

Natural for Mainframes

Natural for MBCS

Version 4.2.6 for Mainframes (Update)

February 2010

This document applies to Natural Version 4.2.6 for Mainframes (Update).

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1979-2010 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Table of Contents

1 Natural for MBCS	1
2 Installing Natural for MBCS under z/OS	3
Prerequisites	4
Installation Tape for Natural for MBCS under z/OS	4
Installation Procedure	6
Installation Verification for Natural for MBCS	7
3 Installing Natural for MBCS under z/VSE	9
Prerequisites	10
Installation Tape for Natural for MBCS under z/VSE	10
Installation Procedure	12
Installation Verification for Natural for MBCS	13
4 Installing Natural for MBCS under VM/CMS	15
Prerequisites	16
Installation Tape for Natural under CMS	16
Copying the Tape Contents to Disk	17
Step 1 - Customize NAT\$LOAD EXEC to Link Natural	17
Step 2 - Load the Natural for MBCS Objects	17
Step 3 - Make Natural Modules Available to the Users	18
5 Options for Box Creation	19
6 Using the Outline Editor	21
Prerequisites	22
Invoking the Outline Editor	24
Using Box Codes	25
Using the End Marker	28
List of Box Codes and BX Parameter Values	28
Example of Box Coding	29
Line Commands and PF Keys	32

1 Natural for MBCS

Natural for MBCS provides the following functionality:

- Converts maps created with Natural Version 3.1 and KAPRI (a product by Beacon IT) into maps that are suitable for compilation and execution with Natural for MBCS.
- Provides features required to compile and execute these converted maps so that the resulting screen layout is the same as for the original maps created and executed with Natural Version 3.1 and KAPRI.
- Provides the outline editor and extended map editor features that can be used to create and maintain boxes for data and text fields (including boxes for blank areas) contained in a map.

	Installing Natural for MBCS under z/OS	How to install Natural for MBCS on a z/OS operating system.
	Installing Natural for MBCS under z/VSE	How to install Natural for MBCS on a z/VSE operating system.
	Installing Natural for MBCS under VM/CMS	How to install Natural for MBCS on a VM/CMS operating system.
	Options for Box Creation	General information on the alternatives provided to create boxes for map fields.
	Using the Outline Editor	Using the outline editor to define boxes for screens created with the Natural map editor.

2 Installing Natural for MBCS under z/OS

■ Prerequisites	4
■ Installation Tape for Natural for MBCS under z/OS	4
■ Installation Procedure	6
■ Installation Verification for Natural for MBCS	7

This section describes step by step how to install Natural for MBCS (product code NKA) under the operating system z/OS using Adabas system files.

See also *z/OS Environment* in the *Operations* documentation.

For installation-related information on Unicode and code page support, refer to *Configuration and Administration of the Unicode/Code Page Environment* in the *Unicode and Code Page Support* documentation.

Notation *vrs* or *vr*:

The notation *vrs* or *vr* stands for the relevant version, release, system maintenance level numbers. For further information on product versions, see *Version* in the *Glossary*.

Prerequisites

- Base Natural must be installed.
- It is possible to use a VSAM file as a system file. In this case, Natural for VSAM must be installed. For more information, see the *Natural for VSAM* documentation.

For further information, refer to the products and versions specified under *Natural and Other Software AG Products* and *Operating and Teleprocessing Systems Required* in the current *Natural Release Notes for Mainframes*.

Installation Tape for Natural for MBCS under z/OS

The installation tape contains the datasets listed in the table below. The sequence of the datasets and the number of library blocks needed are shown in the *Report of Tape Creation* which accompanies the installation tape.

Dataset Name	Contents
NKA <i>vrs</i> .LOAD	Natural for MBCS load modules.
NKA <i>vrs</i> .INPL	Natural for MBCS system objects.

Copying the Tape Contents to a z/OS Disk

If you are using SMA, refer to the *System Maintenance Aid* documentation (included in the current edition of the Natural documentation CD).

If you are *not* using SMA, follow the instructions below.

This section explains how to:

- Copy dataset `COPY.JOB` from tape to disk.
- Modify this dataset to conform to your local naming conventions.

The JCL in this dataset is then used to copy all datasets from tape to disk.

If the datasets for more than one product are delivered on the tape, the dataset `COPY.JOB` contains the JCL to unload the datasets for all delivered products from the tape to your disk.

After that, you will have to perform the individual install procedure for each component.

- [Step 1 - Copy Dataset COPY.JOB from Tape to Disk](#)
- [Step 2 - Modify COPY.JOB on Your Disk](#)
- [Step 3 - Submit COPY.JOB](#)

Step 1 - Copy Dataset COPY.JOB from Tape to Disk

The dataset `COPY.JOB` (Label 2) contains the JCL to unload all other existing datasets from tape to disk. To unload `COPY.JOB`, use the following sample JCL:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* -----
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=tape-volume),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=volume,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*,SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

hilev is a valid high level qualifier

tape-volume is the tape volume name, for example: T12345
volume is the disk volume name

Step 2 - Modify COPY.JOB on Your Disk

Modify the COPY.JOB on your disk to conform to your local naming conventions and set the disk space parameters before submitting this job:

- Set HILEV to a valid high level qualifier.
- Set LOCATION to a storage location.
- Set EXPDT to a valid expiration date.

Step 3 - Submit COPY.JOB

Submit COPY.JOB to unload all other datasets from the tape to your disk.

Installation Procedure

- [Step 1 - Relink the Natural Nucleus to Include the Natural for MBCS Module](#)
- [Step 2 - Load the System Programs](#)

Step 1 - Relink the Natural Nucleus to Include the Natural for MBCS Module

(Jobs I060 and I080 from the Natural installation tape)

Relink your Natural nucleus to include the Natural for MBCS module:

```
INCLUDE NKALOAD(NKANUC)
```

You must link the NKANUC module to all Natural nuclei that should make use of the features supplied by Natural for MBCS. Software AG recommends that you relink Natural with the NKANUC module.

To avoid the need to relink Natural with NKANUC, specify the Natural profile parameters RCA and RCALIAS either dynamically or in the NATPARM parameter module:

```
RCA=NATGWNKA,RCALIAS=(NATGWNKA,NKANUC)
```

Step 2 - Load the System Programs

(Job I061, Step 8200)

Use the Natural utility *INPL* to load the Natural system objects (dataset *NKA vrs. INPL*) into the Natural system file.

Installation Verification for Natural for MBCS

After the last step of the installation procedure has been successfully performed, start Natural and enter the command `EDIT MAP` at the command prompt. The **Edit Map** menu should then appear with the additional menu option **Outline Editor**.

3 Installing Natural for MBCS under z/VSE

- Prerequisites 10
- Installation Tape for Natural for MBCS under z/VSE 10
- Installation Procedure 12
- Installation Verification for Natural for MBCS 13

This section describes step by step how to install Natural for MBCS (product code NKA) under the operating system z/VSE using Adabas system files.

See also *z/VSE Environment* in the *Operations* documentation.

For installation-related information on Unicode and code page support, refer to *Configuration and Administration of the Unicode/Code Page Environment* in the *Unicode and Code Page Support* documentation.

Notation *vrs* or *vr*:

If used in the following section, the notation *vrs* or *vr* stands for the relevant version, release, system maintenance level numbers. For further information on product versions, see *Version* in the *Glossary*.

Prerequisites

- Base Natural must be installed.
- It is possible to use a VSAM file as a system file. In this case, Natural for VSAM must be installed. For more information, see the *Natural for VSAM* documentation.

For further information, refer to the products and versions specified under *Natural and Other Software AG Products* and *Operating and Teleprocessing Systems Required* in the current *Natural Release Notes for Mainframes*.

Installation Tape for Natural for MBCS under z/VSE

The installation tape contains the datasets listed in the table below. The sequence of the datasets and the number of library blocks needed are shown in the *Report of Tape Creation* which accompanies the installation tape.

Dataset Name	Contents
NKA <i>vrs</i> .LOAD	LIBR backup file.
NKA <i>vrs</i> .INPL	Natural for MBCS system objects.

Copying the Tape Contents to Disk

If you are using SMA, refer to the *System Maintenance Aid* documentation (included in the current edition of the Natural documentation CD).

If you are *not* using SMA, follow the instructions below.

This section explains how to:

- Copy dataset COPYTAPE.JOB from tape to disk.
- Modify this dataset to conform with your local naming conventions.

The JCL in this member is then used to copy all datasets from tape to disk.

If the datasets for more than one product are delivered on the tape, the member COPYTAPE.JOB contains the JCL to unload the datasets for all delivered products from the tape to your disk, except the datasets that you can directly install from tape, for example, Natural INPL objects.

After that, you will have to perform the individual install procedure for each component.

- [Step 1 - Copy Dataset COPYTAPE.JOB from Tape to Disk](#)
- [Step 2 - Modify COPYTAPE.JOB](#)
- [Step 3 - Submit COPYTAPE.JOB](#)

Step 1 - Copy Dataset COPYTAPE.JOB from Tape to Disk

The dataset COPYTAPE.JOB contains the JCL to unload all other existing datasets from tape to disk. To unload COPYTAPE.JOB, use the following sample JCL:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                     +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****
*   CATALOG COPYTAPE.JOB TO LIBRARY
* *****
// ASSGN SYS004,nnn                                           <----- tape address
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'                   <----- for catalog
/*
// MTC REW,SYS004

ASSGN SYSIPT,FEC
/*
```

```
/&  
* $$ EOJ
```

where:

nnn is the tape address

lib.sublib is the library and sublibrary of the catalog

Step 2 - Modify COPYTAPE.JOB

Modify COPYTAPE.JOB to conform to your local naming conventions and set the disk space parameters before submitting this job.

Step 3 - Submit COPYTAPE.JOB

Submit COPYTAPE.JOB to unload all other datasets from the tape to your disk.

Installation Procedure

- [Step 1 - Relink the Natural Nucleus to Include the Natural for MBCS Module](#)
- [Step 2 - Load the System Programs](#)

Step 1 - Relink the Natural Nucleus to Include the Natural for MBCS Module

(Jobs I060 and I080 from the Natural installation tape)

Relink your Natural nucleus to include the Natural for MBCS module:

```
INCLUDE NKANUC
```

You must link the NKANUC module to all Natural nuclei that should make use of the features supplied by Natural for MBCS. Software AG recommends that you relink Natural with the NKANUC module.

To avoid the need to relink Natural with NKANUC, specify the Natural profile parameters RCA and RCALIAS either dynamically or in the NATPARM parameter module:

```
RCA=NATGWNKA,RCALIAS=(NATGWNKA,NKANUC)
```

Step 2 - Load the System Programs

(Job I061, Step 8200)

Use the Natural utility *INPL* to load the Natural system objects (dataset *NKA vrs. INPL*) into the Natural system file.

Installation Verification for Natural for MBCS

After the last step of the installation procedure has been successfully performed, start Natural and enter the command `EDIT MAP` at the command prompt. The **Edit Map** menu should then appear with the additional menu option **Outline Editor**.

4 Installing Natural for MBCS under VM/CMS

- Prerequisites 16
- Installation Tape for Natural under CMS 16
- Copying the Tape Contents to Disk 17
- Step 1 - Customize NAT\$LOAD EXEC to Link Natural 17
- Step 2 - Load the Natural for MBCS Objects 17
- Step 3 - Make Natural Modules Available to the Users 18

This section describes step by step how to install Natural for MBCS (product code NKA) under the operating system VM/CMS using Adabas system files.

See also *VM/CMS Environment* in the *Operations* documentation.

For installation-related information on Unicode and code page support, refer to *Configuration and Administration of the Unicode/Code Page Environment* in the *Unicode and Code Page Support* documentation.

Notation *vrs* or *vr*:

If used in the following section, the notation *vrs* or *vr* stands for the relevant version, release, system maintenance level numbers. For further information on product versions, see *Version* in the *Glossary*.

Prerequisites

- Base Natural must be installed.

For further information, refer to the products and versions specified under *Natural and Other Software AG Products* and *Operating and Teleprocessing Systems Required* in the current *Natural Release Notes for Mainframes*.

Installation Tape for Natural under CMS

The installation tape was created under z/OS; it has standard z/OS labels and headers. It contains the datasets listed in the table below. The sequence of the datasets is shown in the *Report of Tape Creation* which accompanies the installation tape.

Dataset Name	Contents
NKA <i>vrs</i> .TAPE	CMS installation material. This dataset is in TAPE DUMP format and must be loaded onto the installation mini disk.
NKA <i>vrs</i> .INPL	Natural system objects.

Copying the Tape Contents to Disk

Create the `.NKAvr.NKAvrs` directory and copy the file `NKAvrs.TAPE` from the tape into this directory. As an alternative, define a minidisk for NKA and load `NKAvrs.TAPE` onto the disk by using the following command:

```
TAPE LOAD * * fm
```

If the sequence number of `NKAvrs.EXEC`, as shown by the *Report of Tape Creation*, is n , you must position over $3n - 2$ tape marks (that is, FSF 1 for the first dataset, FSF 4 for the second, etc.). `fm` is the filemode used for the directory or minidisk.

```
CREATE DIRECTORY .NKAvr.NKAvrs
ACCESS .NKAvr.NKAvrs B
TAPE REW
TAPE FSF n
TAPE LOAD * * fm
```

Step 1 - Customize NAT\$LOAD EXEC to Link Natural

Modify `NAT$LOAD EXEC`: delete the comment for NKA to include `NKANUC TEXT`. Example:

```
LOADLIST = LOADLIST 'NKANUC'
```

Build a new Natural module or a Discontiguous Shared Segment (DCSS) by executing `NATBLDM` or `NATBLDS` respectively.

Step 2 - Load the Natural for MBCS Objects

If the tape drive used when copying the contents of the installation tape to disk was detached from your virtual machine, ask the system operator to attach a tape drive to your virtual machine at address `X'181'` and mount the Natural installation tape.

Before you can start the Natural INPL, the Adabas environment for your CMS machine must have been set up (as described in the Adabas documentation).

Issue the following CMS command:

```
NKAINPL
```

You are prompted for the name of the command to invoke Natural. Enter one of the following:

- the name of the Natural module built in the corresponding step, or
- the name of the Natural DCSS (Discontiguous Shared Segment) bootstrap created in the corresponding step.

NKAINPL then positions the tape and loads the Natural for MBCS system objects.

The Natural system file definitions as coded in the Natural parameter module apply.

Three reports are produced on your virtual printer. Check these reports to ensure that no errors have occurred.

Step 3 - Make Natural Modules Available to the Users

Copy the Natural module and the DCSS bootstrap to a public disk or have your system programmer copy them to the CMS Y-disk.

5

Options for Box Creation

Natural for MBCS provides the option to create boxes for map fields. Boxes help outline particular fields and provide for a more clearly structured screen layout.

Boxes can be created with either the outline editor or the map editor:

■ **Outline Editor**

The outline editor is the primary tool for creating boxes. It provides a convenient way to draw a box directly in the field location within the editing area. This is achieved by setting box codes, for which equivalent BX parameter values are generated into the map source.

Detailed instructions are provided in [Using the Outline Editor](#).

■ **Map Editor**

The map editor can be used to define boxes by adding or changing the BX parameter settings for the field around which you want to draw a box.

The BX parameter can be set in the **BX** field of the extended field editing area, which is invoked with the extended field editing function described in the *Map Editor* documentation. In the **BX** field, you can enter a combination of valid parameter values as described for the BX session parameter and listed in [List of Box Codes and BX Parameter Values](#).

Related Topic:

- *BX - Box Definition* in the *Parameter Reference* documentation

6 Using the Outline Editor

▪ Prerequisites	22
▪ Invoking the Outline Editor	24
▪ Using Box Codes	25
▪ Using the End Marker	28
▪ List of Box Codes and BX Parameter Values	28
▪ Example of Box Coding	29
▪ Line Commands and PF Keys	32

The outline editor is used to define boxes for screens created with the Natural map editor. This is achieved by setting box codes for the fields around which you want to draw a box. Box codes represent values of the BX parameter.

In principle, the outline editor is used for creating boxes and not for defining fields. Field definitions are made with the map editor.

This section covers the following topics:

Prerequisites

The following prerequisites are required for using the outline editor:

- [Suitable Terminal](#)
- [Migration of Maps Created with KAPRI](#)
- [Definition of Delimiter Class D](#)

Suitable Terminal

The terminal or terminal emulation used must support viewing boxes. This is usually a terminal or terminal emulation that supports DBCS (Double Byte Character Set) characters. Otherwise, the boxes generated for a map will not be displayed at execution time.

Related Topic:

For information on the features implemented in Natural to support DBCS, see also *Double-Byte Character Sets* in the *Operations* documentation.

Migration of Maps Created with KAPRI

Maps created with Natural Version 3.1 and KAPRI must be converted into maps that are suitable for compilation and execution with Natural for MBCS. See *Handling of Maps Converted Using Natural for MBCS* in the current *Natural Release Notes for Mainframes*.

 **Caution:** KAPRI remembers the screen position of DBCS (Double Byte Character Set) characters and boxes to display them correctly. Unlike KAPRI, Natural for MBCS uses the real display length and Natural language extensions to determine the correct display position. Therefore, it is essential that maps created with KAPRI are converted before they are executed with Natural and Natural for MBCS. Executing a map that has not been converted may lead to unpredictable results when such a map is displayed.

For the migration of KAPRI maps, we recommend that you keep backup copies and verify the proper layout of the new maps after the migration by using the **Test** function of the map editor or the outline editor.

Because it is not possible to edit converted KAPRI maps with a Natural version that does not support Natural for MBCS, we strongly recommend that you store the converted maps in a different library (or in a different FUSER system file) than the maps created with Natural Version 3.1 and KAPRI.

▶ **To convert single maps created with KAPRI**

- 1 Invoke the **Edit Map** menu of the map editor.
- 2 In the **Edit Map** menu, enter function code X for **Migration of Maps created with KAPRI** and the name of the map to be converted.
- 3 Press ENTER.

A message appears confirming that the map has been successfully converted.

▶ **To convert multiple maps created with KAPRI**

- Use the `CATALL` system command as described in the *System Commands* documentation.

The `CATALL` command also invokes and executes the migration function if the `CATALOG`, `STOW` or `CHECK` option is selected on the **CATALL** screen or specified in a `CATALL` direct command. No migration is performed if the `SAVE` option is specified.

Definition of Delimiter Class D

Delimiter class `D` must be defined in the map settings of each map to be edited with the outline editor. If this delimiter class is missing, you cannot open the outline editor but will receive an appropriate error message instead.

Delimiter class `D` is usually added automatically when you read in the map definition of a map created with KAPRI or when you create a new map with Natural as described in *Delimiter Class D* in the *Map Editor* documentation. However, for an existing map created with Natural, you must add the delimiter class as described in the following instructions.

▶ **To define delimiter class D**

- 1 Invoke the map editor for a map that is stored as a source object in a library in a Natural system file.

The editing area of the map editor appears, which contains the specified map.

- 2 Press `PF2` (Mset) to change the default settings for the map.

The **Define Map Settings** screen appears.

- 3 In the **Delimiters** section, add delimiter class D (or replace an existing delimiter) similar to the example shown below:

```
Delimiters
-----
Cls Att CD Del
T D BLANK
T I ?
A D _
A I )
A N ^
M D &
M I :
O D +
O I (
D I #
```

- 4 Invoke the **Edit Map** menu of the map editor and save the map.

For detailed information on delimiters, see the relevant section in the *Map Editor* documentation.

Invoking the Outline Editor

This section provides instructions for invoking the outline editor.

▶ To invoke the outline editor

- 1 Open the **Edit Map** menu of the map editor.
- 2 Enter function code 0 for **Outline Editor** and the name of the map to be edited with the outline editor. The map must either be contained in the current source work area of the editor or stored as a source object in a library in a Natural system file.

An **Outline Editor** screen similar to the example below appears in split-screen mode for the specified map:

```

Outline Editor
Code BX  Code BX
T  ┌───┐  B  ┌───┐  I  ┌───┐  C  ┌───┐  1  ┌───┐  3  ┌───┐  5  ┌───┐  U  ┌───┐
L  ┌───┐  R  ┌───┐  H  ┌───┐  O  ┌───┐  2  ┌───┐  4  ┌───┐  6  ┌───┐  E  END
001  --010--+-----+-----+-----+-----+-----+-----+-----+-----+
-
#Personal#Information

First Name:  :XXXXXXXXXXXXXXXXXXXXX
Last Name:   :XXXXXXXXXXXXXXXXXXXXX
Street:      :XXXXXXXXXXXXXXXXXXXXX
Zip Code / City:  :99999999 :XXXXXXXXXXXXXXXXXXXXX

Phone Office: :XXXXXXXXXXXXXXXXXXXXX
Home:         :XXXXXXXXXXXXXXXXXXXXX
Mobile:       :XXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit  Test  BoxDi  --  -  +  Full  <  >  Let

```

The upper part of the outline editor screen shows all valid box codes (for example, **O**), symbols for the type of box they create (for example, a square for **O**) and the end marker **E (END)**. This chart is not displayed in full-screen mode.

The lower part contains the editing area with the map definition of the specified map. The characters indicating data fields (such as **X** or **9**) are identical to the characters used in the editing area of the map editor. For details, see *Defining Map Fields* in the *Map Editor* documentation.

Using Box Codes

A box code is a single character which represents a value of the **BX** session parameter or a combination of values. For example, the box code **0** represents a **BX** setting of **TBLR**. For a description of possible parameter values, see [List of Box Codes and BX Parameter Values](#).

The rules and conditions that apply when using box codes are described in the following section. The examples mentioned refer to [Example of Box Coding](#).

A box code is *not* case-sensitive. You can enter a box code for any of the following:

- A data field.

- A text field created with delimiter class `D` (delimiter class `T` is *not* allowed).
- A blank character or a sequence of blank characters.

This section covers the following topics:

- [Entering Box Codes for Data and Text Fields](#)
- [Entering Box Codes for Blank Characters](#)
- [Automatic Box Continuation](#)
- [Changing or Deleting Box Codes](#)

Entering Box Codes for Data and Text Fields

For a data field or a text field of delimiter class `D`, enter a box code in the attribute position of the field for which you want to create a box. The attribute position is immediately in front of the field and is denoted by one of the following:

- The delimiter character as assigned to the field with the map editor.
Example: The text field `Information`.
- A box code that has already been entered for the field. The outline editor then hides the delimiter character and shows the box code instead.
Example: The text field `Personal`.

For a data field that has been defined as an array, you can enter a box code for each occurrence of this array. Example: `Office:`, `Home:` and `Mobile:` are three occurrences of a one-dimensional array (`A20/1:3`). If you want the box code to be valid for all occurrences of this array, you can modify the `BX` parameter with the extended field editing function of the map editor.

Entering Box Codes for Blank Characters

You can create a box for a blank character or a sequence of blank characters. This provides the option to use blank map areas for the box design.

For a blank character or a sequence of blank characters, enter a box code in an empty line position.

The length of the box is determined by the number of blank characters to which the used box code applies and the end marker (described in [Using the End Marker](#)):

- If the end marker is set, the box is drawn around the number of characters from the box code to the end marker.
Examples:
The box codes `2` and `U` in the blank area next to the data field `Last Name:`.
- If no end marker is set, the box is drawn around either the number of characters from the box code to the next field or to the end of the line.

Exceptions: The box codes **L** and **R** only create boxes for a single blank character, regardless of whether or not the end marker is used. Example: The box code **L** between the text fields `Phone` and `Office:`.

A box created for a sequence of blank characters is indicated by a number of periods (.) as shown in *Example of Box Coding after ENTER*. No periods are displayed for a box created with a box code that is immediately followed by the end marker (for example: `0E`) or the box codes **L** and **R**.

A box created for a blank character or a sequence of blank characters is generated as a blank text field of delimiter class `D` into the map definition.

You can reduce the length of a box by replacing any of the periods (.) with the end marker (see also *To place the end marker*) or a box code. The field then ends at the period you replaced.

You can edit a blank text field with the extended field editing function of the map editor.

Automatic Box Continuation

A box code that has a top and/or bottom line but no right vertical line (the box codes **T**, **B**, **I**, **C**, **3** and **5**) remains valid for subsequent fields in this line. The box then ends either at the next box code, the next text field of delimiter class `T` or at the end of the line. Examples: The text fields `Personal` and `Information` and the data fields next to `City:`.

When you enter one of the above box codes, the outline editor places automatically generated box codes in front of all subsequent fields to which this box code applies. The sequence of automatically generated box codes is listed under *Auto-Generated Box Codes* in *List of Box Codes and BX Parameter Values*. Box codes are only generated automatically when you initially enter a box code. You can replace automatically generated box codes.

Changing or Deleting Box Codes

You can change single or multiple box codes by replacing each box code with another box code.

You can delete single or multiple box codes either by replacing each box code with a blank character or by using one of the line commands described in *Line Commands and PF Keys*.

In general, you can only delete the box code of a field and *not* the field itself. A field definition can only be deleted with the map editor. Exception: When you delete the box code of a blank field, you also delete the respective field in the map definition.

Using the End Marker

The end marker **E (END)** determines the length of a box created for a single blank character or a sequence of blank characters.

Exceptions: The box codes **L** and **R** always create boxes for a single blank character only. They can be entered without the end marker. Example: The box code **L** between the text fields **Phone** and **Office:**.

The end marker is *not* case-sensitive.

The end marker is *not* generated as a **BX** parameter value into the map source.

The end marker is valid for the box code and the line where it is placed. You cannot enter a box code in one line and the end marker in another line to extend a box over multiple lines.

▶ To place the end marker

- 1 In an empty line position, enter a box code where you want the box to start and an **E** where you want the box to end.

Example:

The columns **035** and **059** in the empty line above **Last Name:** in [Example of Box Code Settings](#).

- 2 Press **ENTER** to confirm your entries.

The blank characters delimited by box code and end marker (in the example, between the columns **035** and **059**) are replaced by periods (.).

The end marker is no longer indicated on the screen.

- 3 You can reduce the length of a box by overwriting any of the periods (.) with the end marker. Example: **U E**

The box then ends at the period you overwrote.

List of Box Codes and BX Parameter Values

The box codes listed in the editing area of the outline editor are explained in the following table. The table also indicates the sequence of automatically generated box codes (if relevant for a box code) and the equivalent **BX** parameter values (in any order).

Box	Borders	Box Codes	BX Values	Auto-Generated Box Codes	BX Values for Auto-Generated Box Codes
	Top line	T	T	T	T
	Bottom line	B	B	B	B
	Top and bottom line	I	TB	I	TB
	Top, bottom and left vertical line	C	LTB	I	TB
	Top, bottom and right vertical line	1	TRB		
	Top and left vertical line	3	LT	T	T
	Top and right vertical line	4	TL		
	Bottom and left vertical line	5	LB	B	B
	Bottom, left vertical and right vertical line	U	LBR		
	Left vertical line	L	L		
	Right vertical line	R	R		
	Left and right vertical line	H	LR		
	Top, bottom, left vertical and right vertical line	0	TBLR		

Example of Box Coding

This section provides an example of box coding, which is based on the [map definition](#) shown in *Invoking the Outline Editor*. This example is not intended to provide guidance for designing maps with boxes, but rather demonstrates how box codes can be used.

- [Example of Box Code Settings](#)
- [Example of Box Coding after ENTER](#)
- [Example of a Box Layout](#)

Example of Box Code Settings

The example below shows how box codes can be entered in the outline editor:

```

                                O u t l i n e   E d i t o r
Code BX  Code BX
  T  _   B  _   I  _   C  [  1  [  3  [  5  [  U  [
001  L  |   R  |   H  |   O  [  2  [  4  [  6  [  E  END
-----+-----+-----+-----+-----+-----+-----+-----+-----
                                BPersonal#Information

                                First Name:  OXXXXXXXXXXXXXXXXXXXXX
                                Last Name:    2
                                                L:XXXXXXXXXXXXXXXXXXXXX R
                                                U
                                                E
                                Street:       OXXXXXXXXXXXXXXXXXXXXX
                                Zip Code / City: C99999999 :XXXXXXXXXXXXXXXXX1E

                                Phone L Office: OXXXXXXXXXXXXXXXXXXXXX
                                L Home:       :XXXXXXXXXXXXXXXXXXXXX
                                L Mobile:    :XXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit  Test  BoxDi  --  -  +  Full  <  >  Let

```

Example of Box Coding after ENTER

The example below shows how box coding can look after you confirm your box-code entries with ENTER:

```

Outline Editor
Code BX Code BX
T  _  B  _  I  _  C  [  1  [  3  [  5  [  U  [
L  |  R  |  H  |  O  [  2  [  4  [  6  [  E  END
001  --010--+-----+-----030--+-----+-----050--+-----+-----070--+-----+-----

      BPersonalBInformation

      First Name:  OXXXXXXXXXXXXXXXXXXXXX

      Last Name:  2.....
                  L:XXXXXXXXXXXXXXXXXXXXX R
                  U.....

      Street:  OXXXXXXXXXXXXXXXXXXXXX

      Zip Code / City:  C99999999I.IXXXXXXXXXXXXXXXXX1

      Phone L Office: OXXXXXXXXXXXXXXXXXXXXX
             L Home:  :XXXXXXXXXXXXXXXXXXXXX
             L Mobile: :XXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit Test BoxDi --  -  +      Full <  >      Let

```

Example of a Box Layout

The example below shows the box layout of a map when invoked by a program or with the PF4 test function.

Line Command	PF Key	Explanation
..Q	PF3	Exit. Terminates the outline editor and returns to the Edit Map menu. If pressed on the Box-Code Display or the test screen, returns to the editing area of the outline editor.
	PF4	Test. Tests the map definition with the box layout. This function can also be executed from the map editor. Press PF3 or ENTER to return to the editing area.
	PF5	Box Display. Displays the box-code settings of a map: see Box-Code Display below. Press PF3 or ENTER to return to the editing area.
	PF7	Scrolls up one page in the screen.
	PF8	Scrolls down one page in the screen.
	PF9	Switches between full-screen mode and split-screen mode.
	PF10	Moves to the left half of the screen if the map is wider than the screen.
	PF11	Moves to the right half of the screen if the map is wider than the screen.
	PF12	Undoes the last action performed in the editing area <i>before</i> ENTER is pressed.

Box-Code Display

The box-code display function invokes the **Box-Code Display** screen, which is intended for orientation and helps distinguish box codes from text with identical characters.

The **Box-Code Display** screen shows the box codes set and indicates the field positions by a number of periods (.). The screen does not show the E entered for the end marker.

The box-code display screen cannot be used for editing.

Example of a Box-Code Display:

The following example shows the box code settings of a map after PF4 is pressed. The example is based on [Example of Box Code Setting](#).

```

Outline Editor - Box-Code Display
Code BX  Code BX
T  _      B  _      I  =      C  [      1  [      3  [      5  [      U  [
L  |      R  |      H  | |      O  [      2  [      4  [      6  [      E  END
001  --010--+-----+-----030--+-----+-----050--+-----+-----070--+-----+
                                     B.....B.....
                                     .....  O.....
                                     .....  2.....
                                     .....  L.....R
                                     .....  U.....
                                     .....  O.....
                                     .....  C.....I.I.....1
.....  L .....  O.....
      L .....
      L .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit

```