

4. DESKT - Desktop CSCI

4.1 CSCI Overview

The Desktop CSCI provides a desktop environment, which consists of a set of user and client files (called desktop objects, each with a representative icon), a pop-up menu for quick access to specific client applications, specific operations defined for desktop objects, including drag and drop, double-click to open, copy, delete, and rename, and a *desktop manager*, which manages the set of desktop objects. Desktop objects include, but are not limited to: search parameters, result-sets, documents, subscriptions, request-sets, data granules, project folders, application programs, and configuration files. Managing the desktop objects includes viewing them in a hierarchical format, viewing their attributes, e.g., name, creation date, and status. The status is applicable only to certain classes of objects, such as product or search requests, and provides a convenient means of providing information about asynchronous requests from the user.

In Release A, the desktop manager will have two versions. There will be a custom developed desktop manager built with X-Windows Version 11 (X11), Motif, and the Common Desktop Environment (CDE) toolkits. This custom desktop is for users who do not use CDE. At Release A, this is most likely a large part of the community. The other version of the desktop manager will be an off-the-shelf (OTS) version using CDE in its native environment. Thus, the desktop objects will reside on the CDE desktop manager. This will be accomplished by making the files representing the desktop objects comply with the CDE standard. Thus the implementation of the desktop object classes will not change no matter which desktop manager is being used by the user. In Release B, it is likely that a much higher number of users will be using the OTS CDE solution. By later releases, it is planned to eliminate the custom desktop manager based on the number of users that have migrated to the OTS solution.

The desktop is either represented in hierarchical format or iconic format. The hierarchical format is similar to the Microsoft Windows implementation of the File Manager or the Macintosh Desktop implementation when the user selects View By Name. The iconic format is similar to the Microsoft Windows Program Manager implementation or the Macintosh Desktop when in View by Icon. The desktop manager contains desktop objects. There are three types of desktop objects; Container objects, Application objects and Document objects.

Each desktop object will have zero or more actions attached to it. The default action for an object will be executed by double clicking on the icon that represents the object. Other actions can be performed from the Actions pulldown menu. For example, an icon representing a universal reference (UR) to a granule can be ordered by double-clicking the icon. The default action for the granule UR icon is to start the Product Request Tool (PRT) with the associated UR. Another action that could be performed on a granule UR icon includes starting the ESST with the associated granule, which would show the granule on the results screen. This secondary action would be performed by selecting the icon and then pulling down the Actions menu and selecting ESST. Each desktop object is represented with a specific icon on the desktop. CDE provides configuration files that associate appropriate attributes with the object types. Thus, the set of actions can be changed by modifying the CDE configuration files.

Users can add their own tools to the desktop by modifying the appropriate CDE configuration files. These files would have to be modified to create an object type and specify the appropriate actions as well as Drag and Drop specifications. The X/Open CDE specification provides details on how to change these configurations.

4.2 CSCI Context

The Desktop provides no public interfaces for access by any formal track CSCI. **The objects in this CSCI are used by the Workbench CSCI however.**

4.3 CSCI Object Model

This section provides the object model for the Desktop CSCI. The model is graphically depicted in Figure 4.3-1. Each of the objects in this figure are described in detail in the subsequent paragraphs. Since this CSCI is a direct reuse of the Release A development effort, this section includes the latest OMT repository download which corresponds to Phase 2 of the Release A CIDM development stage.

4.3.1 CIDtApplicationObject Class

Parent Class:CIDtDesktopObject

Public:No

Distributed Object:No

Purpose and Description:

Application object is a type of desktop object. It provides default behaviour for objects which represent executable programs.

Attributes:

myCommandToExecute - Command to execute

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

CIDtApplicationObject

Arguments:name:EcTChar*, dispArea:CIDtDisplayArea*, relFilename:RWCString, fullFilename, R WCString, modTime:time_t, actionInfo:, CIDtActionInfo*

ExecuteDefaultAction - Execute the Default Action

Arguments:dtWindow:CIDtDesktopWindow*

Return Type:EcTVoid

Privilege:Private

GetCommandToExecute

Arguments:

Return Type:RWCString

Privilege:Public

UpdateActionsPulldownMenu - Update the pulldownMenu

Arguments:dtWindow:CIDtDesktopWindow*

Return Type:EcTVoid

Privilege:Private

~CIDtApplicationObject - Destructor

Arguments:

Return Type:Void

Privilege:Public

Associations:

The CIDtApplicationObject class has associations with the following classes:

None

4.3.2 CIDtContainerObject Class

Parent Class:CIDtDesktopObject

Public:No

Distributed Object:No

Purpose and Description:

A type of Desktop Object which is capable of holding other desktop objects (containers, applications, and documents). Since containers can contain other container objects, they can be used to implement hierarchical structures. Actions associated with this object are display contents, move into container, remove from container etc.

Attributes:

All Attributes inherited from parent class

Operations:

CIDtContainerObject

Arguments:name:EcTChar*, dispArea:CIDtDisplayArea*, relFilename:RWCString, fullFilename:RW CString, modTime:time_t

CIDtDropFileToDirIcon

Arguments:dragIcon:EcTChar*, dropObj:EcTChar*

Return Type:Void

Privilege:Public

ExecuteDefaultAction - Execute the Default Action

Arguments:dtWindow:CIDtDesktopWindow*

Return Type:Void

Privilege:Protected

UpdateActionsPulldownMenu - Update the pulldownMenu

Arguments:dtWindow:CIDtDesktopWindow*

Return Type:Void

Privilege:Protected

~CIDtContainerObject - Destructor

Arguments:

Return Type:Void

Privilege:Public

Associations:

The CIDtContainerObject class has associations with the following classes:

None

4.3.3 CIDtDesktopObject Class

Parent Class:Not Applicable

Attributes:**myDispArea** - Parent of desktop object

Data Type:CIDtDisplayArea*

Privilege:Protected

Default Value:

myDtObjIcon - Pixmap for the icon

Data Type:Pixmap

Privilege:Protected

Default Value:

myDtObjPixmap

Data Type:Pixmap
Privilege:Protected
Default Value:

myDtObjType - Type of the desktop object

Data Type:EcTInt
Privilege:Protected
Default Value:

myFileSize

Data Type:off_t
Privilege:Protected
Default Value:

myFileSizeLBL

Data Type:Widget
Privilege:Protected
Default Value:

myFullFileName - Full path of the desktop object

Data Type:RWCString
Privilege:Protected
Default Value:

myIconPB - Drawn button to display pixmap

Data Type:Widget
Privilege:Protected
Default Value:

myIconTF - Label to display filename

Data Type:Widget
Privilege:Protected
Default Value:

myModTime - Modification time

Data Type:time_t
Privilege:Protected
Default Value:

myModTimeLBL

Data Type:Widget
Privilege:Protected
Default Value:

myRelFileName - Relative path of the desktop object
Data Type:RWCString
Privilege:Protected
Default Value:

Operations:

CIDtDesktopObject - Constructor
Arguments:name:EcTChar*
Return Type:Void
Privilege:Public

ExecuteDefaultAction - Pure virtual method. Derived classes need to override
Arguments:dtWindow:CIDtDesktopWindow*
Return Type:EcTVoid
Privilege:Protected

FilenameSelected
Arguments:
Return Type:EcTVoid
Privilege:Protected

FilenameSelectedCB
Arguments:w:Widget, client_data:XtPointer, event:XEvent*,
continuedtodispatch:Boolean*
Return Type:EcTVoid
Privilege:Public

GetDisplayArea
Arguments:
Return Type:CIDtDisplayArea*
Privilege:Public

GetDtObjType
Arguments:
Return Type:EcTInt
Privilege:Public

GetIconTF
Arguments:
Return Type:Widget
Privilege:Public

GetRelFileName - returns relative filename of the desktop object

Arguments:

Return Type:RWCString

Privilege:Public

HighlightSelectedIcon

Arguments:

Return Type:EcTVoid

Privilege:Public

IconSelectedCB - Callback called on activating icon

Arguments:w:Widget, client:XtPointer, call:XtPointer

Return Type:EcTVoid

Privilege:Private

IconsSelected - IconSelectedCB calls this method

Arguments:call:XtPointer

Return Type:EcTVoid

Privilege:Protected

RenameFile

Arguments:

Return Type:EcTVoid

Privilege:Protected

RenameFileCB

Arguments:w:Widget, client_data:XtPointer, call_data:XtPointer

Return Type:Void

Privilege:Public

UpdateActionsPulldownMenu - Pure virtual method. Derived classes need to override

Arguments:dtWindow:CIDtDesktopWindow*

Return Type:EcTVoid

Privilege:Protected

Associations:

The CIDtDesktopObject class has associations with the following classes:

CIDtHierarchical (Aggregation)

CIDtIconic (Aggregation)

4.3.4 CIDtDesktopWindow Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The Desktop Window can be either in hierarchical format or iconic format. Desktop Window contains zero or more desktop objects. Desktop Window itself is a Desktop Object i.e., a container object.

Attributes:

myAnimate - Animate Cmd

Data Type:Cmd*

Privilege:Protected

Default Value:

myBaseFORM - Form that holds directory path

Data Type:Widget

Privilege:Protected

Default Value:

myCanvasFORM - Form which holds the work area

Data Type:Widget

Privilege:Protected

Default Value:

myClose - Close Cmd

Data Type:Cmd*

Privilege:Protected

Default Value:

myCopy - Copy Item Cmd

Data Type:Cmd*

Privilege:Protected

Default Value:

myCreateDir - Create Directory Cmd

Data Type:Cmd*

Privilege:Protected

Default Value:

myDeleteItem - Delete Item Cmd

Data Type:Cmd*

Privilege:Protected
Default Value:

myDirPathLBL - Label widget that displays directory path

Data Type:Widget
Privilege:Protected
Default Value:

myDirectoryPath - window's current directory

Data Type:RWCString
Privilege:Protected
Default Value:

myDisplayArea - Ptr to display area

Data Type:CLDtDisplayArea*
Privilege:Protected
Default Value:

myDisplayFORM - Form that holds Display Area

Data Type:Widget
Privilege:Protected
Default Value:

myDuplicate - Duplicate Window Cmd

Data Type:Cmd*
Privilege:Protected
Default Value:

myExit - Exit Cmd

Data Type:Cmd*
Privilege:Protected
Default Value:

myHome - Home Cmd

Data Type:Cmd*
Privilege:Protected
Default Value:

myLoadActions - Load Actions Cmd

Data Type:Cmd*
Privilege:Protected
Default Value:

myMove - Move Item Cmd

Data Type:Cmd*

Privilege:Protected
Default Value:

myOpen - Open Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myOpenInPlace - OpenInPlace Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myOpenNewWindow - OpenNewWindow Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myRefresh - Refresh Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myTutorial - Tutorial Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myUnDelete - Undo Delete Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myUp - Up Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myUserPreferences - User Preferences Cmd
Data Type:Cmd*
Privilege:Protected
Default Value:

myWindowState - whether hierarchical or iconic
Data Type:EcTInt

Privilege:Protected

Default Value:

Operations:

AnimateCmd

Arguments:

Return Type:Cmd*

Privilege:Public

CheckDirectoryAndRefreshCB - Wrapper callback for Refresh

Arguments:w:Widget, client:XtPointer, call:XtPointer

Return Type:EcTVoid

Privilege:Private

CIDtDesktopWindow - Constructor

Arguments:name:Widget, dirPath

Return Type:Void

Privilege:Public

CloseCB - Wrapper callback for Close

Arguments:w:Widget, client:XtPointer, call:XtPointer

Return Type:EcTVoid

Privilege:Private

CloseCmd

Arguments:

Return Type:Cmd*

Privilege:Public

DeleteItemCmd

Arguments:

Return Type:Cmd*

Privilege:Public

ExitCmd

Arguments:

Return Type:Cmd*

Privilege:Public

GetDirectoryOpenMode - Get the Directory Open Mode

Arguments:

Return Type:EcTInt

Privilege:Public

GetDirectoryPath - Gets window's current Directory path

Arguments:

Return Type:RWCString

Privilege:Public

GetDisplayArea - Returns ptr to Display Area

Arguments:

Return Type:CIDtDisplayArea*

Privilege:Public

GetDisplayAreaParent - returns parent of Display Area

Arguments:

Return Type:Widget

Privilege:Public

GetDisplayAreaheight - returns height of Display Area

Arguments:

Return Type:Dimension

Privilege:Public

GetWindowState - Gets window's state

Arguments:

Return Type:CIDtWINDOWSTATE

Privilege:Public

OpenCmd

Arguments:

Return Type:Cmd*

Privilege:Public

OpenInPlaceCmd

Arguments:

Return Type:Cmd*

Privilege:Public

OpenNewWindowCmd

Arguments:

Return Type:Cmd*

Privilege:Public

RefreshCmd

Arguments:

Return Type:Cmd*

Privilege:Public

SetDirectoryOpenMode - Set the Directory Open Mode

Arguments:mode:EcTInt

Return Type:EcTVoid

Privilege:Public

SetDirectoryPathLabel - Sets Directory label for window

Arguments:RWCString

Return Type:EcTVoid

Privilege:Public

SetMessage - Displays the string in the message area

Arguments:message:RWCString

Return Type:EcTVoid

Privilege:Public

SetWindowState - Sets window's state

Arguments:state:CIDtWINDOWSTATE

Return Type:Void

Privilege:Public

UnDeleteCmd -

Arguments:

Return Type:Cmd*

Privilege:Public

UpCmd

Arguments:

Return Type:Cmd*

Privilege:Public

createMenuPanels - creates menu panels

Arguments:

Return Type:EcTVoid

Privilege:Protected

createWorkArea - creates the windows work area

Arguments:w:Widget

Return Type:Widget

Privilege:Protected

~CIDtDesktopWindow - Destructor

Arguments:

Return Type:Void

Privilege:Public

Associations:

The CIDtDesktopWindow class has associations with the following classes:

None

4.3.5 CIDtDisplayArea Class

Parent Class:Not Applicable

Attributes:

myDeskObjects - List of Desktop Objects
Data Type:RWTPtrSlist<CIDtDesktopObject>
Privilege:Protected
Default Value:

myDesktopWindow - Ptr to desktop window
Data Type:CIDtDesktopWindow*
Privilege:Protected
Default Value:

myLastSelectedObject
Data Type:CIDtDesktopObject*
Privilege:Protected
Default Value:

myModTime - Modification time of directory
Data Type:time_t
Privilege:Protected
Default Value:

Operations:

AddCIDtDesktopObject - Adds desktop object to the list
Arguments:dtObject:CIDtDesktopObject*
Return Type:EcTVoid
Privilege:Public

CheckDirectoryAndRefresh

Arguments:

Return Type:EcTVoid

Privilege:Public

CIDtDisplayArea - constructor

Arguments:parent:Widget, name:EcTChar*, dtWindow:CIDtDesktopWindow,
height:Dimension

Return Type:Void

Privilege:Public

CIDtDisplayArea - constructor

Arguments:name:EcTChar*

Return Type:Void

Privilege:Protected

DisplayAreaWindow - window of display area

Arguments:

Return Type:CIDtDesktopWindow*

Privilege:Public

FindCIDtDesktopObject - Find desktop object from the list

Arguments:fullFileName:const EcTChar*

Return Type:CIDtDesktopObject*

Privilege:Public

FindCIDtDesktopObject - Find desktop object from the list

Arguments:widget:Widget

Return Type:CIDtDesktopObject*

Privilege:Public

GetDirectory - Get the current directory

Arguments:

Return Type:const EcTChar*

Privilege:Public

GetLastSelectedObject - Get the DesktopObject which was selected previously.

Arguments:

Return Type:CIDtDesktopObject*

Privilege:Public

GetModificationTime - Returns Modification time of directory

Arguments:

Return Type:time_t

Privilege:Public

GetMultipleSelectedStatus - Gets the status of Multiple Selection, i.e., whether multiple objects are selected or not.

Arguments:

Return Type:Boolean

Privilege:Public

ReadDirectoryInfo - Opens the directory & reads contents

Arguments:

Return Type:EcTVoid

Privilege:Protected

RefreshDisplayArea - Refreshes the Display Area.

Arguments:

Return Type:EcTVoid

Privilege:Public

RemoveCIDtDesktopObject - Removes desktop object from the list

Arguments:dtObject:CIDtDesktopObject*

Return Type:EcTVoid

Privilege:Public

SetLastSelectedObject - Set the desktop object as the last selected object

Arguments:dtObject:CIDtDesktopObject*

Return Type:EcTVoid

Privilege:Public

SetModificationTime - Sets the Modification time of directory.

Arguments:modTime:time_t

Return Type:EcTVoid

Privilege:Public

UnHighlightLastSelectedObject - This function unhighlights the last selected object

Arguments:

Return Type:EcTVoid

Privilege:Public

~CIDtDisplayArea - default destructor

Arguments:

Return Type:Void

Privilege:Public

Associations:

The CIDtDisplayArea class has associations with the following classes:
CIDtDesktopWindow (Aggregation)

4.3.6 CIDtDocumentObject Class

Parent Class:CIDtDesktopObject

Public:No

Distributed Object:No

Purpose and Description:

A type of Desktop object which provides a mechanism for associating multiple types of descriptive data with an object. Applications can be associated with a class of desktop document objects, and this application can be invoked to access the data described in the document object.

Attributes:

myAction - Specific action

Data Type:EcTInt

Privilege:Private

Default Value:

myCommandToExecute - Command to execute

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

CIDtDocumentObject

Arguments:name:EcTChar*, dispArea:CIDtDisplayArea*, relFileName:RWCString,
fullFileName: RWCString, modTime:time_t, fileSize:off_t, actionsInfo:CIDtActionInfo*

ExecuteDefaultAction - Execute the Default Action

Arguments:dtWindow:CIDtDesktopWindow*

Return Type:EcTVoid

Privilege:Private

GetCommandToExecute

Arguments:
Return Type:RWCString
Privilege:Public

UpdateActionsPulldownMenu - Update the pulldownMenu

Arguments:dtWindow:CIDtDesktopWindow*
Return Type:EcTVoid
Privilege:Private

~CIDtDocumentObject - Destructor

Arguments:
Return Type:Void
Privilege:Public

Associations:

The CIDtDocumentObject class has associations with the following classes:
None

4.3.7 CIDtHierarchical Class

Parent Class:CIDtDisplayArea
Public:No
Distributed Object:No
Purpose and Description:
Displays desktop objects in hierarchical format.

Attributes:

myMultipleSelectedStatus - Multiple selected status. 0 represents single selection; 1 represents multiple selection
Data Type:EcTInt
Privilege:Private
Default Value:

Operations:

BuildDirectory - adds children to the Display Area
Arguments:parent:Widget, fullDirName:EcTChar*, manageChildren:Boolean
Return Type:EcTVoid
Privilege:Private

CheckDirectoryAndRefresh - Checks if directory status changed. If so, Refreshes the Directory

Arguments:

Return Type:EcTVoid

Privilege:Public

CIDtHierarchical - Constructor

Arguments:w:Widget,name:EcTChar*, dtWindow:CIDtDesktopWindow*,
height:Dimension

Return Type:Void

Privilege:Public

GetMultipleSelectedStatus - Return the status of multiple selection.

Arguments:

Return Type:EcTBoolean

Privilege:Public

MultipleSelection - Multiple Selection is not supported for hierarchical view. This function displays an error message.

Arguments:event:XEvent*, continuetodispatch:Boolean*

Return Type:EcTVoid

Privilege:Public

MultipleSelectionCB - Wrapper callback for Multiple Selection

Arguments:w:Widget, client_data:XtPointer, event:XEvent*,
continuetodispatch:Boolean*

Return Type:EcTVoid

Privilege:Public

NodeStateChange - Calls BuildDirectory on change of state

Arguments:w:Widget, call:XtPointer

Return Type:EcTVoid

Privilege:Protected

NodeStateChangeCB - Wrapper callback

Arguments:w:Widget, client:XtPointer, call, XtPointer

Return Type:EcTVoid

Privilege:Private

ReadDirectoryInfo - Gets the current working dir, & builds it

Arguments:

Return Type:EcTVoid

Privilege:Protected

RefreshDisplayArea - Re-reads the directory. Redraws the icons

Arguments:

Return Type:EcTVoid

Privilege:Public

~CIDtHierarchical - Destructor

Arguments:

Return Type:Void

Privilege:Public

Associations:

The CIDtHierarchical class has associations with the following classes:

None

4.3.8 CIDtIconic Class

Parent Class:CIDtDisplayArea

Public:No

Distributed Object:No

Purpose and Description:

Displays Desktop objects in Iconic form.

Attributes:

myLastX - X coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myLastY - Y coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myMultipleSelectedFilesList - List of files in the multiple selection in a string form.

Data Type:RWCString

Privilege:Private

Default Value:

myMultipleSelectedStatus - Multiple Selected status. Whether multiple files are selected or not

Data Type:EcTInt

Privilege:Private

Default Value:

myRootLastX - X coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myRootLastY - Y coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myRootStartX - X coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myRootStartY - Y coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

mySelectedIconList - List of the icons selected

Data Type:RWTPtrSlist<CIDtDesktopObject>

Privilege:Private

Default Value:

myStartX - X coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myStartY - Y coordinate

Data Type:EcTInt

Privilege:Private

Default Value:

myXorGC - Xor Graphics context

Data Type:GC

Privilege:Private

Default Value:

Operations:

CheckDirectoryAndRefresh - Checks if directory status changed. If so, Refreshes the Directory

Arguments:

Return Type:EcTVoid

Privilege:Public

CIDtIconic - Constructor

Arguments:w:Widget, name:EcTChar*, dtWindow:CIDtDesktopWindow*,
height:Dimension

Return Type:Void

Privilege:Public

DrawRectangle - Draws the rectangle for selection

Arguments:

Return Type:EcTVoid

Privilege:Private

GetMultipleSelectedFilesList - Get the list of files in a string form

Arguments:

Return Type:RWCString

Privilege:Public

GetMultipleSelectedStatus - Return the multiple selected status. 0 - single selection. 1- multiple selection

Arguments:

Return Type:Boolean

Privilege:Public

GetSelectedIconList - Get the list of selected icons

Arguments:

Return Type:RWTPtrSlist<CIDtDesktop>

Privilege:Public

HighlightIcons - HighLight selected icons

Arguments:

Return Type:EcTVoid

Privilege:Private

MultipleSelection - Callback for MultipleSelection

Arguments:event:XEvent*, continuetodispatch:Boolean*

Return Type:EcTVOID
Privilege:Public

MultipleSelectionCB - Wrapper callback

Arguments:w:Widget, client:XtPointer, event:XEvent*, continuetodispatch:Boolean*

Return Type:EcTVOID

Privilege:Public

RefreshDisplayArea - Re-reads the directory.

Arguments:

Return Type:EcTVOID

Privilege:Public

UnhighlightIcons - Unhighlight selected icons

Arguments:

Return Type:EcTVOID

Privilege:Public

~CIDtIconic - Destructor

Arguments:

Return Type:Void

Privilege:Public

Associations:

The CIDtIconic class has associations with the following classes:

None

4.4 CSCI Structure

Table 4.4-1 provides a summary of the components which make up this CSCI.

Table 4.4-1. DESKT Components

Name	Description	DEV or OTS	Release
Desktop	To manage a set of Desktop Objects	DEV	A

4.5 CSCI Management and Operation

There are no Management and Operation activities that occur for the DESKT CSCI since it resides on the science users workstation. The management of the application is up to the individual user.

This page intentionally left blank.