the trigger documentation tree in a folder named by the workspace code. The typical path to the test case files is:

```
TriggerLibraries\[library name]\Documentation\[trigger category name]\[workspace]
```

For this trigger system, the path is:

```
TriggerLibraries\Retail Banking On-Boarding\Documentation\Category_A_Lack of activity after opening deposit\OA1
```

This folder contains two files that are needed to run the trigger system:

-

```
unica.a.ACCT_PROFILE.20050502000000
```

-

```
unica.a.DDA_TRANS.20050502000000
```

The file names are structured in a very specific way. The 'a' indicates a feed file of the account entity type. The name ACCT_PROFILE and DDA_TRANS are fixed by the Source Name in the data source definition. The date portion is in the format "yyyymmddhhmmss" and must be fixed to the earliest date in the date field of the transaction file. Also because multiple instances of the same feed files can be put in the same run directory, the timestamps on each of the files should be the same.

About entity types in data sources

The Retail Banking On-boarding library is built using data sources of entity type "account", which is represented by the letter a in the data source name. (Look for data sources of entity type "account" in the Data Sources section of the Detect application to see all the data sources of type account.) Detect uses the entity type to mark those data sources that can be shared in the same detection activity. This is an important concept to understand because state history does not allow sharing information for a single entity across different entity types. For example, one cannot build a trigger that detects patterns occurring across both the account and the customer entity types.

Different types of data sources of the same entity type give different types of information about the same entity. For example, in banking the Account Profile, DDA Transactions, and DDA Statement are three data sources that give various pieces of information about the account. The profile gives account summary information such as demographic information, market segment, profitability scores, and type of account. The DDA Transactions gives details about deposits into and withdrawals from the account. The DDA Statement gives summary balance information.

- 💡 The entity type is also referred to as the audience level in other IBM Unica applications.
-

About the transaction feed file

This trigger system uses a DDA (Demand Deposit Account) feed file. The DDA feed file is an extract from the banking system that records transactions that affect the account's balance. In general, every deposit and withdrawal from the account is recorded as a line in a DDA file. The test DDA file for this trigger contains ten test cases, where each entity in the file is a separate test case. The entity ID of each test case is prefixed with the name of the workspace code, and they are labeled OA1001 through OA1009.

- 💡 An actual DDA record has more fields than are listed in the DDA data source definition. The definition contains only those fields that are required by the triggers in the EBM, on-boarding, and alerts libraries.
-

The file contains these fields:

- **ID:** the entity ID, in this case the ID corresponding to the Account
- **Date:** Date of the transaction
- **Trancode:** a unique set of digits that indicates the exact nature of the transaction. For example, the trancode indicates not only that a deposit was made, but whether it was made in person at the bank (over-the-counter), through an ATM, through a wire transfer, as an ACH deposit, or any one of a variety of other means. In order to understand the mapping of trancode to bank event used in the test cases, refer to the trancode lookup table
- **Amount:** the amount of the transaction
- **Check number:** the check number if appropriate

In order to understand the test cases, refer to the test case sections of the trigger's documentation.

About the account profile feed file for the sample trigger system

This OA1 trigger system has an account profile feed file, which contains a number of fields relating to properties of the account. If you opened the ACCT_PROFILE feed you would notice that there are more fields than are needed by this trigger system. The other fields are included because the same feed file is used by all three banking libraries and it was created to contain all of the fields required to test the trigger systems in all of those libraries. The extra fields are ignored by the engine and have no impact on performance. In the case of OA1, the contents of the profile feed can be completely ignored because none of the fields is used by the trigger.

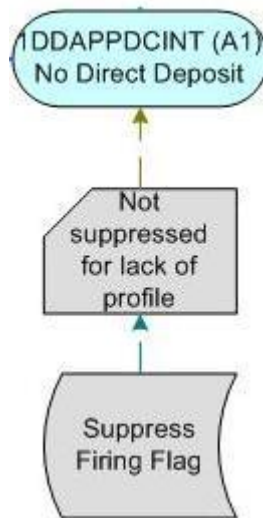
- ⚡ Even though the fields are not used, the profile file must be provided to the system, and it must contain a record for every entity ID that is included in the transaction file (DDA_TRANS). If no profile fields are used for this particular trigger, then each line can have just the EntityID field alone on it. Refer to the *IBM Unica Detect Administration Guide* for more information about profile files.
-

The additional fields appear in the data source definitions in the Detect user interface. If they interfere with your attempts to understand the trigger system, refer to ["Using the Library Management tools to remove unnecessary fields"](#).

About working with missing profile records

In general, a set of data sources for a particular entity type will either require a profile file or not require one. Whether or not the profile file is required is determined by whether a data source of type profile has been created for the entity type. The account entity type, as defined in the IBM Unica Libraries, requires a profile file, which means that every entity in the DDA_TRANS feed file must have a matching record for that entity in the profile file. However, there is a business exception to this rule that involves receiving transactions for which there is no associated profile record. Such a situation could occur when the creation of a profile record involves a back office process that is separate from that which records the opening deposit. Therefore, a bank may generate an opening deposit transaction before it is able to generate an associated profile record for that account ID. Without a way to manage this situation, this trigger would never fire because the opening deposit transaction would be ignored. (It would be ignored because Detect by default discards transactions for entities that do not have a profile record.) To address this situation, Detect can be configured to generate automatically a default profile in case the engine encounters transactions with no associated profile record. With the default profile feature enabled, Detect will process every transaction in DDA_TRANS, even those for which there is no profile record in ACCT_PROFILE. All the triggers in the on-boarding library use this configuration.

Detect also provides a way for the logic to know that a default profile record is being used. The system automatically generates the profile field named "IS_DEFAULT_PROFILE" when the default profile feature is enabled. In the case of OA1, logic is provided in the action that allows you to choose to not fire the trigger if the default profile record is still not present when the Action tries to fire. (From the banking business perspective the reason for this is that it may be acceptable to have a missing profile when the opening deposit occurs, but not acceptable 15 days later. You may need to have a profile to be able to know what number to use to contact the customer, for example.) The logic below checks the "IS_DEFAULT_PROFILE" field on the profile record and suppresses the firing of the trigger if the value in the RULE_PARAMETERS_EBM lookup table is equal to 1. Note that this parameter is indexed in the parameter table by TRIGGER_NAME = "Suppress Firing" and the PARAMETER_NAME = "Suppress Firing when using the default profile". Refer to the *IBM Unica Detect Administration Guide* for more information.



About the lookup tables

This trigger system uses a trancode lookup table and the rule parameters table for event based marketing. These tables are created and populated by running the scripts provided with the library. If you were working with this trigger system, you would run the database scripts included in the library to create and populate the lookup tables that it uses. For more information about lookup tables, refer to [“About lookup tables”](#).

About how you would run the test cases

1. In Detect, select **Engine**.
2. In the Start New Run tab, select the trigger's workspace. In this example, it would be **OA1**.
3. Select the entity data type **Account**.
4. Set the Inactivity End Date to **2005-05-19 (May 19, 2005)**. This is one day after the date of the oldest record in the transaction file. Note that the inactivity end date is identified in the trigger's documentation.
5. In the Run Parameters section, select **Clear State History**. Do not select the other options.
6. Click **Start Run**.

A Overview of the Implementation Process

- About the implementation process
- Building an operational environment
- Transitioning the trigger systems to run against your own data
- Using the results

About the implementation process

The primary purpose of this guide is to help you understand the trigger systems in the libraries. However, it also provides an overview of some of the factors you should consider when you move forward to implement the library in your business. This section provides an overview of the implementation process.

Implementing a trigger system in your business involves the following phases:

1. Assembling, testing, and understanding the trigger systems
2. Transitioning the trigger systems to run against your own data
3. Building an operational environment
4. Integrating the results with downstream processes

✨ This guide explains in detail how to use the trigger libraries to accomplish the first phase. This section provides some suggestions and topics to consider as you move toward implementing a trigger system in your business.

Building an operational environment

In parallel to the process of building and testing triggers, you must develop the operational environment within your organization. This development involves constructing a strategy for extracting transactional data, sizing and assembling a Detect run-time server configuration, and constructing a run plan that takes into account the time needed for transaction construction, running the engine, and preparing the results for downstream consumption.

Transitioning the trigger systems to run against your own data

This section provides some general guidelines to consider when transitioning a trigger system to run against your own data. Among other tasks, you should complete the tasks listed in the following sections.

Development Environment

- Using the Library Management utilities, export the triggers you will deploy from your library environment to your development environment.
- Develop feed files for testing the trigger systems. Before you can run the trigger system against them you might need to update the data source definitions within Detect with names used in your company and with additional fields. You must completely repopulate certain lookup tables, such as the tranocode table in banking, to reflect your company's codes and how they map to the high level categories. You must fine tune the values of other lookup tables, such as the RULE_PARAMETERS_EBM, to ensure that the trigger outcomes and firing volumes meet your needs. IBM Unica Consulting will help you determine the format and content of the data input files you need to provide to the system. For general descriptions of the data files and default formats, refer to the *IBM Unica Detect Administration Guide*. IBM Unica Consulting will also work with you to tailor trigger parameter settings for your environment.
- Test each trigger system and refine it as needed. Refinements might include incorporating your new profile fields (optional) into the additional information section of the action.
- Document your trigger system logic.
- Assemble a number of trigger systems and collected them into a runset. This runset is sometimes referred to as your "runtime library".
- Test the runtime library against feed files comprised of all your test cases

Test Environment

- Using the Library Management utilities, export the trigger systems from your development environment to your test environment
- Extract a statistically significant portion of real data for a select set of entities from your archives
- Test your runtime library against your archival data. First ramp up trigger state by running your trigger systems against a single set of data feeds. (Each feed file here contains transactions that cover the entire historic period. The purpose of this run is to load patterns and containers with data that readies the triggers to fire.) Then run your triggers repeatedly against data feeds that contain daily transactions. The purpose of this step is to analyze trigger outcomes and firing counts

- Validate the results and fine tune your rule parameters accordingly

Production Environment

- Develop a process for procuring the feed files to run against Detect in production.
- Ramp up state. (Ramping up state refers to the process of loading Detect's state history tables with historic information so that triggers are able to start firing on your go live date. Refer to the *IBM Unica Detect Administration Guide* for more information.)
- Prepare the outcome for downstream processing by configuring the OMT (Outcome Management Tool) which comes with Detect.
- Go live

Using the results

How you use the results of running the trigger systems depends largely on the type of trigger systems you are using. The libraries fall into two main categories: libraries for event-based marketing, and libraries for alerts.

Detect for Event Based Marketing

By its very nature, Detect outputs results as soon as they are detected. It does not hold a trigger back if multiple triggers fire within a short time span or judge that one trigger is more appropriate than another, because doing so would take away from the immediacy of the results. Many customers send the results of Detect to Campaign and Optimize in order to manage the outcome stream. This strategy allows them to effectively:

- Prioritize multiple trigger firings for the same entity
- Maintain contact history and decide whether to contact for the latest Detect trigger firing based on this information
- Direct the outcome to the appropriate channel
- Apply campaign management disciplines such as control groups
- Measure the results
- Optimize the number of customer touches

Detect for Alerts

The Alerts library for Retail Banking is intended to provide triggers for individual banking customers and not for the enterprise. The typical library usage flow is different from that of event based marketing and consists of:

- Providing a portal on a banking web site that provides banking customers a way to configure and view their alerts
- For each banking customer, gathering configuration data from the banking portal regarding the alerts that are subscribed to and the parameter settings for each alert

- Populating the profile feed file with the configuration data gathered in the previous step
- Running Detect
- Running Detect's Outcome Management Tool (OMT) on the results
- Sending alerts via email and other channels. One way to approach this notification is through Campaign and eMessage
- Posting the alerts on the banking portal as a kind of Alerts History

B About Data Sources

- [About data sources](#)
- [About transaction data files](#)
- [About the account profile file](#)
- [About lookup tables](#)
- [About data sources and fields used in libraries](#)

About data sources

There are multiple types of data sources that you can provide to the system, and each type has its own purpose.

-
- 💡 The data sources you need in order to test and understand your trigger systems are provided for you within the libraries. The information provided here is intended to help you understand the data sources.
-

- Transaction data files

The system needs data files (transaction files, balance files, customer service files, etc.) in order to recognize the desired activity patterns associated with the triggers and alerts you have chosen to offer. For details, refer to "[About transaction data files](#)".

- Account profile file

The account profile file provides information about customers that is not typically found in transaction files, but may be required in triggers and alert rules. Examples include customer profit score, products used, product packaging information, account holder's name and address, and more.

For customer alerts users, the account profile also indicates which alerts each subscriber has subscribed to and any preferences they have selected, such as alert threshold values. For details, refer to "[About the account profile file](#)".

- Lookup tables

Internal or external database lookup tables can be used as sources of data that enhance transactional information. For details, refer to "[About lookup tables](#)".

About transaction data files

Transaction data files are provided to the system so it can detect patterns of transactions. The default format is ASCII flat file.

Transaction files can originate from multiple systems and default import formats have been defined for many transactional systems. For example: transaction data files can originate from core DDA systems, loan systems, credit card systems, and so on. All must have one field in common, which is the account ID.

The precise files and field information you need to pass to the system depend on the triggers and alerts you choose to make available to your subscribers.

For details about the file format and other requirements for the transaction files, refer to the *Detect Administration Guide*.

About the account profile file

An account profile file contains account-level information and/or customer alerts subscription information needed for each of the triggers and alerts chosen by your enterprise. The account profile file provides information about the customer that is not typically found in transaction files, but is required or desirable in triggers and alert rules. Examples include customer profit score, products used, product packaging information, account holder's name and address, alert subscription information, and more.

For customer alerts, subscriber data must be provided for each online banking customer, whether they had a transaction that day or not, and must indicate whether the subscriber has elected to receive the alert.

Account profile files have similar characteristics as transaction files. They are typically in ASCII flat file format, which provides high engine throughput.


For details about the file format and other requirements for the profile files, refer to the *Detect Administration Guide*.

About lookup tables

The use of the lookup tables can simplify the development and maintenance of trigger systems, and many trigger systems in the libraries use lookup tables. Lookup tables allow trigger systems to access information in database tables. The document that describes each trigger system (or category) typically identifies the lookup table fields used by the trigger system. There are multiple types of lookup tables that may be used by trigger systems in the trigger libraries. When you receive a library, you also receive database scripts (if the library requires lookup tables) that create the tables and populate them with the data required to test the trigger systems. Some of the lookup tables provided in the libraries include:

- Transaction code (trancode) table, which maps DDA transaction codes to high-level banking activities. For example, mapping all the various trancodes that map to a generic deposit
- Merchant Category code table, which maps Merchant Category codes in credit cards transactions to categories of purchases. For example, a category of travel related purchases combines codes for air travel, train travel, rental cars, and hotels
- Rule parameter tables, which control the thresholds of the trigger logic. For example, "Amount greater than X"

Outside of the libraries, lookup tables are sometimes used to enrich the information that is attached to the Action. In an example from retail, a lookup table could be used to attach a product name to an Action by using the product code (SKU) saved in a Container.

 Each lookup table is defined within Detect as a data source, which specifies its column structure. Refer to the *IBM Unica Detect Administration Guide* for information about creating lookup tables.

About rule parameter lookup tables

Many trigger systems use rule parameter tables to manage threshold values within the trigger logic. When a trigger system reaches a decision point, instead of using a hard-coded value embedded in the logic, it may instead pull the value from a lookup table. The use of this table enables you to change and fine tune the behavior of the trigger by changing a value in the rule parameters table. It also means that you have the responsibility to choose and enter the trigger parameters into the parameter lookup table before the trigger will function as desired in your environment. There are two rule parameter tables used by the trigger libraries: one for the EBM and On-boarding libraries (RULE_PARMETERS_EBM), and one for the alerts libraries (RULE_PARMETERS). When either table is used, it is in the RuleSchema, formerly known as CEE4.

When a trigger uses a parameter, it must select on three fields to find it:

- Parameter Name
- Trigger Name
- String/Integer/Real Parameter Value

About the Merchant Category Code lookup table

A Merchant Category Code (MCC) lookup table is used by triggers in the Credit Card trigger library. On their own, merchant category codes provide a way to group each merchant's product or service into exactly one high-level category. Putting them in a lookup table provides a way to create more high-level categories, and to include the codes into multiple categories. For example, one could define a new category called

“Budget Airline Category” and include in it the merchant category codes for Southwest Airlines, AirTran Airways, jetBlue Airlines, and other low cost carriers. Within Detect, one would then use the “Budget Airline Category” to identify budget airline purchases and look for budget airline behaviors, even though each credit card transaction would only have the more granular merchant category code say of jetBlue, for example.

About how trigger systems use lookup tables

A trigger system can draw information from a lookup table by use of a Select function. The Select function specifies the lookup table that you retrieve the information from and the conditions for the retrieval. It behaves much like a SQL select statement.

About the transaction code table

The transaction code lookup table provides a way for the system to correlate the transaction codes used by the enterprise into business-level categories. For example, a trigger system may track an increase in deposits. Rather than develop the trigger system to identify each type of deposit (there can be many types of deposit), the system can rely upon a trancode table to group all types of deposits. The trigger system can refer to the category in the trancode lookup table rather than being written to look for specific codes.

The trancode table is located in the rule schema (formerly CEE4).

Example of using a trancode lookup table

Banking DDA (Demand Deposit Account) transactions, such as those that contain records of deposits, withdrawals, and transfers, typically include a trancode field that indicates the nature of the transaction. Using deposits as an example, the trancode field indicates that a customer made a deposit, and also indicates how the deposit was made. The bank may use different trancodes for a deposit made to the teller at the bank, for a deposit made through an ATM machine, for a deposit made through a direct deposit of a payroll account, and so on. These distinctions are too fine-grained for a trigger that is looking for an increase in deposits, without regard to how the deposit was made. Therefore, a trancode lookup table groups multiple specific trancodes used by the bank’s back office system into business-level category trancodes that the trigger system can use. You can update the trancode table to add or change trancodes without having to change trigger system logic.

Example of a trancode lookup table

The following picture shows an example of portion of a trancode lookup table.

	TRANCODE	TRANSDE	ACH_DEP	ONLINE_PMT	DEPOSIT	NB_ATM_W...	ATM_WITH...
►	EB03	NULL	1	NULL	NULL	NULL	NULL
	0120	ACH Deposit	1	0	1	0	0
	0122	DDA Payroll Deposit	1	0	1	0	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The full trancode table has a row for every possible DDA trancode and a column for every high-level category of activity. In this example, the high-level category is ACH_DEP has a value of “1” in the ACH_DEP column for every trancode that equates to an ACH deposit. In order to use this lookup table, the trigger system needs a Select Function, a Qualifier, and a Simple. The Select Function is defined to return all trancode values from the table whose ACH Deposit field equals 1. The Qualifier will use the “Is Member Of” function to check if the incoming trancode is one the ACH Deposit transcodes. The Simple will use this Qualifier.

In order to use this table in your own environment, you would delete its contents, put in your own transcodes, and for every category, you would put a 1 in every cell where the trancode fits the category.

 The transaction description (TRANSDE) field is optional.

About data sources and fields used in libraries

The directory for each library contains an overview spreadsheet that identifies all of the data sources used by each trigger system in the library. There is an additional master spreadsheet that lists all of the data sources used by all of the libraries, and identifies the fields in each of them. You can use that spreadsheet to determine which fields are available to each library.

Additional resources

Refer to the *Detect Administration Guide* for details on defining additional data sources whether they be at the transaction, account, customer, household, or other level.

C OA1 - Sample Trigger Documentation

■ OA1 - Sample trigger documentation

OA1 - Sample trigger documentation

This section provides a sample of the documentation provided for each trigger system or category of trigger systems. This example is included to support the exercise described in ["Example of how to understand a trigger system"](#).

1. OA1 – No Direct Deposit after an opening deposit

1.1 Trigger Information

Trigger Group: OnBoarding

Trigger Category: A

Trigger Name: No Direct Deposit after an opening deposit

Trigger Outcome String: 1DDAPPDCINT

Description: This trigger is looking for a customer who does not have any direct deposits in X days within the account opening deposit.

1.2 Detailed Description

This trigger tracks account holders who have not made a direct deposit within X days of the opening deposit. It works in the following way:

- An opening deposit activates a forward looking inactivity (timer). The time span of the timer is set to a value taken from the rule parameters table.
- An ACH Deposit (Direct Deposit) will deactivate the timer.
- If X days pass without an ACH Deposit, the trigger will fire.
- The trigger assumes that a profile feed is used. The firing of the trigger may be suppressed if a profile is not available at the moment the timer fires.

1.3 Feed file Input

The following feed is used by the main logic of the trigger system:

DDA Transactions

- Account Number
- Transaction Code

1.4 Trigger Outcome and Additional Information

Trigger Outcome: 1DDAPPDCINT

Additional Information:

- Days of Subsequent Inactivity for Firing – the time span of the FLI. Indicates the number of days to wait for a Direct Deposit to occur from the time of the opening deposit. The value is taken from the RULE_PARMETERS_EBM lookup table.

1.5 Lookup Table Usage

RULE_PARMETERS_EBM

Parameter Name	Value Type	Description	Default Value	Unit
Days of Subsequent Inactivity for Firing	Integer	Controls the time span of the FLI. Indicates the number of days to wait for a Direct Deposit to occur from the time of the opening deposit. It is the value of X in the description.	15	day
Suppress Firing when using Default Profile	Integer	1 = True and 0 = False. The trigger is intended to work with a profile file but assumes that an opening deposit may have been made before the profile for the account was created. This flag configures the behavior to not fire if the profile is also not yet created X days after the account opening. The reason for this is that the missing profile may be accounted for in a different trigger and the lack of direct deposit may be related to the missing profile and therefore of secondary importance.	0 (False)	Boolean

Parameter Name	Value Type	Description	Default Value	Unit
Suppress Firing when using Default Profile	Integer	1 = True and 0 = False. The trigger is intended to work with a profile file but assumes that an opening deposit may have been made before the profile for the account was created. This flag configures the behavior to not fire if the profile is also not yet created X days after the account opening. The reason for this is that the missing profile may be accounted for in a different trigger and the lack of direct deposit may be related to the missing profile and therefore of secondary importance.	0 (False)	Boolean

TRANCODES Lookup table

Parameter Name	Description	Comment
Transaction Code	Code for each transaction	For all workspaces in this category
OPEN_DEP	The opening deposit	Assumes that a deposit is marked with a special trancode to indicate an opening deposit. If the opening deposit is simply the first deposit, then a different strategy has to be used
ACH_DEP	ACH Deposit	An ACH Deposit indicates a direct deposit

1.6 Design Decisions

- The trigger assumes that the opening deposit is indicated by a special trancode.
- The trigger assumes that the period to wait for the direct deposit can be a fixed period of time for all accounts.

1.7 Modification Decisions

- If the opening deposit is indicated by something other than special trancode, then the trigger has to be modified accordingly. For example, if the opening deposit is indicated by the first check paid, then the Opening Deposit simple component should be changed to look for the first check paid. In this case a trigger control should be used to ensure that the trigger fires only once.
- Sophistication could be added to the timespan for the FLI. One suggestion would be to make the timespace vary depending upon some attribute of the account. This could be done by introducing a "Type of Account" lookup table and introducing different timespans for different account types. (In a simple sense, type of account values might be Gold, Silver, and Bronze.)

1.8 Test Information

Workspace: OA1

Feed Directory: ..\Retail Banking On-Boarding\Documentation\ Category_A_Lack of activity after opening deposit\OA1

1.9 Test Objective


- Validate that the trigger fires for users who do not have any direct deposit within X days of the account opening deposit.
- Validate that the trigger does not fire for users who have direct deposits posted within X days of the account opening deposit.
- Validate the correct rule parameter is retrieved by the engine to calculate the number of days (X) since the account opening deposit.
- Validate the correct information is in the output message. The text output is the trigger name and the additional information is the value of X days retrieved from the database.

1.10 Expected Results

These are the test cases for each user in the transaction feed:

- OA1001: Fires - no direct deposit after opening deposit
- OA1002: Fires - direct deposit received after 15 days
- OA1003: Does not fire - direct deposit received on 15th day
- OA1004: Does not fire - direct deposit received on 5th day
- OA1005: Does not fire - no direct deposit after opening deposit, but lacks a real profile, so the action is suppressed.

The engine will output messages for the following list of rules and user ids

 The rule ID shown below may differ from your rule ID, as rule IDs are generated at installation time. However the unique ID for this rule will always be the same.

RULEID	ENTITYID
22612574	OA1001
22612574	OA1002

D Credit Card Event Based Marketing (CCEBM) Library

- [About the Credit Card Event Based Marketing \(CCEBM\) library](#)
- [List of Credit Card Event Based Marketing \(EBM\) trigger systems](#)

About the Credit Card Event Based Marketing (CCEBM) library

This section lists the trigger systems in the Credit Card Event Based Marketing (CCEBM) library. This library contains trigger systems that are used by credit card providers to watch accounts and detect behaviors that are indicative of opportunities to cross-sell, retain, or nurture the customer.

List of Credit Card Event Based Marketing (EBM) trigger systems

Trigger Systems	Description	Workspace
3 Multi Payment Months	Detects when a customer makes multiple payments within the same month three times over a three month period.	C09
Cash Advance then 3 Grocery Spending	This trigger looks for a customer who makes a cash advance followed by 3 grocery purchases within the next 2 weeks.	C13
Credit Utilization above 80%	Card balance has reached 80% of available total credit.	C03
Drop in Card Spending	Detects when a customer has a drop in average spending of 60% over the last four weeks compared with the previous four weeks.	C08

Trigger Systems	Description	Workspace
Jump in Grocery	Tracks grocery purchases and detects when the sum of purchases in the last month is greater than \$600, while the maximum monthly sum of purchases in prior months was less than \$125.	C06
Lack of activity for a new account	Detects varying lack of activity events for accounts opened or activated recently.	C16
Large HMI and Travel Spending	This trigger is looking for a customer who had home improvement and travel expenses that occurred close together (in time).	C12
Multiple Accounts with BT out then Balance Drop	This trigger is looking for a customer who has multiple accounts, one account having balance transfer out then balance drop and, within 2 weeks, the second account does the same thing, that is, a balance transfer out then a balance drop.	C15
No Cash Advance after Account Activation	Detects when a customer activates an account without doing a cash advance.	C05
Payment above 50%	This trigger looks for a customer who makes a payment which is equal or greater than \$5000 and more than 50% of last statement balance amount.	C04
Revolving account makes Large Payment	Detects accounts that frequently pay off only small percentages of their balance, and then suddenly pay off most of their balance.	C10
Unusually High Weekly Card Volume	Detects an unusually high weekly card transaction volume, as compared to the volume collected from prior weeks.	C01
Unusually Large Balance Transfer or ATM Withdrawal	Detects an Unusually Large Balance Transfer or ATM Withdrawal as compared to a historic maximum.	C07
Unusually Large Jump in travel expenses	Tracks travel purchases and detects when the sum of purchases in the last month is greater than \$2000, while the maximum monthly sum of purchases in prior months was less than \$100.	C11
Unusually Large Travel Spend	Unusually large travel spending given the cardholder's "normal" bi-weekly travel spending habits (also known as a "Jump in Travel Spending").	C02

E Retail Banking Alerts Library

- [Retail banking alerts library](#)
- [List of triggers in the Retail Banking Alerts library](#)

Retail banking alerts library

This section lists the alert trigger systems in the Retail Banking Alerts (RBA) library. This library contains trigger systems that provide a watchful eye on the account on behalf of the retail banking customer. Through a bank-provided portal, banking customers can both subscribe to alerts and specify trigger thresholds and sensitivity. For the customers, the outcome of an alert could be an email or SMS message that notifies them of an account behavior that they are concerned about. The Alerts trigger library uses subscriber settings and detects account behaviors. The mechanism for gathering subscriber settings through a portal and the sending of emails is done outside of Detect.

For information about how to implement alerts, refer to ["Additional information about using the Alerts Library for Retail Banking"](#).

List of triggers in the Retail Banking Alerts library

Categories and Trigger Systems	Description	Workspace
Category_A_Balance and specific transaction monitoring	The trigger systems in this category detect behavior in deposit accounts.	
Deposit account: Balance above Threshold	Detects when an account's available balance is greater than a personalized threshold amount.	AA0
Deposit account: Balance below Threshold	Detects when an accounts available balance is less than a personalized threshold amount.	AA0

Categories and Trigger Systems	Description	Workspace
Deposit account: Specific Check Amount Paid	Detects when a check of a personalized amount is paid.	AA0
Deposit account: Specific Check Number Paid	Detects when a check of a personalized check number is paid.	AA0
Deposit account: Deposit Greater than specific amount posted.	Detects when a deposit greater than an account specific amount is posted.	AA0
Deposit account: Transaction Greater than specific amount	Detects when any DDA transaction greater than an account specific amount is posted.	AA0
Category_B_Deposit account transaction occurred	This category is composed of 12 similar triggers which are designed to detect a specific type of the transaction defined in the Trancode table and will fire only if the user is currently subscribed to alerts.	
Payroll Deposit Received	Detects a payroll deposit transaction.	AB0
Electronic Payment	Detects an electronic payment.	AB0
Non-Home Bank ATM Withdrawal	Detects a withdrawal from a non-home bank ATM.	AB0
Foreign Currency ATM Withdrawal	Detects a withdrawal of foreign currency from an ATM.	AB0
ATM Withdrawal	Detects an ATM withdraw.	AB0
Acct to Acct Transfer	Detects account to account transfers.	AB0
Check Cashed by Teller	Detects a check cashed by a teller.	AB0
Interest Posted to Account	Detects interest posted to the account.	AB0
Online Banking Payment	Detects an online banking payment.	AB0
Debit Card Purchase	Detects a debit card purchase.	AB0
Wire Transfer Deposit	Detects a wire transfer deposit.	AB0
Wire Transfer Payment	Detects a wire transfer payment.	AB0

Categories and Trigger Systems	Description	Workspace
Category_C_Approaching balance or overdraft thresholds		
Balance near required minimum	Detects when an account balance is within a marginal amount of that accounts minimum balance requirement	AC0
Overdraft available line balance below threshold	Detects when an account's overdraft available line balance drops below an account specified threshold.	AC0
Overdraft line drawdown above threshold	Detects when an account's overdraft line drawdown is above an account specified threshold.	AC0
Category_D_Overdraft occurred		
Overdraft Occurred	Detects when an overdraft occurs for an account.	AD0
Category_E_Budget Alerts		
xxx Transaction Volume Hits Threshold in Period of Time	There are eight triggers in AE0 workspace that all detect an account in which the volume for a certain type of transaction hits X threshold within Y period of time. The X and Y are customer personalized values obtained from customer profile. There is one trigger system for each of eight transaction types.	
Electronic Payment	This trigger monitors transaction volume the ACH Electronic Payment transaction type.	AE0
ATM Withdrawal	This trigger monitors the ATM Withdrawal transaction type.	AE0
Check Paid	This trigger monitors the Check Paid transaction type.	AE0
Debit Transaction	This trigger monitors the DBT Debit transaction type.	AE0
Deposit Posted	This trigger monitors the deposit posted transaction type.	AE0
Check cashed by Teller	This trigger monitors the OTC Check Cashed by Teller transaction type	AE0
Debit Card Purchase	This trigger monitors the POS Debit Card Purchase transaction type.	AE0
In Branch Withdrawal	This trigger monitors the WDL In Branch Withdrawal transaction type.	AE0

Categories and Trigger Systems	Description	Workspace
xxx Transaction Amount Hits Threshold in Period of Time	There are eight triggers in AE0 workspace that all detect an account in which the transaction amount for a certain type of transaction hits X threshold in Y period of time. The X and Y are customer personalized values obtained from customer profile. There is one trigger system for each of eight transaction types.	
Electronic Payment	This trigger monitors transaction volume the ACH Electronic Payment transaction type.	AE0
ATM Withdrawal	This trigger monitors the ATM Withdrawal transaction type.	AE0
Check Paid	This trigger monitors the Check Paid transaction type.	AE0
Debit Transaction	This trigger monitors the Debit transaction type.	AE0
Deposit Posted	This trigger monitors the Deposit Posted transaction type.	AE0
Check cashed by Teller	This trigger monitors the Check Cashed by Teller transaction type.	AE0
Debit Card Purchase	This trigger monitors the Debit Card Purchase transaction type.	AE0
In Branch Withdrawal	This trigger monitors the In Branch Withdrawal transaction type.	AE0
Category_G_Unusual Account Activity	This group of triggers detects unusual behavior within a particular activity. Each trigger is divided into four branches, each with detecting unusual behavior in different weeks of a month. Each of these weekly branches is further divided into 5 different cases each concerned with a different type of unusual behavior. In effect, each individual trigger does the work of 20 different triggers.	
Unusual Checking Behavior	This trigger detects unusual checking behavior.	AG1
Unusual In-Branch Withdrawal Behavior	This trigger detects unusual in-branch withdrawal behavior.	AG2
Unusual Check Cashed Behavior	This trigger detects unusual check cashed behavior.	AG3
Unusual ATM Withdrawal Behavior	This trigger detects unusual ATM withdrawal behavior.	AG4

Categories and Trigger Systems	Description	Workspace
Unusual Electronic Payment	This trigger detects unusual electronic payment behavior.	AG5
Category_H_Scheduled Payment	The triggers in the AH1, AH2, and AH3 workspaces look for a scheduled payment that was missed and not posted. The three triggers are the same. A user would pick more than one to track more than one scheduled payment. The triggers in AH4, AH5, and AH6 are identical to the AH1-AH3 triggers except that they track deposits and not payments. They look for a scheduled deposit that was not posted to the account.	
Scheduled Payment not posted		AH1
Scheduled Payment not posted		AH2
Scheduled Payment not posted		AH3
Scheduled Deposit not posted		AH4
Scheduled Deposit not posted		AH5
Scheduled Deposit not posted		AH6

F Additional Information About Using the Alerts Library for Retail Banking

- [About alerts and outcome values](#)
- [Subscription information in the account profile file](#)
- [About transaction data files for alerts](#)
- [Outcome message XML](#)
- [Identifying and using transaction variables](#)
- [Deactivating Subscribers](#)
- [Alerts Transaction Code Categories](#)
- [List of alerts and alert-specific variables](#)

About alerts and outcome values

Triggers fired from the Alerts library are intended to be used as the content of alert messages to retail banking customers. Most alerts have associated variables that you can use to generate custom emails for your subscribers. For example, an alert might have a variable to track the sum of deposits in the past week for the subscriber. The values for the alert-specific transaction variables never show up in your emails unless you place them there.

As alert triggers fire, the system inserts rows into the outcome table. Every trigger contains an outcome string that is labeled “TEXT” in the outcome MESSAGE field. The outcome string for every alert trigger is a 9 character code that is intended to indicate the type of trigger that fired. The MESSAGE can also contain fields that carry forth the values of specific fields from the profile or transaction feeds. These fields are referred to as transaction variables and contain information about the customer’s preferences as to how his or her alerts behave. For example, these preferences will indicate things such as sensitivity of the Unusual Activity alerts (high, medium, low) or amount thresholds for

the budget alerts. When you build the alert message, you may want to echo back these values to the alert recipient so that the recipient has a context for why he or she is getting the alert. (The outcome message is defined when the alert is created, and fields can be selected for inclusion in the message by using the Outcome tab on the action component editor.)

This section lists the transaction variables for each alert, describes how to use them, and provides other information you need to use the alerts.

For more complete details about the Outcome table, including its location, how to create a new one, and a description of its contents, refer to the *Detect Administration Guide*.

Subscription information in the account profile file

For customer alerts users, the account profile file at a minimum indicates which alerts each customer subscriber has chosen and passes in any alert preference values the subscribers have chosen. The account profile can also include information such as subscriber name, do not call info, and more.

Subscriber data must be provided for each online banking customer, whether they had a transaction that day or not, and must indicate whether the subscriber has elected to receive the alert.

Example of a subscription file format

Here is an example showing subscription and threshold values:

```
Header record  ID|DEPBALLST_SUB|DEPBALLST_AMT|DEPACHPMT_SUB
               |DEPCHKDIF_SUB|DEPCHKDIF_SENS
Data line     0001|1|200.00|1|1|L
records       0002|0|0|1|0|_
               0003|1|1000.00|0|1|H
```

✧ IBM Unica Consulting will work with you to determine the exact format to use in your account profile file.

In the example, ID is the alphanumeric account number, zero-padded on the left so that all account numbers are the same length.

The parameters that end in _SUB indicate whether the subscriber has subscribed to that alert. Those parameters are optional depending on whether they are set to “1” or not. If they are set to “0”, then these parameters can be omitted. For null values in a numeric field, use a zero (0). For null values in string parameters, include an underscore (_).

So, using this example:

DEPBALLST_SUB is “1” if the customer owning this account has chosen to subscribe this account to the DEPBALLST “Available balance is less than \$[input amount]” alert and “0” if they have not.

DEPBALLST_AMT is the DEPBALLST preference amount in dollars input by the customer for this alert, given as a floating point number with two decimals. Detect will generate an event each day the customer’s balance is below this threshold value.

DEPACHPMT_SUB is “1” if the customer owning this account has chosen to subscribe this account to the DEPACHPMT “Electronic Payment” alert and “0” if they have not. Detect will generate an event each day an ACH Debit transaction posts to the customer’s account.

DEPCHKDIF_SUB is “1” if the customer owning this account has chosen to subscribe this account to the DEPCHKDIF “Unusual Account Activity - Check Paid” alert and “0” if they have not.

DEPCHKDIF_SENS is the DEPCHKDIF unusual activity sensitivity parameter. It determines how “unusual” the check paid pattern has to be to be flagged for notification to the user via this alert. Its values must be set to “H”, “M” or “L”, corresponding to high, medium or low sensitivity. (For example, “H” will fire more readily than “M”, and so on.).

About transaction data files for alerts

Transaction files must contain data for all online banking customers, not just those who have subscribed to Alerts. For alerts that have a time component, such as budget alerts and unusual activity alerts, Detect needs to already have information about the customer before the alert can fire.

By having data for all online banking customers, the system can build historic information over time and be ready to fire relevant triggers as soon as the customer subscribes.

Outcome message XML

The XML in the Outcome Message column of the Outcome table can consist of data from a number of different Detect sources. It can contain data from the feed files and results from the triggers. The XML can be quite complex.

You can extract fields from the outcome messages using the transaction variables and use them in various ways. For example, Alerts users may want to include certain variables in email messages; IBM Unica Leads users may use certain information on Lead summary and detail screens.

Identifying and using transaction variables


You can extract fields from the outcome message using transaction variables, which are specific to the individual alert.

To determine what fields are available for the alerts you selected, refer to the lists of alerts in this section. The fields can also be found in the outcome message, as described in the trigger description documentation provided with each category of alerts. There are at most a few variables available for each alert. You only need to consider the variables related to the triggers you have chosen to implement.

XML for transaction variables

If there is a transaction variable in for your alert's outcome message that you would like to use, you can use an XML string in the following format:

```
<OUTPUT>
  <TEXT>Alert Name</TEXT>
  <DATASOURCE name="Source File" field="Field Name">actual value
</DATASOURCE>
</OUTPUT>
```

 Include a datasource line for each variable you want to include. In this example, Actual value will display the actual value in the output.

For Example:

```
<OUTPUT>
  <TEXT>DEPBALMIN</TEXT>
  <DATASOURCE name="ACCT_PROFILE"
    field="DEPBALMIN_REQ">1500</DATASOURCE>
  <DATASOURCE name="ACCT_PROFILE"
    field="DEPBALMIN_AMT">250</DATASOURCE>
  <DATASOURCE name="DDA_BAL" field="BALANCE">1700</DATASOURCE>
</OUTPUT>
```

Another example shows a message with the contents of a field (Amount_Real) of a container (in this case the container is called "DEPDEPAMT C4 Amount Over Time") are output. A container is a storage structure used during trigger detection. It may contain useful information, such as the sum of the transactions over the last month. The contents of selective containers are sometimes included in the outcome message. (The *Detect User Guide* contains details about containers.)

In this case it is selecting the LAST amount in the container:

```
<OUTPUT>
  <TEXT>DEPDEPAMT</TEXT>
  <DATASOURCE name="ACCT_PROFILE"
    field="DEPDEPAMT_AMT">3</DATASOURCE>
  <DATASOURCE name="ACCT_PROFILE"
    field="DEPDEPAMT_PERIOD">month</DATASOURCE>
  <DATASOURCE name="DEPDEPAMT C4 Amount Over Time"
    field="LAST(Amount_Real)">138128.29 </DATASOURCE>
```

```
<DATASOURCE name="DEPDEPAMT C4 Amount Over Time"
  field="LAST (Count_Real) ">79</DATASOURCE>
</OUTPUT>
```

Deactivating Subscribers

When subscribers close their accounts, elect out of on-line banking, or you choose to discontinue alerts for them, you may eventually remove them from the account profile file.

If you have no inactivity alerts in production (such as Scheduled Payments or Scheduled Deposits) then you can immediately stop including them in the daily account profile file.

If you use any inactivity alerts, you must continue to feed account profile line for the customers for 60 days with the account status attribute changed from active to inactive. This must be done to avoid having Detect Alerts recognize and misinterpret the inactivity for some tracked event for this account.

Alerts Transaction Code Categories

The following table describes the Transaction Code Categories.

Alerts Transaction Code Category Descriptions

Transaction Code Category	Description and Comments
Account to Account Transfer	Book Transfer/Account to Account Transfer debits
ATM Withdrawal	
Check Paid	Checks paid transaction codes AND Check conversion/ACH codes in which the original instrument used by the customer was a check (such as ARC).
Check Cashed By Teller	
Debit Card Purchase	
Deposit Posted	
Electronic Payment	On-line banking withdrawal tran codes and ACH debit tran codes. Exclude check conversion/ACH trans codes like ARC.

Transaction Code Category	Description and Comments
Foreign Currency ATM Withdrawal	If supported by the bank, foreign currency withdrawal tran codes
Home Bank ATM Withdrawal	“On-us” ATM withdrawals
In Branch Withdrawal	
Interest Posted to Account	
Non Home Bank ATM Withdrawal	“Off-us” ATM withdrawals
On-Line Banking Payment	
Overdraft LOC Advance	
Payroll Deposit Received	Note that many banks have only ACH credit tran codes, not distinct codes for payroll deposit.
Scheduled Deposit	Scheduled deposit alerts notify subscribers when a regularly occurring monthly deposit amount they specify is not credited to their account by the day of the month they specify.
Scheduled Payment	Scheduled payment alerts notify subscribers when the regularly occurring debit amount they specify is not posted to their account by the day of the month they specify.
Wire Transfer Deposit	Incoming wire transfer, excluding non-value messages
Wire Transfer Payment	Outgoing wire transfer, excluding non-value messages

List of alerts and alert-specific variables

The tables in this section include a sampling of alerts available through Detect, and the alert-specific (transaction) variables available to each alert. IBM Unica Consulting can provide you with a complete list of alerts.

Deposit Account Alerts

These alerts notify subscribers when certain balance or transaction conditions they specify are met. These alerts generally fall into the following categories:

- Balance threshold alerts – sent when the balance falls above and/or below the amount the subscriber specified.
- Specific transaction posted – sent when a transaction for the amount the subscriber specified is posted or when a check number they specify posts.
- Transaction occurred – sent when the alert transaction type the subscriber selected occurs.

AA0 DEPBALLST

- Available balance is less than \$[input amount]
- Transaction Code Category: Not applicable

Message Variable Description	Field Name	Source File
Customer Input Low Balance Threshold	DEPBALLST_AMT	ACCT_PROFILE
Available Balance	BALANCE	DDA_BAL

AA0 DEPBALGRT

- Available balance greater than \$[input amount]
- Transaction Code Category: Not applicable

Message Variable Description	Field Name	Source File
Customer Input High Balance Threshold	DEPBALGRT_AMT	ACCT_PROFILE
Available Balance	BALANCE	DDA_BAL

AB0 DEPOLBPMT

- On-line banking payment made
- Transaction Code Category: On-Line Banking Payment

Message Variable Description	Field Name	Source File
Customer Input Online Banking Payment	AMT	DDA_TRANS

AA0 DEPCHKNUM

- Check Number [Input check number] Paid
- Transaction Code Category: Check Paid

Message Variable Description	Field Name	Source File
Customer Input Specific Check Number	CHECK_NO	DDA_TRANS
Check Amount	AMT	DDA_TRANS

AA0 DEPCHKEQU

- Check Amount \$[Input Amount] Paid
- Transaction Code Category: Check Paid

Message Variable Description	Field Name	Source File
Customer Input Specific Check Amount	AMT	DDA_TRANS
Check Number	CHECK_NO	DDA_TRANS
Check Amount	DEPCHKEQU_AMT	ACCT_PROFILE

AA0 DEPDEPGRT

- Deposit greater than \$[Input Amount] Posted
- Transaction Code Category: Deposit Posted

Message Variable Description	Field Name	Source File
Customer Input Deposit Threshold	DEPDEPGRT_AMT	ACCT_PROFILE
Deposit Amount	AMT	DDA_TRANS

AA0 DEPTRNGRT

- Transaction greater than \$[Input Amount] Posted
- Transaction Code Category: Not applicable

Message Variable Description	Field Name	Source File
Customer Input Transaction Threshold	DEPTRNGRT_AMT	ACCT_PROFILE

Message Variable Description	Field Name	Source File
Transaction Amount	AMT	DDA_TRANS

AB0 DEPATMFOR

- Non-{Bank name} ATM withdrawal
- Transaction Code Category: Non Home Bank ATM Withdrawal

Message Variable Description	Field Name	Source File
Non Home-Bank ATM Withdrawal Amount	AMT	DDA_TRANS

AB0 DEPATMWDL

- ATM withdrawal
- Transaction Code Category: ATM Withdrawal

Message Variable Description	Field Name	Source File
ATM Withdrawal Amount	AMT	DDA_TRANS

AB0 DEPACHPMT

- Electronic Payment Made
- Transaction Code Category: Electronic Payment

Message Variable Description	Field Name	Source File
Electronic Payment Amount	AMT	DDA_TRANS

AB0 DEPCHKOTC

- Check Cashed by Teller
- Transaction Code Category: Check Cashed By Teller

Message Variable Description	Field Name	Source File
Cashed Check Amount	AMT	DDA_TRANS
Cashed Check Number	CHECK_NO	DDA_TRANS

AB0 DEPATMFXW

- Foreign currency ATM withdrawal

- Transaction Code Category: Foreign Currency ATM Withdrawal

Message Variable Description	Field Name	Source File
FX ATM Withdrawal Amount	AMT	DDA_TRANS

AB0 DEPACHPAY

- Payroll deposit received
- Transaction Code Category: Payroll Deposit Received

Message Variable Description	Field Name	Source File
Payroll Direct Deposit	AMT	DDA_TRANS

AB0 DEPWIRPMT

- Wire transfer payment made
- Transaction Code Category: Wire transfer payment

Message Variable Description	Field Name	Source File
Outgoing Wire Amount	AMT	DDA_TRANS

AB0 DEPWIRCRD

- Wire transfer deposit
- Transaction Code Category: Wire transfer deposit

Message Variable Description	Field Name	Source File
Incoming Wire Amount	AMT	DDA_TRANS

AB0 DEPPOSPMT

- Debit card purchase
- Transaction Code Category: Debit card purchase

Message Variable Description	Field Name	Source File
Debit Card Purchase Amount	AMT	DDA_TRANS

AB0 DEPINTDEP

- Interest Posted to Account
- Transaction Code Category: Interest Posted to Account

Message Variable Description	Field Name	Source File
Interest Posted Amount	AMT	DDA_TRANS

AB0 DEPBKXFER

- Account to account transfer made
- Transaction Code Category: account to account transfer

Message Variable Description	Field Name	Source File
Account to account transfer amount	AMT	DDA_TRANS

Approaching Balance or Overdraft Threshold Alerts

These alerts notify subscribers when an account is approachign a balance or overdraft threshold.

AC0 ODLBALLST

- Overdraft Line balance is less than \$[Input amount]
- Transaction Code Category: Not applicable for balance alerts

Message Variable Description	Field Name	Source File
Customer Input Overdraft Line Balance Threshold	ODLBALLST_AMT	ACCT_PROFILE
Overdraft Line Balance	AVAILABLE_LINE	OD_BAL

AC0 OLDBALGRT

- Overdraft Line drawdown is greater than \$[input amount]
- Transaction Code Category: Overdraft LOC Advance

Message Variable Description	Field Name	Source File
Customer Input Overdraft Line Drawdown Amount Threshold	OLDBALGRT_AMT	ACCT_PROFILE
Overdraft Line Drawdown Amount	AMT	DDA_TRANS

AC0 DEPBALMIN

- Within \$[input amount] of minimum balance requirement
- Transaction Code Category: Not applicable for balance alerts

Message Variable Description	Field Name	Source File
Account Minimum Balance Requirement	DEPBALMIN_REQ	ACCT_PROFILE
Customer Input Minimum Above Balance Requirement Threshold	DEPBALMIN_AMT	ACCT_PROFILE
Available Balance	BALANCE	DDA_BAL

Overdraft Occurred Alert

This alert notifies subscribers when an overdraft has occurred.

AD0 DDABALODL

- Overdraft occurred
- Transaction Code Category: Overdraft LOC Advance

Message Variable Description	Field Name	Source File
Negative Account Balance	BALANCE	DDA_BAL

Budget Alerts

Budget alerts notify subscribers when the selected transactions posted to the account, accumulated over a period of time (selected by the subscriber) reach the threshold they specify. There are alerts that can track the quantity, or amount of transactions that post to accounts within that time period.

AE0 DEPWDLTLY

- [Input No. of Transactions] in-branch withdrawals within x time
- Transaction Code Category: In Branch Withdrawal

Message Variable Description	Field Name	Source File
Customer Input In-Branch Withdrawals Tally Threshold	DEPWDLTLY_NUMBER	ACCT_PROFILE

Message Variable Description	Field Name	Source File
Customer Input Specified Posting Time Period (day, week, month)	DEPWDLTLY_PERIOD	ACCT_PROFILE
Total Amount of In-Branch Withdrawals in Specified Period	LAST(Amount_Real)	DEPWDLxxx C4 Amount Over Time
Tally of In-Branch Withdrawals in Specified Period	LAST(Count_Real)	DEPWDLxxx C4 Amount Over Time

AE0 DEPCHKTLY

- [Input No. of Transactions] checks paid within X time
- Transaction Code Category: Check Paid

Message Variable Description	Field Name	Source File
Customer Input Checks Paid Quantity Tally Threshold	DEPCHKTLY_NUMBER	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPCHKTLY_PERIOD	ACCT_PROFILE
Total Amount of Checks Paid in Specified Period	LAST(Amount_Real)	DEPCHKxxx C4 Amount Over Time
Tally of Checks Paid in Specified Period	LAST(Count_Real)	DEPCHKxxx C4 Amount Over Time

AE0 DEPPOSTLY

- [Input No. of Transactions] Debit Card purchases within X time
- Transaction Code Category: Debit Card Purchase

Message Variable Description	Field Name	Source File
Customer Input Debit Card Purchases Tally Threshold	DEPOSTLY_NUMBER	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPOSTLY_PERIOD	ACCT_PROFILE
Total Amount of Debit Card Purchases Tally in Specified Period	LAST(Count_Real)	DEPPOxxx C4 Amount Over Time
Tally of Debit Card Purchases Amount in Specified Period	LAST(Amount_Real)	DEPPOSxxx C4 Amount Over Time

AE0 DEPDEPTLY

- [Input No. of Transactions] deposits within X time
- Transaction Code Category: Deposit Posted

Message Variable Description	Field Name	Source File
Customer Input Deposits Posted Tally Threshold	DEPDEPTLY_NUMBER	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPDEPTLY_PERIOD	ACCT_PROFILE
Total Amount of Deposits in Specified Period	LAST(Amount_Real)	DEPDEPxxx C4 Amount Over Time
Tally of Deposits in Specified Period	LAST(Count_Real)	DEPDEPxxx C4 Amount Over Time

AE0 DEPOTCTLY

- [Input No. of Transactions] checks cashed by teller within X time
- Transaction Code Category: Check Cashed By Teller

Message Variable Description	Field Name	Source File
Customer Input Checks Cashed by Teller Tally Threshold	DEPOTCTLY_NUMBER	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPOTCTLY_PERIOD	ACCT_PROFILE
Total Amount of Checks Cashed by Teller in Specified Period	LAST(Amount_Real)	DEPOTCxxx C4 Amount Over Time
Tally of Checks Cashed by Teller in Specified Period	LAST(Count_Real)	DEPOTCxxx C4 Amount Over Time

AE0 DEPATMTLY

- [Input No. of Transactions] ATM withdrawals within X time
- Transaction Code Category: ATM Withdrawal

Message Variable Description	Field Name	Source File
Customer Input ATM Withdrawals Tally Threshold	DEPATMTLY_NUMBER	ACCT_PROFILE

Message Variable Description	Field Name	Source File
Customer Input Specified Time Period (day, week, month)	DEPATMTLY_PERIOD	ACCT_PROFILE
Total Amount of ATM Withdrawals in Specified Period	LAST(Amount_Real)	DEPATMxxx C1 ATM in Last Month
Tally of ATM Withdrawals in Specified Period	LAST(Count_Real)	DEPATMxxx C1 ATM in Last Month

AE0 DEPACHTLY

- [Input No. of Transactions] electronic payments within X time
- Transaction Code Category: Electronic Payment

Message Variable Description	Field Name	Source File
Customer Input Electronic Payments Tally Threshold	DEPACHTLY_NUMBER	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPACHTLY_PERIOD	ACCT_PROFILE
Total Amount of Electronic Payments in Specified Period	LAST(Amount_Real)	DEPACHxxx C4 Amount Over Time
Tally of Electronic Payments in Specified Period	LAST(Count_Real)	DEPACHxxx C4 Amount Over Time

AE0 DEPDBTTLY

- [Input No. of Transactions] total debit transactions within X time
- Transaction Code Category: Debit Transaction

Message Variable Description	Field Name	Source File
Customer Input Debits Posted Tally Threshold	DEPDBTTLY_NUMBER	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPDBTTLY_PERIOD	ACCT_PROFILE
Total Amount of Debits Posted in Specified Period	LAST(Amount_Real)	DEPDBTxxx C1 ATM in Last Month
Tally of Debits Posted in Specified Period	LAST(Count_Real)	DEPDBTxxx C1 ATM in Last Month

AE0 DEPWLAMT

- \$[Input No. of Amount] of in-branch withdrawals within X time
- Transaction Code Category: In Branch Withdrawal

Message Variable Description	Field Name	Source File
Customer Input In-Branch Withdrawals Amount Threshold	DEPWLAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPWLAMT_PERIOD	ACCT_PROFILE
Total Amount of In-Branch Withdrawals in Specified Period	LAST(Amount_Real)	DEPWLxxx C4 Amount Over Time
Tally of In-Branch Withdrawals in Specified Period	LAST(Count_Real)	DEPWLxxx C4 Amount Over Time

AE0 DEPCHKAMT

- \$[Input No. of Amount] in checks paid within X time
- Transaction Code Category: Check Paid

Message Variable Description	Field Name	Source File
Customer Input Checks Paid Amount Threshold	DEPCHKAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPCHKAMT_PERIOD	ACCT_PROFILE
Total Amount of Checks Paid in Specified Period	LAST(Amount_Real)	DEPCHKxxx C1 ATM in Last Month
Tally of Checks Paid in Specified Period	LAST(Count_Real)	DEPCHKxxx C1 ATM in Last Month

AE0 DEPPOSAMT

- \$[Input No. of Amount] in debit card purchases within X time
- Transaction Code Category: Debit Card Purchase

Message Variable Description	Field Name	Source File
Customer Input Debit Card Purchases Amount Threshold	DEPPOSAMT_AMT	ACCT_PROFILE

Message Variable Description	Field Name	Source File
Customer Input Specified Time Period (day, week, month)	DEPPOSAMT_PERIOD	ACCT_PROFILE
Total Amount of Debit Card Purchases in Specified Period	LAST(Amount_Real)	DEPPOSxxx C4 Amount Over Time
Tally of Debit Card Purchases in Specified Period	LAST(Count_Real)	DEPPOSxxx C4 Amount Over Time

AE0 DEPDEPAMT

- \$[Input No. of Amount] in deposits within X time
- Transaction Code Category: Deposit Posted

Message Variable Description	Field Name	Source File
Customer Input Deposits Posted Amount Threshold	DEPPOSAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPPOSAMT_PERIOD	ACCT_PROFILE
Total Amount of Deposits Posted in Specified Period	LAST(Amount_Real)	DEPDEPxxx C4 Amount Over Time
Tally of Deposits Posted in Specified Period	LAST(Count_Real)	DEPDEPxxx C4 Amount Over Time

AE0 DEPOTCAMT

- \$[Input No. of Amount] in checks cashed by teller within X time
- Transaction Code Category: Check Cashed By Teller

Message Variable Description	Field Name	Source File
Customer Input In-Branch Checks Cashed Amount Threshold	DEPOTCAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPOTCAMT_PERIOD	ACCT_PROFILE
Total Amount of In-Branch Checks Cashed in Specified Period	LAST(Amount_Real)	DEPOTCxxx C1 ATM in Last Month

Message Variable Description	Field Name	Source File
Tally of In-Branch Checks Cashed in Specified Period	LAST(Count_Real)	DEPOTCxxx C1 ATM in Last Month

AE0 DEPATMAMT

- \$[Input No. of Amount] in ATM withdrawals within X time
- Transaction Code Category: ATM Withdrawal

Message Variable Description	Field Name	Source File
Customer Input ATM Withdrawals Amount Threshold	DEPATMAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPATMAMT_PERIOD	ACCT_PROFILE
Total Amount of ATM Withdrawals in Specified Period	LAST(Amount_Real)	DEPATMxxx C1 ATM in Last Month
Tally of ATM Withdrawals in Specified Period	LAST(Count_Real)	DEPATMxxx C1 ATM in Last Month

AE0 DEPACHAMT

- \$[Input No. of Amount] in electronic payments within X time
- Transaction Code Category: Electronic Payment

Message Variable Description	Field Name	Source File
Customer Input Electronic Payments Posted Amount Threshold	DEPACHAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPACHAMT_PERIOD	ACCT_PROFILE
Total Amount of Electronic Payments Posted in Specified Period	LAST(Amount_Real)	DEPACHxxx C4 Amount Over Time
Tally of Electronic Payments Posted in Specified Period	LAST(Count_Real)	DEPACHxxx C4 Amount Over Time

AE0 DEPDBTAMT

- \$[Input No. of Amount] in total debit transactions within X time
- Transaction Code Category: Debit Transaction

Message Variable Description	Field Name	Source File
Customer Input Debits Posted Amount Threshold	DEPDBTAMT_AMT	ACCT_PROFILE
Customer Input Specified Time Period (day, week, month)	DEPDBTAMT_PERIOD	ACCT_PROFILE
Total Amount of Debits Posted in Specified Period	LAST(Amount_Real)	DEPDBTxxx C1 ATM in Last Month
Tally of Debits Posted in Specified Period	LAST(Count_Real)	DEPDBTxxx C1 ATM in Last Month

Unusual Account Activity Alerts

Unusual activity alerts notify subscribers when account activities they selected post above their normal account usage patterns. The system considers a variety of factors in determining 'unusual activity' including the sensitivity threshold the subscribers selected. The higher the sensitivity, the more likely they are to receive an alert.

The unusual activity alerts look historically for the number and amount of transactions that occur historically in each of four portions of the month over the previous months. From that data, they derive the average amount and count of these transactions for each of those periods. These alerts fire whenever a customer's cumulative amount for one of those periods in the current month or cumulative count exceed those averages by a certain number of standard deviations, determined by parameters in a table which is configured for you by IBM Unica. Those numbers of standard deviations have values for low, medium and high sensitivity choices, something which the subscribers decide upon, depending on how sensitive they want the alert to be. These alerts also fire when the amount of an individual transaction exceeds the maximum in that portion of the month by a ratio again determined by an entry in a table IBM Unica will set up for you.


'Unusual' does not necessarily indicate that unauthorized transactions have occurred. Subscribers that receive an unusual activity alert should review their account activity immediately to determine the origin of this activity to their own satisfaction.

(Includes "sensitivity" dropdown list: high, medium, low.)

 These alerts do not have variables that you would use in a message.

AG2 DEPWLDIF

- Unusual in-branch withdrawals
- Transaction Code Category: In Branch Withdrawal

 This alert does not have variables that you would use in a message.

AG1 DEPCHKDIF

- Unusual Check Behavior Detected
- Transaction Code Category: Check Paid

 This alert does not have variables that you would use in a message.

AG3 DEPOTCDIF

- Cashed checks
- Transaction Code Category: Check Cashed By Teller

 This alert does not have variables that you would use in a message.

AG4 DEPATMDIF

- ATM withdrawals
- Transaction Code Category: ATM Withdrawal

 This alert does not have variables that you would use in a message.

AG5 DEPACHDIF

- Electronic payments
- Transaction Code Category: DEPACHDIF

 This alert does not have variables that you would use in a message.

Scheduled Deposit Alerts

Scheduled deposit alerts notify subscribers when a regularly occurring monthly deposit amount they specify is not credited to their account by the day of the month they specify.

AH4 DDASCHCR1

- Deposit of \$[Input amount] not posted by [input date] of the month
- Transaction Code Category: Scheduled Deposit

Message Variable Description	Field Name	Source File
Specified Payment Amount to Look For	DDASCHCR1_AMT	ACCT_PROFILE
Specified day of the month	DDASCHCR1_DATE	ACCT_PROFILE

AH5 DDASCHCR2

- Deposit of \$[Input amount] not posted by [input date] of the month
- Transaction Code Category: Scheduled Deposit

Message Variable Description	Field Name	Source File
Specified Payment Amount to Look For	DDASCHCR2_AMT	ACCT_PROFILE
Specified day of the month	DDASCHCR_DATE	ACCT_PROFILE

AH6 DDASCHCR3

- Deposit of \$[Input amount] not posted by [input date] of the month
- Transaction Code Category: Scheduled Deposit

Message Variable Description	Field Name	Source File
Specified Payment Amount to Look For	DDASCHCR3_AMT	ACCT_PROFILE
Specified day of the month	DDASCHCR1_DATE	ACCT_PROFILE

Scheduled Payment Alerts

Scheduled payment alerts notify subscribers when the regularly occurring debit amount they specify is not posted to their account by the day of the month they specify.

AH1 DDASCHPY1

- Payment of \$[Input amount] not posted by [input date] of the month
- Transaction Code Category: Scheduled Payment

Message Variable Description	Field Name	Source File
Specified Payment Amount to Look For	DDASCHPY1_AMT	ACCT_PROFILE
Day of the Month Payment is Due	DDASCHPY1_DATE	ACCT_PROFILE

AH2 DDASCHPY2

- Payment of \$[Input amount] not posted by [input date] of the month
- Transaction Code Category: Scheduled Payment

Message Variable Description	Field Name	Source File
Specified Payment Amount to Look For	DDASCHPY2_AMT	ACCT_PROFILE
Day of the Month Payment is Due	DDASCHPY2_DATE	ACCT_PROFILE

AH3 DDASCHPY

- Payment of \$[Input amount] not posted by [input date] of the month
- Transaction Code Category: Scheduled Payment

Message Variable Description	Field Name	Source File
Specified Payment Amount to Look For	DDASCHPY3_AMT	ACCT_PROFILE
Day of the Month Payment is Due	DDASCHPY3_DATE	ACCT_PROFILE

G Retail Banking Event Based Marketing (RBEEM) Library

- [About the Retail Banking Event Based Marketing \(RBEEM\) library](#)
- [List of Retail Banking Event Based Marketing \(RBEEM\) trigger systems](#)

About the Retail Banking Event Based Marketing (RBEEM) library

This section lists the trigger systems in the Retail Banking Event Based Marketing (RBEEM) library. This library contains trigger systems that monitor the behavior of mature retail banking customers. The triggers identify behaviors that indicate opportunities for cross-selling, retention, and nurturing. You can distribute the outcome of these triggers, for example through Campaign, to a member of your organization who will use the content of the triggers as a basis for contact; or you can feed the triggers directly to Campaign to be used as content for a marketing campaign.

List of Retail Banking Event Based Marketing (RBEEM) trigger systems

Categories and Trigger Systems	Description	Workspace
Category_A_Unusually Large Transaction Amount	This group of triggers looks for a customer whose cumulative calendar week's transaction volume is more than specified percentage and greater than historical weekly maximum. The trigger-specific volume threshold and sufficient historical activity must exist for trigger to fire. There are 10 different types of Transactions that can be identified within this group, each one is based on a specific type of transaction.	
Unusually Large Check Paid	This trigger detects when an unusually large Check Paid transaction has occurred.	EA1

Categories and Trigger Systems	Description	Workspace
Unusually Large ACH Debit	This trigger detects when an unusually large ACH debit transaction has occurred.	EA2
Unusually Large ACH Credit	This trigger detects when an unusually large ACH credit transaction has occurred.	EA3
Unusually Large Outgoing Wire Transfer (Debit)	This trigger detects when an unusually large outgoing wire transfer (debit) transaction has occurred.	EA4
Unusually Large Outgoing Wire Transfer (Credit)	This trigger detects when an unusually large incoming wire transfer (credit) transaction has occurred.	EA5
Unusually Large Disbursement	This trigger detects when an unusually large ACH Credit transaction has occurred.	EA6
Unusually Large Deposit	This trigger detects when an unusually large deposit transaction has occurred.	EA7
Unusually Large Credit Card Purchase	This trigger detects when an unusually large credit card purchase transaction has occurred.	EA8
Unusually Large Debit Card Purchase	This trigger detects when an unusually large debit card purchase transaction has occurred.	EA9
Unusually Large Bill Payment	This trigger detects when an unusually large bill pay transaction has occurred.	EAa
Category B_and_C_Unusually Large Transaction Volume or Aggregate Amount	This group of triggers is looking for a customer who has an unusually large transaction for a specific transaction code. The category B triggers target transaction volume, while the category C triggers target+B30 aggregated transaction amount. The target value must be greater than a user defined percentage over the maximum weekly value from the previous 13 months. Because the majority of rules in the two categories are similar, to reduce state size and improve performance, two triggers are merged into one workspace prefixed with "EB" and share the same container. There are total 10 workspaces, one for a specific transaction type. Each workspace contains two triggers: one for the Unusually Large Volume and one for the Unusually Large Aggregate Amount. Likewise, there are two outcome strings.	

Categories and Trigger Systems	Description	Workspace
	Unusually Large Check Paid Volume	EB1
	Unusually Large Aggregate Check Paid Amount	
	Unusually Large ACH Debit Volume	EB2
	Unusually Large Aggregate ACH Debit Amount	
	Unusually Large ACH Credit Volume	EB3
	Unusually Large Aggregate ACH Credit Amount	
	Unusually Large Outgoing Wire Transfer Volume	EB4
	Unusually Large Aggregate Outgoing Wire Transfer Amount	
	Unusually Large Incoming Wire Transfer Volume	EB5
	Unusually Large Aggregate Incoming Wire Transfer Amount	
	Unusually Large Disbursement Volume	EB6
	Unusually Large Aggregate Disbursement Amount	
	Unusually Large Deposit Volume	EB7
	Unusually Large Aggregate Deposit Amount	

Categories and Trigger Systems	Description	Workspace
Unusually Large Credit Card Purchase Volume		EB8
Unusually Large Aggregate Credit Card Purchase Amount		
Unusually Large Debit Card Purchase Volume		EB9
Unusually Large Aggregate Debit Card Purchase Amount		
Unusually Large Bill Payment Volume		EBa
Unusually Large Aggregate Bill Payment Amount		
Unusually Large VRU Inquiry Volume	This trigger looks for a customer who has an unusually large VRU inquiry volume. The volume must be greater than a user defined percentage over the maximum monthly value from the previous 13 months.	EBb
Category_D_Interruption in Regular Deposit	The triggers in this category look for a customer who has had regular deposits of a specific type, and then has as an interruption in those deposits.	
Interruption in Regular ACH Deposit		ED1
Interruption in Regular ACH Payroll Deposit		ED2
Interruption in Regular ACH Social Security Deposit		ED3
Interruption in Regular ACH Social Security Disability Deposit		ED4
Category_E_Lack of Activity	This category of triggers looks for the lack of a specific type of activity after an opening deposit.	
No Check Paid after an Opening Deposit	This trigger looks for no check paid activities X days after the opening deposit.	EE1

Categories and Trigger Systems	Description	Workspace
No Deposit activities after an Opening Deposit	This trigger looks for no deposit activities X days after the opening deposit.	EE2
No Debit Card Usage After an Opening Deposit	This trigger looks for no debit card activities X days after the opening deposit.	EE3
No Online Transaction Activities after an Opening Deposit	This trigger looks for no online transaction activities X days after the opening deposit.	EE4
No Bill Pay after an Opening Deposit	This trigger looks for no bill pay activities X days after the opening deposit. The customer has to be enrolled in bill pay before this trigger will fire.	EE5
Category_F_Transaction on Greater Than Fixed Amount	This category of triggers fire if a customer has not been performing a certain type of transaction recently when a large transaction of that type comes in.	
Credit Card Purchase	This trigger fires if a customer has not been using the credit card recently when a large purchase came in.	EF1
No Debit Card Purchase after an Opening Deposit	This trigger fires if a customer has not been using the debit card recently when a large purchase came in.	EF2
ACH Debit	This trigger fires if a customer has not have ACH debit recently when a large ACH debit came in.	EF3
Outgoing Wire Trans	This trigger fires if a customer has not have outgoing wire transfers recently when a large outgoing wire transfer came in.	EF4
Category_G_Shift from Bank-owned ATM Usage		
Shift from Bank-owned ATM Usage	This trigger is looking for high-usage customers of bank owned ATMs, who then have a drop in usage of bank owned ATMs and an increase in usage of non bank owned ATMs.	EG1
Category_H_Drop in Account Activity	The triggers in this category look for a drop in a particular type of account activity after it has been determined that there was high activity on the account.	
Drop in Bill Pay Activity		EH1

Categories and Trigger Systems	Description	Workspace
Drop in Branch Activity		EH2
Drop in Credit Card Activity		EH3
Drop in ATM Activity		EH4
Drop in Debit Card Activity		EH5
Drop in Check Card Usage		EH6
Drop in Check Volume		EH7
Drop in VRU Inquiries		EH8
Drop in Disbursements		EH9
Category_I_And_J_Large Deposit followed by Large Disbursement		
Large Deposit followed by Large Disbursement	This trigger looks for a customer who has a reasonably large Deposit transaction, followed by a large Disbursement within 3 days of that deposit.	EI1
Multiple Large Deposit followed by Large Disbursement	This trigger looks for a customer who has had a large deposit followed by a large disbursement three times within a two month period.	EJ1
Category_K_Draw on Lines of Credit		
HELOC Debit	This trigger identifies a Home Equity Line of Credit (HELOC) Debit by looking for a Debit or Credit Indicator of "D" in the HELOC Transactions data source.	EK1
OD LOC Advance	This trigger looks for an Overdraft LOC Advance.	EK1
HELOC	Debit This trigger identifies a Home Equity Line of Credit (HELOC) Debit by looking for a Debit or Credit Indicator of "D" in the HELOC Transactions data source.	EK1
Category_L_Draw on Lines of Credit-2		

Categories and Trigger Systems	Description	Workspace
HELOC Debit	This trigger looks for a Home Equity Line of Credit (HELOC) Debit.	EL1
Credit Card Cash Advance	This trigger looks for a Credit Card Advance	EL1
HELOC Debit and Credit Card Cash Advance	This trigger looks for HELOC Debit and Credit Card Cash Advance occurring within 4 days of each other.	EL1
HELOC Above Threshold	This trigger looks for a HELOC balance posting with an amount greater than a threshold.	EL2
Credit Card Cash Advance	This trigger looks for a credit card cash advance.	EL2
HELOC High Balance and Credit Card Cash Advance	This trigger looks for a HELOC High Balance and Credit Card Cash Advance occurring within 4 days of each other.	EL2
Category_M_HELOC and Increased Credit Card Usage		
Increased Credit Card Usage	This trigger looks for an increase in credit card usage	EM1
HELOC Debit	This trigger identifies Home Equity Line of Credit (HELOC) Debit.	EM1
HELOC and Increased Credit Card Usage	This trigger looks for a Home Equity Line of Credit (HELOC) Debit and an Increased Credit Card Usage occurring within 1 week of each other.	EM1
Category_N_Shift in Home ATM and Declining Balance		
Shift in Home ATM and Declining Balance	This trigger detects when a customers monthly balance is falling at the same time a shift in ATM usage has occurred. To reduce complexity and redundant processing, components from the EG1 workspace have been reused for this trigger. NOTE: The EN1 workspace is not complete without being merged with the EG1 workspace.	EN1
Category_O_Check Drop with no Change in ACH Debit or Bill Pay		

Categories and Trigger Systems	Description	Workspace
Check Drop with no Change in ACH Debit or Bill Pay	This trigger detects a Drop in Check Writing, combined with no significant increase in either ACH Debit or Bill Pay activity. To reduce complexity and redundant processing, components from the EB2, EBa, and EH7 workspaces have been reused for this trigger. See the respective design documents for more details on these components.	EO1
Category_P_Bill Pay drop with no Change in ACH Debit or Check Writing		
Bill Pay drop with no Change in ACH Debit or Check Writing	This trigger detects a drop in bill pay usage after a high, combined with no significant increase in either ACH Debit or check writing activity. To reduce complexity and redundant processing, components from the EB1, EB2, and EH1 workspaces have been reused for this trigger. See the respective design documents for more details on these components.	EP1
Category_Q_Drop in Balance Combo		
Drop in OTC Deposits and Balance	This trigger detects a drop in OTC (Over the Counter) deposits and significantly decreased balance for frequent OTC depositors.	EQ1
Drop in Debit Card Purchase and Balance	This trigger detects drop in Debit Card Purchase and significantly decreased balance.	EQ2
Category_R_Large Withdrawal Transactions		
Large CC Purchase	This trigger detects the occurrence of a large credit card purchase, where the customer has not used the credit card recently.	ER1
Large ACH Debit	This trigger detects the occurrence of large ACH debits, where the customer has not used ACH debits recently.	ER2
Large Outgoing Wire Transfer	This trigger detects the occurrence of a large outgoing wire transfer, where the customer has not had outgoing wire transfers recently.	ER3
Category_S_First Overdraft		

Categories and Trigger Systems	Description	Workspace
First Overdraft after Some Time for High Value Customer	This trigger looks for a high value customer who has an overdraft item after no overdraft for some time.	ES1
Category_T_Account Balance Close to Minimum		
Account Balance Close to Minimum	This trigger looks for a customer to have a balance on their account drop to within a percentage amount of the required minimum balance for the account type. If the same account is within the range for three consecutive months, the trigger will fire.	ET1
Customer near balance minimum	This trigger looks for a customer who is near the balance minimum in two or more accounts.	ET1
Customer near balance minimum in two accounts	This trigger looks for a customer who is near the balance minimum in two accounts.	ET2

H Additional Information About Using the Retail Banking Event Based Marketing Library

- [More About Event Based Marketing Libraries](#)
- [Account profile file for RBEBM triggers systems](#)
- [Transaction code categories for RBEBM triggers](#)

More About Event Based Marketing Libraries

This appendix provides information specific to Retail Banking Event Based Marketing (RBEBM) triggers. It describes the account profile file and lists the transaction code categories.

Account profile file for RBEBM triggers systems

For customers who only use retail banking event-based marketing trigger systems, the profile file is typically used to further refine out-of-the-box trigger rules. Trigger rules are almost always refined as part of the implementation process.

Below is a sample of account profile data fields which might be used by a bank using the Retail Banking Event Based Marketing (RBEBM) library. For a complete list of data sources and fields, refer to the overview spreadsheet provided within the library directory.

Sample of account profile data fields used by Retail Banking Event Based Marketing triggers

field name	notes
CUSTOMER_ID	Customer Key
BRANCH	Default Customer Branch
NAME	Customer Name
ADDRESS	Customer Address
PRIMARY_TEL_NUM	Primary Telephone Number
SECONDARY_TEL_NUM	Secondary Telephone Number
SEGMENTATION_CODE	Customer Segment Code
PROFIT_SCORE	Profit Score
ATTRITION_SCORE	Attrition Risk Score
POTENTIAL_SCORE	Value Score
TOTAL_INVESTABLE_ASSETS	Customer Assets
DO_NOT_SOLICIT	Do Not Solicit Indicator
NUM_OF_ACCOUNTS	Number of accounts for customer

Transaction code categories for RBEBM triggers

The following table describes the Transaction Code Categories for Event Based Marketing triggers.

Transaction Code Category	Description and Comments
ACH Debit	Exclude check conversion/ACH transaction codes like ARC.
ACH Deposit	ACH Credit
ACH Payroll Deposit	Note that many banks have only ACH credit transaction codes, not distinct codes for payroll deposit.

Transaction Code Category	Description and Comments
ACH Social Security Deposit	Note that many banks have only ACH credit tran codes, not distinct codes for Social Security deposits.
ACH Social Security Disability Deposit	Note that many banks have only ACH credit tran codes, not distinct codes for Social Security deposits.
ATM Withdrawal	
Branch Activity	
Check Card Usage	
Check Paid	Checks paid transaction codes AND Check conversion/ACH codes in which the original instrument used by the customer was a check (such as ARC).
Debit Card Purchase	
Deposit Posted	
Disbursement	
Electronic Payment	On-line banking withdrawal tran codes and ACH debit tran codes. Exclude check conversion/ACH transactions like ARC.
Home Bank ATM Withdrawal	"On-us" ATM withdrawals
Non Home Bank ATM Withdrawal	"Off-us" ATM withdrawals
On-Line Banking Payment	
On-line Transaction	Any on-line transaction - uses to monitor drop in general customer activity.
Opening Deposit	Used for on-boarding triggers to determine account opening data. Trigger can be easily modified if no such tran code exists; work with IBM Unica Consulting.
Over the Counter Deposit	

Transaction Code Category	Description and Comments
Wire Transfer Payment	Incoming wire transfer, excluding non-value messages

I Retail Banking On-boarding Library

- [About the Retail Banking On-boarding library](#)
- [Retail Banking On-boarding Trigger System Library](#)

About the Retail Banking On-boarding library

This section lists the trigger systems in the Retail Banking On-boarding (RBONB) library. This library contains trigger systems that provide a mechanism for monitoring the behavior of newly-acquired retail banking accounts with the intent of following up with individual customers who are not using banking services as expected. These trigger systems are intended to help identify a new account that has not fully come onboard with the bank, possibly because a back office process may have gone wrong when making a new customer into a fully functioning account, or possibly because the customer has not yet decided to fully stop using their older account with their previous bank and move over to this new account.

Retail Banking On-boarding Trigger System Library

Category_A_Lack of activity after opening deposit

Categories and Trigger Systems	Description	Workspace
No Direct Deposit after an opening deposit	This trigger looks for a customer who does not have any direct deposits within X days of the account opening deposit.	OA1
No Check and Bill Pay activities after an opening deposit	This trigger looks for a customer who does not have any checks paid and bill pay activities within X days of the account opening deposit.	OA4

Category_B_Balance less than X

Categories and Trigger Systems	Description	Workspace
Closing balance is less than a specific amount	This trigger looks for a customer whose 'Closing Account Balance' is less than \$Y.	OB1
Closing Balance less than package minimum	This trigger looks for a customer whose 'Closing Account Balance' is less than their Bank Package Minimum Balance.	OB2
Average Balance is less than a specific amount	This trigger looks for a customer whose statement's 'Average Balance' is less than \$Y.	OB3
Average Balance is less than package minimum	This trigger looks for a customer whose statement's 'Average Balance' is less than their Bank Package Minimum Balance.	OB4

Category_C_Lack of On-line banking activity

Categories and Trigger Systems	Description	Workspace
No On-line Banking Activity	This trigger will fire when the customer has signed up for on-line banking and has not had any on-line banking activity within x days of signing up.	OC1

Category_D_Check Number and Activity Monitoring

Categories and Trigger Systems	Description	Workspace
The triggers in this set monitor check numbers and activity.		
No Permanent Check Book	This trigger looks for a customer who does not have any checks paid with serial numbers greater than 100 within X days of the account opening deposit.	OD1
Counter Check Used	This trigger looks for a customer who has a check posted with "zero" or blank check number within X days of the account opening deposit. A blank check in the feed may be denoted by having a check number of "0000", "0", "_" or "". This trigger should only fire once for the first blank check paid.	OD1
No Permanent Checks and Counter Check Used	This trigger is the combination of triggers of the other two triggers in this workspace. It looks for a customer who does not have any checks paid with serial numbers greater than 100 but has a blank check posted within X days of the account opening deposit.	OD1

Category_E_X days after account opening

Categories and Trigger Systems	Description	Workspace
15 Days after account was opened	This trigger fires 15 days after account opened and fires only once for an account.	OE1
30 Days after account was opened	This trigger fires 30 days after account opened and fires only once for an account.	OE1
45 Days after account was opened	This trigger fires 45 days after account opened and fires only once for an account.	OE1
60 Days after account was opened	This trigger fires 60 days after account opened and fires only once for an account.	OE1
90 Days after account was opened	This trigger fires 90 days after account opened and fires only once for an account.	OE1
180 Days after account was opened	This trigger fires 180 days after account opened and fires only once for an account.	OE1

Category_F_X days after last customer contact

Categories and Trigger Systems	Description	Workspace
15 Days Since Last Contact	For an account opened within the last y days, this trigger fires 15 days after last customer contact. Customer contact date determined by incoming file fed into Detect. Y default is 180.	OF1
30 Days Since Last Contact	For an account opened within the last y days, trigger fires 30 days after last customer contact. Customer contact date determined by incoming file fed into Detect. Y default is 180.	OF1
45 Days Since Last Contact	For an account opened within the last y days, trigger fires 45 days after last customer contact. Customer contact date determined by incoming file fed into Detect. Y default is 180.	OF1
60 Days Since Last Contact	For an account opened within the last y days, trigger fires 60 days after last customer contact. Customer contact date determined by incoming file fed into Detect. Y default is 180.	OF1

Category_G_Activity approaching package threshold

Categories and Trigger Systems	Description	Workspace
ATM activity approaching package threshold	This trigger looks for a customer whose ATM activities reached within Y (default is 1) transactions of the package threshold. It may fire only once per statement cycle but it may fire multiple times within the first Z (default is 4) months of account opening.	OG1
Branch activity approaching package threshold	This trigger looks for a customer whose branch activities reached within Y (default is 1) transactions of the package threshold. It may fire only once per statement cycle but it may fire multiple times within the first Z (default is 4) months of account opening.	OG2
Checks paid activity approaching package threshold	This trigger looks for a customer whose checks paid reached within Y (default is 1) transactions of the package threshold. It may fire only once per statement cycle but it may fire multiple times within the first Z (default is 4) months of account opening.	OG3
Disbursement activity approaching package threshold	This trigger looks for a customer whose disbursements reached within Y (default is 1) transactions of the package threshold. It may fire only once per statement cycle but it may fire multiple times within the first Z (default is 4) months of account opening.	OG4

Category_I_Lack of profile after X days

Categories and Trigger Systems	Description	Workspace
Profile Missing	Categories and Trigger Systems	OI1

J Telco Event Based Marketing, Account Based (TELA) Library

- Telco Event Based Marketing, Account Based (TELA) library
- About data requirements for the TELA library
- List of Telco EBM, Account Based (TELA) trigger systems

Telco Event Based Marketing, Account Based (TELA) library

This section lists the trigger systems in the Telco Event Based Marketing, Account Based (TELA) library and provides some information about the data they require. The TELA library contains event-based marketing triggers for a telephone service provider (telco). They watch for behaviors that indicate some change in the pattern of voice call or recharge usage. Unlike the number-based triggers, these triggers are account based. The entityID is an account number and there is nothing in the feed files that indicates the calling or the called number.

About data requirements for the TELA library

Each library comes with the data sources you need in order to test and understand the trigger systems within them. This section provides some details about the data sources used by the trigger systems in the TELA library.

There are three types of data feeds:

Profile fields: These fields are used mostly to attach information to the Actions. The Account_Type field indicates whether it is a prepaid or contract account.

Acct_Cdr: This is another form of a call data record (CDR). It records various details about the call. Unlike the number based CDR, there is no calling number, called number, or indication whether the call was made in or out of the network. Other fields that are included are:

- **IMEI:** The ID of the mobile phone that was involved in the call
- **PRODUCT CATEGORY:** Indicates whether it was a voice, SMS, or MMS call. In the number-based CDRs the voice and SMS records were broken into separate feeds
- **CALL DIRECTION:** Indicates whether it was in incoming or outgoing call
- **MANUFACTURER:** The maker of the cell phone

Recharge: This feed indicates the date and amount of money paid to recharge a prepaid account. This feed is used in a trigger to watch for a lapse in recharging the phone.

List of Telco EBM, Account Based (TELA) trigger systems

This section lists the triggers that are included in the Telco EBM, account based trigger system library, which is also known as the TELA library.

Categories and Trigger Systems	Description	Workspace
A Large Recharge Event	Detects when a customer recharges their account with an amount more than 99.00	TA1
A Sony-Ericsson Handset Call	Detects when a customer makes a call on a Sony-Ericsson handset.	TA2
Account Started Active Roaming	This trigger looks for an increase in roaming compared to past behavior.	TA8
IMEI Change	This trigger looks for accounts that have used their old phone for a period of time and are now using a new phone, with a new IMEI (International Mobile Equipment Identity).	TA7
No Recharge No Calls	This trigger looks for Prepaid accounts that have not recharged within the average of their recharge intervals and have not made any voice call recently.	TA5
Numerous Customer Care Calls	Detects a pattern of calls to the customer care number, aiming to retain potentially unhappy customers.	TA3
Usage Monitoring	These three triggers detect when there is an unusual rise or drop in account usage, where the minutes used are more (or less) than 2 standard deviations of the historical minutes used.	
Rise in Average Usage	Average Minutes Last Week are 2 standard deviations greater than the average minutes in the previous 10 weeks.	TA6

Categories and Trigger Systems	Description	Workspace
Drop in Average Usage	Average Minutes Last Week are 2 standard deviations less than the average minutes in the previous 10 weeks.	TA6
Unusual Usage Activity	Average Minutes Last Week are 2 standard deviations more or less than the average minutes in the previous 10 weeks.	TA6

K Telco Event Based Marketing, Phone Number Based (TELP) Library

- Telco Event Based Marketing, Phone Number Based (TELP) library
- About data requirements for the TELP library
- List of Telco EBM, Phone Number Based (TELP) trigger systems

Telco Event Based Marketing, Phone Number Based (TELP) library

This section lists the trigger systems in the Telco Event Based Marketing, Phone Number based (TELP) library, and provides some information about the data it requires. The TELP library contains event based marketing triggers for a telephone service provider (telco). They watch for behaviors that indicate some change in the pattern of voice call or SMS usage. These trigger systems are based on the phone number, rather than the account.

About data requirements for the TELP library

Each library comes with the data sources you need in order to test and understand the trigger systems within them. This section provides some details about the data sources used by the trigger systems in the TELP library.

There are three types of data feeds:

- **Profile:** This feed contains general information about the customer, including the start date of the customer's service and various plans to which that the customer has subscribed. The entity ID is the phone number.

- **Voice_CDR (Voice Call Data Record):** This feed contains a record that a phone call was made. The record indicates that a caller (CALLING_NUMBER) called someone (CALLED_NUMBER) on this DATE for so many seconds (SECONDS_USED). The entityID is the calling number. The calling number is a customer of the Telco. It includes a flag that indicates whether the call went outside of or stayed within the Telco's network.
- **SMS_CDR:** This feed contains a record that an SMS (text) message was sent. Conceptually it works like a phone CDR except that there is no duration field.

Phone numbers that begin with a '6' are mobile numbers. Phone numbers that begin with a '9' are land line numbers. The CALLED NUMBER '666' is a usage request, to determine how many minutes have been used so far.

There are two types of customers:

- **Contract customers:** These customers pay a monthly services charge and have a certain plan. This is the standard usage in the United States.
- **Prepaid customers:** These customers pay for a set number of minutes on the phone. They can only use the phone so long as their meter has not run out.

List of Telco EBM, Phone Number Based (TELP) trigger systems

This section lists the triggers that are included in the Telco EBM, phone number based trigger system library, which is also known as the TELP library.

Categories and Trigger Systems	Description	Workspace
Called Competitor	This set of trigger systems monitor calls to competitors.	
Called Competitor	Detects when a customer has called a number belonging to a competitor.	TNh
Called Competitor 3 Times	Detects when a customer has called a number belonging to a competitor three times within one week.	TNh
Competitor Calls total more than 10 minutes	Detects when a customer has made calls to a competitor numbers with a total duration of more than 10 minutes in a week.	TNh
Decreasing On-net/Off-net Minutes Ratio	This trigger detects a decrease in the ratio of On-net to Off-net minutes.	TNb
Happy First Anniversary	Reports when a customer has reached their first year anniversary within a 5 day window.	TNg

Categories and Trigger Systems	Description	Workspace
High Off-Net Usage	Detects possible attrition of a high-value customer, due to a high number of calls and SMS sent through another company's network.	TNd
No calls to Land Line numbers in 60 days	Detects when no calls have been made to a landline number in 60 days.	TN7
No calls to List of Preferred Numbers	This trigger is designed to detect a person who subscribes to the "ListOfPreferredNumbers" service and does not use it for a month (30 days).	TN6
No calls to Preferred Number	This trigger is designed to detect a customer who subscribes to the "PreferredNumber" service and does not use it within 30 days.	TN5
Significant Voice SMS Traffic	This trigger is looking for a spike but is not comparing current usage to past usage. It is collecting a count of SMS messages and numbers of voice seconds used and triggering when the voice usage is greater than 60 and SMS message count is greater than 2.	TNe
SMS plan	This trigger consists of 3 actions and tracks a user's SMS usage and tells the user which plan would be appropriate based on the number of messages sent. The actions each include a call to the call center. The actions also each track average SMS calls per month in the following ranges: 20-50, 50-100, and over 100.	TNa
Three Long Calls Occurred in 1 Week	Detects when a customer has 3 long calls within 1 week.	TNf
Usage Downturn after Complain	Detects a phone usage reduction after a pattern of calls to the call center.	TN9
Usage Monitoring	The triggers in this set monitor changes in usage patterns.	
Down Trend and Low Value	Detects when there is a downward Spike of 30% in the sum of monthly minutes compared to the previous month's sum for a low value contract.	TN1
Usage Requests greater than 10	Detects when a low value contract makes a high number of usage requests to determine how many minutes have been used so far on their plan.	TN1
Low Value Contract Jeopardy	This trigger system looks for a pattern of the other two triggers in this category combined. This trigger fires when both events (a downward trend on a low value contract is combined with more than 10 usage requests) occur within a in a two week period.	TN1

L Terminology Used in this Guide

■ Terminology used in this guide

Terminology used in this guide

The following terms are used when discussing Detect.

- **ACH (Automated Clearing House):** An ACH is an electronic fund transfer from a checking or savings account
- **Action:** An Action in Detect is a component that writes an outcome to the outcome table.
- **Alert:** An alert is a type of trigger system for which the result is typically sent to the institution's customers.
- **Container:** A data structure within Detect. It is used to store transactional data or the value of calculations that need to be stored over time.
- **Container Manipulator (CM):** A component in Detect used for testing data in containers or inserting/deleting data from containers.
- **DDA (Demand Deposit Account):** A DDA is a banking system that captures deposits and withdrawals (credits and debits) to an account
- **Entity:** An entity is the object of the detection. It is generally a customer, account, or household, but it could be anything that identifies the object that is being watched. For simplicity, an entity is generally referred to as a customer.
- **EBM (Event Based marketing):** EBM is a form of marketing which reacts to a customer's needs (events) as indicated by his or her transactional patterns. Within Detect it is a methodology of using triggered events as a call to action to service your customer.
- **Event:** Event is a very generic term within Detect. It is used to indicate something important in the incoming transaction stream (incoming or simple events), it is used to indicate a partial completion of a trigger (internal or complex events), and it is used to indicate the outcome (triggered events or trigger).
- **Event-detection engine (engine):** The engine is the core of the Detect application. It takes incoming transactions, captures past relevant actions in state history, uses the library to detect patterns across the incoming transactions and state history, and triggers when a behavior is detected.

- **FLI (Forward Looking Inactivity):** An FLI is a component in Detect that acts as a timer, firing in the future if the designated tracked event does not occur in the intervening time. For example, an FLI may be configured to fire its timer 15 days after an Opening Deposit, unless a Payroll Deposit (tracked event) occurs.
- **Inactivity:** Indicates a pattern of inaction. For example, a customer who has opened an account but never used it is exhibiting a behavior of inactivity.
- **Library:** A library a collection of trigger systems
- **Merchant Category Code (MCC):** A field on a credit card transaction that indicates the category of the merchant from whom you made the purchase. Examples of categories include airlines, hotels, rental car agencies, clothing, and groceries.
- **NSF (Non-sufficient Funds):** An NSF is a banking transaction that indicates that insufficient funds were available to cover a check (bounced check)
- **On-boarding:** On-boarding is a marketing approach that involves watching the behavior of an account during its first 3 to 6 months to ensure that the customer is properly becoming a fully engaged customer of the bank with this account.
- **Outcome Table:** A database table that contains the outcomes fired by Detect. An outcome table is named according to the convention [Rule User].[vendor]_[workspace_code]_outcome.
- **OMT (Outcome Management Tool):** A utility in Detect that post-processes the outcome table, generating a separate table for each type of trigger and exploding the contents of the XML Message field into separate columns.
- **Profile:** A profile is a collection of optional pieces of information that indicate relatively static information that is known about the entity (customer or account). It includes details such as demographic information, market segment indicators, monthly balances, credit scores, etc. It contains information about the entity that is static and not normally included on transactions.
- **Profile File:** A profile file is a file that contains profile information that is known about a customer.
- **Simple Event:** A Simple Event is a component in Detect that is used to interface with the transaction stream. A Simple component, (often referred to itself as a Simple Event), fires an event when the transaction it encounters meets the criteria set in its conditions.
- **State history:** State history is the essence of Detect's speed. It is used to hold the partial state of the event detection pattern as the entity is interacting with your enterprise over time.
- **Transaction Feed File:** An input file to Detect. Transactions may either come from an operations system or consist of an extract from a data warehouse.
- **Transactions:** Transactions are the input to the detection engine. Each transaction represents some completed action of the customer (such as purchase, deposit, or account activation) or some discrete fact about the customer (such as periodic balance, overdraft, credit score, or profitability index).

- **Trigger:** A trigger is something actionable that was detected, which might also be referred to as a detected event or triggered event. Physically, it is just a row in a table indicating that a particular behavior was detected about a particular entity (customer or account) at this point in time. The term can also be used generically to mean a rule that causes an event to be inserted in the outcome table. The results of triggers are typically sent to internal institutional personnel.
- **Trigger system:** A trigger system is an event detection algorithm. It is comprised of all the components in Detect required to generate a particular trigger or outcome. A trigger system operates as a process occurring over time that generates various intermediate events and finally culminates in a trigger.
- **XML field (Additional outcome):** The Message field on the Detect outcome record that contains information which was accumulated during the detection process.