

Getting Started with Inform 7

PART ONE: Creating Your World

The map, rooms, pathways, room descriptions, properties, and traveling messages.

ROOM

The room object is the core of IF as it creates geographic locations (indoors or outdoors) that are part of a map.

Rooms can be connected by 12 possible directions: north, south, east, west, northwest, northeast, southwest, southeast, in (inside of), out (outside of), up (up from), or down (down from).

The first room is created like this:

```
The Foyer is a room.
```

Note too that the first room asserted will also be the players starting location.

PATHWAYS (the 8 cardinal compass directions)

We create (and connect) additional rooms by stating the new room's directional relationship to a previously declared room:

```
The Kitchen is west of the Foyer.
```

or

```
The Kitchen is a room. It is west of the Foyer.
```

or

```
A room called the Kitchen is west of the Foyer.
```

or

```
West of the Foyer is the Kitchen.
```

This creates what we call a **pathway** that Inform will understand as a two-way connection. In this example, going west will take us from the Foyer room to the Kitchen room and going east from the Kitchen room will take us back to the Foyer room.

PATHWAYS (up and down & out and in)

We can also move up, down, out, or in. The wording for these directions is worded slightly different:

```
The Attic is up from the Foyer.
The Basement is down from the Foyer.
The Closet is inside from the Foyer.
The Front Porch is outside from the Foyer.
```

DESCRIBING THE ROOMS

Room descriptions (the text displayed to the player when they first enter a room or whenever the player types 'look' or 'L') are created by placing the description between double quotations immediately following the room declaration.

```
The Foyer is a room. "The Foyer is an austere space containing
no furnishings. You can see exits east, south, and stairs that
lead up."
```

```
The Kitchen is west of the Foyer. "The kitchen is a modern
design and is remarkably clean. There is an exit west."
```

Room descriptions are (by default) only displayed the first time the player enters a room. After that, room descriptions will be displayed only if the player explicitly types 'look' or 'L'.

It should also be noted that seasoned players know that they can override this default behavior by typing 'verbose' at any time during play. This will always display room descriptions even if they've been there before. Typing 'brief' (or 'terse') will revert the game back to its default mode of displaying room descriptions.

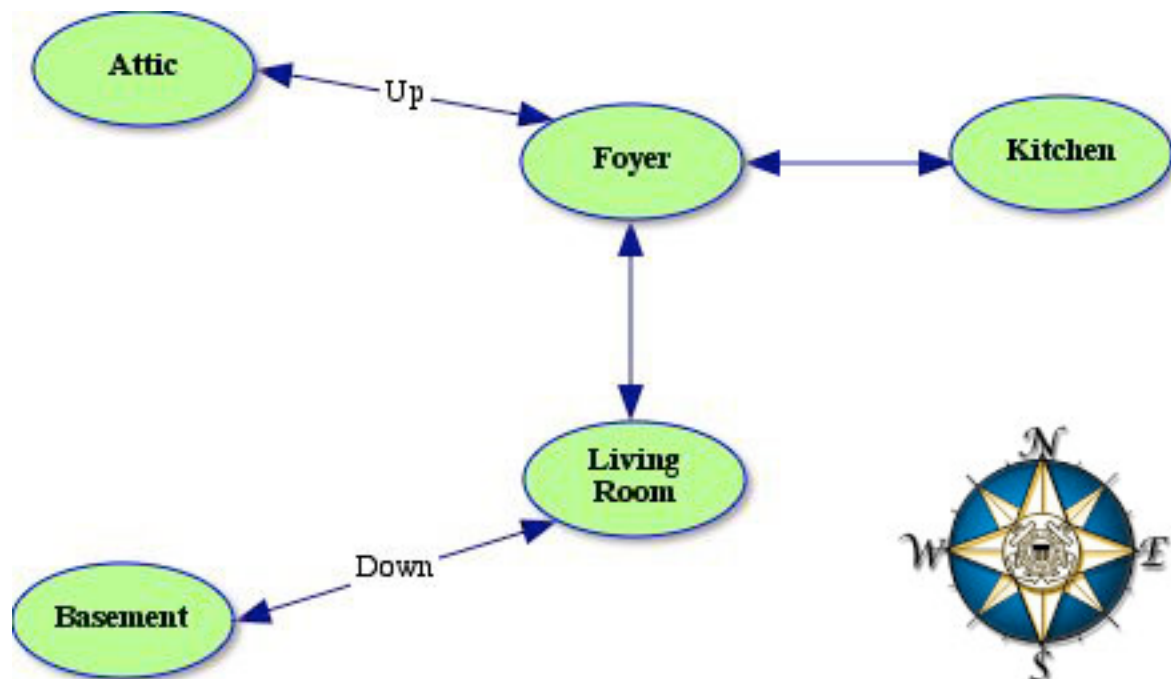
You may wish to override this default behavior so that room descriptions will always be displayed by placing the following command near the beginning of your code:

```
Use full-length room descriptions.
```

This is purely optional but may be helpful when testing your game. Just remember to remove it before releasing your game.

A QUICK EXAMPLE

THE MAP:



THE CODE:

"A Simple Map" by John Timmons

The Foyer is a room. "The Foyer is an austere space containing no furnishings. You can see exits east, south, and stairs that lead up."

The Kitchen is east of the Foyer. "The kitchen is of a modern design and is remarkably clean. There is an exit west."

The Living Room is south of the Foyer. "Tastefully furnished, the living room seems like a very relaxing place. There is an exit north and stairs leading down."

The Attic is up from the Foyer. "This attic is filthy and has a strong musky odor. Stairs lead down from here."

The Basement is down from the Living Room. "The basement is comprised of concrete walls and floor illuminated by a solitary light bulb. Stairs lead up from here."

COMPLICATING THE MAP

This section deals with more sophisticated (and complex) methods of creating your story world. These are purely optional but you may find them helpful in generating ideas for more interesting maps.

PROPERTIES

Room objects have, by default, a property of **lit**. This means that rooms always have light and we will be able to see the room description and any other objects located within the room.

Should you wish to create a room that is in total darkness do this:

```
The Foyer is a dark room. "The Foyer is an austere space
containing no furnishings. You can see exits east, south, and
stairs that lead up."
```

or

```
The Foyer is a room. It is dark. "The Foyer is an austere
space containing no furnishings. You can see exits east,
south, and stairs that lead up."
```

Here is how this room will look no matter what the room description says:

Darkness

It is pitch dark, and you can't see a thing.

>

We will discuss ways for the player to illuminate a dark room in a later chapter.

ONE-WAY EXITS

Sometimes we may want to create a pathway that cannot be retraced (a one-way connection.)

We can accomplish this by the following:

The Foyer is a room. "The Foyer is an austere space containing no furnishings. You can see exits east, south, and stairs that lead up."

The Kitchen is west of the Foyer. **East of the Kitchen is nowhere.** "The kitchen is of a modern design and is remarkably clean. There is an exit west."

Now when the player goes west from the Foyer into the Kitchen they will not be able to go back east to the Foyer as they normally would.

Here is how this would look in play:

Foyer

The Foyer is an austere space containing no furnishings. You can see exits east, south, and stairs that lead up.

>west

Kitchen

The kitchen is of a modern design and is remarkably clean.

>east

You can't go that way.

>

This is a situation where the player enters a room and the door slams shut behind them or they go down stairs which collapse preventing them from going back the way they came. Now the player will have to find (and the author must create) another way out.

"MAGIC" EXITS

Pathways can be configured to override the default two-way connections.

In the following example, you can enter the Closet but, when you return the way you came in, you will find yourself in another room.

The Foyer is a room. "This is an austere place barren of any furniture. There is an exit to the north and to the west is a closet."

The Kitchen is north of the Foyer. "Red walls, marble counter tops and a silver stove, this kitchen reminds you of The Jetsons."

The Living Room is south of the Foyer. "This living room looks like it came right out of a JC Penney catalog."

The Closet is west of the Foyer. **East of the Closet is the Living Room.** "Nothing here but a bare light bulb and one wire coat hanger."

Here is how it might look in play:

Foyer

This is an austere place barren of any furniture. There is an exit to the north and to the west is a closet.

>west

Closet

Nothing here but a bare light bulb and one wire coat hanger.

>east

Living Room

This living room looks like it came right out of a JC Penney catalog.

>

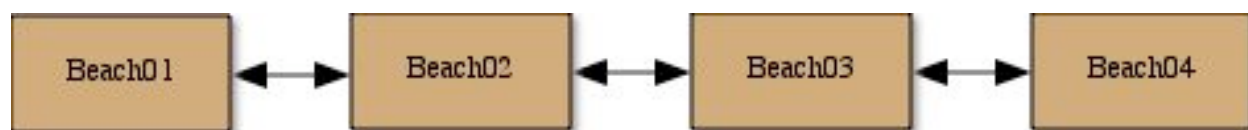
VAST SPACES

Creating the illusion of a large space (like a cathedral, a football field, a meadow, a stretch of beach, a long hallway, etc.) is a common issue in creating IF maps. One way to create the reality of traversing a large space over time is to subdivide that space into smaller 'rooms.' The desired effect is produced through a combination of room descriptions and something new called the 'printed name' property.

The Printed Name Property

Rooms have a property called 'printed name.' This property allows you to display a name for the room that is different from the coded name of the room.

Consider this map that outlines a stretch of beach running east to west:



What we are going for is something that looks like this in play:

On the Beach

The beach continues to the west.

>go west

On the Beach

Off to the west, you can see some kind of structure rising above the horizon. The beach continues east and west.

>go west

On the Beach

Just to the west, you can see a light house. The beach continues east and west.

>go west

On the Beach

An immense light house stands at the edge of the beach to your north. The beach continues to the east.

>

Note how the room description changes when the player moves into another room but the short description of the room remains "On the Beach".

For example:

```
The Foyer is a room. "This room is an austere space. You can
see stairs leading up."
```

will display like this:

```
Foyer
```

```
This room is an austere space. You can see stairs leading up.
```

```
>
```

Now, look carefully at this code:

```
The Foyer is a room with printed name "The Gilded Entryway".
"This room is an austere space. You can see stairs leading
up."
```

This will produce the following during play:

```
The Gilded Entryway
```

```
This room is an austere space. You can see stairs leading up.
```

```
>
```

The coded room name is still 'Foyer' but the player will see it as 'The Gilded Entryway'. Understanding this printed name property will be integral to creating the illusion of traversing a vast space.

Here is the basic code structure for the map of a stretch of beach using the printed name property:

```
Beach01 is a room with printed name "On the Beach".
```

```
Beach02 is a room with printed name "On the Beach". It is west
of Beach01.
```

```
Beach03 is a room with printed name "On the Beach". It is west
of Beach02.
```

```
Beach04 is a room with printed name "On the Beach". It is west
of Beach03.
```

Here is how this basic code will look in play:

```
On the Beach
```

```
>go west
```

```
On the Beach
```

```
>go west
```

```
On the Beach
```

```
>go west
```

```
On the Beach
```

```
>
```

At this point, the map is minimal and appears that the player is not moving. What is important to understand is that the displayed room names remain constant. Each room has to have a unique code name (so Inform knows the difference) but we have also told Inform to print out the same name for each of those rooms. This is the significance of the printed name property: we want it to appear that the player is still within the same (vast) area.

Let's embellish the map further with room descriptions and (hopefully) things should start to make sense.

```
OnBeach01 is a room with printed name "On the Beach". "The beach continues to the west."
```

```
OnBeach02 is a room with printed name "On the Beach". It is west of OnBeach01. "Off to the west, you can see some kind of structure rising above the horizon. The beach continues east and west."
```

```
OnBeach03 is a room with printed name "On the Beach". It is west of OnBeach02. "Just to the west, you can see a light house. The beach continues east and west."
```

```
OnBeach04 is a room with printed name "On the Beach". It is west of OnBeach03. "An immense light house stands at the edge of the beach to your north. The beach continues to the east."
```

Here is how it will now look in play:

On the Beach

The beach continues to the west.

>go west

On the Beach

Off to the west, you can see some kind of structure rising above the horizon. The beach continues east and west.

>go west

On the Beach

Just to the west, you can see a light house. The beach continues east and west.

>go west

On the Beach

An immense light house stands at the edge of the beach to your north. The beach continues to the east.

>

The important point here is that, although it appears that the player is always 'on the beach,' it is the room descriptions that are providing information as to where the player is and what direction the player can or should go. In other words, **it is the room descriptions themselves that makes each room unique to the player.**

TRAVELING MESSAGES

A traveling message refers to some information the player will see as they are moving from one room to another.

Here is an example of this:

Foyer

This room is an austere space. You can see stairs leading up.

>go up

Carefully testing each step for it's sturdiness, you slowly make your way up the stairs.

Attic

This place is filthy with a strong musky odor. Stairs lead down from here.

>

In this example, the player has some tactile sense of the stairs they are traversing instead of just instantly moving from the Foyer to the Attic. It may also provide a clue about the construction of the stairs in this place that may come into play later.

On the Landing

The castle's sheer scale intimidates you. You can see the entrance to the castle north of here.

>go north

Stepping inside, the large noisy crowd gathered here suddenly grows silent at your arrival. A young boy emerges from among the hushed spectators and strolls deliberately towards you.

Main Entrance to the Castle

This immense hall seems to stretch forever. The walls are covered with painted portraits of stern-looking individuals alongside mounted animal heads.

A young boy stands facing you.

>

This second example shows how a traveling message can create a mood and "set the stage" for events to come.

We will create these traveling messages by employing the 'report rule.'

The Report Rule

The report rule simply means to report something to the player when a certain action occurs. In other words, display some text on the screen after the player specifically does something.

Here is how the code looks in the first example about climbing the stairs:

```
Report going up:
  say "Carefully testing each step for it's sturdiness,
  you slowly make your way up the stairs."
```

This says: when the player types 'go up', print out this message.

Let's break this down.

We type 'report' then the direction (going east, going out, etc.) and then a colon.

```
Report going up:
```

On the next line we type 'say' (which means to display on the screen) and then what we want to be displayed (including the concluding period) within double quotations.

```
  say "Carefully testing each step for it's sturdiness,
  you slowly make your way up the stairs."
```

So, whenever the player travels up display this text.

Here is how the code looks in the second example about entering the castle:

```
Report going north:
  say "Stepping in, the large noisy crowd gathered here
  suddenly grows silent at your appearance. A young boy
  emerges from the hushed spectators and strolls
  deliberately towards you."
```

Here are all twelve possibilities for report rules for traveling messages:

Report going east:	Report going west:	Report going north:	Report going south:
Report going northeast:	Report going northwest:	Report going southeast:	Report going southwest:
Report going up:	Report going down:	Report going in:	Report going out:

Seems pretty simple right? This will work but there is a potential problem. These report rules will be invoked whenever the player moves up or moves north respectively anywhere in the game. Probably not what we would want to happen.

From our first example, we can make it specific to the room:

```
Report going up from the Foyer:
  say "Carefully testing each step for it's sturdiness,
  you slowly make your way up the stairs."
```

or in the second example:

```
Report going north from On the Landing:
  say "Carefully testing each step for it's sturdiness,
  you slowly make your way up the stairs."
```

The addition of the phrase 'from the [room]' narrows the occurrence of the report rule down to a specific direction from a specific room.

But what if this is a pathway that the player must traverse several times? Is the message necessary each time? Probably not.

Here is how we can prevent this redundancy:

```
Report going up from the Foyer the first time:
  say "Carefully testing each step for it's sturdiness,
  you slowly make your way up the stairs."
```

The addition of the phrase 'the first time' tells Inform to only print out this message once and no more. For the record, there are two other equivalent ways to invoke this report rule:

```
Report going up from the Foyer for the first time:
Report going up from the Foyer 1 time:
```

Lastly, what if we want the message to print out more than once?

We can do it like this:

```
Report going up from the Foyer 2 times:
  say "Carefully testing each step for it's sturdiness,
  you slowly make your way up the stairs."
```

This will invoke the report rule only twice and then stop.

Be sure to use numeric values (2, 3, 4, etc.) and not text (two, three, etc.)