

---

Article

[Ben Spread](#) · Sep 30, 2019 2m read

## Tips & Tricks: Use self-destructing debugging code

Every developer has made the mistake of accidentally leaving temporary debug code in place when they meant to remove it after debugging is complete. The great thing about writing in ObjectScript is that there is a way to make temporary code be truly temporary and automatically self-destruct! This can also be done in such a way that the code has no change of making it into your source control stream, which can be helpful as well.

The secret to this lies in making use of the "Intermediate code" (.INT) which is generated when you compile classes (.CLS), routine code (.MAC) or CSP pages (.CSP). The INT code is then used to create the OBJ code which actually executes.

(see

[https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GORIENTchintro#GORIENTin\\_troworktogether](https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GORIENTchintro#GORIENTin_troworktogether) for more details on this compilation flow)

INT code can actually be edited directly in the IDE, but since it is generated code it is generally not source controlled. Once an INT file has been edited it can then be compiled in order to create new OBJ code, which impacts the execution logic. However, these edits are self-destructing as they will automatically be overwritten when an upstream controlled configuration item is compiled (e.g. CLS, MAC, CSP, etc).

Leveraging the self-destructing nature of generated INT files can be helpful in the following situations:

- Adding debugging code
- Adding a breakpoint to interactive debugging
- Adding additional logging (e.g. DO LOG^%ETN)
- Inserting a short-term fix into a non-development environment without checking out the source file for an emergency edit
  - \*ONLY USE THIS IF\* your environments receive source directly and automatically from source control ... if not then the "temporary" fix could be longer lived than intended

With this approach, any time new source gets synced or pushed and then compiled it will rebuild all INT code and blow away anything temporary there-in. Very helpful to ensure that these temporary edits are indeed temporary!

[#Debugging](#) [#Development Environment](#) [#Tips & Tricks](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

---

Source URL: <https://community.intersystems.com/post/tips-tricks-use-self-destructing-debugging-code>