



z/OS Learning Center: Introduction to ISPF

Unit 1: The Basics of ISPF and Data Sets Module 3: ISPF Data Set Basics

```
l. 3444  
t. Score  
PrintNumber  
  
o. 219  
DrawShape  
  
ah, 1  
16h  
GetKey
```



```
xor di, di  
mov cx, 2000  
mov ax, 700h  
rep stosw  
  
call DrawBorders  
  
mov di, 184  
mov si, offset sNext  
call PrintText  
mov di, 272  
mov si, offset sHiScore  
call PrintText
```

```
mov al, 0- mov ah, 7  
Clear screen and set color 7  
  
mov di, 3430  
mov si, offset sStop  
call PrintText  
mov di, 450  
mov si, offset sSpeed  
call PrintText
```

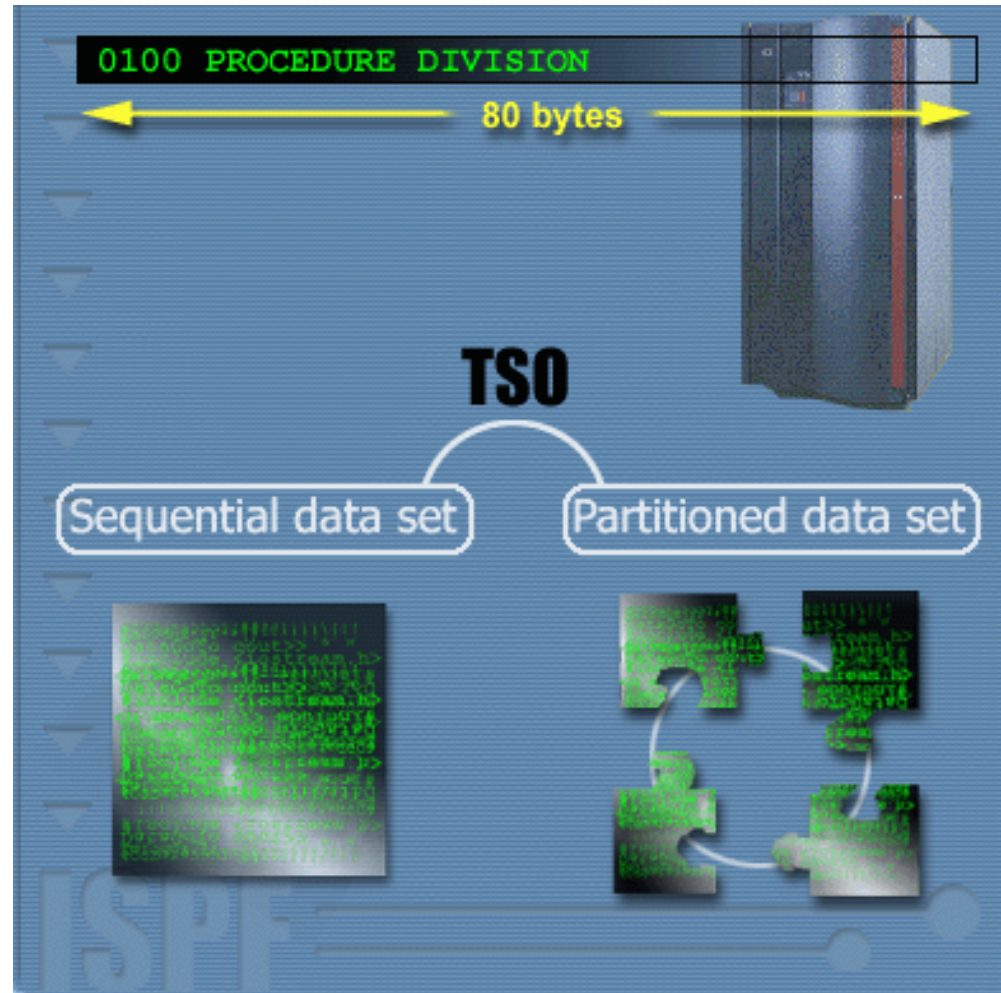
```
mov di, 292  
mov ax, HiScore  
call PrintNumber  
  
mov Score, 0  
  
call ChooseGame  
  
call Rand  
mov NextShape, ax  
call NewShape  
call DrawNextShape
```



Data Set Basics – Introduction

This module, ISPF Data Set Basics, introduces you to the two most common types of data sets, sequential and partitioned and discusses how data sets are named.

Time to complete: 10 – 15 minutes



Data Set Basics – Objectives

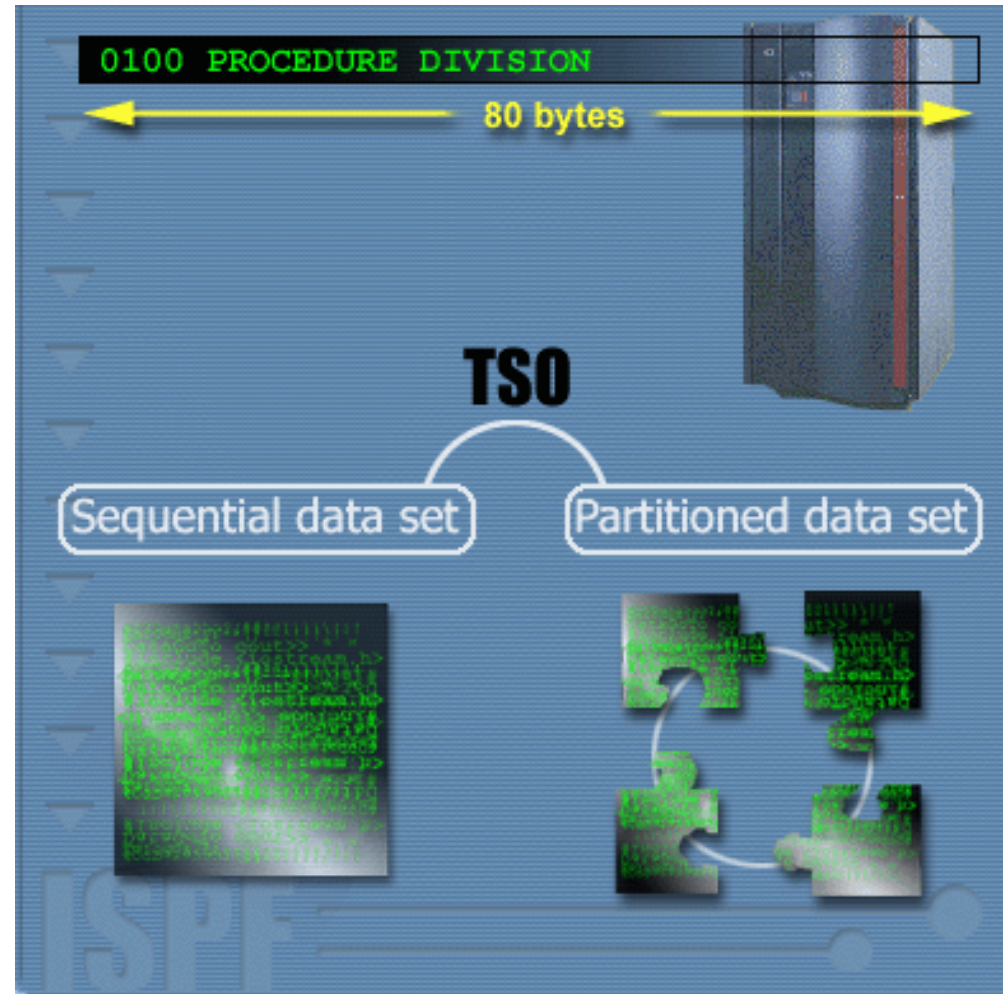
Upon completion of this module, you should be able to:

- **Describe the nature of sequential and partitioned data sets**
- **State one advantage to using a partitioned data set**
- **Identify the parts of a data set name**

Data Set Basics – Types of Data Sets

In the z/OS environment, files are known as data sets. TSO supports several types of data sets. The two most common types are sequential and partitioned.

JCL and program source code are maintained as files of 80-byte records. Every statement is a full 80 bytes long even if it is completely blank. This is for compatibility with older MVS systems from the era of 80-column punched cards. Data files can be of a different fixed length or variable length.



Data Set Basics – Sequential Data Sets

In a sequential data set such as a log file, transaction file, or a list file (as shown here) records are stored in physical order. New records are appended to the end of the data set.

The records in a sequential data set can be retrieved only in the same order they were written. To process a record somewhere within the file, the system must scan past all of the preceding records.

```

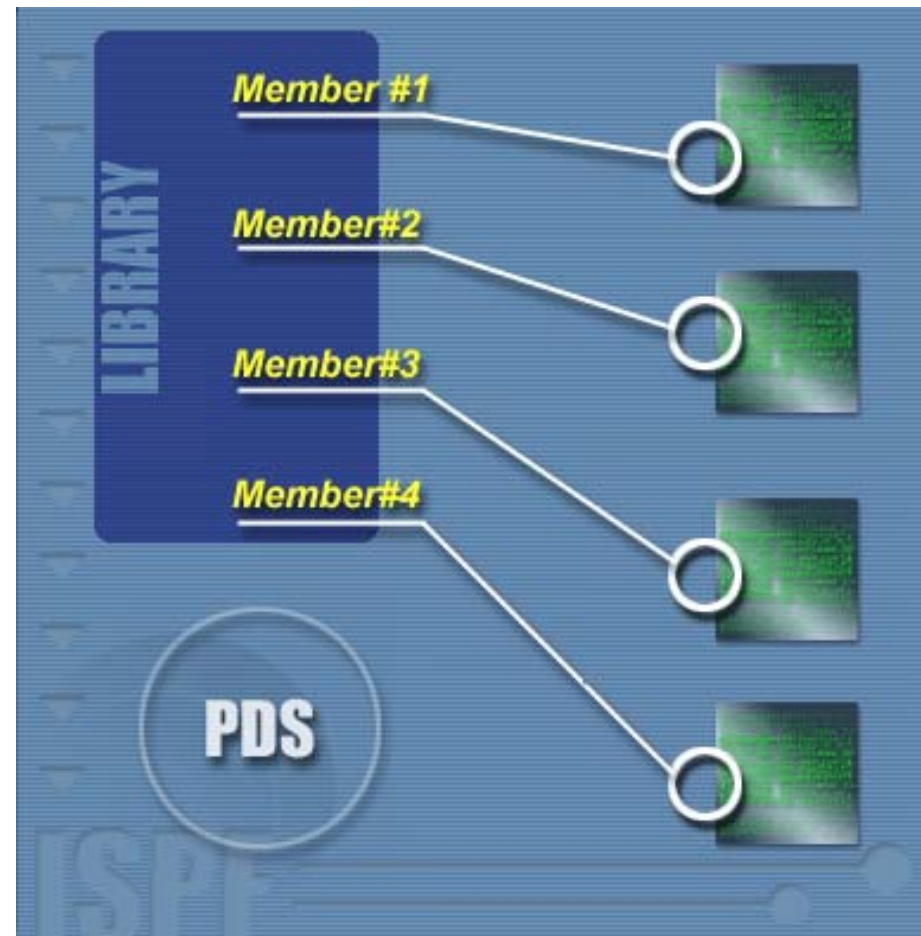
Menu  Utilities  Compilers  Help
BROWSE  SMCHUGH.DASD.LIST  Line 00000000 Col 001 080
Command ==>  Scroll ==> PAGE
***** Top of Data *****
3800 PKIODF 3390 PRIVATE
3801 TOTSP5 3390 PRIVATE
3802 TOTDBP 3390 PRIVATE
3803 KMTRV1 3390 PRIVATE
3804 KMTRV2 3390 PRIVATE
3805 NW3805 3390 PRIVATE
3806 NW3806 3390 PRIVATE
3807 SCAR04 3390 PRIVATE
3808 WSDS08 3390 PRIVATE
3809 NW3809 3390 PRIVATE
380A NW380A 3390 PRIVATE
380B NW380B 3390 PRIVATE
380C 010SM3 3390 PRIVATE
380D BS0X37 3390 PRIVATE
380E TOTSTC 3390 PRIVATE
380F XSMS04 3390 PRIVATE
3810 NW3810 3390 PRIVATE
F1=Help F2=Split F3=Exit F5=Rfind F7=Up F8=Down F9=Swap
F10=Left F11=Right F12=Cancel
MF a 04/015

```


Data Set Basics – Partitioned Data Sets

Every file processed by ISPF is stored either on a host as a sequential data set or a member of a partitioned data set, or as a workstation file.

A partitioned data set (PDS) consists of a directory and one or more members. A PDS is also called a library. Each member is functionally the same as a sequential data set, and you can process an individual member of a partitioned data set as if it were a sequential data set. In addition, there are some operations—such as print, delete, rename, and compress—that you can perform on the entire PDS as if it were a single file.



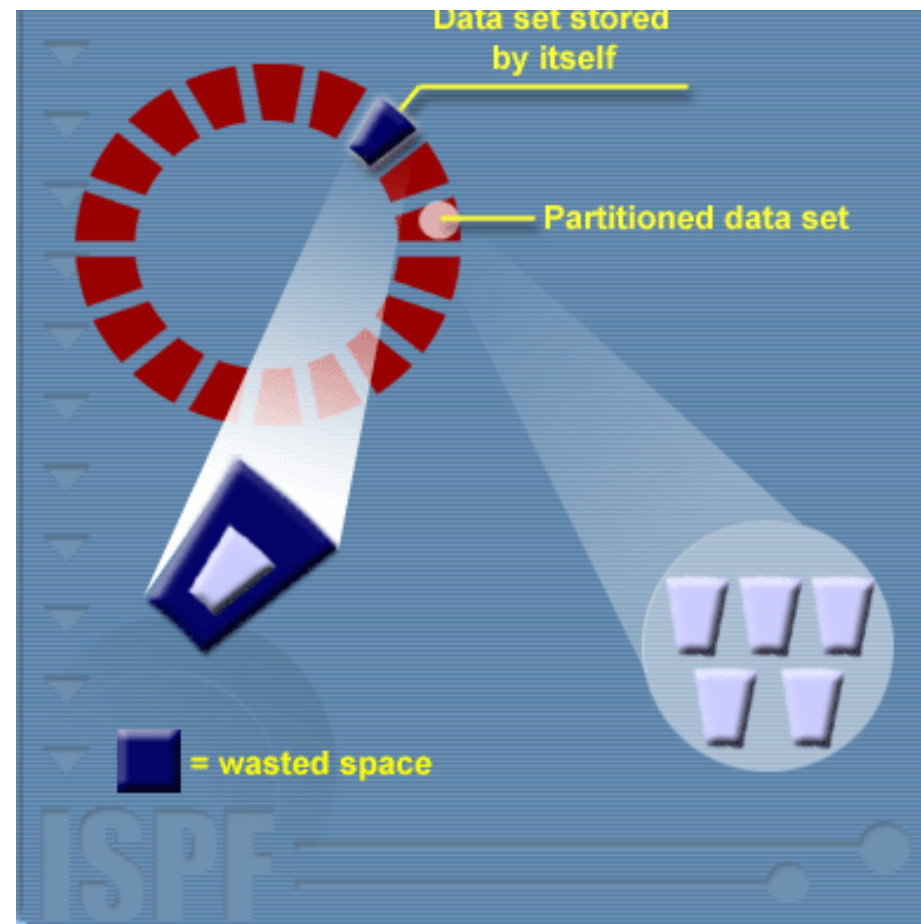
Data Set Basics – Partitioned Data Set Members

Partitioned data sets are good for keeping like things together. For example, all of your panel definitions could be members of a partitioned data set. Another PDS could contain all the message definitions, and another could contain all the programs in your application.

Each partitioned data set contains a directory and data area. The directory maintains housekeeping information about the members stored in the data area. It stores an index of member names and their locations.

Data Set Basics – Advantages of Partitioning

The minimum allocation for a data set is one track of disk space, and this can usually amount to a significant amount of space. For a small file that contains only a small amount of information, much space can be wasted if it is stored in a data set all by itself. Instead it can be stored as one of many members in a partitioned data set. This is how most source code and JCL is stored under z/OS.



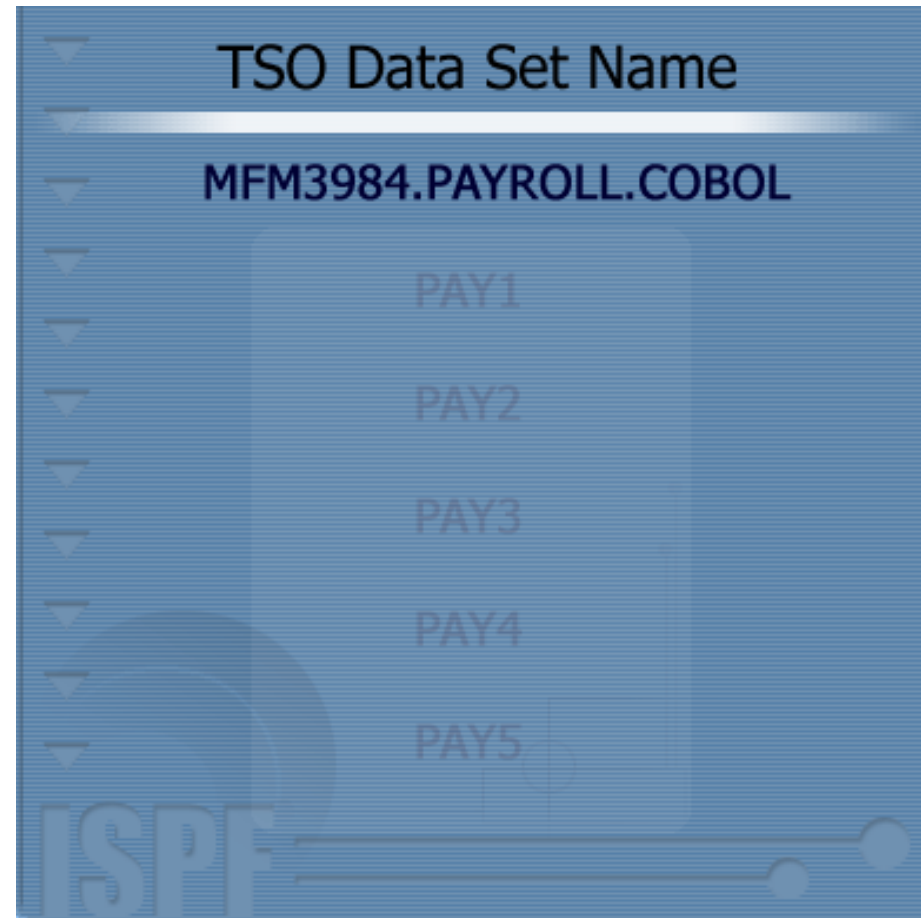
Data Set Basics – Parts of the TSO Data Set Name

Standard TSO data set naming calls for three qualifiers within the name of the data set:

- High-level Qualifier
- User Determined Qualifier
- Type of Data

Qualifiers can be up to eight characters long. When you specify a data set name, you separate the qualifiers by periods. A data set name can be up to 44 characters long, including periods.

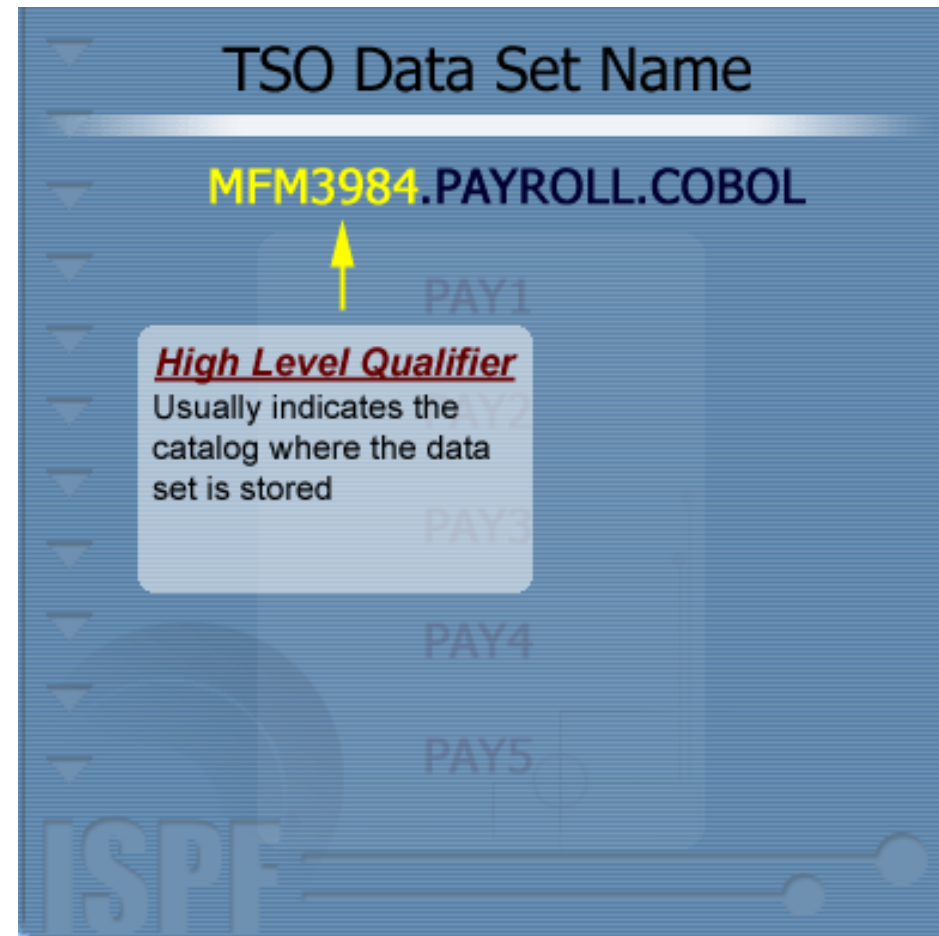
An example of a data set name is shown here.



Data Set Basics – High-level Qualifier

The first part of the data set name is the high-level qualifier. The high-level qualifier is also referred to as the project.

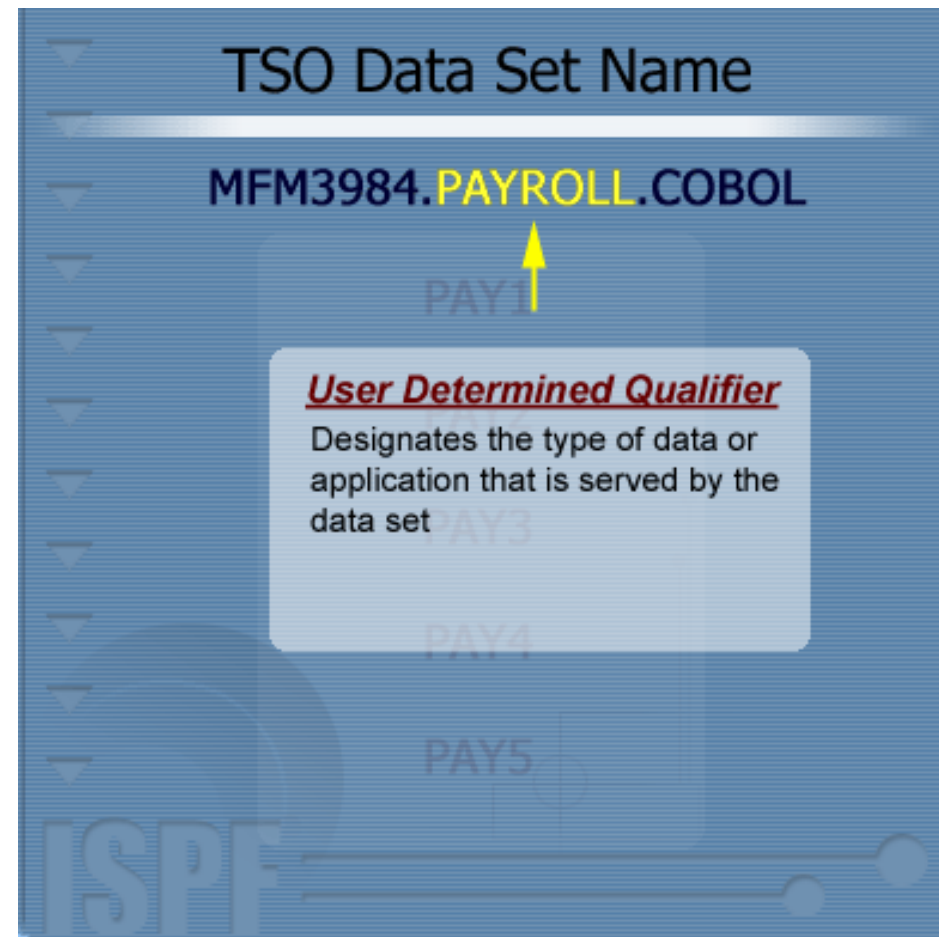
The project is the common identifier for all ISPF libraries belonging to the same programming project. The high-level qualifier is either the name of a project that is predefined in the MVS master catalog, or your TSO userid.



Data Set Basics – User Determined Qualifier

The second part of the data set name is the user determined qualifier, which is also referred to as the group.

The group identifies a particular set of ISPF libraries, that is, the level of the libraries within the library hierarchy. For example, if you need to keep separate copies of a software project for development, testing, and production, the user determined qualifiers for the three data sets might be DEV, TEST, and PROD, respectively.



Data Set Basics – Type of Data Qualifier

The third part of the data set name is the type qualifier.

The type identifies the type of information in the ISPF library, such as the source code language (PL/I, COBOL, C, for example). Other examples are **PANELS** for panel definitions and **MSGS** for message definitions.

Some types of qualifiers are shown here.

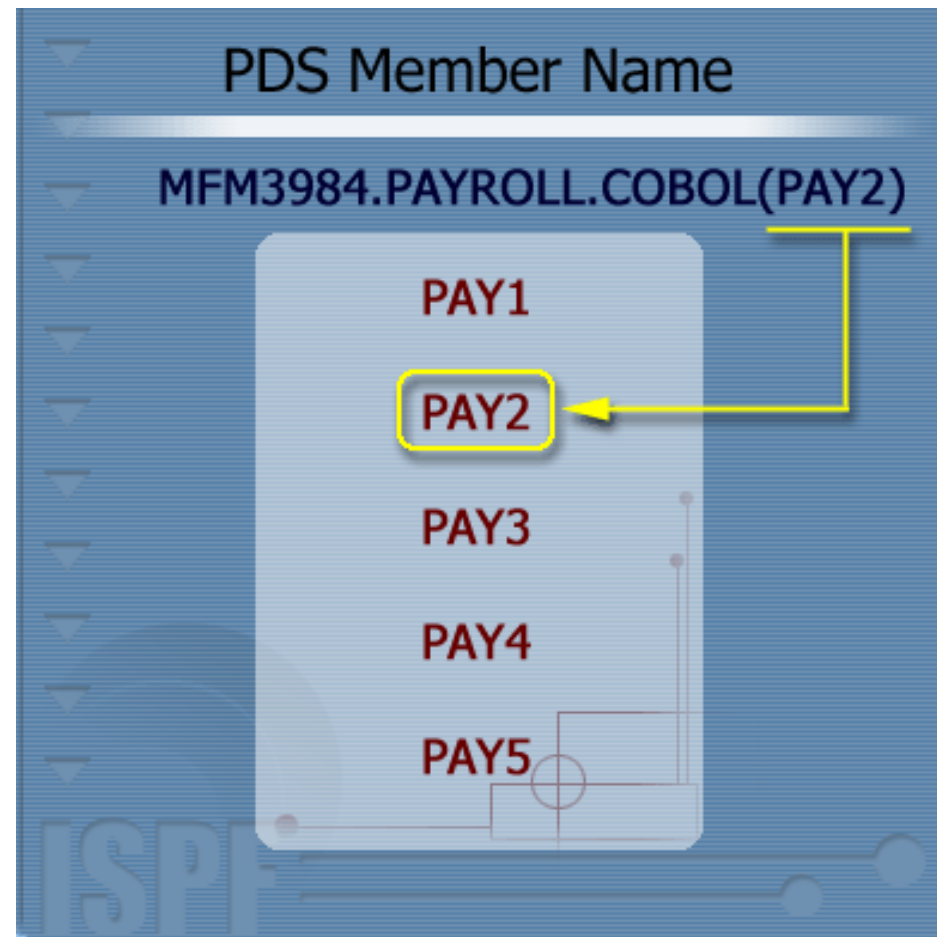
The diagram illustrates the structure of a TSO Data Set Name. At the top, the text "TSO Data Set Name" is displayed. Below it, the example name "MFM3984.PAYROLL.COBOL" is shown. The "COBOL" portion is highlighted in yellow, and a yellow arrow points to it from a list of qualifiers below. The list of qualifiers is organized into two columns. Each entry consists of a red underlined code followed by its description.

TSO Data Set Name	
MFM3984.PAYROLL.COBOL	
<u>ASM</u>	Assembler language source code
<u>CLIST</u>	CLIST procedure
<u>CNTL</u>	JCL job stream used for batch job facility
<u>COBOL</u>	COBOL source code
<u>DATA</u>	Uppercase text data
<u>EXEC</u>	REXX procedure
<u>FORT</u>	FORTRAN source code
<u>LOAD</u>	Executable program module
<u>OBJ</u>	Object module
<u>PLI</u>	PL/I source code
<u>TEXT</u>	Upper and lowercase text data

Data Set Basics – PDS Member Name

In order to deal with a specific member of a partitioned data set, the member name is placed at the end of the data set name inside parentheses.

The data set name shown at the right is a partitioned data set. The member name, which is the name in parentheses, can be up to eight characters long, and must start with a letter.

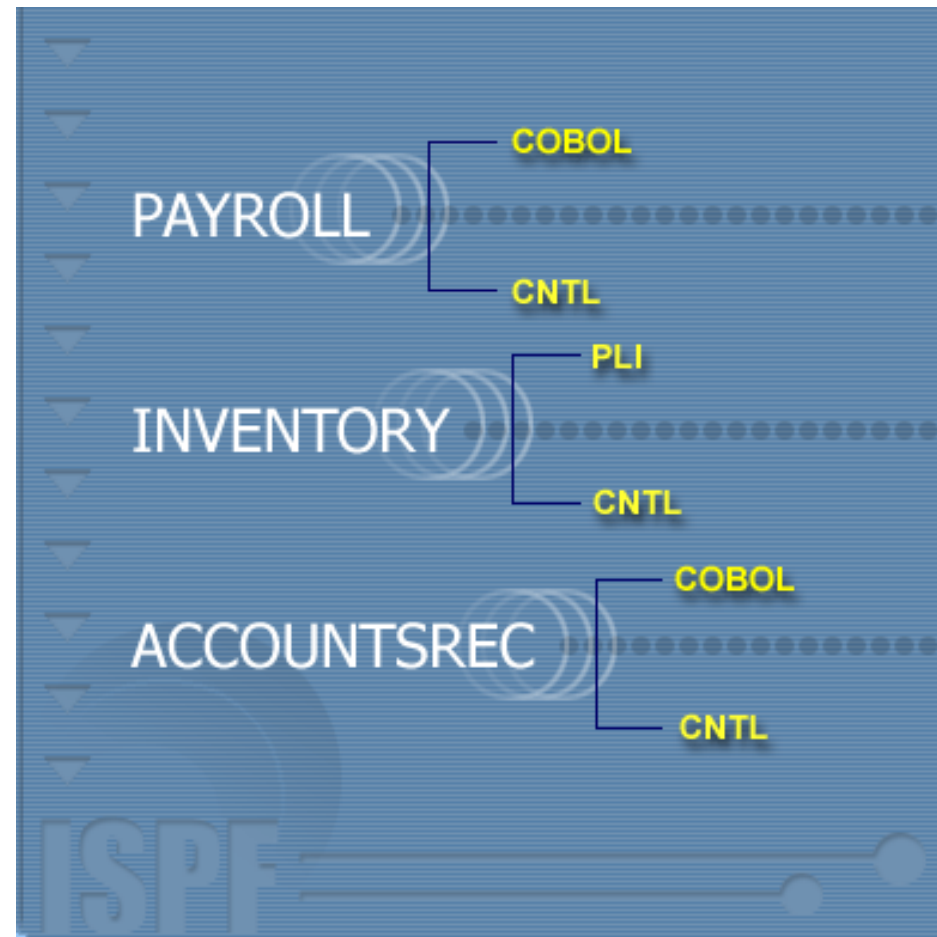


Data Set Basics – Managing the PDS

Programmers usually maintain several PDSs for a project. One may be for source code and include the code language as the third qualifier (COBOL, C, FORT, PLI, or ASM, for example). The standard names cause ISPF to default to convenient screen presentation formats for each language.

Another PDS may be for object code, and have OBJ for the third qualifier, while another may contain the JCL statements and have CNTL as the third qualifier.

For a large development project, you might have dozens of member types including PANELS, MSGS, SKELS, ASM, EXEC, CLIST, and LOAD to name just a few.



Data Set Basics – Summary

In this unit, Data Set Basics, you learned:

- **z/OS lets you maintain two types of data sets under TSO: sequential and partitioned.**
- **In a sequential data set, the records of a file can be retrieved only in the same order they were written.**
- **A partitioned data set (PDS) consists of a directory and one or more members. A PDS is also called a library.**
- **The parts of the data set name are separated by periods and include the project, group, and type.**
- **The PDS member name is included in parentheses after the data set name.**