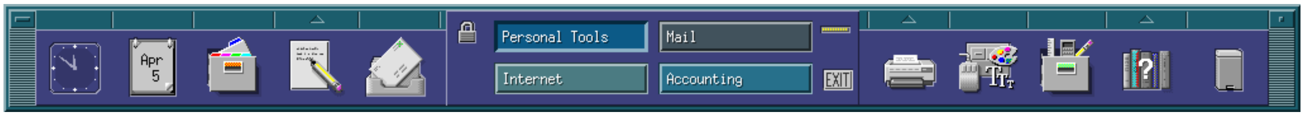


APPLICATION INTEGRATION FOR THE COMMON DESKTOP ENVIRONMENT

July 31, 1995

This paper provides an overview of the steps needed to integrate applications into the Common Desktop Environment (CDE) on Digital platforms. This document is intended only as a guide. It is directed to explain *why* something is necessary and the results in terms of user actions, not the implementation steps (the how). To implement the steps, use the details contained in the resource materials listed at the end of this article.

Introduction

The most important step in integrating an existing application into CDE may be to realize that nothing is required to run existing applications in the CDE environment. Since the graphical infrastructure is based on Motif, applications based on Motif 1.2.3 continue to function in the CDE environment.

This is not to suggest that additional steps should not be taken. CDE offers a rich environment that is strengthened with each application that uses the CDE features. However, because existing applications continue to operate, integration of CDE features can proceed at a pace appropriate to the needs of a given application.

The integration steps are divided into three catagories. The first catagory should be implemented in its entirety. The other two catagories contain steps which can be implemented independent of the other steps contained in the catagories. As a result, the steps can be implemented in an order appropriate to the application.

The catagories are as follows:

- Minimal Integration **-** outlines the steps to simplify the activation of an application in the CDE environment. Generally, all of the steps in this category should be performed.
- Recommended Integration **-** outlines the steps to be taken by all applications to integrate with CDE. These steps should be performed, but can proceed in the order appropriate for the application.
- Application Specific **-** outlines steps that should be taken only by applications that have a need for the services described

Minimal Integration

The steps outlined in this section configure the desktop for an application. While application code changes are not required, some code changes such as style changes and creating direct access to application functions can enhance the results of these steps.

Background Requirements

CDE maintains and builds upon the guidelines contained in the Motif Style Guide. Compliance with those guidelines are an important first step to achieve minimal integration.

Icon Presentation

To provide a visual representation of an application's services, icons of various types are needed:

- Application icon **-** Used for launching the application.
- Data file icons **-** Lets users activate the application through data file icons. For each type of data file used by your application of which the user is aware, consider providing a separate icon. By doing so, users can activate your application not only via the application icon, but also via data files used by the application. This has the added benefit of starting the application in the context of the data file. A good example of this would be to activate an icon representing a mail folder file and having the mailer startup with that mail folder as the initial context.
- Specific application function icons **-** Lets users access the application from more than one function. Using the mail example again, there could be a send mail icon that when activated, would place the user in the mailer but in the context of sending a mail message. Further, dragging a text file onto such an icon would cause the mailer to start with that text being the intended mail contents.

Each of the previous types require more than one icon. Icons can be displayed on a monochrome or color monitor. In addition, icons can be displayed in more than one context including the file manager (both large and small icon displays) and on the front panel. Because of this, 3 sizes are required (48X48, 32X32, 16X16) in both monochrome and color for a total of 6 icons.

Short cuts: Existing icons can be used. Also, the icon editor provides a simple method for creating icons from existing displays.

File Typing

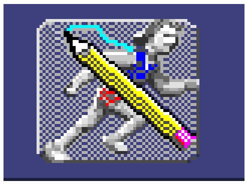
After the icons have been created, the icon representations are bound to the files using the following two clauses:

- Data Type **-** Identifies the file type based on characteristics such as pathname, filename, mode, etc. This clause is used for both data files and the application.
- Data Attributes **-** Connects files of a specific type to the attributes used to operate and represent (such as icons) the file type.

These clauses are specified in one or more rule files you provide with your application. These files have names of the form *.dt. Please note you only need a data type clause for each type of file (and icon) you define. The desktop takes care of selecting the icon size and color for you.

The third type of clause needed is the action clause described in the next section.

Actions



The action that results when the user activates any of the file types defined in the previous section is specified in an action clause which can include the following characteristics:

- The action execution **-** which is either a message or a shell command.

  ⇒ If it is a message, it uses the ToolTalk® messaging facility to transmit one of a number of set messages to an application.

  ⇒ If it is a shell command, anything that is possible through a shell script can be performed. Typically, the shell command in an action clause activates a program with appropriate arguments.

- The form of the activation **-** which has at least the following forms:

  ⇒ Simple activation. In this case, no further context is supplied.

  ⇒ Data activation. In this case, the action was initiated through a file with the specific data type. For example, when a mail folder file is activated, the mailer is started with the data file supplied as the initial context.

⇒ Drag and Drop activation. In this case, the icons for one or more data files are dropped on another icon representating an action such as when more than one file is copied to some destination.

- The action selection by context - which drives the choice in action based on the current context.. For example, a number of different types of files can be edited, but each type of file requires a different editor. An icon file requires the icon editor while a text file is modified with the standard text editor (dtpad). In another example, one version of an application might need only one argument while another version might require more than one argument (files dropped).

Short cut: Initial rules can be generated through the create action tool.

Recommended Integration

All of the recommended integration steps outlined in this section require code changes in the application.

Session Management

The CDE session manager provides a mechanism that permits an application to preserve its state before logout and then restarts the application to the saved state at the next user login. This is accomplished using the session manager's support of the ICCCM Session Management Protocol.

With an application enabled for such support, the following steps occur:

1. The user starts an exit from the session. The session manager saves resource and other information about the environment, and then the session manager notifies all applications enabled to receive the exit notification.
2. Each application notified saves its state using whatever means is appropriate (interaction with the user is not permitted however due to timing and other constraints - the user is, after all, trying to log out), notifies the session manager that the save is complete, and provides the session manager with the restart command line.
3. When the user next logs in, the session manager restores the environment and then restarts the applications with the supplied command line and the previous workspace and geometry information.

Help

CDE provides an entirely new help mechanism which includes:

- Help widgets and API
- Applications for creating help volumes
- Applications for viewing help volumes

To use the CDE help system, the application must be adapted to use the help API and convert the help information used by the application. Through the use of the hyperlink mechanisms, the application can link into other help volumes as well as provide running examples using links to executables.

Drag and Drop

CDE provides a wrapper around the existing Motif drag and drop functions. Some of the benefits include:

- Simplicity. Provides more features in an easier to use API when compared with the standard Motif drag and drop.
- Policy. Provides drag icons (what the icon looks like while being dragged), animation (the result of dropping at an incorrect location), targets, etc.
- Appearance. Provides many services with the representation built-in

resulting in the user viewing a consistent appearance.

Standard Font Names

While CDE does not supply fonts, it uses a set of standard font names of the form -dt-application-*. The benefit to the application developer is that by specifying font names of this type, there is a guarantee that correctly functioning fonts for the platform are used. The user benefits because all text displayed from various operating applications have the same font.

Color

Similar to fonts, use of the standard color map within CDE enhances the consistency viewed by the user. When utilized by the application, changing colors in the style manager is reflected in the application.

Application Specific Integration

Workspace Management

Not only does CDE provide the user with multiple workspaces and the ability to manipulate them, the same services are available for use by applications through the use of provided APIs.

Note that this API is not intended for casual use. While it is true that an application can change the current workspace being viewed by the user, careless use of such features can lead to serious user dissatisfaction.

There are some contexts where use of this API may be appropriate. For example, under realtime conditions, it may be important to notify the user of a realtime event (“1 minute before detonation”). In this case, changing the current window to provide additional data may be appropriate. Another example might be in a turnkey system where all of the workspaces have a known organization in which case, placing windows in specific workspaces may be appropriate.

Messaging Services

CDE provides both a messaging service as well as a set of defined messages. Through the use of these messages, an application can interact with existing applications without either application having prior knowledge of the other application.

Calendar Services

CDE provides a calendar application and a calendar API that is accessible to other applications within CDE. Using this API, a project management application can automatically schedule a meeting with respective members of a project at the completion of a milestone. This is accomplished by examining and comparing the calendars of all the respective members and then scheduling a meeting time that accommodates all of the attendees.

Front Panel

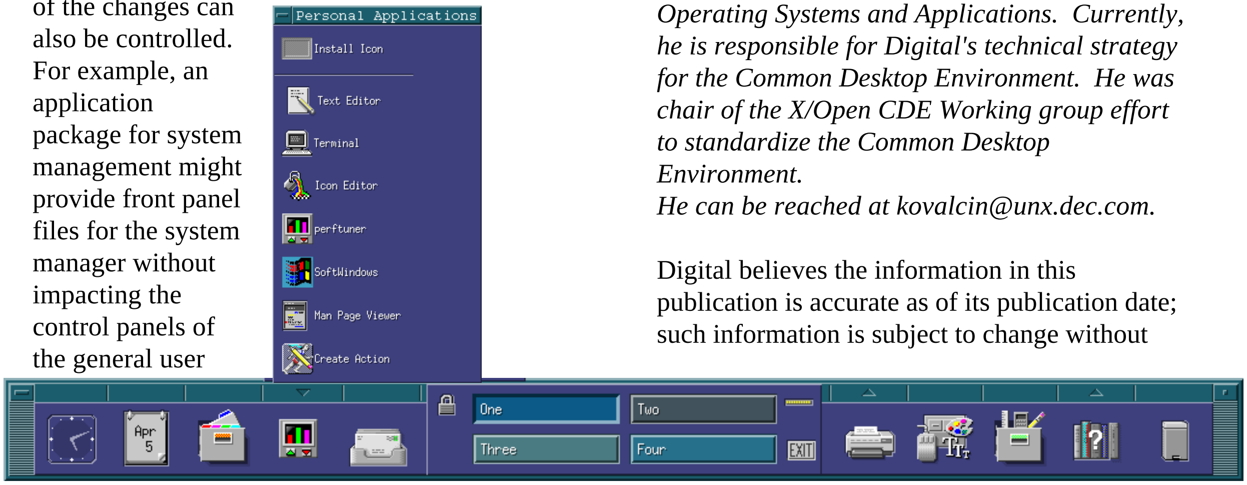
CDE provides a control panel that is present in all workspaces by default. It is strongly recommended that the installation of an application not automatically change the front panel. This recommendation maintains the consistency of the front panel across implementations in the default case. However, providing tools to permit an explicit user decision to modify their front panel is appropriate.

For example, the default control panel contains the following fields: clock, calendar, file manager, personal applications, mail, workspaces, printer, style manager, applications, help and trashcan. In environments where the system is configured for a specific task, the general services provided by the CDE front panel may be inappropriate and other services not present might be important.

There are a couple of mechanisms in which the front panel can be modified to solve a particular application problem (these mechanisms are in addition to the tools available to individual users to reconfigure the control panel dynamically):

- The existing front panel can be replaced. In this case, the existing front panel descriptor file (.fp) is replaced with a new front panel file that addresses the need of the application.
- The existing front panel can be augmented by providing additional front panel files. Existing fields in the default front panel file can be replaced by providing new descriptions of the identified fields in additional front panel files. Also, the fields are given relative placement. By identifying a new field in the control panel, existing fields are shifted to accommodate the new field. This method has the added attraction of not obscuring previously made local changes.

In addition to the above mechanisms, the scope of the changes can also be controlled. For example, an application package for system management might provide front panel files for the system manager without impacting the control panels of the general user population. CDE permits such changes by permitting individual front panels to be independently configured.



Resource materials

CDE is included as an Advanced Developer's Kit with DEC OSF/1 version 3.2 which includes documentation in PostScript® form. The following manuals are a particular aid in understanding application integration into the CDE environment:

> "Common Desktop Environment: Advanced User's and System Administrator's Guide" see chapters on creating actions, data types and front panael customization.
>
> "Common Desktop Environment: Programmer's Overview"
>
> "Common Desktop Environment: Programmer's Guide"

A description of the ICCCM Session Management Protocol (Inter-Client Communications Convention Manual) is available in "X Window System" by Robert W. Scheifler and James Gettys published by Digital Press, order number EY-E755E-DP.

About the Author:

David Kovalcin has been with Digital Equipment Corporation for over 15 years. During that time he has worked on a variety of platforms in a number of areas including Operating Systems and Applications. Currently, he is responsible for Digital's technical strategy for the Common Desktop Environment. He was chair of the X/Open CDE Working group effort to standardize the Common Desktop Environment.
He can be reached at kovalcin@unx.dec.com.

Digital believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. Digital is not responsible for any inadvertent errors. Digital conducts its business in a manner that conserves the environment and protects the safety and health of its employees, customers, and the community.