

Unlock Subroutine

Description

Releases locked records.

Syntax

Unlock (*tablename,rowIDs,columnList,locktype*)

Parameters

The function has the following parameters:

Parameter	Description
<i>tablename</i>	The tablename containing the row to be unlocked.
<i>rowIDs</i>	An @fm delimited list of keys to unlock. If null the entire table is unlocked.
<i>columnList</i>	An @fm delimited list of columns to unlock. If the columnList is null the entire record is unlocked.
<i>lockType</i>	Returns the prior locktype of the record.

Remarks

Since the Unlock routine uses the Unlock statement, it is more efficient to use the Unlock statement in a BASIC+ script.

The Unlock routine uses Set_Status for error handling. In order to process errors use the [Get_Status\(\)](#) function.

See also

[UnLock statement](#), [Lock\(\)](#)

Example

```
/* Lock and unlock specific columns within a row */
```

```
declare subroutine lock, unlock
```

```
table = 'PRODUCTS'  
row = '3542-5-310-1'  
cols = 'DESCRIPTION':@fm:'PRICE'  
locktype = ""
```

```
Lock(table,row,cols,locktype)  
status = Get_Status(errCodes)  
If status Then  
    FMsg(errCodes)  
End
```

```
* Do Some Processing
```

```
call Unlock(table,row,cols,locktype)  
status = Get_Status(errCodes)  
If status Then  
    FMsg(errCodes)  
End
```

```
/* Lock and unlock a row */
```

```
declare subroutine lock, unlock
```

```
table = 'PRODUCTS'  
row = '3542-5-310-1'  
cols = ''  
locktype = ""
```

```
Lock(table,row,cols,locktype)  
status = Get_Status(errCodes)  
If status Then  
    FMsg(errCodes)  
End
```

```
* Do Some Processing
```

```
Unlock(table,row,cols,locktype)  
status = Get_Status(errCodes)  
If status Then  
    FMsg(errCodes)  
End
```