

Clairol 2000 Custom Solution Database Integration API Functional Specification

Clayton Weimer
David Seeger

3/23/98

Revision 3

Approved By: _____

Date: _____

This is a preliminary document and may be changed prior to final commercial release. This document is provided for informational purposes only, and MobilePoint makes no warranties, either express or implied, in this document.

Copyright © 1998 MobilePoint Corporation. All rights reserved.

Summary of Revisions

Revision 0	First edition.
Revision 1	3/5 meeting with Clay, Dave, Roger, Tanya, Fred. Change some terminology. Change how multiple-entry fields are handled. Change how unresolved links (relationships) are handled.
Revision 2	3/18 and 3/19 meetings with Clay, Dave, Fred, Nancy, Sylvestre, Tom, and Kevin. Change name and purpose of this document. Change how multiple-entry fields are handled. Change how unresolved links (relationships) are handled.
Revision 3	3/20 comments from Dave and Kevin and 3/23 comments from Fred. Added examples and made them more similar to Kevin's examples. Changed "multiple-entry" field to "list" field. Dropped references to "new" and "add" if data is received by the enterprise with no Enterprise Key: This data could be new to this tablet, but not new to the enterprise, so the Enterprise Key associated with this new Tablet Key could conceivably be an existing Enterprise Key and not a newly-generated one (for example, if one tablet user crossed into another tablet user's territory).

Table of Contents

I. Purpose/Audience 4

II. Database Integration 4

A. Transaction Files (.TRN) 4

1. Transaction Identification Fields 5
2. Business Data Fields 5
3. Transactions From the Enterprise Data Server to the HQ Server 6
4. Relationship (Key) Fields 7
5. List Fields 7
6. From the HQ Server to the Enterprise Data Server 8

B. Key Association Table Files (.KAT) 8

III. Customer Requirements 9

I. Purpose/Audience

The purpose of this document is to specify how MobilePoint and the Clairol System Integrator will exchange information between the tablet computers running e-case and the enterprise computers that hold the corporate customer and product database. The audiences for this document are MobilePoint and the Clairol System Integrator (one or more people who perform system administration, database administration, and other information management functions).

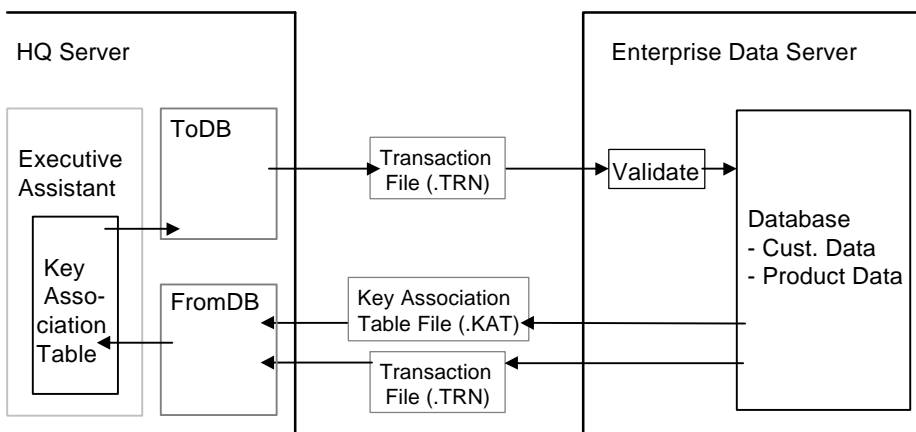
II. Database Integration

This section describes the data transaction files and the key association files that must be used to exchange data between the tablet databases and the enterprise database.

Database information must be exchanged using a transaction file (.TRN). See the following section.

If the tablet sends a transaction to the enterprise that does not have an Enterprise Key, then the enterprise data server must return a message to MobilePoint with the Enterprise Key for that database record. This message is a key association table (.KAT) file; see page 8.

MobilePoint keeps a database of all keys, pairing each Tablet Key with an Enterprise Key. There is no need for the enterprise data server to keep a copy of this key association file or to keep any record of Tablet Key/Enterprise Key pairs.



A. Transaction Files (.TRN)

A transaction file is a tab-delimited text file of database information.

- New lines separate the rows of the file.
- Tabs separate the columns (fields) of a transaction.
- A transaction consists of a set of transaction identification fields, usually followed by business data fields of a database record (only the delete transaction and the make inactive transaction have no data in the business data fields). A single transaction can span multiple rows (see “5. List Fields” on page 7). In a multiple-row transaction, only the first row contains data in the transaction identification fields. A single transaction includes all the data between two Transaction Type characters (see “Table 1: Transaction Identification Fields”).
- Transaction files have exactly the same format whether they are coming from the tablet or from the enterprise.
- Transactions for different kinds of records can be sent in the same transaction file.
- Transactions can be sorted by any of the transaction identification fields.
- Transactions are atomic and are not grouped. Transactions are not grouped by logical unit of work or by any other criteria.

The name of each transaction file has the form *UserID_TimeStamp*.TRN, where:

UserID is a unique three-character string that identifies each tablet user and therefore each tablet.

TimeStamp is a string of the form *YYYYMMDD_HHMMSS* that identifies the exact time the file was created. This helps to ensure that transactions are addressed in the correct order so that older changes do not overwrite newer changes.

The following is an example of a valid transaction file name: STEPH_19980209_084527 .TRN

1. Transaction Identification Fields

Each transaction begins with a set of six transaction identification fields: fields that identify this record in the database.

The following six transaction identification fields are mandatory for all transactions:

Table 1: Transaction Identification Fields		
Transaction Type	1A	This is a one-character string that tells the database what to do with the business data in this transaction. It can have only two values: “M” for modify (update or add) or “D” for delete. Every transaction, whether from the tablet or from the enterprise, has a Transaction Type. A single transaction includes all the data between two Transaction Type characters.
Tablet Key	32A	This key uniquely identifies a record in any of the tablet databases. If the same record is stored on more than one tablet (such as a product or distributor record), then that record has a different Tablet Key on each tablet. Transaction files coming from the HQ Server to the enterprise have a Tablet Key for every transaction. Transaction files coming from the enterprise to the HQ Server do not have any Tablet Keys. MobilePoint matches the Enterprise Key to the Tablet Key using its Key Association Table (see page 8).
Enterprise Key	64A	This is the key the enterprise uses to uniquely identify a record in its database. Transaction files coming from the enterprise to the HQ Server have an Enterprise Key for every transaction. Transaction files coming from the HQ Server to the enterprise do not have Enterprise Keys for transactions that have new Tablet Keys. If the Tablet Key is found in MobilePoint’s Key Association Table (see page 8), then the matching Enterprise Key is sent with that transaction along with the Tablet Key.
Category	64A	The category uniquely identifies a record type in the enterprise database. Every transaction, whether from the tablet or from the enterprise, has a Category.
Modification Time Stamp	14A	The time this record of business data was last updated. This value is a string of the form: YYYYMMDDHHMMSS. Every transaction, whether from the tablet or from the enterprise, has a Modification Time Stamp.
Inactive	1A	This character identifies this record as either active or inactive. It can be one of two values: a null value for an active record, or a “1” for an inactive record. An inactive record cannot be viewed or manipulated by the tablet user through the standard tablet database interface, but that data still exists in the tablet database.

2. Business Data Fields

These fields must be fully specified in a separate document.

Business Data Fields are defined by the order specified in the above-referenced documentation. Null values indicate fields that have not changed or that are not being added at this time.

A transaction does not need to be padded with tabs after the last modified field; only fields up to the last field for which data is being sent need to be accounted for (with a null value or data). Therefore, a Delete transaction does not need to contain any fields other than the transaction identification fields. Similarly, a transaction with the Inactive field set does not need to contain any fields other than the transaction identification fields.

Clearing a Field in a Record

To clear the data in a field in a record, send only a single tilde (~) character in that field in the transaction.

3. Transactions From the Enterprise Data Server to the HQ Server

Example 1 below is an example of a transaction file being transmitted from an enterprise data server to the HQ Server. Note that none of the transactions has a value in the **Tablet Key** field, but every transaction has a value in the **Enterprise Key** field. Similarly, the Contact transaction has no value in the **Unresolved Reference** field, but has a value in the **Company Key** field.

The enterprise data server provides its Enterprise Key for each record in the enterprise database. MobilePoint tries to pair this Enterprise Key with a Tablet Key for this database record on the target tablet. If this is a modification or deletion of an existing record, then a Tablet Key for this record on the specified tablet already exists in MobilePoint's Key Association Table. If this is a new database record for this tablet, then MobilePoint assigns a new Tablet Key for this record.

This example is shown as a table, rather than in the correct tab-delimited format, to make it easier to read the information in each column. The two rows of bold column headings are *not* a part of the transaction file; they are provided only to make the example more readable. The first six fields are the transaction identification fields. All remaining fields are business data fields; real records have many more fields than are shown in this example.

Example 1								
Transaction File: From the Enterprise Data Server to the HQ Server								
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Name	Address	Telephone Number
M		EK1	Company	19980209012235			123 Main Street	~
D		EK2	Company	19980209012241				
M		EK3	Company	19980209012307	1			
M		EK4	Company	19980209012715		New Ventures	678 First Street	408-123-4000
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Name	Unresolved Reference	Company Key
M		EK5	Contact	19980209012738		Claire Roll		EK4

In the above example:

- The first transaction is an update to an existing enterprise database record. Only two business data fields have data in them. The street address is being changed, and the telephone number is being cleared.
- The second transaction has no data or tabs in the business data fields because it is a Delete transaction.
- The third transaction has no data or tabs in the business data fields because the only modification being done is the record is being changed to an inactive record.
- The fourth transaction is an add transaction. All the business data fields have data in them.
- The fifth transaction is adding a Contact person to the Company just added. The Contact's **Company Key** is the same as the **Enterprise Key** of the Company that this person is a Contact for. The **Unresolved Reference** field is null because

this new record is coming from the enterprise database, so the reference is already an enterprise key and is not unresolved. See the following section “4. Relationship (Key) Fields” for more information.

4. Relationship (Key) Fields

Any business data field that is a key (that relates this record to another database record) is represented in the transaction file as a *pair* of fields: an **Unresolved Reference** field immediately followed by a key field. A null value in the **Unresolved Reference** field indicates this is *not* an unresolved reference, and a “Y” in the **Unresolved Reference** field indicates this *is* an unresolved reference. The key field contains an **Enterprise Key** if this is *not* an unresolved reference, or a **Tablet Key** if this *is* an unresolved reference.

The **Unresolved Reference** field in this pair of fields is always null for transactions coming from the enterprise, and it is often null for transactions coming from the tablet. The only time the **Unresolved Reference** field in this pair of fields contains a “Y” is when it is in a transaction that is adding a new record from the tablet database. A new tablet record is defined to be a record whose Tablet Key is not yet listed in MobilePoint’s Key Association Table, so there is no corresponding Enterprise Key to substitute for it. In this case, the enterprise must match the Tablet Key in the key field that immediately follows this “Y” to this same Tablet Key in the **Tablet Key** field of another record in the transaction file; this is the record being referenced. When an Enterprise Key is assigned to the record being referenced, that same Enterprise Key must be substituted for all instances of this Tablet Key in this transaction file. See the examples in the next section.

5. List Fields

A field of a database record can contain a list of values. For example, a Call can have many Contacts, as shown in the example below.

To modify or remove any single entry of a field that is allowed to have a list of entries, or to add a new entry to a field that is allowed to have a list of entries, all entries in the list must be transmitted, including entries that are not changing. Send one of the entries in the first row of the transaction, and send each additional entry in a separate row that contains only tabs and that particular entry (and a value in the associated Unresolved Reference field if necessary).

If you are *not modifying* any entry in a particular field that is allowed to have a list of entries, then *do not transmit any* of those entries. If you transmit one entry but not the others, you are saying you want the other entries deleted.

Sending any entry in a field that is allowed to have a list of entries causes the entire existing list to be overwritten with whatever is sent.

In the following example, a new Company record, two new Contact records, and a new Call record are coming from the tablet. The **Contact Key** field in the Call record is a list field:

Example 2								
Transaction File: From the HQ Server to the Enterprise Data Server with a List Field								
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Name	Unresolved Reference	Company Key
M	TK1		Company	19980209012738		Curls R Us		
M	TK2		Contact	19980209012738		Lucy Jones	Y	TK1
M	TK3		Contact	19980209012738		Roberta Black	Y	TK1
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Unresolved Reference	Contact Key	
M	TK4		Call	19980209012738		Y	TK2	
						Y	TK3	

As a result of the above Add transactions, the enterprise sent the following .KAT file (see page 8):

```
TK1    EK6
TK2    EK7
TK3    EK8
TK4    EK9
```

In Example 3 below, a new company is being added that is the parent company of Curls R Us (TK1/EK6). The Company record for Curls R Us is modified to add the new parent Company (TK5). No other fields in that record need to be sent since they are not changing in this transaction. The primary contact at the new company is an existing contact (Lucy Jones: TK2/EK7). Notice that the entire list of Company Keys for Ms. Jones must be resent, not just the new key. If only the new reference key was sent, the transaction would mean to replace the existing Company Key with this one new one.

Example 3								
Transaction File: From the HQ Server to the Enterprise Data Server with a Modified List Field								
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Name	Unresolved Reference	Company Key
M	TK5		Company	19980209012738		Parent Curls		
M	TK1	EK6	Company	19980209012738			Y	TK5
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Name	Unresolved Reference	Company Key
M	TK2	EK7	Contact	19980209012738				EK6
							Y	TK5

Example 4 shows how to handle a record that contains more than one list field that is being updated. In Example 4, a call is made on a known company, two known contacts are seen, and two new orders are taken:

Example 4											
Transaction File: Multiple List Fields											
Trans. Type	Tablet Key	Enterprise Key	Category	Modification Time Stamp	Inactive	Unres. Ref.	Company Key	Unres. Ref.	Contact Key	Unres. Ref.	Order Key
M	TK6		Call	19980209012738			EK6		EK7	Y	TK7
									EK8	Y	TK8

6. From the HQ Server to the Enterprise Data Server

When the transaction file is being transmitted from the HQ Server to the enterprise (rather than from an enterprise data server to the HQ Server), it looks almost exactly the same as the one shown in “3. Transactions From the Enterprise Data Server to the HQ Server” on page 6. The only difference is that the Tablet Key field is filled out and the Enterprise Key field may not be filled out. The Enterprise Key field is not filled in if this is a new Tablet Key; this Tablet Key is not yet in the Key Association Table, so it does not yet have an associated Enterprise Key. If the enterprise database assigns an Enterprise Key to this record, then the enterprise must return that Enterprise Key, along with the Tablet Key, in a .KAT file; see “B. Key Association Table Files (.KAT)” on page 8.

Also, an Enterprise Key within the business data fields of a database record (creating a relationship between two records) cannot be supplied if the association is to a new Tablet Key. In this case, the field contains the Tablet Key for the record being referenced so that the enterprise can locate the correct record to find the newly-assigned Enterprise Key.

For more examples of transactions coming from the tablet, see “5. List Fields” on page 7.

B. Key Association Table Files (.KAT)

If the enterprise data server receives a transaction that does not have an Enterprise Key (the transaction only has a Tablet Key), the enterprise must return to MobilePoint the Enterprise Key that was assigned to that record.

If a transaction from a tablet has a Tablet Key that the enterprise has never seen before, then MobilePoint cannot substitute an Enterprise Key for that Tablet Key before sending the transaction on to the enterprise. So the enterprise receives a transaction that has a Tablet Key but no Enterprise Key. If the enterprise adds this record and assigns an Enterprise Key to it, then the enterprise must return this Enterprise Key to MobilePoint, along with its original Tablet Key, so that MobilePoint can substitute the correct Enterprise Key the next time an update to this record comes from the tablet.

Whenever the enterprise receives a Tablet Key instead of an Enterprise Key, it must return that Tablet Key to MobilePoint, along with a corresponding Enterprise Key, so that MobilePoint can update its Key Association Table. The enterprise must send this key pair in the form of a key association table file (.KAT) that contains Tablet Key/Enterprise Key pairs. The Tablet Key is simply copied from the transaction file that was sent from the tablet. The Enterprise Key must identify this record uniquely in the enterprise database.

The enterprise data server puts the key association table file in the FromDB directory on HQ Server. There is no need for the enterprise data server to keep a copy of this key association table file or to keep any record of Tablet Key/Enterprise Key pairs. There is no need to send this information for any transactions except transactions the enterprise receives that do not have an Enterprise Key.

A key association table file is a tab-delimited text file of database key pairs.

- Tabs separate the two columns of the file.
- The Tablet Keys are in the first column; the Enterprise Keys are in the second column.
- New lines separate the rows of the file.

The name of each key association table file has the form: *UserID_TimeStamp.KAT*, where *UserID* and *TimeStamp* are defined the same as for transaction files, and *UserID* is the same *UserID* that was in the name of the transaction file that contained the add transaction for which this new Enterprise Key is being supplied.

Following is an example of a key association table file:

```
TK1    EK6
TK2    EK7
TK3    EK8
```

III. Customer Requirements

This section lists the requirements Clairol must meet in order to work with the API described in this document.

- Clairol must provide a unique User ID for each tablet user. This User ID is a 3A field and should not contain special characters (it will be used in filenames and directory names).
- Clairol must provide documentation of the records that are to be shared between the enterprise database and the tablet database. For each of these records, Clairol must provide a programmer's description of each field that is to be shared between the two databases and specify the order of these fields in the transaction.
- Clairol must provide a Category name for each table in the database. The Category for each table must uniquely identify that table in the enterprise database. This name must fit a 64A field.
- Clairol must provide an Enterprise Key for each record in the enterprise database. The Enterprise Key for each record must uniquely identify that record in the enterprise database. This key must fit a 64A field.
- If new data is sent from the tablet to the enterprise, the enterprise must return both the Enterprise Key for that record and the Tablet Key that was sent along with the new data so that MobilePoint can maintain the integrity of the correspondence between the tablet databases and the enterprise database.
- Clairol must perform any data validation or cleansing that is required. MobilePoint does not perform any data checking. MobilePoint simply passes the data between the tablets and the enterprise.