

# Dancik Open-WebService Reference

## Function ID/Path : /ar/getARLedger

This function provides a list of ledger data.

### Inputs :

Variable	Type	Req?	Description
start_row	decimal(9,0)	Y	Start at Row
rows_per_screen	decimal(9,0)	Y	How Many Records to Display
parm_Company	string(1)	Y	Company
parm_Account	string(6)	Y	Company and Account
parm_Branch	string(3)	Y	Branch
parm_Invoice	string(6)	Y	Invoice
parm_Order	string(6)	Y	Order
parm_Check	string(14)	Y	Check Number
parm_Custpo	string(14)	Y	Customer PO
parm_Invamt	string(14)	Y	Invoice Amount
parm_Chkamt	string(14)	Y	Check Amount
parm_Trancode	string(2)	Y	Transaction Code
parm_PostMonth	string(6)	Y	Post Month
parm_GLAccount	string(5)	Y	GL Account
parm_Paid	string(1)	Y	Show current paid items (Y/N)
parm_Aging	string(1)	Y	Aging Option
parm_Roll	string(1)	Y	Roll Credits(Y/N)
parm_Age	string(2)	Y	Aging Days (15,30,60,90)
parm_Adar	string(1)	Y	Include ADs (Y/N)
parm_SortField	string(25)	Y	Sort on column
parm_SortDirection	string(4)	Y	Sort (ASC/DESC)
parm_printFlag	string(1)	Y	Print Report (Y/N)
parm_Title	string(500)	Y	Report Title
parm_FileName	string(500)	Y	xls File Name

#### Example (XML) :

```
<dancik-api-request>
  <start_row> XXX </start_row>
  <rows_per_screen> XXX </rows_per_screen>
  <parm_Company> XXX </parm_Company>
  <parm_Account> XXX </parm_Account>
  <parm_Branch> XXX </parm_Branch>
  <parm_Invoice> XXX </parm_Invoice>
  <parm_Order> XXX </parm_Order>
  <parm_Check> XXX </parm_Check>
  <parm_Custpo> XXX </parm_Custpo>
  <parm_Invamt> XXX </parm_Invamt>
  <parm_Chkamt> XXX </parm_Chkamt>
  <parm_Trancode> XXX </parm_Trancode>
  <parm_PostMonth> XXX </parm_PostMonth>
  <parm_GLAccount> XXX </parm_GLAccount>
  <parm_Paid> XXX </parm_Paid>
  <parm_Aging> XXX </parm_Aging>
  <parm_Roll> XXX </parm_Roll>
  <parm_Age> XXX </parm_Age>
  <parm_Adar> XXX </parm_Adar>
  <parm_SortField> XXX </parm_SortField>
  <parm_SortDirection> XXX </parm_SortDirection>
</dancik-api-request>
```

#### Example (JSON) :

```
{
  "start_row": " XXX ",
  "rows_per_screen": " XXX ",
  "parm_Company": " XXX ",
  "parm_Account": " XXX ",
  "parm_Branch": " XXX ",
  "parm_Invoice": " XXX ",
  "parm_Order": " XXX ",
  "parm_Check": " XXX ",
  "parm_Custpo": " XXX ",
  "parm_Invamt": " XXX ",
  "parm_Chkamt": " XXX ",
  "parm_Trancode": " XXX ",
  "parm_PostMonth": " XXX ",
  "parm_GLAccount": " XXX ",
  "parm_Paid": " XXX ",
  "parm_Aging": " XXX ",
  "parm_Roll": " XXX ",
  "parm_Age": " XXX ",
  "parm_Adar": " XXX ",
  "parm_SortField": " XXX ",
  "parm_SortDirection": " XXX ",
}
```

# Dancik Open-WebService Reference

## Function ID/Path : /ar/getARLedger

This function provides a list of ledger data.

```

<parm_printFlag> XXX </parm_printFlag>
<parm_Title> XXX </parm_Title>
<parm_FileName> XXX </parm_FileName>
</dancik-api-request>

```

```

"parm_printFlag" : " XXX " ,
"parm_Title" : " XXX " ,
"parm_FileName" : " XXX "
}

```

## Outputs :

Layout name : records

Variable	Type	Description
INVOICE		
branch		
invdatesrt		
invdate		
ARCO		
ARACCT		
ordernum		
invamt		
pdtran		
pdamnt		
pdamnt\$		
running_balance		
discount		
discount\$		
bank_code		
check		
paydate		
paydatesrt		
custpo		
dispute		
ivtran		
invdate_days		
ARPAYD		
duedate_days		
due_date		
sort_due_date		
terms_code		
termdays		

Layout name : aging

Variable	Type	Description
aging15		
aging30		

# Dancik Open-WebService Reference

## Function ID/Path : /ar/getARLedger

This function provides a list of ledger data.

Variable	Type	Description
aging45		
aging60		
aging90		
aging120		
agingInterest		
agingCurr		

### Example (XML) :

```
<dancik-api-response>
  <records>
    <INVOICE> XXX </INVOICE>
    <branch> XXX </branch>
    <invdatesrt> XXX </invdatesrt>
    <invdate> XXX </invdate>
    <ARCO> XXX </ARCO>
    <ARACCT> XXX </ARACCT>
    <ordernum> XXX </ordernum>
    <invamt> XXX </invamt>
    <pdtran> XXX </pdtran>
    <pdamnt> XXX </pdamnt>
    <pdamnt$> XXX </pdamnt$>
    <running_balance> XXX </running_balance>
    <discount> XXX </discount>
    <discount$> XXX </discount$>
    <bank_code> XXX </bank_code>
    <check> XXX </check>
    <paydate> XXX </paydate>
    <paydatesrt> XXX </paydatesrt>
    <custpo> XXX </custpo>
    <dispute> XXX </dispute>
    <ivtran> XXX </ivtran>
    <invdate_days> XXX </invdate_days>
    <ARPAYD> XXX </ARPAYD>
    <duedate_days> XXX </duedate_days>
    <due_date> XXX </due_date>
    <sort_due_date> XXX </sort_due_date>
    <terms_code> XXX </terms_code>
    <termdays> XXX </termdays>
  </records>
  <aging>
    <aging15> XXX </aging15>
    <aging30> XXX </aging30>
    <aging45> XXX </aging45>
    <aging60> XXX </aging60>
    <aging90> XXX </aging90>
    <aging120> XXX </aging120>
    <agingInterest> XXX </agingInterest>
    <agingCurr> XXX </agingCurr>
  </aging>
</dancik-api-response>
```

### Example (JSON) :

```
{
  "records" : {
    "INVOICE" : " XXX ",
    "branch" : " XXX ",
    "invdatesrt" : " XXX ",
    "invdate" : " XXX ",
    "ARCO" : " XXX ",
    "ARACCT" : " XXX ",
    "ordernum" : " XXX ",
    "invamt" : " XXX ",
    "pdtran" : " XXX ",
    "pdamnt" : " XXX ",
    "pdamnt$" : " XXX ",
    "running_balance" : " XXX ",
    "discount" : " XXX ",
    "discount$" : " XXX ",
    "bank_code" : " XXX ",
    "check" : " XXX ",
    "paydate" : " XXX ",
    "paydatesrt" : " XXX ",
    "custpo" : " XXX ",
    "dispute" : " XXX ",
    "ivtran" : " XXX ",
    "invdate_days" : " XXX ",
    "ARPAYD" : " XXX ",
    "duedate_days" : " XXX ",
    "due_date" : " XXX ",
    "sort_due_date" : " XXX ",
    "terms_code" : " XXX ",
    "termdays" : " XXX "
  }
  "aging" : {
    "aging15" : " XXX ",
    "aging30" : " XXX ",
    "aging45" : " XXX ",
    "aging60" : " XXX ",
    "aging90" : " XXX ",
    "aging120" : " XXX ",
    "agingInterest" : " XXX ",
    "agingCurr" : " XXX "
  }
}
```

### History :

Date	Description
12/30/2015	New Call