Error Message Guidelines

An error message is text that is displayed to describe a problem that has occurred that is preventing the user or the system from completing a task. The problem could result in data corruption or loss. Other message types include confirmations, warnings, and notifications. The guidelines in this topic are intended to help you write clear error messages that are easy to localize and useful for customers.

Poorly written error messages can be a source of frustration for users and can increase technical support costs. A well-written error message provides the following information to the user:

- What happened and why?
- What is the end result for the user?
- What can the user do to prevent it from happening again?

The length of the text is not an issue as long as the developer handles buffer sizes correctly. It is important that the user have all the information necessary to solve the problem. If a message has multiple audiences, you may need to provide separate text for administrators, end users, and developers.

Best Practices

The following are ways to improve your error messages:

- Avoid error conditions. If you can predict that an error will occur when a user performs a specific action, rewrite your code so that the user cannot cause the error.
- Write a separate error message for each known cause of the error. Do not use a single, generic message to explain every possible reason for the error unless you cannot determine the cause of the error when it occurs.
- State the problem clearly and, if it will be helpful to the user, explain what caused the problem. Whenever possible, replace the generic messages from the system message-table resources with a detailed message that is specific to the problem.
- Provide the user with a solution to the problem. If the solution has more than one step, refer to a help topic the explains the task in detail.
- Display only the product, component, or wizard name in the title bar of the message. This helps the user determine where the problem is. Do not summarize the problem in the title bar or include the word "error".
- Do not use technical jargon, use terminology that your audience understands. Do not use slang or abbreviations.
- Use the appropriate command buttons, such as OK, Cancel, Yes, No, and Retry. You can use combinations of these buttons. The Yes and No buttons must always be used in combination and must always be preceded by a question.
  - To stop an operation and close the message box, use the **Cancel** button.
  - To close a message box, use the **Close** button.
  - To provide more information about the cause of the error, use the **Details** button.
  - To provide more information about the solution to the problem, use the **Help** button.
  - If a user action is included in the message, use the **OK** button to close the message box.
  - **Yes** and **No** buttons must be used in combination and must always be preceded by a question.
- If the error is a critical error, write it to the event log.

Style Considerations

- Use complete but simple sentences.
- Use the present tense to describe the conditions that caused the problem or a state that still exists. You can use past tense to describe a distinct event that occurred in the past.
- Use active voice whenever possible. You can use passive voice to describe the error condition.
- Avoid uppercase text and exclamation points.
- Do not make the user feel at fault even if the problem is the result of a user error.

- Do not anthropomorphize. Do not imply that programs or hardware can think or feel.
- Do not use colloquial words or phrases. Do not use terms that may be offensive in certain cultures.
- Do not compound several nouns without adding a preposition or subclause to clarify the meaning. For example, "Site Server LDAP Service directory server" should be changed to "Directory server for the LDAP Service of the Site Server".
- Insert descriptors before a term to clarify the meaning of the sentence. For example, "Specify InfID when Detect is set to No." should be changed to "Specify the InfID parameter when the Detect option is set to No".
- Avoid the word "bad". Use more descriptive terms to tell the user what is wrong. For example, avoid messages such as "Bad size". Instead, tell the user what criteria to use when specifying a size.
- Avoid the word "please". It can be interpreted to mean that a required action is optional.
- Place words that are both in the index and relevant to the central meaning at the beginning of the message string.