Processing

This chapter covers the following topics:

- Data Verification
- Data Compression

Data Verification

ADACMP checks each field defined with format P (packed) or U (unpacked) to ensure that the field value is numeric and in the correct format. If a value is empty, the null characters must correspond to the format specified for the field (see *Representing SQL Null Values* in the *COMPRESS Essential Data Definition Syntax* section.

Alphanumeric (A)	blanks (hex ’40’)
Binary (B)	binary zeros (hex ’00’)
Fixed (F)	binary zeros (hex ’00’)
Floating Point (G)	binary zeros (hex ’00’)
Packed (P)	decimal packed zeros with sign (hex ’00’ followed by ’0F’, ’0C’, or ’0D’ in the rightmost, low-order byte)
Unpacked (U)	decimal unpacked zeros with sign (hex ’F0’ followed by ’C0’ or ’D0’ in the rightmost, low-order byte)

Any record that contains invalid data is written to the ADACMP error (DD/FEHL) dataset and is not written to the compressed dataset.

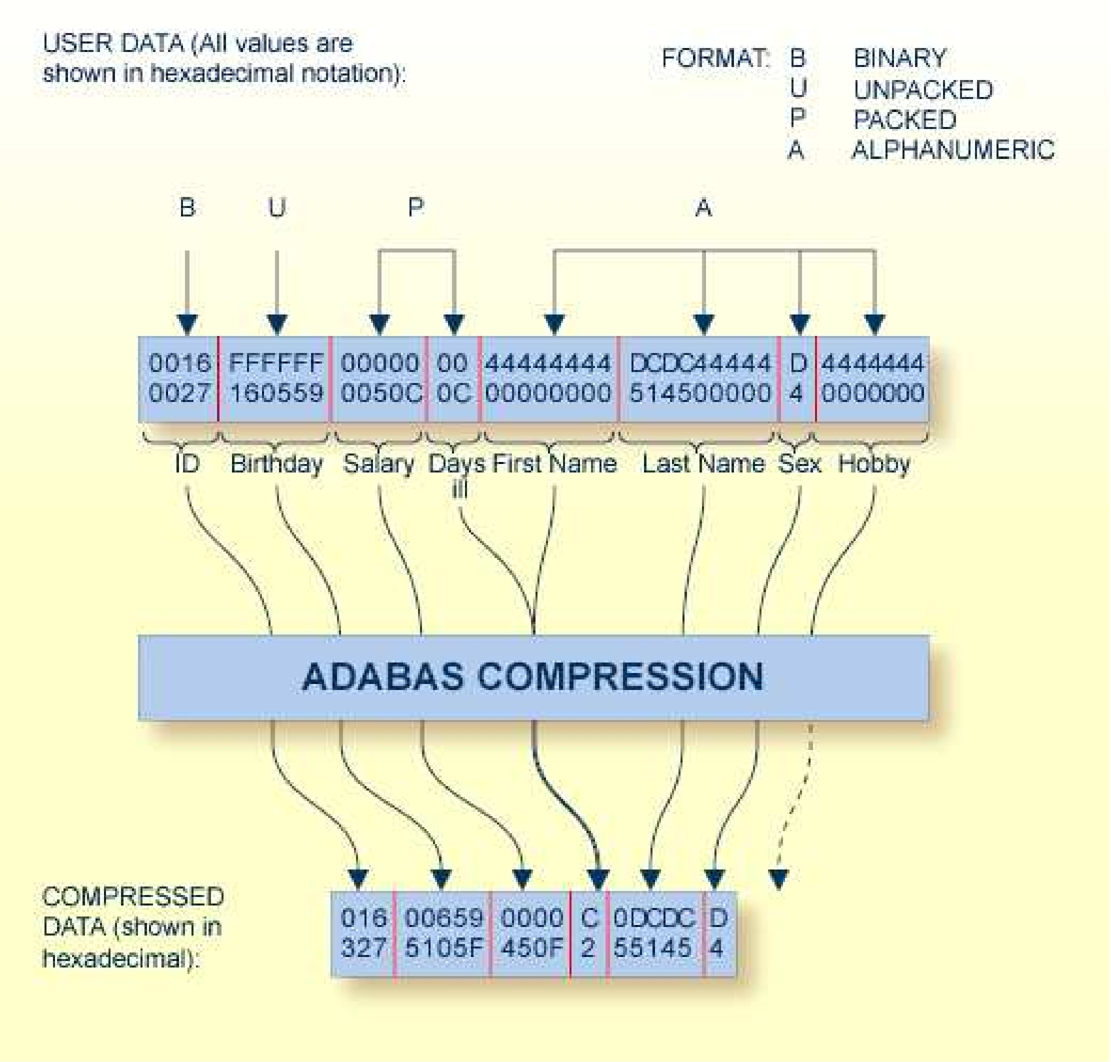
Data Compression

The value for each field is compressed (unless the FI option is specified) as follows:

- Trailing blanks are removed for fields defined with A format.
- Leading zeros are removed for numeric fields (fields defined with B, F, P or U format).
- If the field is defined with U (unpacked) format, the value is converted to packed (P) format.
- Trailing zeros in floating-point (G format) fields are removed.
- If the field is defined with the NU option and the value is a null value, a one-byte indicator is stored. Hexadecimal ’C1’ indicates one empty field follows, ’C2’ indicates that two empty fields follow, and so on, up to a maximum of 63 before the indicator byte is repeated. For SQL null value (NC option field) compression, see *Representing SQL Null Values* in the *COMPRESS Essential Data Definition Syntax* section.

- Empty fields located at the end of the record are not stored, and therefore not compressed.

Example of Data Compression



ADACMP Compression

The graphic shows how the following field definitions and corresponding values would be processed by ADACMP:

```
FNDEF='01,ID,4,B,DE'
FNDEF='01,BD,6,U,DE,NU'
FNDEF='01,SA,5,P'
FNDEF='01,DI,2,P,NU'
FNDEF='01,FN,9,A,NU'
FNDEF='01,LN,10,A,NU'
FNDEF='01,SE,1,A,FI'
FNDEF='01,HO,7,A,NU'
```