TUESDAY, MARCH 08, 2011

New Language / Compiler Features

Here are some proposed compiler / language options that most modern programming languages can benefit from. These are based on common errors that have occurred on systems that I have used and/or tested. Consider implementing these if you feel moved to do so.

Feel free to add your own in the comments.

|  |  |
| --- | --- |
| **Feature** | **Preventing** |
| ***Find all files*** - When enabled, this will allow the program to search not only the local computer, but all computers on the network and then through files available through major search engines. One optional parameter is a timeout, given in hours. | "File Not Found" error |
| ***Instantiate all Objects*** - Each object referenced in the project will be instantiated and when the condition is detected where an object appears to have no instance provided, a suitable instance from the pre-instantiated collection of each object type will be retrieved. Care should be taken when this feature is used in conjuction with 'Universal Cast' | "Object reference not set to an instance of an object" error |
| ***Universal Cast*** - Allow the casting between any two types. How hard could that be? | "Cannot implicitly convert type 'a' to 'b'" and "Type Mismatch" error |
| ***Expect the Unexpected*** - When enabled, the compiler will just ignore anything it doesn't expect. | "Unexpected character "?"" error |
| ***Trust Everyone***- As a society, we are trusing each other too little. This can cause all sorts of issues with regards to full and complete access. When this feature is enabled, all means (legal and otherwise) are used to obtain the information or resources requested. | "Access Denied" and "Unauthorized" errors |
| ***Pass Mac*** - Give the MAC a passing grade, even when it hasn't really tried all that hard. Eventually, this will be someone else's problem. | "Vewstate Mac Failed" errors |
| ***Handle all Exceptions***- Surely this error has been seen before. When enabled, the program will search online for a solution related to handling this exception and do that. | "Unhandled Exception" errors |
| ***All Knowing*** - Nothing happens by accident. So finding out the error can't be all that difficult, right? | "Unknown Error" errors |
| ***Virtual Valium*** - Instead of freaking out when there's an issue, give the kernel some virtual valium when it starts to get confused. | "Kernel Panic" error (aka BSOD) |
| ***Allow Infinity*** - When enabled, allow infinity to be used in calculations. For the purposes of this feature, there is no distinction between a positive infinity and a negative infinity. | "Index out of Range" and "Attempt to Divide by Zero" errors |
| ***Ethics Committee*** - When this is enabled, the Heap is investigated for corruption prior to the corruption starting to ensure that all ethics guidelines are followed. | Heap Corruption error |
| ***Stack Bowl*** - when enabled, the stack will be placed in a bowl to catch any overflow. This bowl will also be partially filled to fill in the stack when 'underflow' occurs. | Stack Overflow\Underflow |
| ***Auto-initialize everything*** - For times when you think "I just declared that, why can't I start to use it?" (Thanks to Shmuel Gershon for this new feature) | Errors when using variables before initializing |
| ***Do/Catch*** - As Yoda says "Do or not do, there is no try". (Thanks to Dwain) | Try/Catch |
| **.correctSpelling, .toPresentTense, .toPastTense, .toFutureTense, .toSingular, .toPlural, .toFirstPerson, .toSecondPerson, .toThirdPerson etc.** - Allows you to validate responses (such as from security challenges) with users that have difficulty remembering the spelling, tense, etc. of their original answers. (Thanks to Chris/Chad) | Security challenge response validation issues. |