Maps

As an alternative to specifying screen layouts dynamically, the `INPUT` statement offers the possibility to use predefined map layouts which makes use of the Natural object type "map".

This chapter covers the following topics:

- Benefits of Using Maps

- Types of Maps

- Creating Maps

- Starting/Stopping Map Processing

Benefits of Using Maps

Using predefined map layouts rather than dynamic screen-layout specifications offers various advantages such as:

- Clearly structured applications as a result of a consequent separation of program logic and display logic.

- Map layout modifications possible without making changes to the main programs.

- The language of an applications's user interface can be easily adapted for internationalization or localization.

The benefit of using programming objects such as maps will become obvious when it comes to maintaining existing Natural applications.

Types of Maps

Maps (screen layouts) are those parts of an application which the users see on their screens.

The following types of maps exist:

- **Input Map**
  The dialog with the user is carried out via input maps.

- **Output Map**
  If an application produces any output report, this report can be displayed on the screen by using an output map.

- **Help Map**
  Help maps are, in principle, like any other maps, but when they are assigned as help, additional checks are performed to ensure their usability for help purpose.

The object type "map" comprises

- the map body which defines the screen layout and

- an associated parameter data area (PDA) which, as a sort of interface, contains data definitions such as name, format, length of each field presented on a specific map.

Related Topics:

- For information on selection boxes that can be attached to input fields, see *SB - Selection Box* in the `INPUT` statement documentation and *SB - Selection Box* in the *Parameter Reference*.

- For information on split screen maps where the upper portion may be used as an output map and the lower portion as an input map, see *Split-Screen Feature* in the `INPUT` statement documentation.

Creating Maps

Maps and help map layouts are created and edited in the map editor. The appropriate LDA is created and maintained in the data area editor.

Depending on the platform on which Natural is installed, these editors have either a character user interface or a graphical user interface.

Related Topics:

- For information on using the data area editor, see *Data Area Editor* in the platform-specific *Editors* documentation.

- For information on using the map editor, see *Map Editor* in the platform-specific *Editors* documentation.

- For a comprehensive description of the full range of possibilities provided by the Natural map editor (character-user-interface version), see *Map Editor Tutorial*.

- For information on input processing using screen layouts specified dynamically, see *Syntax 1 - Dynamic Screen Layout Specification* in the `INPUT` statement documentation.

- For information on input processing using a map layout created with the map editor, see *Syntax 2 - Using Predefined Map Layout* in the `INPUT` statement documentation.

Starting/Stopping Map Processing

An *input map* is invoked with an `INPUT USING MAP` statement.

An *output map* is invoked with a `WRITE USING MAP` statement.

Processing of a map can be stopped with an `ESCAPE ROUTINE` statement in a processing rule.