

# Application Frames

The following topics are covered below:

- Frame Overview
- Browse Dialog
- Deletion Subprogram
- Key Dialog
- Maintain Dialog
- Mass Processing Dialog
- Modal Window
- Nonstandard Dialog
- Subdialog
- Background Program
- Load Objects Subprogram
- Unload Objects Sub

## Frame Overview

Frame	Description
Browse dialog	Displays a list of objects. The maintain or mass processing dialog or the deletion subprogram can be invoked for selected objects.
Deletion subprogram	Deletes an object, displaying a message box to request confirmation of the deletion from the user. It requires an associated key dialog.
Key dialog	Used to provide open object, save as and active help windows. It allows the input of an object ID or the selection of an object from a list.
Maintain dialog	Allows the creation, display, and update of individual objects. It requires an associated key dialog.
Mass processing dialog	Allows the creation, display, update and deletion of multiple objects. This dialog does not use preliminary copies.
Modal window	Multi-purpose modal window. No default dialog layout/data access is produced. The dialog includes standard navigation and command handling logic.
Nonstandard dialog	Includes standard logic for navigation and command handling.
Subdialog	Used in association with a maintain dialog. (The link must be manually coded.)
Background program	This program is used to implement background programs which start from an online program using subprogram ZXBG010N.
Load objects subprogram	This subprogram is used to load objects of a specified type into the database from a workfile.
Unload objects subprogram	This subprogram is used to unload objects of a specified type from the database to a workfile.

For each dialog type except for modal windows, there is both a production and prototype variant of the program frame.

Production versions of each dialog type (except modal windows) include data access and a default dialog layout for the accessed data.

Prototype versions do not include data access or a default layout for the data accessed.

## Browse Dialog

### Description

Search and overview functions are provided by this frame.

As result of a search, data records are displayed in a list box. By selecting data records and choosing an action, the corresponding function is called for each of the selected data records.

This frame distinguishes between the mass processing action and all other actions.

For the mass processing action, the function is only called once for all selected data records. The mass processing and browse dialogs then communicate with one another using commands.

For all other actions, a separate function is started for each selected data record.

### Links with Other Dialogs

Called:	Directly through the command system
Calls:	Maintain dialog Deletion subprogram Mass Processing dialog

### Dialog Layout

Component	Comments
Search key	Up to 5 search key components. If the key includes more than 2 components, it is advisable to improve the layout using the dialog editor.
List box	Displays fields read using the multiple object access module.
Push buttons	Search

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

<b>Name</b>	<b>Prototype</b>	<b>Production</b>
Z_ACCESS_DATA	X	
Z_ASSIGN_INPUT_TO_KEY		
Z_CMD_EXEC_END		
Z_CMD_EXEC_START		
Z_CUSTOM_CMD		
Z_FILL_ITEM		X
Z_INITIALIZE		
Z_PASS_KEY		X
Z_PROCESS_ITEM		X
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		
Z_SET_KEY_RANGE		X
Z_UPDATE_ITEM		X

## Commands Supported

Command	Explanation	Components Activated
Z_LB_CLICK	Selection in list box changed	
Z_LB_FILL	Fill-event in list box occurred	Z_ACCESS_DATA Z_FILL_ITEM
Z_LB_SELECT	Process reaction on change of selection in list box	
Z_CANCEL_DLG	Mass processing gives notice of ending	
Z_CANCEL_KEY	Key dialog gives notice of ending	
Z_CANCEL_TMR	Cancel timer activated after runtime error	
Z_CLOSE	Close dialog	
Z_EXIT	End application Comments: A timer is activated that triggers the command Z_EXIT_TMR.	
Z_EXIT_TMR	Close dialog due to ending of application	
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_KEY	Key dialog sends selected key	Z_RECEIVE_KEY
Z_ITEM_ADD	Another dialog sends new data records for insertion in the list box	Z_UPDATE_ITEM Z_PASS_KEY
Z_ITEM_DEL	Another dialog sends keys for deletion from the list box	Z_UPDATE_ITEM Z_PASS_KEY
Z_ITEM_MOD	Another dialog sends data records for modification in the list box	Z_UPDATE_ITEM Z_PASS_KEY
Z_ITEM_NEXT	Mass processing expects key IDs of the next data record selected	Z_PASS_KEY Z_SELECT
Z_ITEM_PREV	Mass processing expects key IDs of the data record previously selected	Z_PASS_KEY Z_SELECT
Z_SELECT_ALL	Select all list box entries	
Z_SEARCH	Search is started anew Z_ACCESS_DATA Z_FILL_ITEM Z_PASS_KEY	Z_SET_KEY_RANGE
Z_START_KEY	Start with key	Z_INITIALIZE
Z_START_NKEY	Start without key	Z_INITIALIZE
Type of command Action	Function start for marked data records (1) Only when list box is empty (2 ) Only when list box is not empty	(1)Z_ASSIGN_INPUT_TO_KEY (2) Z_PASS_KEY (2) Z_SELECT

## Associated Variables

Name	Value	Comment
LZ_ALIGN_HORIZ	<b>True</b>	When changing the dialog size, the list box size is adjusted vertically.
	False	No vertical adjustment of the list box size.
LZ_ALIGN_VERT	<b>True</b>	When changing the dialog size, the list box size is adjusted horizontally.

Name	Value	Comment
	False	No horizontal adjustment of the list box size.
LZ_ADD_EVERY	True	Newly added data records are included in the list box, independent of whether they fit in the current range of records in the list box.
	<b>False</b>	Only newly added data records that fit in the current range of records in the list box are included in it.
LZ_ADD_ON_EMPTY	<b>True</b>	If the list box is empty, each newly added data record is included in the list box.
	False	Newly added data records are only included in an empty list box if there are no data (EOD).
LZ_BOX		Group consisting of LZ_BOX_ITEM and LZ_BOX_ITEM_KEY. The group can be used for MOVE BY NAME statements as a simple way of transferring fields in or out of database fields.
LZ_BOX_ITEM		Matches the attribute STRING of a list box item. If several columns are displayed in the list box, this variable must be redefined accordingly.
LZ_BOX_ITEM_KEY		If the key is not completely contained in the list box, the missing parts must be defined here.
LZ_DOUBLE.LZ_CMD_ID		Sequence of commands (actions) for double-click on a list box item. The first action allowed in the table counts as a command that is executed with a double-click.
LZ_START_USER_INPUT	True	If the list box is empty or no entry selected and an action (not type add) chosen, the subroutine Z_GET_KEY_USER_INPUT is called and then the function is started with the key LZ_SELECT_KEY.
	<b>False</b>	No reaction when an action is chosen and the list box is empty or no entry is selected.
LZ_KEY_IN_LISTBOX	<b>True</b>	The list box contains the entire key of a data record.
	False	The key of a data record is <b>not</b> completely contained in the list box.
LZ_MARGIN_RIGHT	<b>40</b>	Right margin in pixels between dialog and list box (only when LZ_ALIGN_HORIZ = TRUE).
LZ_MARGIN_BOTTOM	<b>40</b>	Bottom margin in pixels between dialog and list box (only when LZ_ALIGN_VERT = TRUE).
LZ_REC_EOD	True	No further data present.
	False	Further data present.
LZ_REC_FOUND		Number of data records that are found with an access.
LZ_REC_IND		Index of the current data record in the table of a multiple-record access module.
LZ_REC_NUM_FILL	<b>10</b>	Number of data records to be read following an event on list box.
LZ_REC_NUM_SEARCH	<b>0</b>	Number of data records that are read for command Z_SEARCH. Depending on the current screen resolution, the variable is set by the frame to a multiple of LZ_REC_NUM_FILL.
LZ_SELECT_KEY		Key value for the current data record. If the key consists of several components, LZ_SELECT_KEY is to be redefined accordingly.

Name	Value	Comment
LZ_START_SEARCH	True	The search begins directly after the dialog is started. (Default for production frame)
	False	The start of the search must be explicitly triggered by the user. (Default for prototype frame)

## Variables for Controlling Frame Behavior

The following variables for controlling frame behavior do not appear in the suggested code. Values deviating from the default can be assigned in the customizable component `Z_INITIALIZE`.

- LZ\_ALIGN\_VERT,
- LZ\_ALIGN\_HORIZ,
- LZ\_START\_USER\_INPUT,
- LZ\_KEY\_IN\_LISTBOX,,
- LZ\_MARGIN\_RIGHT,
- LZ\_MARGIN\_BOTTOM,
- LZ\_REC\_NUM\_FILL,
- LZ\_REC\_NUM\_SEARCH,
- LZ\_ADD\_EVERY,
- LZ\_ADD\_ON\_EMPTY

# Deletion Subprogram

## Description

Deletes an object, displaying a message box to request confirmation of the deletion from the user. It requires an associated key dialog.

## Links with Other Dialogs

Called:	Directly through the command system from browse, maintain and mass processing dialogs.
Calls:	None

## Dialog Layout

Standard deletion confirmation box.

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_DELETE	X	
Z_LOCK_RECORD		X
Z_INITIALIZE		X

## Associated Variables

Name	Value	Comment
LZ_DELETE_KEY		Record key to be deleted. This variable is to be redefined to correspond to the data transferred.
PZ_MSG.PZ_MSG_FILL (1)		This variable should be set to the value of the key is to be used in display format.

# Key Dialog

## Description

Used to provide open object, save as and active help windows. It allows the input of an object ID or the selection of an object from a list.

## Links with Other Dialogs

Called:	To select object (maintain dialog, deletion subprogram) to specify key of new object (maintain dialog, mass processing dialog) to specify key of object SAVE AS (maintain dialog)
Calls:	None

## Dialog Layout

Component	Comments
Search key	Up to 5 search key components. If the key includes more than 2 components, it is advisable to improve the layout using the dialog editor.
List box	Displays fields read using the multiple object access module.
Push buttons	OK, Cancel, Search

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_ACCESS_DATA	X	
Z_CMD_EXEC_END		
Z_CMD_EXEC_START		
Z_CUSTOM_CMD		
Z_FILL_ITEM		X
Z_INITIALIZE		X
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		
Z_RETURN_KEY		X
Z_SELECT		X
Z_SET_KEY_RANGE		

## Available Commands

Command	Explanation	Components Activated
Z_LB_CLICK	Selection in list box changed	
Z_LB_DOUBLE occurred	Double-click in list box Z_RETURN_KEY	Z_SELECT
Z_LB_FILL	Fill event of list box occurred Z_FILL_ITEM	Z_ACCESS_DATA
Z_LB_SELECT	Return selected list box item	Z_SELECT
Z_CANCEL	Close dialog	
Z_CANCEL_KEY	Key input was interrupted	
Z_CLOSE	Close dialog	
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_KEY	Key dialog sends selected key Comments: (1) Only when previous command was Z_READ. (2) For change of action within a dialog, additionally the same components as under 2. are run through.	1. Selection of a foreign key: Z_RECEIVE_KEY  2. Selection of a key at the start of the processing: Z_INITIALIZE (1) Z_CHECK_EXISTENCE Z_LOCK RECORD Z_ADD_PREL_REC Z_FILL_DIALOG  3. Input of a key at the close of the processing: Z_CHECK_EXISTENCE Z_LOCK RECORD Z_UPDATE_PREL_KEY Z_VALIDATE Z_UPDATE_PREL_REC Z_ACTIVATE_PREL_REC (2)
Z_OK	Pass value to calling dialog Close dialog	Z_RETURN_KEY
Z_SEARCH	Start of a search	Z_SET_KEY_RANGE Z_ACCESS_DATA Z_FILL_ITEM
Z_START_SEL	Start as selection dialog	Z_INITIALIZE
Z_START_SAVE	Start as storage dialog	Z_INITIALIZE

## Associated Variables

Name	Value	Comment
LZ_BOX		Group consisting of LZ_BOX_ITEM and LZ_BOX_ITEM_KEY. The group can be used for MOVE BY NAME statements as a simple way of transferring fields in or out of database fields.
LZ_BOX_ITEM		Matches the attribute STRING of a list box item. If several columns are displayed in the list box, this variable must be redefined accordingly.
LZ_BOX_ITEM_KEY		If the key is not completely contained in the list box, the missing parts must be defined here.
LZ_KEY_IN_LISTBOX	<b>True</b>	The list box completely contains the key of a data record.
	False	The key of a data record is <b>not</b> completely contained in the list box.
LZ_REC_NUM_FILL	<b>10</b>	Number of data records to be read following a FILL event on list box.
LZ_REC_NUM_SEARCH	<b>0</b>	Number of data records that are read for command <b>Z_SEARCH</b> . Depending on the size of the list box, the variable is set by the frame to a multiple of LZ_REC_NUM_FILL.
LZ_SELECT		Group to enable transfer of input fields using MOVE BY NAME into other fields. Contains as only field LZ_SELECT_KEY
LZ_SELECT_KEY		This variable must be redefined to correspond to the key to be transferred.
LZ_START_SEARCH	<b>True</b>	The search begins directly after dialog start. (Default for production Frame)
	False	The start of the search must be explicitly triggered by the user. (Default for prototype Frame)

## Variables for Controlling Frame Behavior

The following variables for controlling frame behavior do not appear in the suggested code. Values deviating from the default are to be assigned in the customizable component Z\_INITIALIZE.

- LZ\_KEY\_IN\_LISTBOX,
- LZ\_REC\_NUM\_FILL,
- LZ\_REC\_NUM\_SEARCH

# Maintain Dialog

## Description

Maintenance functionality is implemented in this frame. A dialog of this type implements "new", "modify" and "display" actions.

If not all attributes of the object to be processed can be presented in one dialog, these can be distributed among further dialogs. These dialogs should be implemented using the frame for subdialogs or for modal windows. The maintain dialog communicates with subdialogs.

Modifications of data are made to preliminary copies of the data records. The original data are only modified when a Save command is issued.

## Links with Other Dialogs

Called:	Directly through the command system from a browse dialog
Calls:	Key dialog (required) Deletion subprogram Subdialogs (handcoded link)

## Dialog Layout

Component	Comments
Data entry	Allows update of fields read using the single object access module, excluding the search key.
Push buttons	Close, confirm

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_ACTIVATE_PREL_REC	X	
Z_ADD_PREL_REC		X
Z_ASSIGN_DEFAULT_KEY		X
Z_ASSIGN_SUBDIALOG		
Z_CHECK_EXISTENCE		X
Z_CMD_EXEC_END		
Z_CMD_EXEC_START		
Z_CUSTOM_CMD		
Z_FILL_DIALOG		X
Z_INITIALIZE	X	X
Z_LOCK_RECORD		X
Z_NAVIGATE_ON_ERROR		
Z_READ_PREL_REC		X
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		
Z_UPDATE_PREL_KEY		X
Z_UPDATE_PREL_REC		X
Z_VALIDATE		

The following standard commands are available:

- Z\_APPLSTART,
- Z\_CALL,
- Z\_GET\_GLOBAL,
- Z\_HELP,
- Z\_HELPCONTNT,
- Z\_HELPUSE,
- Z\_HELPSEARCH

## Available Commands

Command	Explanation	Components Activated
Z_CANCEL	Interrupt processing	
Z_CANCEL_DLG	Subdialog gives notice of closing	
Z_CANCEL_KEY	Key input was interrupted	
Z_CANCEL_TMR	Cancel timer activated after runtime error	
Z_CLOSE	Close dialog Comments: (1) The same component as for the command Z_SAVE.	(1)
Z_CONF_DLG	Subdialog was confirmed	
Z_CONFIRM	Update preliminary files Z_UPDATE_PREL_REC	Z_VALIDATE
Z_DATA_MOD	Data has been modified	
Z_EXIT	End application Comments: A timer is activated that triggers the command Z_EXIT_TMR.	
Z_EXIT_TMR	Close dialog due to ending of application Comments: (1) The same component as for the command Z_SAVE.	(1)
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_KEY	Key dialog sends selected key Comments: (1) Only when previous command was Z_READ. (2) For change of action within a dialog, additionally the same components as under 2. are run through.	1. Selection of a foreign key: Z_RECEIVE_KEY  2. Selection of a key at the start of the processing: Z_INITIALIZE (1) Z_CHECK_EXISTENCE Z_LOCK RECORD Z_ADD_PREL_REC Z_FILL_DIALOG  3. Input of a key at the close of the processing: Z_CHECK_EXISTENCE Z_LOCK RECORD Z_UPDATE_PREL_KEY Z_VALIDATE Z_UPDATE_PREL_REC Z_ACTIVATE_PREL_REC (2)
Z_MOD_DLG	Subdialog was modified	
Z_OPEN	Start new processing	
Z_READ	Read in new data record Comments: (1) The same component as for the command Z_SAVE.	(1)
Z_REFRESH	Reverse changes since last update of the preliminary copies	Z_READ_PREL_REC Z_FILL_DIALOG
Z_RESET_DLG	Subdialog was refreshed	

Command	Explanation	Components Activated
Z_SAVE	Store modifications in original data Comment: Only for update functions. For insert with key definition, the key dialog is opened instead.	Z_VALIDATE (1) Z_UPDATE_PREL_REC (1) Z_ACTIVATE_PREL_REC (1)
Z_SAVEAS	Store data under another key with the original data Comment: The key dialog is opened.	
Z_START_KEY	Start with key Z_ASSIGN_CONTROL Z_ASSIGN_SUBDIALOG Comments: (1) Only for update functions. If the key can not be processed, the key dialog is opened.	Z_INITIALIZE Z_CHECK_EXISTENCE Z_LOCK_RECORD (1) Z_ADD_PREL_REC Z_FILL_DIALOG
Z_START_NKEY	Start without key Comments: (1) Only for insert with key definition. Otherwise, the key dialog is opened.	Z_INITIALIZE Z_ASSIGN_SUBDIALOG Z_ASSIGN_DEFAULT_KEY (1) Z_ADD_PREL_REC (1) Z_FILL_DIALOG (1)
Local command allocated to subdialog	Open subdialog or bring into the foreground	
Command of type action with subtype display or modify	Change of action Z_INITIALIZE (2) Comments: (1) The same components as for the command Z_SAVE. (2) Only when switching between actions within the same dialog. (3) Only for update functions	(1) Z_CHECK_EXISTENCE Z_LOCK_RECORD (2) (3) Z_ADD_PREL_REC (2)

## Associated Variables

Name	Value	Comment
LZ_USE_DEFAULT_KEY	<b>True</b>	When inserting, an artificial key is used until the first store.
	False	When inserting, before processing, the key dialog for input of a key is displayed.
LZ_MAIN_CONFIRM_ALL	<b>True</b>	When confirming the input in the maintain dialog, the input of the subdialogs is also confirmed.
LZ_CHECK_MODIFY	<b>True</b>	The frame controls the activation of dialog elements allocated to commands using the CHANGE- or CLICK-EVENTS of the dialog elements.
LZ_GEN_TITLE	<b>True</b>	The dialog title is generated by the frame.
LZ_DLG_TYPE		The dialog type allocated to the dialog.
LZ_SUB_DLG_CMD_ID(*)		Commands which cause a subdialog to be started.
LZ_SUB_DLG_NAME(*)		Natural names of the subdialogs that are available for using commands
LZ_SUB_DLG_MAX		Number of subdialogs available for using commands
LZ_KEY_DLG_NAME		Natural name of the key dialog
LZ_ACTIVATE_MODULE		Natural name of the activation module
PZ_SEL_KEY		Key passed by key dialog
PZ_KEY		Technical key
PZ_OBJ_ID		Object type of the object to be processed
LZ_LOCK_OBJ_ID		Object type of the data record to be locked
LZ_LOCK_KEY		Key of the data record to be locked
LZ_VAL_ERR	True	During processing, an error has occurred
LZ_FOCUS		Handling of the dialog element to be focussed
PZ_MSG.PZ_MSG_FILL (*)		Additional information for message
PZ_MSG.PZ_MSG_NUM (*)		SYSERR error number for message
PZ_ERR_FLD_POS		Identification of the erroneous field in the interface of the access module concerned
LZ_SUB_DLG_CMD_ID_SEL		Command to start a subdialog that contains an erroneous field
PZ_CMD_ID		Command
PZ_CMD_TYPE_MAIN		Command type
PZ_ACT_TYPE_CUR		Current Action type
PZ_DLG_ID		Natural dialog ID
LZ_KEY_NEW		New key when storing under a new key
LZ_PTS_NEW		New processing time stamp when storing under a new key
LZ_PREL_KEY		Key for access to preliminary data record
LZ_FRAME_CMD_ID		Command to be processed by the frame

## Variables Controlling Frame Behavior

The following variables which control frame behavior do not appear in the suggested code. Values deviating from the default are to be assigned in the customizable component `Z_INITIALIZE`.

- `LZ_CHECK_MODIFY`,
- `LZ_USE_DEFAULT_KEY`,
- `LZ_GEN_TITLE`,
- `LZ_MAIN_CONFIRM_ALL`

# Mass Processing Dialog

## Description

A function for mass processing data can be produced either as a combination of a browse dialog with a mass processing window or as a stand-alone mass processing window. The latter can be started directly from the function list. All processing takes place in the quick-input window.

The mass processing function is usually called from a browse dialog: here, the processing of a single object can be triggered by a double-click. With multiple selection of objects, the processing is triggered using the menu bar. As input, the function receives the (first) key value selected in the browse dialog. The data can then be modified and stored. The modifications are transferred into the browse dialog, in so far as the data concerned are visible there. In addition to these modification functions, the following actions are possible:

- reading a new object,
- deleting an object,
- saving a new object,
- requesting the next object (for multiple selection),
- requesting the previous object (for multiple selection), and
- requesting a new object to be processed by double-clicking in the list in the browse dialog.

Data-modifying actions (record, modify, delete) are transferred into the database direct if save is chosen.

## Links with Other Dialogs

Called:	Directly through the command system from a browse dialog.
Calls:	None

## Dialog Layout

Component	Comments
Data entry	Allows update of fields read using the single object access module, excluding the search key.
Push buttons	Close, Save, New, Read, Next, Previous, Help

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_CHECK_EXISTENCE	X	
Z_CLEAR_INPUT_FIELDS		X
Z_CMD_EXEC_END		
Z_CMD_EXEC_START		
Z_CUSTOM_CMD		
Z_FILL_DIALOG		X
Z_INITIALIZE		X
Z_LOCK_RECORD		X
Z_NAVIGATE_ON_ERROR		X
Z_PASS_KEY		X
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		
Z_UPDATE		X
Z_VALIDATE		

## Available Commands

Command	Explanation	Components Activated
Z_CANCEL_DLG	Ending of the browse dialog Comment: (1) Components of Z_SAVE	(1)
Z_CANCEL_KEY	Key input was interrupted	
Z_CLEAR	New Comment: (1) Components of Z_SAVE	(1) Z_CLEAR_INPUT_FIELDS
Z_CLOSE	Close Comment: (1) Components of Z_SAVE	(1)
Z_DATA_MOD	Data has been modified	
Z_EXIT_TMR	Close dialog due to termination of application	(1)
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_KEY	Key dialog sends selected key	Z_RECEIVE_KEY
Z_KEY_MOD	Key value has been modified	
Z_LIST_MOD	New selection from search dialog Comment: (1) Components of Z_SAVE (2) Components of Z_START_KEY	(1) (2)
Z_NEXT	Request next record from the selection list Comment: (1) Components of Z_SAVE Afterwards the next key from the selection list is requested from the browse dialog (2) Components of Z_START_KEY	(1) (2)
Z_NEW_REC	New record from browse dialog Comment: (1) Components of Z_SAVE (2) Components of Z_START_KEY	(1) (2)
Z_PREVIOUS	Request previous record from the selection list Comment: (1) Components of Z_SAVE Afterwards the previous key from the selection quantity is requested from the browse dialog (2) Components of Z_START_KEY	(1) (2)
Z_READ	Read	Z_PASS_KEY Z_CHECK_EXISTENCE Z_LOCK_RECORD Z_FILL_DIALOG
Z_SAVE	Save Comment: (1) Only for new record (2) Only in case of error	Z_PASS_KEY (1) Z_CHECK_EXISTENCE (1) Z_LOCK_RECORD (1) Z_VALIDATE Z_UPDATE Z_NAVIGATE_ON_ERROR (2)

Command	Explanation	Components Activated
Z_SCRATCH	Delete Comment: (1) Only in case of error (2) Not in case of error	Z_UPDATE Z_NAVIGATE_ON_ERROR (i) Z_CLEAR_INPUT_FIELDS (2)
Z_START_KEY	Start the function with selection	Z_INITIALIZE Z_CHECK_EXISTENCE Z_LOCK_RECORD Z_FILL_DIALOG
Z_START_NKEY	Start the function without selection	Z_INITIALIZE Z_CLEAR_INPUT_FIELDS
Z_START_SOLO	Start the function without selection and not from the browse dialog	Z_INITIALIZE Z_CLEAR_INPUT_FIELDS

### Associated Variables

Name	Value	Comment
LZ_CLEAR_AFTER_SAVE	<b>False</b>	Retain the input fields after saving
	True	Delete the input fields after saving
LZ_CLEAR_BEFORE_READ	<b>False</b>	Retain the input fields before reading
	True	Delete the input fields before reading

# Modal Window

## Description

This frame can be used to implement every kind of modal window that is called from a higher level dialog.

Access to preliminary data is not possible from this frame. It is therefore not a suitable substitute for a subdialog, but rather for optional marginal functionality which does not use much business data.

Parameters that should be passed to the frame can be supplied through the standard interface (PZ\_DATA). The return of data to the calling dialog also takes place through this interface.

## Links with Other Dialogs

Called:	Directly through the command system from a browse dialog
Calls:	Deletion Key dialog (required) Modal Window (handcoded link) Subdialogs (handcoded link)

## Dialog Layout

Component	Comments
Push buttons	OK, Close, Help

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_CMD_EXEC_START		
Z_CMD_EXEC_END		
Z_CUSTOM_CMD		
Z_FILL_DIALOG		X
Z_INITIALIZE		
Z_NAVIGATE_ON_ERROR		
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		
Z_RETURN_PARMS		
Z_VALIDATE		

## Available Commands

Command	Explanation	Components Activated
Z_CANCEL	Interrupt processing	
Z_CANCEL_KEY	Key input was interrupted	
Z_CLOSE	Close the window	Frame performance Z_VALIDATE Z_RETURN_PARMS
Z_DATA_MOD	Data has been modified	
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_KEY	Key dialog sends selected key	Z_RECEIVE_KEY
Z_NAV_ERR	Following batch error	Z_NAVIGATE_ON_ERROR
Z_START_KEY	Call the window with key	Z_ASSIGN_CONTROL Z_INITIALIZE Z_FILL_DIALOG
Z_START_NKEY	Call the window without key	Z_ASSIGN_CONTROL Z_INITIALIZE Z_FILL_DIALOG

## Additional Information

### Predefined Command Buttons

If you remove predefined command buttons from the dialog, you must also remove the corresponding instructions, which reference these dialog elements, from the Z\_ASSIGN\_CONTROL component.

### Additional Processing

Where required additional processing, such as existence checking or storing data, can be included in the Z\_VALIDATE component.

### Locking Data

The frame contains a subroutine for locking data, but does not execute this processing.

If additional records should be logically locked in a subdialog, the variables LZ\_LOCK\_OBJ\_ID and LZ\_LOCK\_KEY must be set and subsequently the subroutine Z\_CHECK\_AND\_LOCK\_RECORD must be called.

### Data Transfer

To transfer data to the modal window or from the modal window to the calling dialog, you can use the user buffer PZ\_DATA in the standard interface. If you wish to transfer data during processing to the calling dialog, then use the command SEND EVENT and include the user buffer in the data sent.

## Nonstandard Dialog

### Description

This frame supports navigation and command interpretation. It contains the standard interface for parameters, through which there is communication within the whole application.

Any further functionality must be implemented individually.

### Links with Other Dialogs

Called	Directly through command system
Calls	None

### Dialog Layout

Component	Comments
Data entry	Allows update of fields read using the single object access module, excluding the search key.
Push buttons	None

### Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_CMD_EXEC_END		
Z_CMD_EXEC_START		
Z_CUSTOM_CMD		X
Z_INITIALIZE		
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		

## Available Commands

Command	Explanation	Components Activated
Z_CANCEL	Interrupt processing	
Z_CANCEL_KEY	Key input was interrupted	
Z_CANCEL_TMR	Cancel timer activated after runtime error	
Z_CLOSE	Close dialog	
Z_DATA_MOD	Data has been modified	
Z_EXIT Z_EXIT_TMR.	End application Comments: A timer is activated that triggers the command	
Z_EXIT_TMR	Close dialog due to termination of application	
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_KEY	Key dialog sends selected key	Z_RECEIVE_KEY
Z_START_KEY	Start with key	Z_INITIALIZE
Z_START_NKEY	Start without key	Z_INITIALIZE

## Associated Variables

Name	Value	Comment
LZ_FRAME_CMD_ID		Command to be processed by the frame.
PZ_SEL_KEY		Key transferred by key dialog.
PZ_KEY		Technical key.
PZ_OBJ_ID		Object type of the object to be processed.
LZ_VAL_ERR	True	During processing an error has occurred.
LZ_FOCUS		Handling of the dialog element to be activated.
LZ_DLG_TYPE		The dialog type allocated to the dialog.

## Variables for Controlling Frame Behavior

The following variables for controlling frame behavior do not appear in the suggested code. Values deviating from the default are to be assigned in the customizable component Z\_INITIALIZE.

- LZ\_DLG\_TYPE

## Locking Data

The frame contains a subroutine for locking data, but does not execute this processing.

If additional records should be logically locked in a subdialog, the variables LZ\_LOCK\_OBJ\_ID and LZ\_LOCK\_KEY are to be set and subsequently the subroutine Z\_CHECK\_AND\_LOCK\_RECORD is to be called.

# Subdialog

## Description

Subdialogs process additional attributes not included in the main dialog. Subdialogs are not modal, so for one main dialog, several subdialogs can be processed at once. The communication between the main dialog and the subdialogs is implemented internally within the frame. Modifications of the data are stored in preliminary copies of the data records.

Several modal windows can be called by a subdialog. At one time, however, only one modal window can be open. The calls to the modal windows are coded individually.

Multiple actions (add, modify, display) can be provided by one dialog.

## Links with Other Dialogs

Called	From maintain dialog (manually link)
Calls	Modal window (manually link)

## Dialog Layout

Component	Comments
Data entry	Allows update of fields read using the single object access module, excluding the search key.
Push buttons	OK, Cancel, Confirm, Refresh, Help

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

<b>Name</b>	<b>Prototype</b>	<b>Production</b>
Z_CMD_EXEC_END		
Z_CMD_EXEC_START		
Z_CUSTOM_CMD		
Z_FILL_DIALOG		X
Z_INITIALIZE		X
Z_NAVIGATE_ON_ERROR		
Z_READ_PREL_REC		X
Z_RECEIVE_DATA		
Z_RECEIVE_KEY		
Z_UPDATE_PREL_REC		X
Z_VALIDATE		

## Available Commands

Command	Explanation	Components Activated
Z_CANCEL	Interrupt processing	
Z_CANCEL_KEY	Key input was interrupted	
Z_CLOSE	Close dialog Comments: (1) Only for update functions.	Z_VALIDATE (1) Z_UPDATE_PREL_REC (1)
Z_CONFIRM	Update preliminary files	Z_VALIDATE Z_UPDATE_PREL_REC
Z_DATA_MOD	Data has been modified	
Z_ENTER_SUB	Subdialog is activated	
Z_GET_DATA	Modal window sends data	Z_RECEIVE_DATA
Z_GET_FOCUS	Dialog is given the focus	
Z_GET_KEY	Key dialog sends selected key	Z_RECEIVE_KEY
Z_INIT	Initialization when maintain processing type changes Z_FILL_DIALOG	Z_INITIALIZE Z_READ_PREL_REC
Z_NAV_ERR	Following back error	Z_NAVIGATE_ON_ERROR
Z_OK	Update preliminary files and close dialog Comments: (1) Only for update functions	Z_VALIDATE (1) Z_UPDATE_PREL_REC (1)
Z_REFRESH	Reverse changes since last update of the preliminary copies	Z_READ_PREL_REC Z_FILL_DIALOG
Z_SAVEAS	Update dialog after 'Save as'	Z_INITIALIZE
Z_START_KEY	Start of dialog	Z_INITIALIZE Z_READ_PREL_REC Z_FILL_DIALOG

## Associated Variables

Name	Value	Comment
LZ_CHECK_MODIFY	<b>True</b>	The frame controls the activation of dialog elements allocated to commands using the CHANGE- or CLICK-EVENTS of all dialog elements
LZ_GEN_TITLE	<b>True</b>	The dialog title is generated by the frame
LZ_DLG_TYPE		The dialog type allocated to the dialog
PZ_SEL_KEY		Key passed by key dialog
PZ_KEY		Technical key
PZ_OBJ_ID		Object type of the object to be processed
LZ_LOCK_OBJ_ID		Object type of the data record to be locked
LZ_LOCK_KEY		Key of the data record to be locked
LZ_VAL_ERR	True	During processing, an error has occurred
LZ_FOCUS		Handle of the dialog element to be focussed
PZ_MSG.PZ_MSG_FILL (*)		Additional information for message
PZ_MSG.PZ_MSG_NUM (*)		SYSERR error number for message
PZ_ERR_FLD_POS		Identification of the erroneous field in the interface of the access module concerned
PZ_CMD_ID		Command
PZ_CMD_TYPE_MAIN		Command type
PZ_ACT_TYPE_CUR		Current Action type
PZ_DLG_ID		Natural dialog ID
LZ_PREL_KEY		Key for access to preliminary data record
LZ_FRAME_CMD_ID		Command to be processed by the frame

## Variables for Controlling Frame Behavior

The following variables for controlling frame behavior do not appear in the suggested code. Values deviating from the default are to be assigned in the customizable component Z\_INITIALIZE.

- LZ\_CHECK\_MODIFY,
- LZ\_DLG\_TYPE,
- LZ\_GEN\_TITLE

### Locking Data

The frame contains a subroutine for locking data, but does not execute this processing.

If additional records should be logically locked in a subdialog, the variables LZ\_LOCK\_OBJ\_ID and LZ\_LOCK\_KEY are to be set and subsequently the subroutine Z\_CHECK\_AND\_LOCK\_RECORD is to be called.

# Background Program

## Description

This program is used to implement background programs which start from an online program using subprogram ZXBG010N.

It contains the standard interface for the program with which the background program communicates.

Any additional functionality must be individually implemented.

## Links with Other Dialogs

Called by:	Dialogs using subprogram ZXBG010N
Calls:	None

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Implementation	Description
Z_BG_INIT		Subroutine called once to execute program dependent initialization.
Z_STORE_RESTART_DATA		Subroutine used to write restart data.
Z_BG_CODING	X	Main subroutine to implement the background functionality.

## Associated Variables

Name	Description
LZ_PGM_PARM	Parameters passed from the dialog.
LZ_RESTART_DATA	Data required for a restart.

The variables are defined in the background program. They can be redefined to fit the data passed from the dialog. The variable LZ\_RESTART\_DATA is set during the execution of the background program. It is needed in case of a restart.

# Load Objects Subprogram

## Description

This subprogram is used to load objects of a specified type into the database from a workfile.

## Links with Other Dialogs

Called by:	The load data dialog in the application shell.
Calls:	None

## Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_UL_INIT_UNLOAD		X
Z_UL_FILL_VIEW		X
Z_UL_STORE_VIEW		X#

## Associated Variables

Name	Description
PZ_UL_KEY_FROM	The start key for the range of records to be loaded.
PZ_UL_KEY_THRU	The end key for the range of records to be loaded.
PZ_UL_WF_DATA(*)	The data of the record to be loaded.

## Unload Objects Subprogram

### Description

This subprogram is used to unload objects of a specified type from the database to a workfile.

### Links with Other Dialogs

Called by:	The unload data dialog in the application shell.
Calls:	None

### Customizable Components

Components marked with an "X" contain executable code while unmarked components contain only suggested code.

Name	Prototype	Production
Z_UL_INIT		
Z_UL_SELECT_TOP		X
Z_UL_SELECT_LOW		X
Z_UL_WRITE_WORKFILE		X
Z_UL_DELETE_RECORDS		X
STORE_REFERENCED_VIEW		

### Associated Variables

Name	Description
PZ_UL_KEY_FROM	The start key for the range of records to be unloaded.
PZ_UL_KEY_THRU	The end key for the range of records to be unloaded.
PZ_UL_WF_DATA(*)	The data of the record to be unloaded.