**Introduction:**

  A data flow diagram (DFD) maps out the flow of information for any

process or system. It uses defined symbols like rectangles, circles and arrows, plus

short text labels, to show data inputs, outputs, storage points and the routes

between each destination. Data flowcharts can range from simple, even hand-

drawn process overviews, to in-depth, multi-level DFDs that dig progressively

deeper into how the data is handled.

They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow

software and systems, they are less applicable nowadays to visualizing interactive,

real-time or database-oriented software or systems.

**Symbols and notations used in DFD’s:**

Three common systems of symbols are named after their creators:

* Yourdon and Coad
* Yourdon and DeMarco
* Gane and Sarson

One main difference in their symbols is that Yourdon-Coad and Yourdon-

DeMarco use circles for processes, while Gane and Sarson use rectangles with

rounded corners, sometimes called lozenges. There are other symbol variations in

use as well, so the important thing to keep in mind is to be clear and consistent in

the shapes and notations you use to communicate and collaborate with others.

Using any convention’s DFD rules or guidelines, the symbols depict the four

components of data flow diagrams.

**1. External entity:** An outside system that sends or receives data,

communicating with the system being diagrammed. They are the sources and

destinations of information entering or leaving the system. They might be an

outside organization or person, a computer system or a business system. They

are also known as terminators, sources and sinks or actors. They are typically

drawn on the edges of the diagram.

**2. Process:** any process that changes the data, producing an output. It might

perform computations, or sort data based on logic, or direct the data flow

based on business rules. A short label is used to describe the process, such

as“Submit payment.”

**3. Data store:** files or repositories that hold information for later use, such as a

database table or a membership form. Each data store receives a simple

label, such as “Orders.”

**4. Data flow:** the route that data takes between the external entities, processes

shown with arrows, typically labeled with a short data name, like “Billing

details.”

**DFD rules and tips:**

* Each process should have at least one input and an output.
* Each data store should have at least one data flow in and one data flow out.
* Data stored in a system must go through a process.
* All processes in a DFD go to another process or a data store.

**How to make data flow diagram:**

You can create your own DFD online with Lucidchart. Use our DFD

examples and specialized notations to visually represent the flow of data

through your system. Our data flow diagram maker is simple, yet powerful.

Get started with a template, and then use our shapes to customize your

processes, data stores, data flows and external entities.

Data flow diagramming is quick and easy with Lucidchart. Start a free trial today

to start creating and collaborating.

**Examples of how DFDs can be used:**

Data flow diagrams are well suited for analysis or modeling of various types of

systems in different fields.

DFD in software engineering: This is where data flow diagrams got their main start

in the 1970s. DFDs can provide a focused approached to technical development, in

which more research is done up front to get to coding.

DFD in business analysis: Business analysts use DFDs to analyze existing systems

and find inefficiencies. Diagramming the process can uncover steps that might

otherwise be missed or not fully understood.

DFD in business process re-engineering: DFDs can be used to model a better,

more efficient flow of data through a business process. BPR was pioneered in the

1990s to help organizations cut operational costs, improve customer service and

better compete in the market.

DFD in agile development: DFDs can be used to visualize and understand business

and technical requirements and plan the next steps. They can be a simple yet

powerful tool for communication and collaboration to focus rapid development.

DFD in system structures: Any system or process can be analyzed in progressive

detail to improve it, on both a technical and non-technical basis.

|  |  |
| --- | --- |
|  |  |