

# **Expense Reporting EFT – Easy as ABC**

**Technical Assistance Documentation**

**Atlanta OAUG**

**May 21, 1999**

**Presented By: Allen Webb, The Kaylor Group, Inc.**

**Table of Contents**

Sample EFT PL/SQL Script.....	3
Sample Remittance Alert SQL Script .....	19



```

        fh_reference_code      varchar2(8)      := 'AP EFT';

/*****/
/*****/
/*****/
        ch_record_type_code    varchar2(1)      := '5';
        ch_service_class_code  varchar2(3)      := '200';
        ch_company_name        varchar2(18)     := 'SAMPLE CORPORATION';
        ch_company_disc        varchar2(20)     := 'AP EXPENSE';
        ch_company_id          varchar2(10)     := null;
        ch_entry_class         varchar2(3)      := 'PPD';
        ch_company_entry_desc  varchar2(10)     := 'AP EXPENSE';
        ch_company_desc_date   varchar(6)       := null;
        ch_company_entry_date  varchar(6)       := null;
        ch_settlement_date_jul varchar2(3)      := ' ';
        ch_originator_status   varchar2(1)      := '1';
        ch_originating_dfi_id  varchar2(8)      := null;
        ch_batch_number        varchar2(7)      := '0000001';

/*****/
/*****/
/*****/
        de_record_type_code    varchar2(1)      := '6';
        de_transaction_code    varchar2(2)      := null;
        de_individual_bank_id  varchar2(8)      := null;
        de_check_digit         varchar2(1)      := null;
        de_bank_acct_number    varchar2(17)     := null;
        de_amount              varchar2(10)     := null;
        de_individual_id       varchar2(15)     := null;
        de_individual_name     varchar2(22)     := null;
        de_bank_disc           varchar2(2)      := ' ';
        de_addenda_rec_ind     varchar2(1)      := '0';
        de_trace_number        varchar2(15)     := null;

/*****/
/*****/
/*****/
        cc_record_type_code    varchar2(1)      := '8';
        cc_service_class       varchar2(3)      := '200';
        cc_entry_count         varchar2(6)      := null;
        cc_entry_hash          varchar2(10)     := null;
        cc_total_debits        varchar2(12)     := null;
        cc_total_credits       varchar2(12)     := null;
        cc_company_id          varchar2(10)     := null;
        cc_filler1             varchar2(19)     := ' ';
        cc_filler2             varchar2(6)      := ' ';
        cc_originating_fin     varchar2(8)      := null;
        cc_batch_number        varchar2(7)      := '0000001';

/*****/
/*****/
/*****/
        fc_record_type_code    varchar2(1)      := '9';
        fc_batch_count         varchar2(6)      := '000001';
        fc_block_count         varchar2(6)      := null;

```

```
fc_entry_hash          varchar2(10)  := null;
fc_total_debits        varchar2(12)  := null;
fc_total_credits       varchar2(12)  := null;
fc_reserved            varchar2(39)  := ' ';
```

```
/***/
```

```
l_company_identifier   varchar2(13)  := 'CO IDENTIFIER';
l_amount_paid          number          := null;
l_checkrun_name        varchar2(30)  := '&1';
dummy_date             date;
```

```
f_check_id             number(15)    := null;
f_vendor_id            number(15)    := null;
f_vendor_site_id      number(15)    := null;
```

```
l_vendor_num           varchar2(30)  := null;
l_vendor_name          varchar2(80)  := null;
l_check_amount         number        := null;
l_check_date           varchar2(6)   := null;
l_bank_account_num     varchar2(30)  := null;
l_bank_account_type    varchar2(25)  := null;
l_bank_num             varchar2(25)  := null;
```

```
l_nb_account_num       varchar2(30)  := null;
l_nb_transit_num       varchar2(30)  := null;
l_transmission_date    varchar2(6)   := null;
l_transmission_time    varchar2(4)   := null;
```

```
l_validate_count      number        := null;
l_rec_count            number := 0;
l_rec_seq              number := 0;
l_tot_rec_count        number        := 0;
l_entry_count          number        := 0;
l_entry_hash           number        := 0;
l_total_debits         number        := 0;
l_total_credits        number        := 0;
```

```
vin                    varchar2(8)   := null;
vout                   varchar2(8)   := null;
```

```
p_procedure_name       varchar2(50)  := null;
p_debug_text           varchar2(300) := null;
p_debug_ref            varchar2(80) := null;
p_debug_code           varchar2(10)  := null;
p_debug_mesg          varchar2(50) := null;
p_debug_mode           varchar2(1)   := ";
```

```
error_record           exception;
```

```
/***/
```

```
/***/ CURSOR DECLARATIONS ***/
```

```
/***/
```

```
/***/
```

```

/*****
cursor chk_curs is
select t1.selected_check_id,
       t1.vendor_id,
       t1.vendor_site_id,
       t1.vendor_num,
       t1.check_amount,
       t1.vendor_name,
       t1.bank_account_num,
       t1.bank_account_type,
       t1.bank_num /*TRANSIT NUM*/
from ap_selected_invoice_checks t1
where t1.checkrun_name = l_checkrun_name
and t1.ok_to_pay_flag = 'Y'
order by t1.check_number;

chk_rec chk_curs%rowtype;

/*****
/***** PROCEDURE DECLARATIONS *****/
/*****

/*****
/* Abort Processing of the Record */
/*****

procedure abort_record is
begin
  raise error_record;
end abort_record;

/*****
/* Display Debug Messages */
/*****

procedure print_debug is
begin

  dbms_output.put_line('*** ERROR ****');
  dbms_output.put_line('Debug Proc: '||p_procedure_name);
  dbms_output.put_line('Debug Rec: '||to_char(f_check_id));
  dbms_output.put_line('Debug Code: '||p_debug_code);
  dbms_output.put_line('Debug Text: '||p_debug_text);
  dbms_output.put_line('Debug Ref: '||p_debug_ref);

  abort_record;

end print_debug;

/*****
/* Debug Process Flow */
/*****

procedure display_process is
begin
  if p_debug_mode = 'D' then
    dbms_output.put_line('Debug Process Flow: '||p_procedure_name);
  end if;
end display_process;

```

```

end display_process;

/*****
/* Reset transaction Variables */
*****/
procedure reset_vars is
begin
  p_procedure_name := 'BEG reset_vars';
  display_process;
  p_debug_text      := null;
  p_debug_ref       := null;
  p_debug_code      := null;
  p_debug_mesg      := null;

  l_rec_count := 0;
  l_rec_seq   := 0;

  p_procedure_name := 'END reset_vars';
  display_process;
end reset_vars;

/*****
/* Validate Bank Information */
*****/
procedure validate_bank_info is
begin
  p_procedure_name := 'BEG validate_bank_info';
  display_process;

  if length(l_nb_transit_num) <> 9
  then
    p_debug_code := 'XXX00015';
    p_debug_text := 'Nations Bank Transit# <> 9 characters in length';
    p_debug_ref  := l_nb_transit_num;
    print_debug;
  end if;

  if length(l_bank_num) <> 9
  then
    p_debug_code := 'XXX00016';
    p_debug_text := 'Employee Bank Transit# <> 9 characters in length';
    p_debug_ref  := l_bank_num;
    print_debug;
  end if;

  if l_bank_account_type not in ('S','C')
  then
    p_debug_code := 'XXX00017';
    p_debug_text := 'Employee Bank Account Type <> Savings or Checking';
    p_debug_ref  := l_bank_num;
    print_debug;
  end if;

  p_procedure_name := 'END validate_bank_info';
  display_process;

```

```

/*****
/* Calculate Check Digit
*****/

procedure calculate_check_digit
(value_in      in   varchar2,
 value_out     out  varchar2)
is

    i            integer := 0;
    total_weight integer := 0;
    digit        integer := null;
    next_multiple integer := 0;

begin

for i in 1..8
loop
    digit := substr(value_in,i,1);

    if i in (1,4,7)
    then
        total_weight := total_weight + digit * 3;
    elsif i in (2,5,8)
    then
        total_weight := total_weight + digit * 7;
    elsif i in (3,6)
    then
        total_weight := total_weight + digit * 1;
    end if;

end loop;

    next_multiple := round(total_weight,-1);

    if next_multiple < total_weight
    then
        next_multiple := next_multiple + 10;
    end if;

    value_out := ltrim(rtrim(to_char(next_multiple - total_weight)));

end calculate_check_digit;

/*****
/* Validate Payment Batch Status
*****/

procedure validate_payment_batch is
begin
    p_procedure_name := 'BEG validate_payment_batch';
    display_process;

    select count(t1.rowid)

```

```

from ap_invoice_selection_criteria t1,
     ap_check_stocks t2
where t1.payment_method_lookup_code = 'EFT'
     and t1.check_stock_id = t2.check_stock_id
     and t2.name = 'XXX: EFT'
     and t2.inactive_date is null
     and t1.status in ('BUILT','FORMATTING')
     and t1.checkrun_name = l_checkrun_name;

if l_validate_count = 0
then
  p_debug_code := 'ERR00005';
  p_debug_text := 'Payment Batch Selected not available for processing';
  p_debug_ref := l_checkrun_name;
  print_debug;
end if;

p_procedure_name := 'END validate_payment_batch';
display_process;
end validate_payment_batch;

/*****
/* Get Check Date and Bank Account */
*****/

procedure get_check_date_bank_acct is
begin
  p_procedure_name := 'BEG get_check_date_bank_acct';
  display_process;

  select to_char(t1.check_date,'YYMMDD') check_date,
         to_char(sysdate,'YYMMDD') transmission_date,
         to_char(sysdate,'HH24MM') transmission_time,
         t2.bank_account_num          nb_account_num,
         t3.bank_num                  nb_transit_num
  into l_check_date,
       l_transmission_date,
       l_transmission_time,
       l_nb_account_num,
       l_nb_transit_num
  from ap_invoice_selection_criteria t1,
       ap_bank_accounts_all         t2,
       ap_bank_branches            t3
  where t1.checkrun_name = l_checkrun_name
       and t2.bank_branch_id = t3.bank_branch_id
       and t1.bank_account_name = t2.bank_account_name;

```

```

p_procedure_name := 'END get_check_date_bank_acct';
display_process;
exception
  when no_data_found then
    p_debug_code := 'ERR00003';
    p_debug_text := 'Error getting Check Date and Bank Account';
    p_debug_ref := to_char(f_vendor_id);
    print_debug;
  when error_record then
    raise;
  when others then
    p_debug_code := to_char(sqlcode);
    p_debug_text := substr(sqlerrm,1,50);
    print_debug;

end get_check_date_bank_acct;

```

```

/*****
/* Insert File Header Record */
*****/
procedure process_fh_rec is
begin
  p_procedure_name := 'BEG process_fh_rec';
  display_process;

  l_rec_seq := 0;
  l_tot_rec_count := l_tot_rec_count + 1;

  fh_immediate_destination := concat(' ',l_nb_transit_num);
  fh_immediate_origin := ' '||substr(fh_immediate_origin,2,9);
  fh_transmission_date := l_transmission_date;
  fh_transmission_time := l_transmission_time;

  select to_char(XXXap_nacha_file_id_s.nextval)
  into fh_file_id_modifier
  from dual;

  if fh_file_id_modifier is null
  then
    fh_file_id_modifier := '0';
  end if;

```

```

insert into XXXap_nacha_interface
(
  checkrun_name,
  record_type_code,
  record_sequence,
  record_layout
)
values
(
  l_checkrun_name,
  fh_record_type_code,
  l_rec_seq,
  fh_record_type_code           ||
  fh_priority_code             ||
  fh_immediate_destination     ||
  fh_immediate_origin          ||
  fh_transmission_date         ||
  fh_transmission_time         ||
  fh_file_id_modifier          ||
  fh_record_size               ||
  fh_blocking_factor           ||
  fh_format_code               ||
  rpad(fh_destination,23, ' ') ||
  rpad(fh_origin,23, ' ')      ||
  rpad(fh_reference_code,8, ' ')
);

p_procedure_name := 'END process_fh_rec';
display_process;
end process_fh_rec;

/*****
/* Insert Company Header Record */
*****/
procedure process_ch_rec is
begin
  p_procedure_name := 'BEG process_ch_rec';
  display_process;

  l_rec_seq := 0;
  l_tot_rec_count := l_tot_rec_count + 1;

  ch_company_id      := l_company_identifier;
  ch_company_desc_date := l_check_date;
  ch_company_entry_date := l_check_date;
  ch_originating_dfi_id := substr(l_nb_transit_num,1,8);

```

```

insert into XXXap_nacha_interface
(
  checkrun_name,
  record_type_code,
  record_sequence,
  record_layout
)
values
(
  l_checkrun_name,
  ch_record_type_code,
  l_rec_count,
  ch_record_type_code           ||
  ch_service_class_code        ||
  rpad(ch_company_name,16,' ') ||
  rpad(ch_company_disc,20,' ') ||
  ch_company_id                ||
  ch_entry_class                ||
  rpad(ch_company_entry_desc,10,' ') ||
  ch_company_desc_date         ||
  ch_company_entry_date        ||
  ch_settlement_date_jul       ||
  ch_originator_status         ||
  ch_originating_dfi_id        ||
  ch_batch_number
);

p_procedure_name := 'END process_ch_rec';
display_process;
end process_ch_rec;

/*****
/* Insert Detail Entry Record */
*****/

procedure process_de_rec is
begin
  p_procedure_name := 'BEG process_de_rec';
  display_process;

  l_rec_count := l_rec_count + 1;
  l_tot_rec_count := l_tot_rec_count + 1;

  if l_check_amount >= 0
  then
    if l_bank_account_type = 'Savings'
    then
      de_transaction_code := '32';
    else
      de_transaction_code := '22';
    end if;
  end if;
end if;

```

```

if l_check_amount < 0
then
  if l_bank_account_type = 'Savings'
  then
    de_transaction_code := '37';
  else
    de_transaction_code := '27';
  end if;
end if;

l_entry_count := l_entry_count + l_rec_count;
l_entry_hash := l_entry_hash + to_number(substr(l_bank_num,1,8));

if l_check_amount < 0
then
  l_total_debits := l_total_debits + l_check_amount;
else
  l_total_credits := l_total_credits + abs(l_check_amount);
end if;

de_individual_bank_id := substr(l_bank_num,1,8);

/* not being calculated - assume Payroll has correct check digit entered */
/* calculate_check_digit(substr(l_bank_num,1,8),vout); */
/* de_check_digit := vout; */

de_check_digit := substr(l_bank_num,9,1);

de_bank_acct_number := substr(l_bank_account_num,1,17);
de_amount := to_char(abs(l_check_amount),'FM00000000V00');
de_individual_id := substr(l_vendor_num,1,15);
de_individual_name := substr(l_vendor_name,1,22);
de_trace_number := substr(l_nb_transit_num,1,8)||
to_char(l_rec_count,'FM0000000');

```

```

insert into XXXap_nacha_interface
(
  checkrun_name,
  record_type_code,
  record_sequence,
  record_layout
)
values
(
  l_checkrun_name,
  de_record_type_code,
  l_rec_count,
  de_record_type_code           ||
  de_transaction_code          ||
  de_individual_bank_id        ||
  de_check_digit               ||
  rpad(de_bank_acct_number,17,' ') ||
  de_amount                    ||
  rpad(de_individual_id,15,' ')   ||
  rpad(de_individual_name,22,' ') ||
  de_bank_disc                  ||
  de_addenda_rec_ind           ||
  de_trace_number
);

p_procedure_name := 'END process_de_rec';
display_process;
end process_de_rec;

/*****
/* Insert Company Control Record */
*****/

procedure process_cc_rec is
begin
  p_procedure_name := 'BEG process_cc_rec';
  display_process;

  l_rec_seq := 0;
  l_tot_rec_count := l_tot_rec_count + 1;

  cc_entry_count := to_char(l_rec_count, 'FM000000');
  cc_total_credits := to_char(l_total_credits,'FM0000000000V00');
  cc_total_debits := to_char(l_total_debits, 'FM0000000000V00');
  cc_entry_hash := to_char(l_entry_hash, 'FM0000000000');
  cc_company_id := l_company_identifier;
  cc_originating_fin := substr(l_nb_transit_num,1,8);

```

```

insert into XXXap_nacha_interface
(
  checkrun_name,
  record_type_code,
  record_sequence,
  record_layout
)
values
(
  l_checkrun_name,
  cc_record_type_code,
  l_rec_seq,
  cc_record_type_code           ||
  cc_service_class             ||
  cc_entry_count               ||
  cc_entry_hash                ||
  cc_total_debits              ||
  cc_total_credits             ||
  cc_company_id                ||
  lpad(cc_filler1,19,' ')     ||
  lpad(cc_filler2,6,' ')      ||
  cc_originating_fin           ||
  cc_batch_number
);

p_procedure_name := 'END process_cc_rec';
display_process;
end process_cc_rec;

/*****
/* Insert File Control Record */
*****/

procedure process_fc_rec is
begin
  p_procedure_name := 'BEG process_fc_rec';
  display_process;

  l_rec_seq := 0;
  l_tot_rec_count := l_tot_rec_count + 1;

  fc_entry_count := to_char(l_rec_count, 'FM00000000');
  fc_total_credits := to_char(l_total_credits,'FM0000000000V00');
  fc_total_debits := to_char(l_total_debits, 'FM0000000000V00');
  fc_entry_hash := to_char(l_entry_hash, 'FM0000000000');
  fc_block_count := to_char(ceil(l_tot_rec_count / 10),'FM000000');

```

```

insert into XXXap_nacha_interface
(
  checkrun_name,
  record_type_code,
  record_sequence,
  record_layout
)
values
(
  l_checkrun_name,
  fc_record_type_code,
  l_rec_seq,
  fc_record_type_code           ||
  fc_batch_count               ||
  fc_block_count               ||
  fc_entry_count               ||
  fc_entry_hash                ||
  fc_total_debits              ||
  fc_total_credits             ||
  lpad(fc_reserved,39,' ')
);

p_procedure_name := 'END process_fc_rec';
display_process;
end process_fc_rec;

/*****
/* Process EFT Payments */
*****/

procedure process_eft is
begin
  p_procedure_name := 'BEG process_eft';
  display_process;

  open chk_curs;
  loop
    fetch chk_curs into chk_rec;
    exit when chk_curs%notfound;
    begin

      f_check_id           :=      chk_rec.selected_check_id;
      f_vendor_id         :=      chk_rec.vendor_id;
      f_vendor_site_id    :=      chk_rec.vendor_site_id;

      l_vendor_num        :=      chk_rec.vendor_num;
      l_vendor_name       :=      chk_rec.vendor_name;
      l_check_amount      :=      chk_rec.check_amount;
      l_bank_account_num  :=      chk_rec.bank_account_num;
      l_bank_account_type :=      chk_rec.bank_account_type;
      l_bank_num          :=      chk_rec.bank_num;
    end;
  end loop;
end;

```

```

/*
dbms_output.put_line('NB Transit#: '||l_nb_transit_num);
dbms_output.put_line('NB Account#: '||l_nb_account_num);
dbms_output.put_line('EFT Date: '||l_check_date);
dbms_output.put_line('EFT Amount: '||to_char(l_check_amount));
dbms_output.put_line('Bank Transit#: '||l_bank_num);
dbms_output.put_line('Bank Account#: '||l_bank_account_num);
dbms_output.put_line('Bank AcctType: '||l_bank_account_type);
dbms_output.put_line('Vendor Number: '||l_vendor_num);
dbms_output.put_line('Vendor Name: '||l_vendor_name);
*/

validate_bank_info;
process_de_rec;

exception
when error_record then
    raise;
when others then
    p_debug_code := to_char(sqlcode);
    p_debug_text := substr(sqlerrm,1,50);
    print_debug;
end;
end loop;
close chk_curs;

p_procedure_name := 'END process_eft';
display_process;
end process_eft;

/*****
/***** MAIN PROCESSING BLOCK *****/
*****/
begin
dbms_output.enable(1000000);

reset_vars;
validate_payment_batch;
get_check_date_bank_acct;

process_fh_rec;
    process_ch_rec;
        process_eft;
    process_cc_rec;
process_fc_rec;

```

```
exception
when error_record then
  dbms_output.put_line('*****');
  dbms_output.put_line('*** ERROR: Process is Terminating ***');
  dbms_output.put_line('*****');
  dbms_output.put_line('*** Refer to previous Debug Text to resolve ***');
  dbms_output.put_line('*** the error. ***');
  dbms_output.put_line('*** Ignore next message - used to force the ***');
  dbms_output.put_line('*** termination of the SQL Process ***');
  dbms_output.put_line('*****');
  select to_date('abcde') into dummy_date from dual;
when others then
  p_debug_code := to_char(sqlcode);
  p_debug_text := substr(sqlerrm,1,50);
  print_debug;
end;
/
commit;
exit;
```

## Sample Remittance Alert SQL Script

```
select
  t2.bank_account_name      bank_account_name,
  t1.bank_account_num      bank_account_num,
  t1.amount                 deposit_amount,
  t1.check_date             deposit_date,
  t7.email_address          email_address,
  t1.vendor_name            employee_name,
  t3.amount                 ilamt,
  t3.expenditure_item_date  ildate,
  substr(t3.description,1,35)  ildesc,
  'XXX Remmitance Advice' remmitance_title
into
  &bank_account_name,
  &bank_account_num,
  &deposit_amount,
  &deposit_date,
  &email_address,
  &employee_name,
  &ilamt,
  &ildate,
  &ildesc,
  &remmitance_title
from
  ap_checks_all              t1,
  po_vendor_sites_all        t2,
  ap_invoice_distributions_all t3,
  ap_invoice_payments         t4,
  po_vendors                  t6,
  hr_employees                t7
where
  t1.payment_method_lookup_code = 'EFT'
and t1.status_lookup_code = 'NEGOTIABLE'
and t1.check_id = t4.check_id
and t3.invoice_id = t4.invoice_id
and t6.employee_id = t7.employee_id
and t1.vendor_site_id = t2.vendor_site_id
and t6.vendor_id = t2.vendor_id
and translate(t7.email_address,',';*) not like '%*%'
and t7.email_address is not null
and t1.creation_date >
  to_date(:DATE_LAST_CHECKED,'DD-MON-YYYY HH24:MI:SS')
order by
  t2.bank_account_name,
  t1.bank_account_num,
  t1.amount,
  t1.check_date,
  t7.email_address ,
  t1.vendor_name,
  t3.expenditure_item_date ,
  t3.amount ,
```