CALL TRANSACTION - Calling a Transaction

Basic form 4

```
CALL TRANSACTION tcod.
```

Additions:

1. `... AND SKIP FIRST SCREEN`
2. `... USING itab`

2a. `... OPTIONS FROM opt`
2b. `... MODE mode`
2c. `... UPDATE f`
2d. `... MESSAGES INTO itab`

In an ABAP Objects context, a more severe syntax check is performed that in other ABAP areas. See Incorrect transaction call.

Effect

Calls the transaction `tcod`; you can specify `tcod` either as a literal or a variable. The called transaction ends and control returns to the point from which it was called when the `LEAVE PROGRAM` statement is reached. You can only use one of the two additions in each statement.

Notes

- Please consult Data Area and Modularization Unit Organization documentation as well.

- You can use a namespace prefix with transactions.

Example

```
CALL TRANSACTION 'SP01'.
```

Addition 1

```
... AND SKIP FIRST SCREEN
```

Effect

Skips the initial screen of the transaction (as long as all of its required fields can be filled using the `SPA/GPA` mechanism.

Addition 2

```
... USING itab
```

Effect

Calls transaction **tcod** and passes the internal table **itab** to it. **itab** contains one or more screens in batch input format.
If the transaction sends a message, it is placed in the fields **SY-MSGID**, **SY-MSGTY**, **SY-MSGNO**, **SY-MSGV1**, ..., **SY-MSGV4**.

The return code is set as follows:

SY-SUBRC = 0:
Processing successful.
SY-SUBRC <> 0:
An error occurred and the transaction terminated.

Notes

- A called transaction ends successfully in the following circumstances:
  1. **COMMIT WORK**
  2. Next screen = 0
  3. **LEAVE TO TRANSACTION SPACE**
- You cannot use either **MODE** or **UPDATE** with the **OPTIONS FROM** addition.

Addition 2a

```
... OPTIONS FROM opt
```

Effect

Allows you to control processing using the values of the componetns of the structure **opt**, which must have the ABAP Dictionary type **CTU_PARAMS**. The components have the following meanings:

DISMODE
Display mode (like the **MODE** addition)

UPDMODE
Update mode (like the **UPDATE** addition)

CATTMODE
CATT mode (controls a CATT)
CATT mode can have the following values:
' ' No CATT active
'N' CATT without single-screen control
'A' CATT with single-screen control

DEFSIZE
Use default window size

RACOMMIT
Do not end transaction at **COMMIT WORK**

NOBINPT
No batch input mode (that is, **SY-BINPT = SPACE**)

NOBIEND
No batch input mode after the end of **BDC** data.

The components **DEFSIZE** , **RACOMMIT**, **NOBINPT**, and **NOBIEND** always take the following values:
'X' Yes
' ' No

If you do not use the **OPTIONS FROM** addition, the following control parameter settings apply:

DISMODE
From addition **MODE**
UPDMODE
From addition **UPDATE**
CATTMODE
No CATT active
DEFSIZE
Do not use default window size
RACOMMIT
Successful end on **COMMIT WORK**
NOBINPT
Batch input mode active (**SY-BINPT = X**
NOBIEND
Batch input mode remains active after the **BDC data**

Addition 2b

```
... MODE mode
```

Effect

The display mode **mode** can take the following values:
'A' Display the screens
'E' Only display if an error occurs
'N' Do not display
If you do not specify the **MODE** addition, the display mode is **'A'**.
If you called a transaction in display mode **E** and a screen appears because there is no more **BDC** data, the system automatically switches to display mode **A**.

Addition 2c

```
... UPDATE f
```

Effect

The specified update mode **f** determines the update type. It can take the following values:

`'A'` (asynchronous update)
`'S'` (synchronous update)
`'L'` (local update)
If you do not specify the **`UPDATE`** addition, the default update mode `'A'` is used.

Addition 2d

```
... MESSAGES INTO itab
```

Effect

Any messages that occur during **`CALL TRANSACTION USING ...`** are collected in the specified internal table. The internal table must have the structure **`BDCMSGCOLL`**.

Example

```
DATA: BDCDATA TYPE TABLE OF BDCDATA.

DATA: ITAB TYPE TABLE OF BDCMSGCOLL.

DATA: PROGRAM LIKE SY-REPID,

   WA_BDCDATA TYPE BDCDATA.

WA_BDCDATA-PROGRAM  = 'SAPMS38M'.
WA_BDCDATA-DYNPRO   = '0100'.
WA_BDCDATA-DYNBEGIN = 'X'.
APPEND WA_BDCDATA TO BDCDATA.
CLEAR WA_BDCDATA.
WA_BDCDATA-FNAM     = 'RS38M-PROGRAMM'.
WA_BDCDATA-FVAL     = PROGRAM.
APPEND WA_BDCDATA TO BDCDATA.
...
CALL TRANSACTION 'SE38'  USING BDCDATA  MODE 'N'
                         MESSAGES INTO ITAB.
```

Notes

Runtime errors:

- **`CALL_TRANSACTION_NOT_FOUND`**: Transaction unknown
- **`CALL_TRANSACTION_IS_MENU`**: Transaction is a menu
- **`CALL_TRANSACTION_USING_NESTED`**: Recursive **`CALL TRANSACTION USING`**
- **`CALL_TRANSACTION_LOCKED`**: Transaction is locked

Related

`SUBMIT`

CALL DIALOG

Additional help

Calling Transactions