



## Data Flow Diagram with Examples - Video Rental System Example

Written Date : February 16, 2015

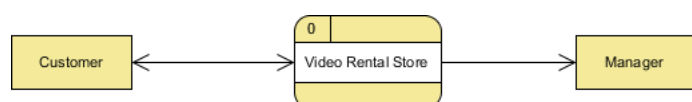
Data Flow Diagram (DFD) provides a visual representation of the flow of information (i.e. data) within a system. By drawing a Data Flow Diagram, you can tell the information provided by and delivered to someone who takes part in system processes, the information needed in order to complete the processes and the information needed to be stored and accessed. Data Flow Diagram has a wide use in software engineering. This article describes and explain Data Flow Diagram (DFD) by using a video rental system as an example.

---

### The Video Rental System Example

#### Context DFD

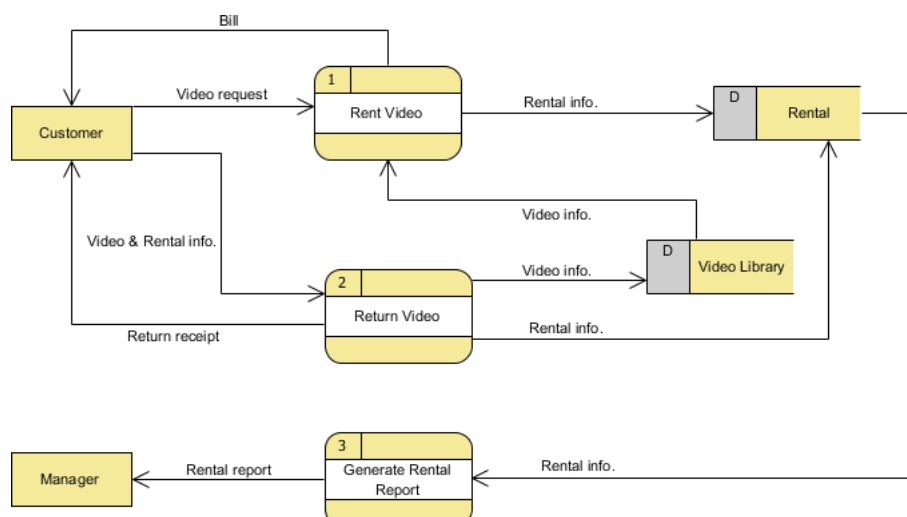
The figure below shows a context Data Flow Diagram that is drawn for a video rental system. It contains a process (shape) that represents the system to model, in this case, the "*Video Rental Store*". It also shows the participants who will interact with the system, called the external entities. In this example, there are two external entities, namely *Customer* and *Manager*. In between the process and the external entities, there are data flow connectors indicating the existence of information exchange between customer and the system.



Context DFD is the entrance of a data flow model. It contains one and only one process and does not show any data store, which makes the diagram simple.

#### Level 1 DFD

The figure below shows the level 1 DFD, which is the decomposition (i.e. break down) of the video rental system that is shown in the context DFD. Read through the diagram and then we will introduce some of the key concepts based on this diagram.



The Video Rental System Data Flow Diagram example contains three processes, two external entities and two data stores. Although there is no design guideline that governs the positioning of shapes in a Data Flow Diagram, we tend to put the processes in the middle and data stores and external entities on the sides to make it easier to comprehend.

Based on the diagram, we know that a *Customer* makes a *Video request* to the *Rent Video* process. The *Rent Video* process also receives *Video info.* from the *Video Library* data store. As a result, the process produces a *Bill* to the *Customer*, and stores the *Rental info.* into the *Rental* data store.

A *Customer* can *Return Video* by providing *Video & Rental info.* The process stores the *Video info.* into the *Video Library* data store and *Rental info.* into the *Rental* data store. As a result, *Return receipt* is delivered to the *Customer*. Although we said that the receipt is delivered as a result of the *Return Video* process, the Data Flow Diagram implies no such thing. It is our common sense that lead us to interpret the diagram in the way that we understand it naturally. Strictly speaking, the diagram only tells us the *Return Video* process receives *Video & Rental info.* and produces *Video info.*, *Rental info.*, and *Return receipt*, with no order specified. Note that Data Flow Diagram does not answer in what way and in what order the information is being used throughout a system. If this information is important and worth mentioning, consider to model it with diagrams like [BPMN Business Process Diagram](#) or [UML Activity Diagram](#).

Finally, a *Manager* can receive *Rental report* from the *Generate Rental Report* process and the information involved is provided by the *Rental* data store.

## Data Flow Diagram Tips and Cautions

Be aware of the level of details

In this Data Flow Diagram example, the word "info" is used many times when labeling data. We have "rental info" and "video info". What if we write them explicitly as "rental date, video rented, person rent", "video id, video name and video status"? Is this correct? Well, there is no definite answer to this question but try to ask yourself a question when making a decision. Why are you drawing a DFD?

In most cases, Data Flow Diagram is drawn in the early phase of system development, where many details are yet to be confirmed. The use of general terminologies like "details", "information", "credential" certainly leave room for discussion. However, using general terms can be kind of lacking details and make the design lost its usefulness. So it really depends on the purpose of your design.

**Don't mix up data flow and process flow**

Some designers may feel uncomfortable when seeing a connector connecting from a data store to a process, without seeing the step of data request being shown on the diagram somehow. Some of them will try to represent a request by adding a connector between a process and a data store, labeling it "a request" or "request for something", which is wrong.

Keep in mind that Data Flow Diagram was designed for representing the exchange of information. Connectors in a Data Flow Diagram are for representing data, not for representing process flow, step or anything else. When we label a data flow that ends at a data store "a request", this literally means we are passing a request as data into a data store. Although this may be the case in implementation level as some of the DBMS do support the use of functions, which intake some values as parameters and return a result, in Data Flow Diagram, we tend to treat data store as a sole data holder that does not possess any processing capability. If you want to model the system flow or process flow, use [UML Activity Diagram](#) or [BPMN Business Process Diagram](#) instead. If you want to model the internal structure of data store, use [Entity Relationship Diagram](#).

**Resources**

1. [Video-Rental-Store.vpp](#)



[Visual Paradigm home page](https://www.visual-paradigm.com/)  
(<https://www.visual-paradigm.com/>)

[Visual Paradigm tutorials](https://www.visual-paradigm.com/tutorials/)  
(<https://www.visual-paradigm.com/tutorials/>)