
Article

[Kurro Lopez](#) · Mar 1, 2022 5m read

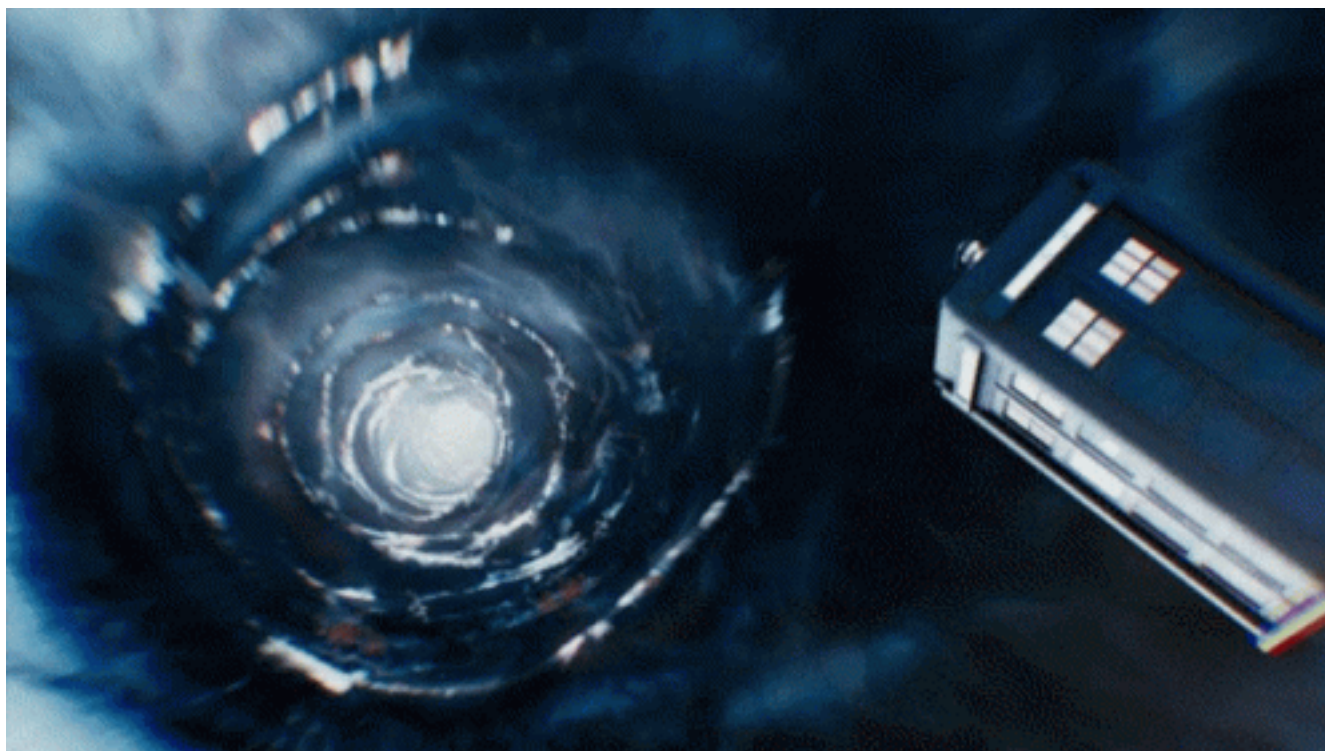
How to become a time lord - The birth

Good men don ' t need rules.

The Doctor.

It's not an easy task to be a master of dates and times, it is always a problem and sometimes confusing in any programming language, we are going to clarify and put a few tips to make this task as simple as possible.

Get on the TARDIS and I'm going to turn you into a Time lord



Let's start with the basics

If you usually use other languages, bear in mind that the dates in InterSystems Object Script (hereinafter IOS, not to be confused with an Apple mobile) are a bit peculiar.

When we run the command [\\$HOROLOG](#) in the terminal, to have the current date and time, you'll see that it is divided into two parts:

```
WRITE $HOROLOG
```

```
> 66149,67164
```

The first value is the day, to be exact the number of days since December 31st, 1840, that is, the value 1 is January 1st, 1841 and the second are the seconds since 00:00 of today.

In this example, 66149 corresponds to 09/02/2022 (February 9th in european format dd/mm/yyyy) and 67164 to 18:39:24. We will call this format internal format of datet and time.

Confused? Well, we are going to start revealing the great secrets of the universe (of dates and times).

How can I convert the internal format to a clearer format?

For that, we will use the command [\\$ZDATETIME](#)

The basic command would be

```
SET RightNow = $HOROLOG
WRITE RightNow

> 66149,67164
WRITE $ZDATETIME(RightNow)

> 02/09/2022 18:39:24
```

By default, it uses the American format (mm/dd/yyyy). If you want to use the date in another format, we will use the second parameter, such as the European one (dd/mm/yyyy), in that case we will give it the value 4 (for more formats, see the documentation [\\$ZDATETIME.dformat](#))

```
SET RightNow = $HOROLOG
WRITE RightNow

> 66149,67164

WRITE $ZDATETIME(RightNow,4)

> 09/02/2022 18:39:24
```

This option uses as separator and year format what we have defined in the local variables

If we also want to put another time format, for example in 12-hour format (AM/PM) instead of 24-hour format, we use the third parameter with the value 3, if we don't want to show the seconds, we will use the value 4 (see the documentation [\\$ZDATETIME.tformat](#))

```
SET RightNow = $HOROLOG
WRITE RightNow

> 66149,67164

WRITE $ZDATETIME(RightNow,4,3)

> 09/02/2022 06:39:24PM

WRITE $ZDATETIME(RightNow,4,4)

> 09/02/2022 06:39PM
```

Is it clearer now? So let's go deeper

ODBC Format

This format is independent of your local configuration, it will always be shown as yyyy-mm-dd, its value is 3. It's recommended to use it if we want to create data that is going to be exported in files, such as CSV, HL7 files, etc.

```
SET RightNow = $HOROLOG
```

```
WRITE RightNow
```

```
> 66149,67164
```

```
WRITE $ZDATETIME(RightNow,3)
```

```
> 2022-02-09 18:39:24
```

Day of the week, name of the day, day of the year

Value	Description
10	The day of the week will be a value between 0 and 6, 0 for Sunday and 6 for Saturday.
11	The name of the abbreviated day of the week, it will return it according to the local configuration you define, the default installation of IRIS is enuw (English, United States, Unicode)
12	The name of the day of the week in long format. same as 11
14	The day of the year, so that, the number of days since January 1

If we just want to treat dates and times separately, we should use the \$ZDATE and \$ZTIME commands respectively. The parameters for the formats are the same as those defined in [\\$ZDATETIME.dformat](#) and [\\$ZDATETIME.tformat](#)

```
SET RightNow = $HOROLOG
```

```
WRITE RightNow
```

```
> 66149,67164
```

```
WRITE $ZDATE(RightNow,10)
```

```
> 3
```

```
WRITE $ZDATE(RightNow,11)
```

```
> Wed
```

```
WRITE $ZDATE(RightNow,12)
```

```
> Wednesday
```

And how do I convert a date to internal format?

Well, now we're going to see the opposite step, that is, have a text with a date and convert it into IOS format. For this task, we'll use the command [\\$ZDATETIMEH](#)

This time, we have to indicate what format the date and time are in (if we are using \$ZDATETIMEH) or the date (\$ZDATEH) or the time (\$ZTIMEH) separately.

The formats are the same, that is, if we have a string with the date in ODBC format (yyyy-mm-dd), then we'll use the value 3

```
SET MyDatetime = "2022-02-09 18:39:24"
SET Internal = $ZDATETIMEH(MyDatetime, 3, 1) // ODBC Format

SET MyDatetime = "09/02/2022 18:39:24"
SET Interna2 = $ZDATETIMEH(MyDatetime, 4, 1) // European format

SET MyDatetime = "02/09/2022 06:39:24PM"
SET Interna3 = $ZDATETIMEH(MyDatetime, 1, 3) // American format with time in 12h AM/PM
WRITE Internal,!,Interna2,!,Interna3

> 66149,67164
66149,67164
66149,67164
```

Logically, if we say that the string is using a specific format and we give it the wrong parameter, anything can happen, like instead of February 9th it understands as September 2nd.

Do not mix formats, that later come the problems.

```
SET MyDatetime = "09/02/2022"

/// American format
SET InternalDate = $ZDATEH(MyDatetime, 1)

/// European format
SET OtherDate = $ZDATETIME(InternalDate, 4)
WRITE InternalDate,!,OtherDate

> 66354
02/09/2022
```

Needless to say, if we try to set a European date and try to transform it into an American one... What would happen on Valentine's Day?

```
SET MyDatetime = "14/02/2022"
SET InternalDate = $ZDATEH(MyDatetime, 1) // American format. month 14 doesn't exist
!!!
^
<ILLEGAL VALUE>
```

Well, like all Valentine's day... broken hearts, well... in this case, broken code.

Well, let's do something with what you've already learned.

```
READ !,"Please indicate your date of birth (dd/mm/yyyy): ",dateOfBirth
SET internalFormat = $ZDATEH(dateOfBirth, 4)
SET dayOfWeek= $ZDATE(internalFormat, 10)
SET nameOfDay = $ZDATE(internalFormat, 12)
WRITE !,"The day of the week of your birth is: ",nameOfDay
IF dayOfWeek = 5 WRITE "you always liked to party!!!" // was born on friday
```

Later we will see other ways of doing things, and how to handle errors.

Next chapter: How to travel in time

Curiosity

If you want to know why the value of 01/01/1841 is taken as the value 1, it comes because this date was chosen because it was the non-leap year before the birth of the oldest living US citizen, who was a civil war veteran 121 years old when [the MUMPS programming language](#) was designed, from which it extends Object Script

[#ObjectScript](#) [#Tips & Tricks](#) [#Caché](#)

Source URL:<https://community.intersystems.com/post/how-become-time-lord-birth>