

**Proprietary Software ChangedName**

Last edited: 6/01/01

# Table of Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1. What is Proprietary Software ChangedName? .....	1
1.2. Purpose/Assumptions .....	1
1.3. How This Manual is Organized .....	1
1.4. Typographical Conventions .....	2
<b>2. INSTALLATION .....</b>	<b>3</b>
2.1. System Requirements .....	3
2.1.1. Minimum Hardware Requirements .....	3
2.1.2. Software Requirements .....	3
2.1.3. System and Host Access Requirements .....	3
2.1.4. Installing the Software .....	3
<b>3. QUICK TOUR - SETTING UP A DEVELOPMENT PROJECT .....</b>	<b>5</b>
3.1. A Few Important Concepts Before Getting Started .....	5
3.1.1. Component Locking .....	5
3.1.2. The Proprietary Software Application (CSD) File .....	5
3.2. Development Group Dialog .....	8
3.2.1. General Tab .....	9
3.2.2. The New Screens Tab .....	11
3.3. Setting up a Development Project .....	12
3.3.1. Step 1: Creating The Initial Project File .....	12
3.3.2. Step 2: Start Proprietary Software ChangedName .....	12
3.3.3. Step 3: Creating and Saving a Master CSD .....	15
3.3.4. Step 4: Creating a Local CSD .....	17
3.3.5. Step 5: Developing a Local CSD .....	18
3.3.6. Step 6: Distributing the Completed Application .....	24
<b>4. MESSAGE PROCESSORS .....</b>	<b>25</b>
4.1. Creating a Message Processor .....	25
4.2. Message Handler Prefixes .....	26
4.3. Checking In Message Processors .....	27
4.4. Checking Out Message Processors .....	27
4.5. Deleting Message Processors .....	28
4.6. Recreating Message Processors in the Master CSD .....	28
<b>5. GLOBAL FUNCTIONS .....</b>	<b>29</b>
5.1. Creating a Global Function .....	29
5.2. Global Function Prefixes .....	31
5.3. Checking In Global Functions .....	31
5.4. Checking Out Global Functions .....	32
5.5. Deleting Global Functions .....	32
5.6. Recreating Global functions in the Master CSD .....	32
5.7. Viewing the Latest Global Functions - Get All .....	33
<b>6. RECORDS .....</b>	<b>35</b>
6.1. Record Characteristics .....	35
6.2. Creating Containers or Records at the Same Time as Another Developer .....	35
6.3. Records (CB) .....	35
<b>7. FAQs AND RECOMMENDATIONS .....</b>	<b>37</b>
7.1. Frequently Asked Questions: .....	37
7.2. Recommendations for using ChangedName .....	40

<b>8. TIPS &amp; TROUBLESHOOTING .....</b>	<b>43</b>
8.1. Missing files from the Master CSD .....	43
8.2. Unable to Add Screens to the Master CSD .....	43
8.3. Keyboard Mapping.....	43
8.4. Renaming/Deleting Screens From Master CSD Allowed .....	44
8.5. Screen Names .....	44
8.6. Keeping the Master CSD Current .....	44
<b>9. INDEX .....</b>	<b>45</b>

# 1. Introduction

---

## 1.1. What is Proprietary Software ChangedName?

*Proprietary Software ChangedName* is used for large development projects when a team of developers will be building *Proprietary Software Enterprise* solutions. *ChangedName* provides a full development platform for group development.

### ***ChangedName* features:**

- No limit to the number of developers on a project
- Check-in/check-out development environment
- Version control

*ChangedName* allows your company to capitalize on all of the existing business rules while adding new business logic at the desktop. *ChangedName* facilitates shared development. *Proprietary Software* applications, stored on a common server, are based upon an object-based architecture. Each developer checks-out the objects they require. As the object modules are completed, the developer checks-in the changed modules. Templates can be created to ensure consistent development standards.

As the scope of a company's projects grow, *Proprietary Software Enterprise* can easily be upgraded to *Proprietary Software ChangedName*.

## 1.2. Purpose/Assumptions

This Proprietary Software manual describes the operation of the Proprietary Software ChangedName product and is intended for users who have had Proprietary Software training and understand how to create Proprietary Software applications. We also assume that users know how to organize files and folders using Microsoft Windows.

## 1.3. How This Manual is Organized

This manual will guide you through the software installation and then give you a quick tour that will help you to set up your first development project. After setting up the initial project, this manual will guide you through creating Message Processors and Global Functions.

Be sure to read the Frequently Asked Questions, Recommendations, Tips and Troubleshooting. These sections will give you valuable tips that will be helpful when developing your applications.

## 1.4. Typographical Conventions

To help with the readability of this User Manual, we have used certain typographical conventions as follows:

System responses such as alerts will appear in "*Quotes, underlined and italicized*" and user input options appear bolded, for example: **OK, Cancel**

Notes, as indicated below, provide important information and it is strongly recommended that you read all notes



**Note:** Notes contain important information. It is strongly recommended that you read all notes.

Warnings as indicated below provide crucial information. Not heeding the information contained in a warning may result in irreparable damage to your ChangedName project.



**Warning:** Warnings contain crucial information. Ignoring a warning may cause irreparable damage to your ChangedName project.

# 2. Installation

---

## 2.1. System Requirements

### 2.1.1. Minimum Hardware Requirements

RAM: 64 MB  
Hard Disk: 60 MB of free space  
Processor Speed: 300 MHz or better

### 2.1.2. Software Requirements

**Operating System:** Windows 98, Windows NT or Windows 2000

**Browser:** Internet Explorer 5.0 or better

Before installing, Proprietary Software products please install Internet Explorer 5.0. Internet Explorer 5.0 contains the JScript, VB Script engines and the XML COM object that Proprietary Software products utilize.

### 2.1.3. System and Host Access Requirements

Workstations running Proprietary Software must have access to the host computer in order to capture host screens.

If you plan to access the host through TCP/IP, you can use Owner's integrated TN/3270 or TN/5250 emulators. You may also use any of the host-access software listed below:

<u>32 Bit</u>
<ul style="list-style-type: none"><li>• Advantis Passport</li><li>• Eicon Aviva</li><li>• Extra PC</li><li>• Hummingbird</li><li>• IBM PComm Client Access</li><li>• IBM Personal Communication</li><li>• NetSoft NS/Elite</li><li>• Reflection</li><li>• WallData Rumba</li></ul>

### 2.1.4. Installing the Software

To install Proprietary Software ChangedName, follow these steps:

1. Close all open programs

2. Insert the Proprietary Software CD which should start automatically.  
If the CD does not start automatically or if you must restart the installation, follow the steps below:  
Access your CD-Rom using Windows Explorer  
Doubleclick Setup.exe
3. On the Welcome screen click **Next**
4. Read the License Agreement and click **Yes** if you agree. If you do not agree click **No**. The install program will Exit.
5. Insert the Customer Information and Serial Number



**Note:** Your key ID is on the back of the CD package

6. Select a Destination folder and click **Next**
7. Select the Setup Type:

Type Of Install	Brief Explanation	Files Installed
Typical	Recommended for most users	Program Files, Help files, Samples, Setup Applications
Compact	Programs will be installed with minimum required options	Program Files
Custom	You may choose the options you want to install. Recommended for advanced users.	Program Files, Help files, Samples, Event Logging support, Setup Applications

8. Select the Program Folder and click **Next**
9. Select **Yes, I want to restart my computer now** and click **Finish**.

# 3. Quick Tour - Setting up a Development Project

---

## 3.1. A Few Important Concepts Before Getting Started

Before we start using Proprietary Software ChangedName, let's review some of its important concepts.

### 3.1.1. Component Locking

Proprietary Software ChangedName lets each developer in a team work on one component of an application without compromising each other's efforts. While a developer works on a component, Proprietary Software ChangedName locks other developers out of that component so that it can't be modified by anyone else.

### 3.1.2. The Proprietary Software Application (CSD) File

The Proprietary Software Application file, called the CSD, contains all of the objects and code necessary for your application. Proprietary Software objects include:

- Captured screen objects with their associated Screen Initialization and Auto Screen Handler SmartCode modules
- Fields, and Containers
- Message Processors
- Keyboard mappers
- Unexpected Screen Handler
- Menu specifications
- Timers
- Global Functions
- System-wide Windows
- File exchange procedures

The Proprietary Software reference manual describes each of these components in more detail.

There are two types of CSD files associated with Proprietary Software ChangedName : a Master CSD and one or more Local CSD files.

## The Master CSD

The Master CSD contains all of the objects created by the development team. You may assign the CSD any name, followed by the extension “.csd”.



**Note:** If your system does not support long file names, remember to use an appropriate 8.3 format file name.

To minimize the likelihood of error, we recommend naming the CSD file something like **MasterX.csd**, which identifies the file unambiguously.

There are two files associated with the Master CSD -- MasterX.log and MasterX.csg.

**MasterX.log** This file tracks all of the objects checked in or out of the Master CSD.



**Note:** This log file is created only after a Local CSD is created and after an object is checked out.

**MasterX.csg** This file coordinates all changes to the Master CSD from all Local CSDs



**Warning:** Do NOT modify the CSG file!

```
File Edit Search Help
12/21/1998 14:16:30, corread, CheckOut, Screen 2
12/21/1998 14:16:38, corread, CheckIn, Screen 2
12/21/1998 14:16:52, corread, CheckOut, Screen 2
12/21/1998 14:16:55, corread, CheckIn, Screen 2
12/21/1998 14:16:56, corread, CheckOut, Screen 2
12/21/1998 14:16:57, corread, CheckIn, Screen 2
```

Figure 1- .log File

The system needs both the .log and .csg files to identify the Master CSD.

## Local CSDs

Developers use their own Local CSDs to update and edit the objects checked out from a Master CSD. You can create a Local CSD from a Master CSD at any time.



**Warning:** Never update the Master CSD directly. This causes application errors. Instead, edit a Local CSD and have the Local CSD update the Master CSD automatically when you check the object in. To prevent accidental editing of the Master CSD, we ***strongly recommend*** keeping the Master and Local CSDs in separate directories.

Remember that the Local CSD does not have to reside locally. It can reside anywhere on the network.

---

## 3.2. Development Group Dialog

Before we begin developing a project, let's take a quick tour of the Development Group Dialog. You will access this dialog throughout your development project and it is important to become familiar with its features. Under this section we will discuss the **General** and **New Screens** tab of the Development Group Dialog. These tabs are required for the Quick Tour. The **Global Function** and **Message Processor** tabs will be discussed a little later in Sections 4 and 5.

Access the Group Development dialog by selecting **Group Development** from the **File** Menu:

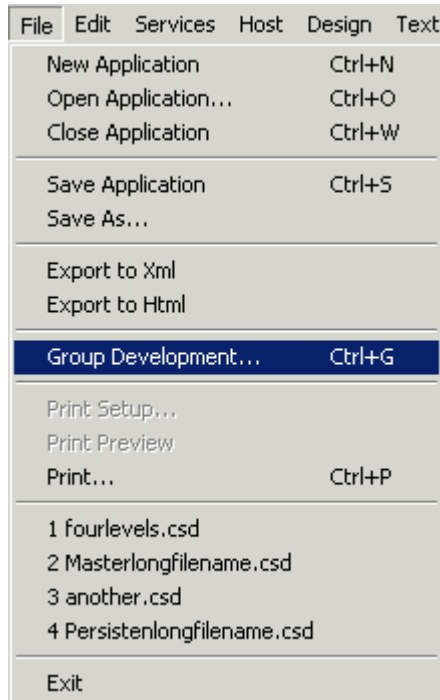


Figure 2 - Group Development Menu Item

The Group Development window entitled **ChangedName Dialog** will open and the **General** tab will be selected.

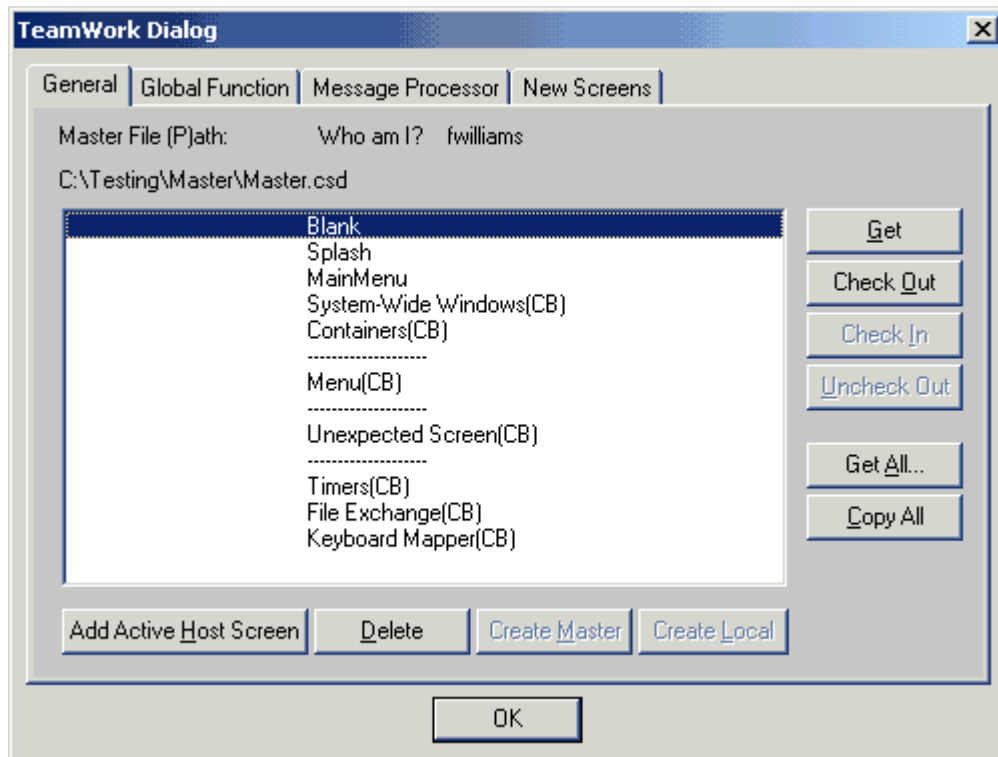


Figure 3 - ChangedName Dialog, General Tab

### 3.2.1. General Tab

Directly beneath the General tab Proprietary Software ChangedName will display the **Master File Path:** (in Figure 3 above, the path is C:\Testing\Master\Master.csd), and the name of the user as indicated to the right of **Who am I?** (in the above figure the user is fwilliams)

The central area of the screen lists the objects in the Master CSD that are available for check out.

#### **Group Dialog Command Buttons for the General Tab:**

The command buttons are described in order of screen appearance from left to right then from top to bottom as follows:

**Add Active Host Screen:** Allows you to add a screen to the Master CSD if the screen to be added has the same name as a screen currently existing in the Master. To add such a screen to the Master CSD, the screen must be the active screen behind the Group Development dialog window. This button is not enabled for screens that were checked out.

**Delete:** Deletes the selected screen, i.e., the item highlighted in the central area containing the list of objects. Deleting a screen from the Master CSD also removes the associated Screen Initialization and Auto Screen

Handler SmartCode modules.

You may not delete the following global objects, i.e., those with a “CB” next to them:

- System Wide Windows
- Containers
- Menu
- Message Processor
- Unexpected Screen
- Global Functions
- Timers
- File Exchange
- Keyboard Mapper

Additionally, you may NOT delete screen objects that have been checked out

<b>Create Master</b>	Allows you to create a Master CD
<b>Create Local</b>	Allows you to create a Local CSD from a Master CSD
<b>Get</b>	Updates the copy of an object in the Local CSD with the current version of the object in the Master CSD. The <b>Get</b> button is disabled if you attempt to get a screen that you have already checked out.
<b>Check Out</b>	Checks out the highlighted object in the list of objects. While the object is checked out, no other developer can work on it
<b>Check In</b>	Checks a revised copy of an object from the Local CSD back into the Master CSD so that the new version of the object is available to others in the group
<b>Uncheck Out</b>	Will “undo” any changes to the object in the Local CSD and leaves the copy of the object in the Master CSD unchanged
<b>Get All</b>	Updates the copy of all objects in the Local CSD with the current version of those objects in the Master CSD
<b>Copy All</b>	Overwrites the current content of the Local CSD with the current content of the Master CSD



**Note:** If you delete a screen object from the Master CSD, the name of that screen remains on the New Screens tab until the screen is deleted from the Local CSD. This is because Proprietary Software ChangedName treats the “deleted” screen as if it were a new screen waiting to be added to the Master CSD. This gives you a chance to add the screen to the Master CSD again – in case you removed the screen from the Master CSD by mistake.

## 3.2.2. The New Screens Tab

The **New Screens** tab in the **Group Development** dialog shows:

- Screens that were captured but not yet added to the Master CSD.
- Screens that were deleted from the Master CSD but reside in the Local CSD.
- The new name of a checked out screen after the name is changed.

The **Add** button lets you add, to the Master CSD, any newly captured screens which you've added to the Local CSD.

When you click on the **New Screens** tab, the system automatically identifies screens in the Local CSD that are not in the Master CSD.

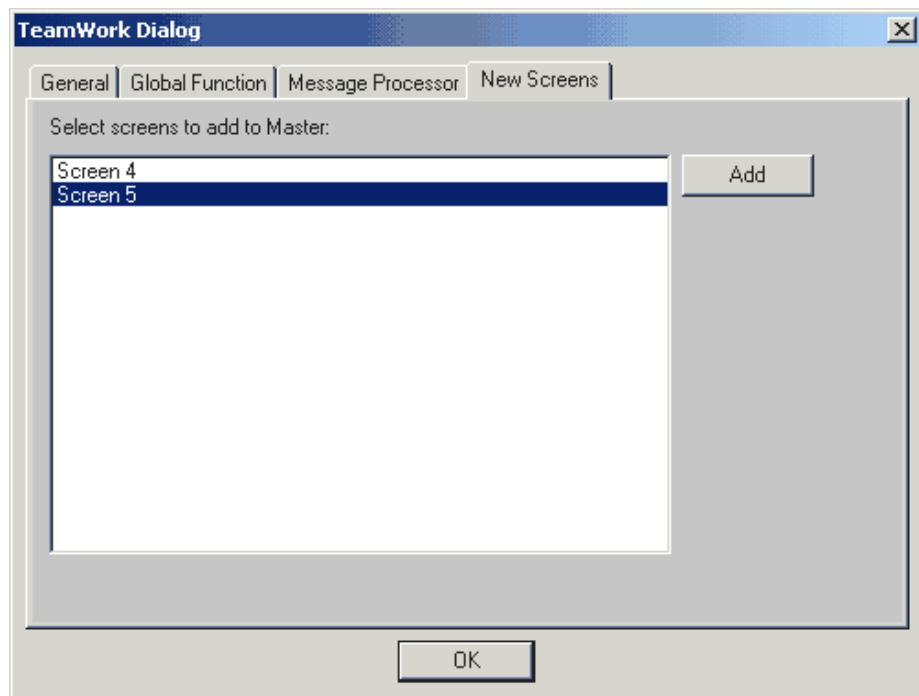


**Note:** You may NOT rename or otherwise modify any screen that was not first checked out of the Master CSD.

## Adding Screens to the Master CSD

To add screens to the Master CSD:

1. Highlight the screen you want to add (refer to the sample screen below).
2. Click **Add**.
3. When you have finished adding screens, click **OK**.



*Figure 4 - Adding New Screen*

## 3.3. Setting up a Development Project

This section presents an overview of how to set up a development project with two or more developers.

After you have installed Proprietary Software ChangedName (For installation instructions see section 2.1.4 Installing the Software), follow the next 5 steps as outlined below:

### 3.3.1. Step 1: Creating The Initial Project File

Before you create the Master CSD and the Local CSDs, you must set up your Initial Project File. In most cases you will want to start with a blank file. However, sometimes you might be starting with an existing application, or you might have to capture only a few screens before creating a Master CSD.

### 3.3.2. Step 2: Start Proprietary Software ChangedName

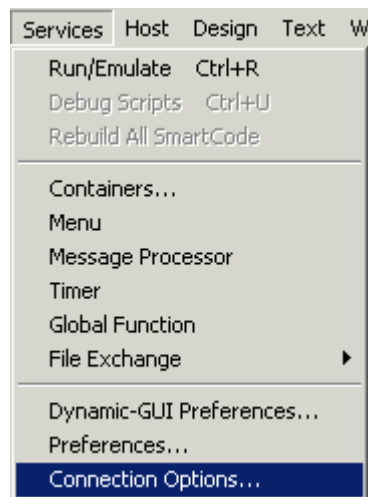
To start Proprietary Software ChangedName , do the following:

1. After clicking **Start** from the Windows Taskbar, select **Owner, Inc** from within the **Program** files.
2. Within the Proprietary Software Program Group, select **Proprietary Software ChangedName**.
3. Once you are in Proprietary Software ChangedName, select the **File** menu.
4. If you are starting a new project, click **New Application**. You will see a blank host screen as indicated in Figure 5 below.



*Figure 5 - New Application Blank Screen*

5. Select **Connection Options** from the **Services** Menu to display the **Connection Options** screen.



*Figure 6 - Services Menu*

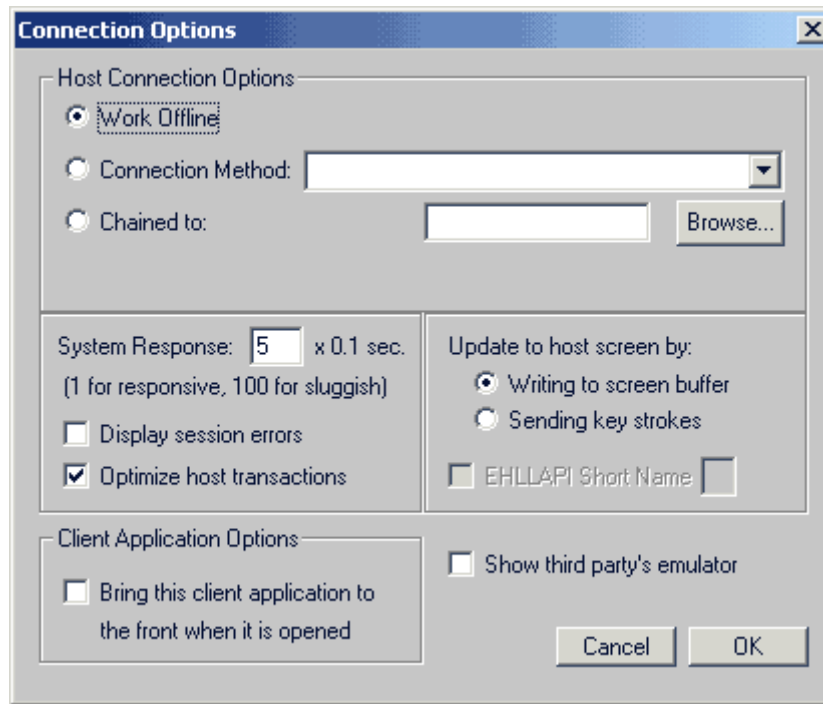


Figure 7 - Connection Options Window

6. Select the desired connection method from the dropdown list. Refer to the connection method's Help file for help in choosing a connection method.
7. From the **Services** menu, select **Preferences** (shown below).

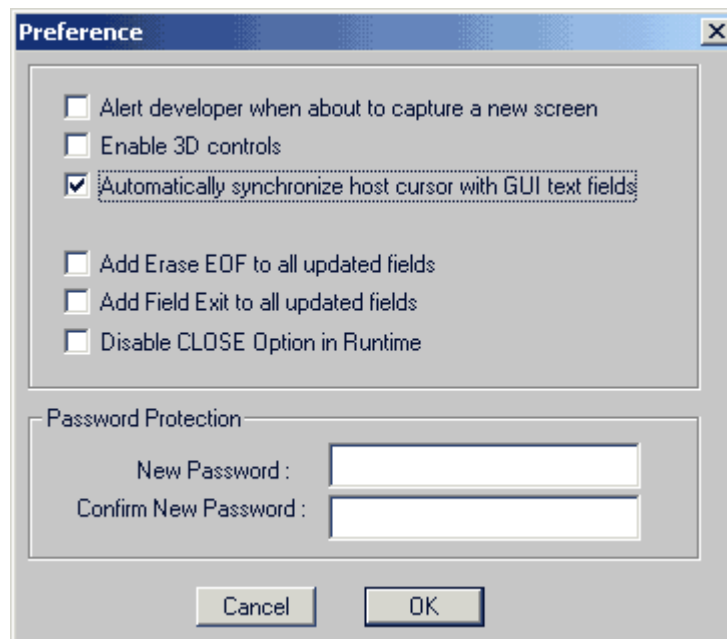


Figure 8 - Preference Window

8. The following check boxes should be checked:
  - Automatically synchronize host cursor with GUI text fieldsBy setting the Connection Options **Preferences** before you create the Local CSD, they will be set uniformly for all newly created Local CSDs.
9. Press **OK** to close the Preference dialog and display the Main screen again.
10. Select **Run/Emulate** from the **Services** menu and capture screens. The first Host screen is displayed. (For information on capturing screens and creating a Proprietary Software application file, consult the Proprietary Software manual and the training guide.)
11. Save the application.
12. Type the name of the new application and click **Save**.
13. Create a backup of the Initial File by selecting **Save As** and saving the file under a different name. The renamed file is now the active file.
14. From the **File** menu select **Close Application** to close the backup file.
15. From the **File** menu select **Open Application** to open the original Initial File.

You are now ready to proceed to Step 3: Creating and Saving a Master.CSD

### 3.3.3. Step 3: Creating and Saving a Master CSD

In this section, you will create the Master CSD from the Initial Project File and save the Master CSD in a location that is accessible to all the developers on the project. When setting up and naming the Master CSD, make sure to follow the guidelines given in the section titled The Master CSD on page 6.

When you create the Master CSD, the system automatically creates a **.csg** file with the same name as the Master CSD. A similarly named **.log** file is automatically created when the first object is checked out of the Master CSD and into a Local CSD. By default, the **.csg** and **.log** files are placed in the same directory where the Master CSD is.



**Warning: Never move, delete or rename these files!** Without these files the system can't identify your Master CSD and connection between the Local and Master CSD is lost.

**Never work directly on the Master CSD!** The Master CSD communicates with each of the Local CSDs and manages the work so that developers do not overwrite each other's code. To make changes to the Master CSD, check out the object you want to change and change it in the Local CSD. Then check the changed object in the Local CSD back into the Master CSD.

---

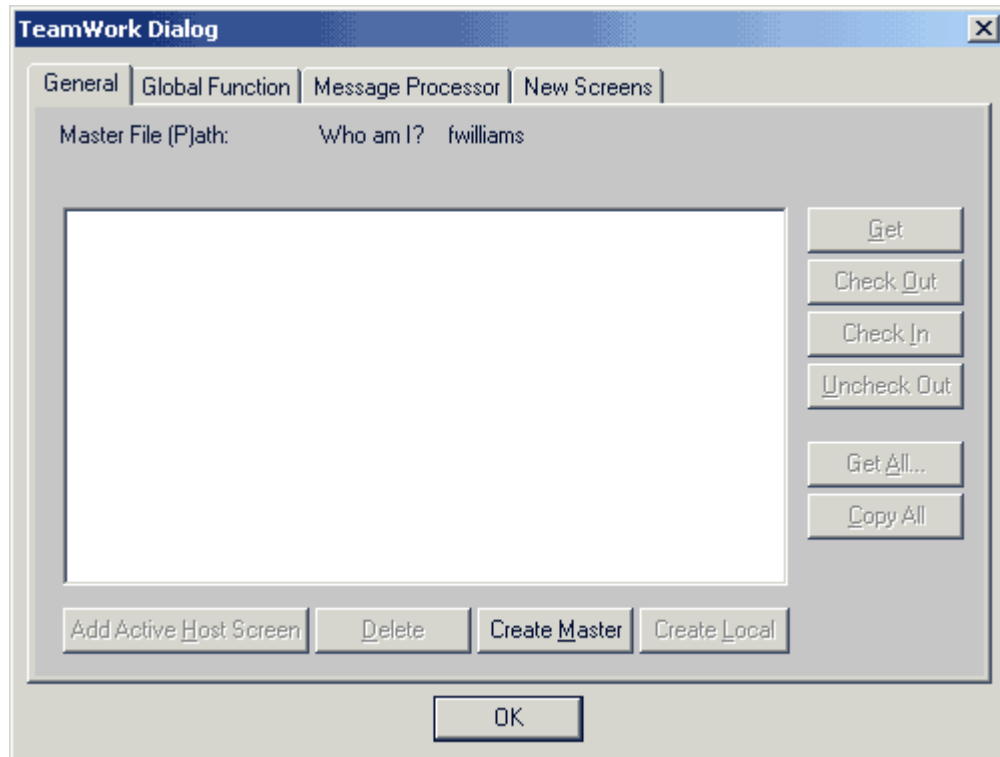
To create a Master CSD, do the following:

1. From the **File** menu, select **Group Development**. The ChangedName dialog is displayed (See Figure 9).

2. In the **General tab**, select **Create Master**. Notice that all other buttons are disabled.



**Note:** If the **Create Master** button is disabled, it means you are trying to create a Master CSD from another Master



*Figure 9 - ChangedName Dialog*

After selecting the **Create Master** button, you will see the File Management dialog entitled **Open** shown below:

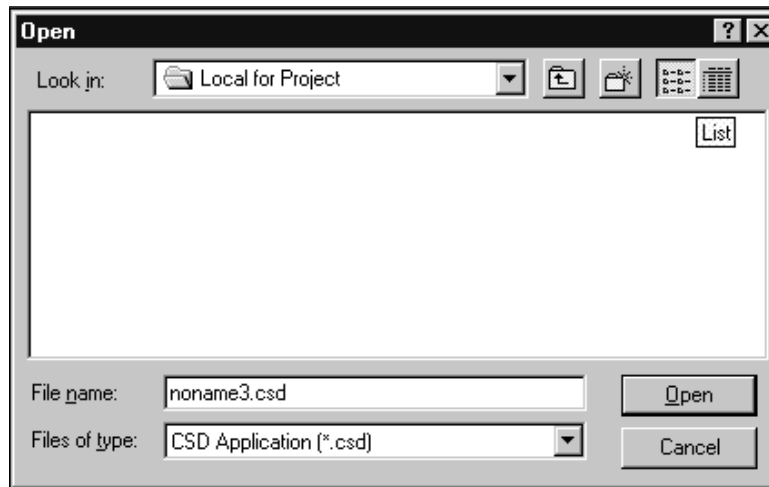


Figure 10 - Open Dialog

3. Select the directory where your Master CSD will reside, remembering that it should be on a drive accessible to the whole development team.
4. Type the name of the Master CSD and click **Open**.



**Note:** The Master CSD is created on the network, and the Initial CSD becomes Local to the Master that was just created. The Local CSD is the currently open application.

Now you are ready to go to Step 4, Creating a Local CSD

### 3.3.4. Step 4: Creating a Local CSD

Each developer can create a Local CSD by following the procedure outlined below:

1. Open the Master CSD.
2. From the **File** menu, select **Group Development**. The ChangedName dialog is displayed (See Figure 9).
3. Select the **Create Local** button.

Notice that all other buttons are disabled. If the Create Local button is disabled, it means you are not working with a Master CSD.

4. After selecting **Create Local**, the system displays the File Management dialog shown below.

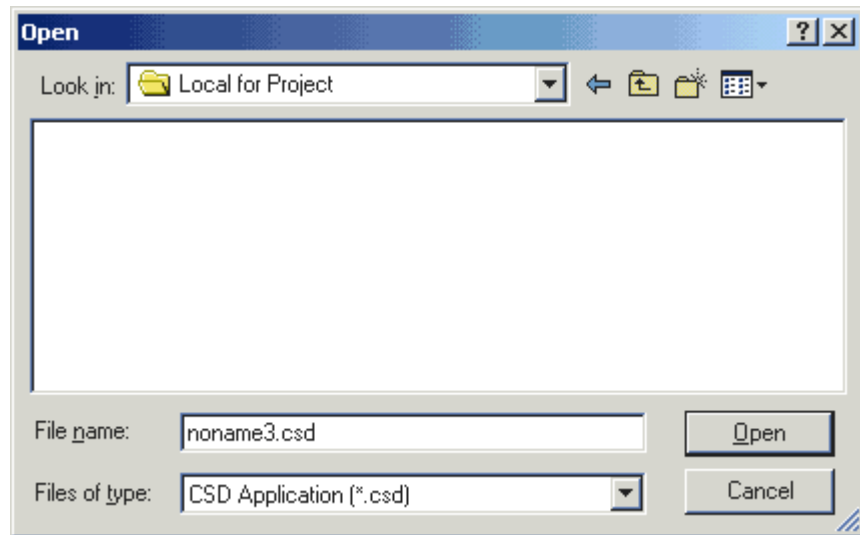


Figure 11 -Open Dialog

5. Select the directory where your Local CSD will reside. (You don't need to save the Local CSD on your local hard drive. You may place it anywhere on the network.)
6. Type the name of the Local CSD, and click **Open**.



**Note:** Once you have set up your project, you can't change the location of the Master CSD. Plan your projects accordingly. Place your Master and Local CSDs in a "safe" area on your network.

Now that we've created the Master CSD and also created a Local CSD from the Master, we are ready to proceed to step 5 and Develop the Local CSD

### 3.3.5. Step 5: Developing a Local CSD



**Note:** Each developer must work on his/her own Local CSD and not on the Master CSD or on someone else's Local CSD. The name of the file that is open always appears on the upper left-hand corner of the Proprietary Software Title Bar.

Proprietary Software ChangedName identifies Local CSDs by location, even if the names are identical. Therefore, to eliminate confusion within a network, we recommend that the development team establish a reasonable and unambiguous naming convention for Local CSDs. For more details regarding naming conventions, see section entitled **The Master CSD** on page 6.

- Each developer checks out the objects they will be developing.
- After modifying an object, the developer checks the modified object into the Master CSD to make the new object available to others.
- During development each developer can update an object in their Local CSD with the corresponding object from the Master CSD by using the Get or Get All buttons. (as described in section 3.2.1)
- Before deploying the finished application, all developers must check in their screen objects.

## Checking Objects Out/In

### Checking Objects Out

Each developer checks out the objects they plans to use. You must do this periodically to get the latest version of objects that were checked in by other developers. To check out an object:

1. Select **Group Development** from the **File** menu to access the Group Development dialog.
2. Highlight the item you want to check out.
3. Click **Check Out**

Your user name appears on the left-hand side of the object. In addition, the object in your Local CSD becomes editable. (see Figure 12)

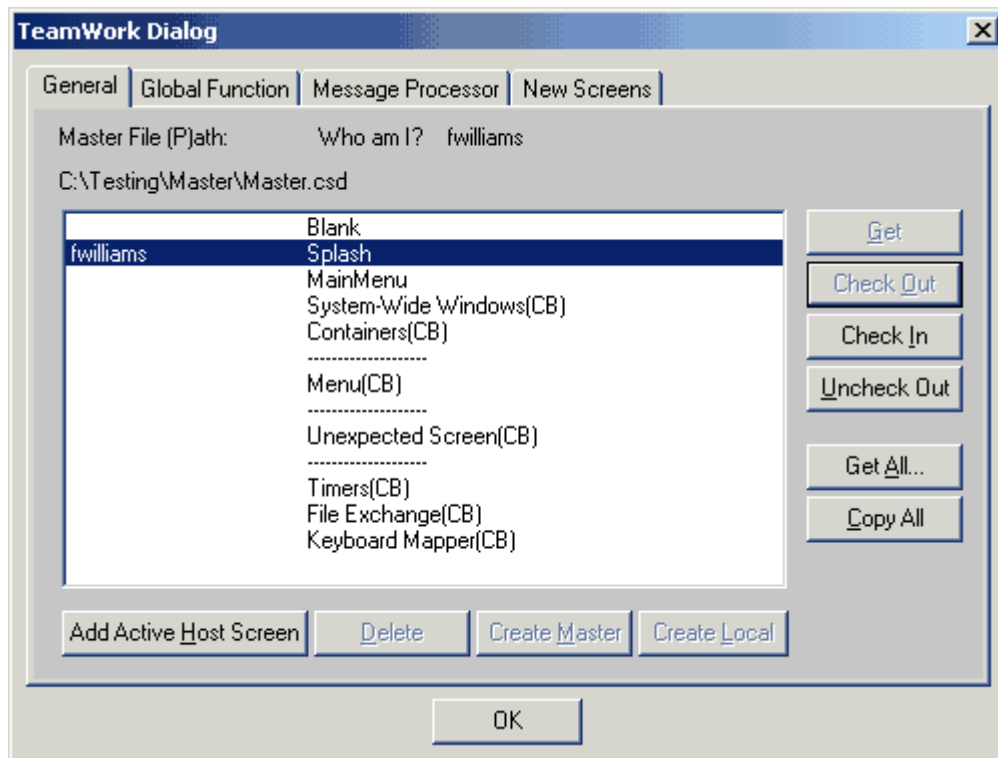


Figure 12 - Object Check Out

## Properties of Checked Out Objects

Object	Properties when checked out to a Local CSD
Host Screen, including the Blank Screen	You can create fields and records, modify GUIs, Screen Tasks, Screen Initialization, Auto Screen Handler, and Screen Recognition functions.
System Wide Window	You can create GUIs that are active for all host screens with associated SmartCode.
Containers (CB)	This is a locking mechanism to prevent others from creating Containers. To create Containers, the user should: <ol style="list-style-type: none"> <li>1. Check out Containers.</li> <li>2. Select the <b>Services</b> menu.</li> <li>3. Click the <b>Containers</b> option.</li> <li>4. Use the dialog box to create Containers.</li> </ol> <p><b>Note:</b> The renaming feature has been disabled because of the impact the number of users has on the development project.</p>
Menu (CB)	After you select Menu on the Services menu, this dialog is accessible.
Message Processor (CB)	After you select <b>Message Processor</b> from the <b>Services</b> menu, this dialog is accessible. This menu lets you create Message Processors that can be shared by all users when once the processors are checked into the Master CSD.
Unexpected Screen (CB)	From the <b>Host</b> menu select <b>Unexpected Screen Handler</b> . When this Screen Handler is checked out, it is editable.
Global Function (CB)	When you select Global Function from the Services menu, this dialog is accessible.
Timers (CB)	When you select <b>Timer</b> from the <b>Services</b> menu, this dialog is editable. You may now add timers that can be shared when checked into the Master CSD.
File Exchange (CB)	From the <b>Services</b> menu select, <b>File Exchange</b> and click <b>Import</b> or <b>Export</b> . When a File Exchange CB is checked out, this dialog is editable. Make sure files in a Network area are accessible to all users.
Keyboard Mapper (CB)	When you select <b>Keyboard Mapping</b> from the <b>Host</b> menu, this dialog is editable. <p><b>Note:</b> The Keyboard Mapping settings are stored in a file in the same directory as your Local CSD. If you plan to move the CSD to a new location, you must move the keyboard mapping file too, or use SmartCode to remap the keyboard.</p>

## Notification of Checked-Out Items

When a master CSD containing checked-out items is opened, an alert message informs the user that items have been checked out. The master CSD can then only be opened in read-only mode until all items have been checked in.

## Checking Objects In

When you're done coding an object in your Local CSD, you must check it back into the Master CSD. This copies your code and/or GUI objects to the Master CSD and makes your revisions available to other developers. To check in an object, do the following:

1. Access the **Group Development** dialog,
2. Highlight the item you want to check in, and
3. Select the **Check In** (See the sample screen on page 20).

Your user name disappears from the left-hand side of the screen, and the object is no longer editable in your Local CSD.

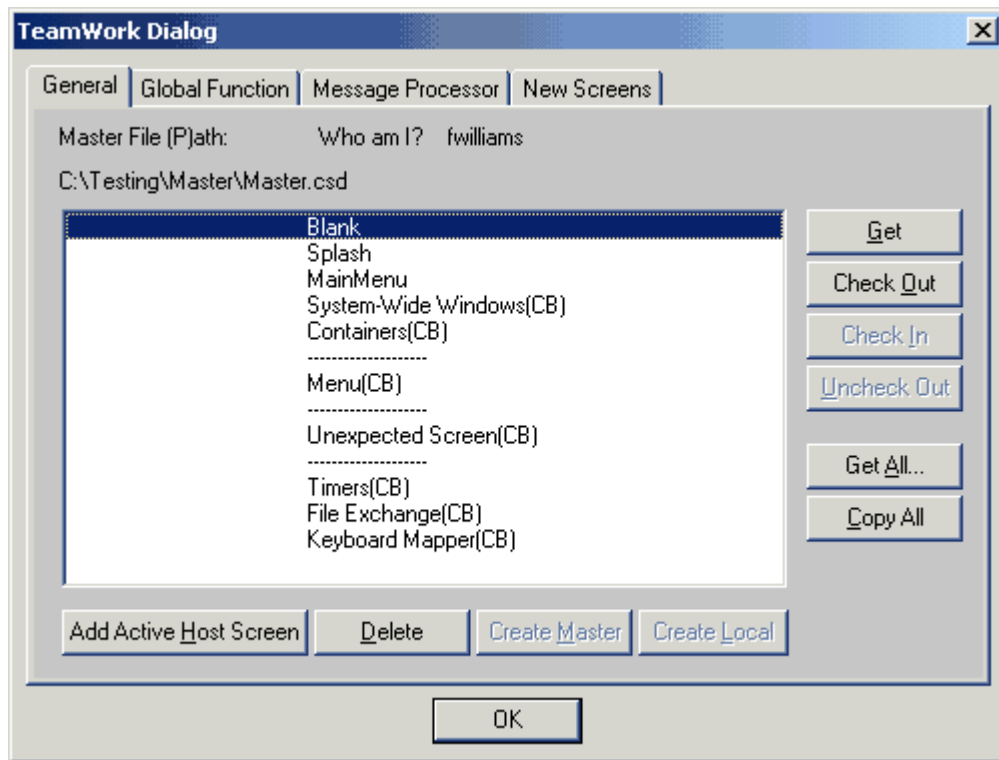


Figure 13 - Object Checked In

To access a modified object in the Master CSD, other developers can:

1. Highlight the object in Group Development, and
2. Select the Get button to copy the object to their Local CSD.



**Note:** If you check out an object from the Master CSD into your Local CSD, this checked-out object updates / overwrites any previous version of the same object you may have in your Local CSD. However, when you check a modified copy of an object into the Master CSD, any copies of that object in other users' Local CSDs are NOT automatically updated.

## Adding Objects

### *Adding A Screen to the Master CSD*

To copy a newly captured screen to the master, follow these steps:

1. Open your Local CSD.
2. Highlight the desired screen on the Screen Object list -- so you can see the host screen. (See the sample screen on page 22.)
3. From the **File** menu select **Group Development**.
4. Select **Add Active Host Screen** to add the selected screen to the Master CSD.
5. Click **OK** button to exit the Group Development dialog.
6. To add another screen to the Master CSD, repeat steps 2 through 5.



**Note:** After adding each screen, you must exit the Group Development dialog before you can add another screen to the Master CSD.

### *Adding Multiple Screens To the Master CSD:*

To add multiple screens to the Master, do the following:

1. From the **File** menu select **Group Development**.
2. Select the **New Screens** tab to view the list of the newly captured screens.
3. While pressing and holding the **Ctrl** key, highlight the screens you want to add.
4. Click **Add**.

Once you've added the screens, select the General tab. The newly added screens will appear on the list..

### *Adding a Deleted Screen Back Into the Master CSD:*

1. From the **File** menu select Group Development.
2. Select the **New Screens** tab.
3. Select the screen you want to add, i.e., the screen previously deleted from the Master CSD but which is still on the **New Screens** list.
4. Click the **Add** button.

### *Recovering a Screen Accidentally Deleted From the Local CSD*

1. From the **File** menu select **Group Development** in the Local CSD.
2. Select the screen you want to copy to your Local CSD.

3. Select **Get**. This copies the screen object and all of its associated code from the Master CSD to the Local CSD.

## Deleting Objects

### *Deleting a Screen From the Master CSD*

1. From the **File** menu select **Group Development**.
2. Select the name of the screen object you want to delete. If the screen is checked out, it cannot be deleted until it is checked in by the developer.
3. Select **Delete**.

This deletes the screen from the Master CSD.



**Note:** If the screen is available in the user's Local CSD but has been deleted from the Master CSD, the screen name remains in the New Screen list.

4. Delete the screen from the Local CSD.

### *Deleting a Screen From the Local CSD:*

1. From the screen list (the left drop-down box on the toolbar), select the host screen you want to delete. The screen you select becomes visible on your monitor.



**Note:** Make sure the name of that screen is the screen in the screen list window.

2. Click anywhere on the host screen to select it.
3. From the Host menu select Delete Host Screen.



---

**Warning:** Never open the Master CSD directly to make changes. This can cause irreparable damage to the project. If you must work directly on the Master CSD, all developers must check in all of their objects. After changes are made directly to the Master CSD, all developers must create New Locals to access any changes made to the Master CSD.

---

### 3.3.6. Step 6: Distributing the Completed Application

Before deploying the finished application, all developers must check in their screen objects. Make a copy of the Master and test it thoroughly.



**Note:** All Local CSDs will NOT be pointing to this file.

Once testing is complete:

1. Install the Runtime version of Proprietary Software on all user machines.
2. Distribute the application (Master CSD) and any relevant files to the user machines.
3. Open the file with the runtime version of Proprietary Software.
4. Keep a separate copy of the Master CSD together with the **log** and **csg** files for future updating.

# 4. Message Processors

---

Proprietary Software ChangedName allows developers to create, Check In, Check Out, and Delete Global Functions and Message Processors. Multiple users manipulate individual Global Functions or Message Processors at any time.

## 4.1. Creating a Message Processor

To create a Message Processor:

1. Select **Message Processor** from the **Services** menu.
2. In the **Define Message Handlers** window, select **New**.
3. Enter a name for the Message Processor.
4. Click **OK**.

The Script Editor opens for the new Message Processor. Note that the name of the Message Processor is on the Title line and that the Script Editor is enabled. When you create a Message Processor, it is automatically checked out to the user who is creating it.

5. Enter the code for any of the 3 scripting languages supported by Proprietary Software products
6. To compile the code, select **Compile** from the **Script** menu if you are using JavaScript, VBScript or SmartCode. If you're using SmartCode, you can select the **Compile** button on the SmartCode palette. When you compile the code, the Message Processor is still checked out to the developer who created it.

To verify that the Message Processor is still checked out by the developer, do one of the following:

1. Select the **Message Processor** from the **Services** menu. When the **Define Message Handlers** dialog appears, you'll see the name of the Message Processor that was just created, prefixed by an asterisk "\*" (See Figure 14). The asterisk indicates that the developer in this Local CSD has the Message Processor checked out.



Figure 14 - Define Message Processor Dialog

2. Select **Group Development** from the **File** menu.
3. Select the **Message Processor** tab.

On the ChangedName Dialog, you'll see the name of the Message Processor that was just created and, to the left, the Network User ID of the developer who checked out the Message Processor. (See below)

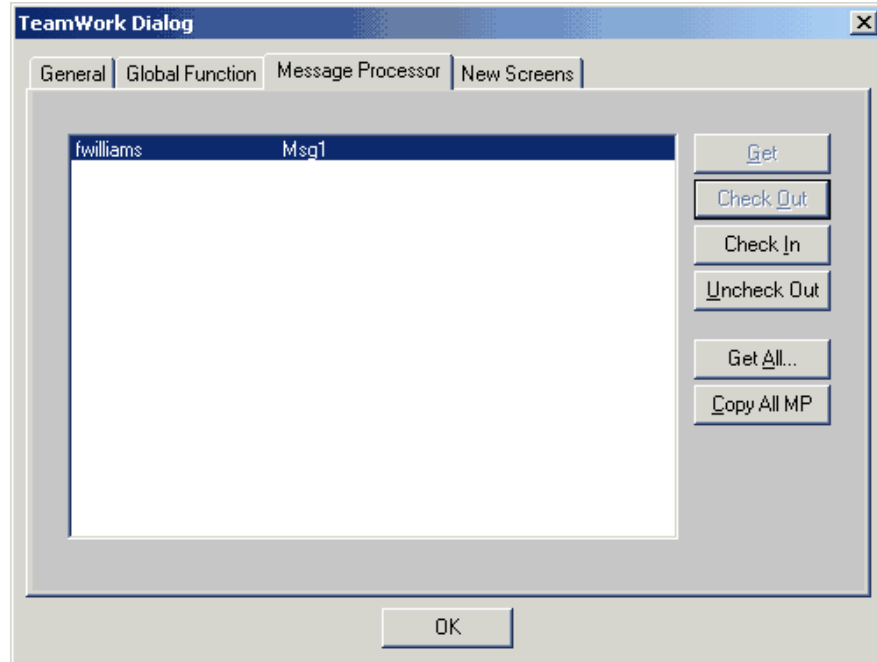


Figure 15 - Message Processor Tab of ChangedName Dialog

## 4.2. Message Handler Prefixes

The Define Message Handlers dialog (accessed via Services, Message Processor) lists all of the Message Processors that were created. Each prefix, ("\*", ".", "?" or blank) indicates a different Message Processor status as follows:

<b>Prefix</b>	<b>A Message Processor with the specified prefix indicates:</b>
*	The currently open Local CSD has the Message Processor checked out.
.	The Message Processor is checked out by another user.
?	This Message Processor was deleted from the Master CSD by another developer but still resides in the current Local CSD
<b>(blank)</b>	This Message Processor is available for check out. The Message Processor tab of the Group Development dialog shows the Network User ID of the developer who has the Message Processors checked out.

## 4.3. Checking In Message Processors

You can check Message Processors in and out from both the **Define Message Handler** dialog and from the Group Development dialog. To check in the Message Processor from the Message Handler dialog:

1. Select **Message Processor** from the **Services** menu.
2. Select a Message Processor that has an “\*” prefix.
3. Click **Check In**. The “\*” is removed from the list to show that the Message Processor is checked in to the Master CSD. Any developer may check out the Message Processor or get the updated code from the Message Processor.

To check in the Message Processor from the Group Development dialog:

1. Select **Group Development** from the **File** menu.
2. In the **ChangedName Dialog** Select the **Message Processor** tab.
3. Select the Message Processor with the Network User ID of the current developer who is logged into the computer.
4. Select **Check In**. The Network User ID is removed from the list to show that the Message Processor was checked in to the Master CSD. Any developer may check out the Message Processor or get the updated code from the Message Processor.

## 4.4. Checking Out Message Processors

To check out a Message Processor from the Message Handler dialog:

1. Select **Message Processor** from the **Services** menu.
2. Select a Message Processor that does not have a special prefix.
3. Select **Check Out**. The “\*” is added to the left of the Message Processor name in the list to indicate that the current developer of this Local CSD has the Message Processor checked out. To modify the Message Processor, select the **Script** button.

To check out the Message Processor from the Group Development dialog:

1. Select **Group Development** from the **File** menu.
2. In the **ChangedName Dialog** Select the **Message Processor** tab.
3. Select the Message Processor that does not contain a Network User ID.
4. Select **Check Out**. The Network User ID is added to the Message Processor name to show that the Message Processor was checked out of the Master CSD. The Message Processor is now available to the current developer to modify. To modify the Message Processor, select the **Script** button.

## 4.5. Deleting Message Processors

You may delete a Message Processor only from the **Message Handler Dialog** and only if the object is not checked out of the Master CSD. To delete a Message Processor, do the following:

1. Select **Message Processor** from the **Services** menu.
2. Select a Message Processor that does not have a special prefix.
3. Click **Delete**. The system displays a dialog, "Delete the message "xxx" and its actions?" **Cancel, Delete**
4. Click **Delete** to delete the Message Processor from the Master CSD and from the Local CSD.

## 4.6. Recreating Message Processors in the Master CSD

When you delete a Message Processor from the Master CSD, that Message Processor may still be available in a Local CSD. To add a Message Processor back into the Master CSD from a Local CSD, do the following:

1. Select **Message Processor** from the **Services** menu.
2. Select a Message Processor name that has the "?" prefix.
3. Select **Add**. The "?" prefix disappears from the Message Processor name to indicate that the processor was added to the Master CSD.

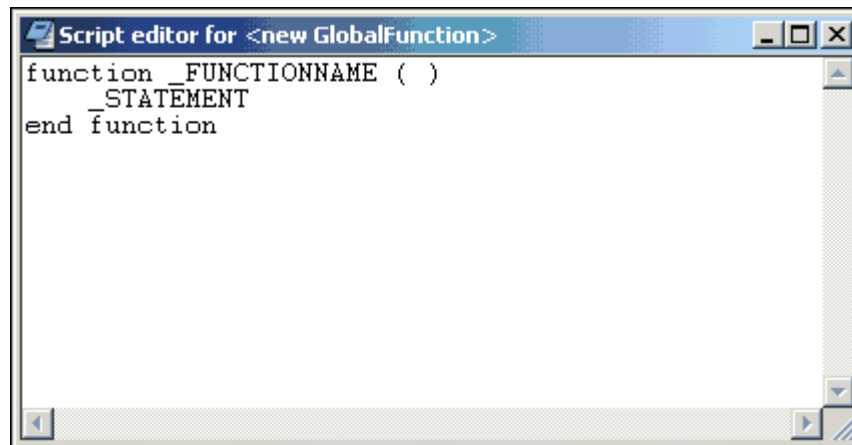
# 5. Global Functions

---

## 5.1. Creating a Global Function

To create a Global Function:

1. Select **Global Function** from the **Services** menu.
2. From **Define Global Functions** dialog select **New**.
3. The Script editor opens for the new Global Function.



*Figure 16 - Script Editor*

4. Name the Global Function by replacing “\_FUNCTIONNAME” with the name you want to give the new Global Function.
5. Enter the code for any of the 3 scripting languages supported by Proprietary Software products.
6. Compile the code by selecting **Compile** from the **Script** menu if you are using JavaScript, VBScript or SmartCode. If you’re coding in SmartCode, you can select the **Compile** button on the SmartCode palette. While you compile the code, the Global Function remains checked out by the developer who created it.

To verify that the Global Function is still checked out, do one of the following:

1. Select **Global Function** from the **Services** menu. When the dialog appears, you’ll see, on the list, the name of the Message Processor that was just created, prefixed by an asterisk “\*”. The asterisk indicates that a developer in this Local CSD has the Global Function checked out.

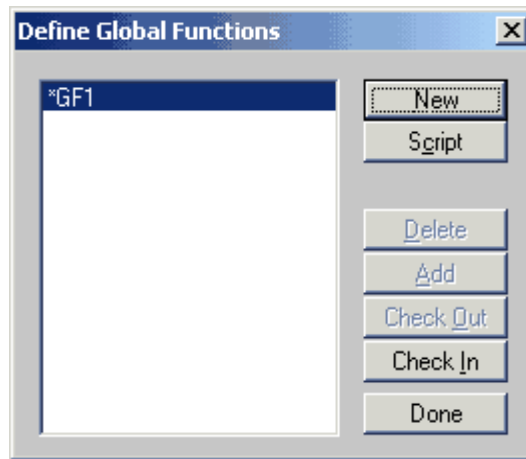


Figure 17 - Define Global Function Window

OR

2. Select **Group Development** from the **File** menu.
3. In the **ChangedName Dialog** select the **Global Function** tab.
4. The system displays the name of the Global Function that was just created and the left, the Network User ID of the developer who checked out the Global Function (see below).

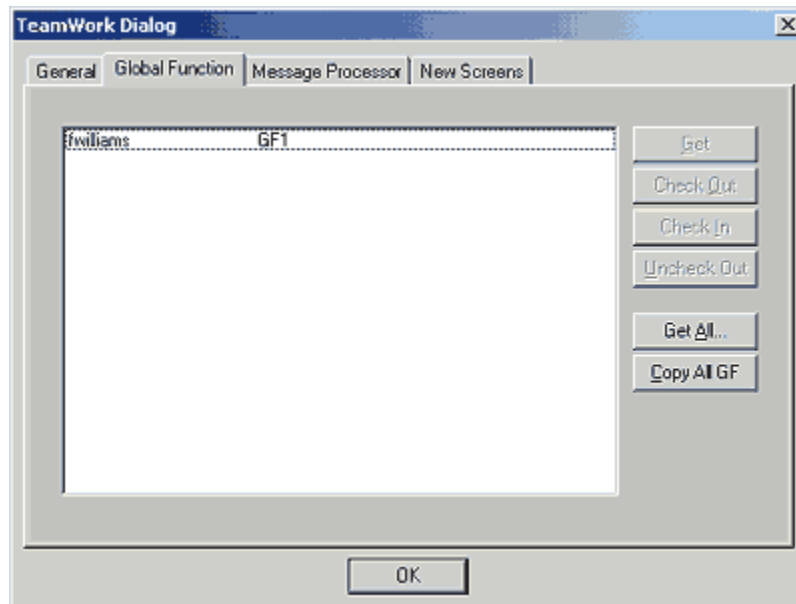


Figure 18 - Global Functions Tab of ChangedName Dialog

## 5.2. Global Function Prefixes

The Define Global Functions dialog lists all of the Global Functions that were created. Each prefix, (“\*”, “.”, “?” or blank) indicates a different Global Function status as follows:

<b>Prefix</b>	<b>A Global Function with the specified prefix indicates:</b>
*	The currently open Local CSD has the Global Function checked out.
.	The Global Function is checked out by another user.
?	This Global Function was deleted from the Master CSD by another developer but still resides in the current Local CSD.
<i>(blank)</i>	It is available for checking out. The Global Function tab of the Group Development dialog shows the Network User Id of the developer who has the Global Functions checked out.

## 5.3. Checking In Global Functions

You can check in and check out Global Functions from both the Message Handler dialog and the Group Development dialog. To Check In the Global Function from the Message Handler dialog:

1. Select **Global Function** from the **Services** menu.
2. Select the Global Function that has an “\*” prefix.
3. In the Define Global Function dialog select **Check In**. The “\*” is removed from the list to show that the Global Function was checked in to the Master CSD. Any developer may check out the Global Function or get the updated code from the Global Function.

To check in the Global Function from the Group Development dialog:

1. Select **Group Development** from the **File** menu.
2. In the **ChangedName Dialog** select the **Global Function** tab.
3. Select the Global Function with the Network User ID of the current developer that is logged in to the computer.
4. Select **Check In**. The Network User ID is removed from the list to show that the Global Function was checked in to the Master CSD. Any developer may check out the Global Function or get the updated code from the Global Function.

## 5.4. Checking Out Global Functions

To check out a Global Function from the Message Handler dialog:

1. Select **Global Function** from the **Services** menu.
2. From the **Define Global Functions** dialog select a Global Function that does not have a special prefix.
3. Select **Check Out** An "\*" is added to the left of the Global Function name in the list to indicate that the current developer of this Local CSD has the Global Function checked out. To modify the Global Function, select the **Script** button.

To check out the Global Function from the Group Development dialog:

1. Select **Group Development** from the **File** menu.
2. From the **ChangedName Dialog** select the **Global Function** tab.
3. Select the Global Function that does not have a Network User ID next to it.
4. Click on the Check Out button. The Network User ID is added to the Global Function name to indicate that the Global Function was checked out of the Master CSD. The function is now available for the current developer to modify. To modify the Global Function, select the Script button.

## 5.5. Deleting Global Functions

You can delete a Global Function from the Message Handler Dialog only when the object is not checked out of the Master CSD. To delete a Global Function, follow these steps:

1. Select Global Function on the Services menu.
2. Select a Global Function that does not have a special prefix.
3. Select the Delete button.  
The system displays a dialog, asking if you want to delete the Message Processor.
4. Select the Delete button, and the Global Function is deleted from the Master CSD and from the Local CSD that performed the delete operation.

## 5.6. Recreating Global functions in the Master CSD

When you delete a Global Function from the Master CSD, that Global Function may still be available in a Local CSD. To add the Global Function back into the Master CSD, do the following:

1. Select Global Function on the Services menu.
2. Select the Global Function name with the "?" prefix.

3. Select the Add button. The “?” prefix disappears from the Global Function name, which indicates that the function was added to the Master CSD.

## **5.7. Viewing the Latest Global Functions - Get All**

When a Developer modifies a Global Function and checks it back into the Master CSD, the modified code is not available to other Developers unless they either check out the Global Function or execute the **Get All** operation from each tab (General, Global Function and Message Processor) of the Group Development dialog.



# 6. Records

---

## 6.1. Record Characteristics

All records created on a host screen have a default characteristic of “Tied to Current Host Screen”. If you decide to change the scope of a record from the Record Characteristics option of the Host menu, you are prompted as follows:

*“If the record is changed to a different type, you cannot go back to “Tied to the Current Host Screen”. Do you want to change the type?”* **Yes, No**

If you select **Yes**, the characteristics change. When you check the characteristics for that record, the “Tied to Current Host Screen” option is disabled. If you select **No**, nothing changes.

Another message was added. When two developers try to change the characteristics of a record at the same time, one developer may change the characteristics of the record. The second developer gets a message, saying that another user has checked out the records. When the first developer has finished making changes, the second developer may then change the characteristics of the record.

## 6.2. Creating Containers or Records at the Same Time as Another Developer

When two developers try to create a Container and/or records at the same time, one developer may create the Container and/or records. The second developer gets a message saying that another user has checked out the Container and/or records. When the first developer has finished working with the Container and/or records, the second developer may then work on them.

Trying to Create a Container that Exists in the Master CSD

If you try to create a Container with the same name as an existing Container, the following message appears at compile time:

*“The Container is already in the master. Do you want to use the same name?”* **Yes, No**

If you select **Yes**, the code compiles. If you select **No**, the SmartCode window remains open. This gives you a chance to change the Container name.

## 6.3. Records (CB)

In the Group Development dialog window, the Records (CB) object was replaced with dashes. Developers are no longer able to check out records due to users not releasing them in time for other users to continue working.



# 7. FAQs and Recommendations

---

## 7.1. Frequently Asked Questions:

**Q: A message says I can't create containers when I have a screen checked out and try to compile the SmartCode from SmartCode Editor. Why?**

*A: Another user may have Containers (CB) checked out. To resolve this problem, follow these steps:*

- *Save your work in the SmartCode object.*
- *Go to the Group Development dialog box to see if someone has the Container (CB) object checked out.*
- *Ask the user to check the Container (CB) object back into the Master CSD.*
- *Go to the SmartCode object and compile the code you previously worked on, and continue working.*

**Q: After I rename a screen and look at the Group Development dialog box, the new name does not appear in the list. Why?**

*A: The screen with the old name is being used by others. The new name will not appear in the New Screens tab until the old screen is checked back in. Then the new screen name will appear in the Group Development dialog box and will be removed from the New Screens tab*

**Q: Why do I get a message saying that I cannot make a connection to the Master CSD?**

*A: There are 3 reasons why this message appears:*

1. *Someone may have the Master CSD open.*
2. *The Master CSD was moved to a different directory.*
3. *The drive mapping may have been changed.*

**Q: After I delete a screen from the Master CSD through the Group Development dialog box, I can still see it in my Local CSD. Why?**

*A: There is nothing wrong. You must delete the screen from your Local CSD. . If you have unintentionally deleted a screen from the Master CSD, you can put it back into the Master CSD again from your Local CSD. If you delete the screen from the Master CSD and the name disappears from the Group Development list, the screen has indeed been deleted from the Master CSD*

**Q: When I delete a screen from my Local CSD from the “Host/Delete Host Screen” option I still see the name of the screen in the Group Development dialog box. Why?**

*A: You have deleted the screen from your Local CSD, not from the Master CSD. To delete the screen from the Master CSD, delete it from the Group Development dialog box. Please remember that the Master CSD and the Local CSD are separate entities. The Group Development dialog box communicates the changes you want to make to the Master CSD from the Local CSD*

**Q: What do I do if I get an error in the Group Development dialog box?**

*A:*

- 1. Close Proprietary Software ChangedName.*
- 2. Launch Proprietary Software ChangedName.*
- 3. Open the Master CSD.*
- 4. Create a new Local CSD.*
- 5. Delete the old Local CSD.*

**Note:** *If you continue working with the old Local CSD, you may corrupt the Master CSD*

**Q: What do I do if I have to move my project?**

*A: If you must move the project files, follow the procedure below:*

- 1. Have all developers check in their objects.*
- 2. Copy the Master CSD, .csg and .log files to the new directory.*
- 3. All developers must create new Local CSDs from the new location of the Master CSD.*

***Q. I'm missing one of my files from the Master CSD. I can't find my .log file. What do I do?***

*A: If you are missing any files from your Master (this includes .csd, .csg or .log), it is important that all developers currently working with the Master are notified to stop development immediately, then contact Owner's Customer Care Department at 1-800-622-2684 extension 324.*

## 7.2. Recommendations for using ChangedName

**Recommendation:** *Don't work on the Master CSD, work on your Local CSD.*

**Explanation:** Never edit the Master CSD directly until the end of the project. Working directly on the Master CSD while others are checking objects in and out could corrupt the Master and cause lost work.

**Recommendation:** *Save often and back up files regularly*

**Explanation:** Back up all of your files, Local CSD, Master CSD and Master.csg, every day. If you don't have a regular backup system, copy these files to diskettes, always noting the paths where the files reside. As you work, save the files regularly to your hard disk.

To back up the Master CSD:

1. Open the Master CSD.
2. From the File menu select Save As.
3. Choose a name like Master\_backup.csd to remind you that this is a backup.
4. Select the Save button.
5. Make a copy of the Master.csg file and place it in the same directory as the Master\_backup.csd. Rename the Master.csg file to Master\_backup.csg.

To back up a Local CSD, follow steps 1 to 4, substituting Local CSD for Master CSD.

**Recommendation:** *Add Screen Name to Record Name*

**Explanation:** To eliminate potential problems, make a list showing which screens are associated with each record. Prefix all record names with the name of the screen followed by the name of the record. For example, if the screen is named Main Menu, record names could be:

```
r.MainMenu_command  
r.MainMenu_edit
```



**Note:** Remember that spaces are not allowed in record or field names (i.e., `r.name` or `f.name` respectively).

***Recommendation: Retain original path and configuration***

***Explanation:*** The Local CSDs are linked to the Master CSD in the location where the Master was originally created. Therefore keep the Master CSD in its original location until the project is done. If the Master is moved, its link to the Local CSD is lost. If this happens, you must create a new Local CSD



# 8. Tips & Troubleshooting

---

## 8.1. Missing files from the Master CSD

If you are missing any files from your Master (this includes .csd, .csg or .log), it is important that all developers currently working with the Master are notified to stop development immediately, then contact Owner's Customer Care Department at 1-800-622-2684 extension 324.

If the original Master CSD, CSG or LOG file is lost/deleted, and a new Master CSD is created from a Local CSD, developers should be aware that all previous lock unit information will be erased and the existing local CSD's will not work with the new Master. Developers should re-create their own local CSD's from the new Master CSD

## 8.2. Unable to Add Screens to the Master CSD

If your application contains two screens with the same screen recognition, you will not be able to check-in/add the second screen to the master CSD. Proprietary Software prevents two screens with the same screen recognition attributes from appearing in the same CSD. This behavior applies even if the screens are named differently.

### *Solution*

Since Proprietary Software was designed to prevent two screens with the same screen recognition from co-existing in the same CSD, you will need to alter either one of the screen recognition attributes.

1. Check-out the screen that has the same screen recognition as the newly-captured screen;
2. Modify the screen recognition on the checked-out screen, ensuring that the new screen recognition is unique and different from the newly-captured screen;
3. Check-in the modified screen.

You should now be able to add the newly captured screen to the master CSD.

## 8.3. Keyboard Mapping

When you do not have Keyboard Mapping checked out, the Reset and Reset All buttons are enabled. If you select them, the mappings may appear to be reset, but they are not! When you refresh the window, you'll see that the settings are unchanged.

## **8.4. Renaming/Deleting Screens From Master CSD Allowed**

When a Developer attempts to rename or delete a screen from the Master CSD while another Developer has that screen checked out, a confirmation box will appear. If the Developer answers, **Yes**, the screen will be renamed or deleted in the Master CSD.

The Developer who renames or deletes the screen should inform other Developers of this action.

## **8.5. Screen Names**

If someone tries to assign a screen name that is already used by a screen in the Master CSD an alert message will appear.

## **8.6. Keeping the Master CSD Current**

After a development task is done, check the latest version of all related objects in all Local CSDs back into the Master CSD. This ensures that the Master CSD is always up-to-date.

# 9. Index

---

<b>"*" Prefix</b>		
Global Functions .....	31	
Message Handler .....	26	
<b>"." Prefix</b>		
Global Functions .....	31	
Message Handler .....	26	
<b>"?" Prefix</b>		
Global Functions .....	31	
Message Handler .....	26	
<b>Adding</b>		
objects .....	22	
screens to the Master CSD .....	11, 22	
<b>Checking In</b>		
Global Functions .....	31	
Message Processors .....	27	
objects .....	20	
<b>Checking Out</b>		
Global Functions .....	32	
Message Processors .....	27	
objects .....	19	
<b>Completed Application</b>		
distribution .....	24	
<b>Connection Options .....</b>	<b>13</b>	
<b>Containers</b>		
creating at same time as another developer .....	35	
<b>Creating</b>		
Containers .....	35	
Global Functions .....	29	
Local CSD .....	17	
Master CSD .....	15	
Message Processors .....	25	
Project File .....	12	
Records .....	35	
<b>Deleting</b>		
a screen from the Local CSD .....	23	
a screen from the Master CSD .....	23	
Global Functions .....	32	
Message Processors .....	28	
<b>Developing</b>		
Local CSD .....	18	
<b>Development Group Dialog .....</b>	<b>8</b>	
General Tab .....	9	
Global Function Tab .....	30	
Message Processor Tab .....	26	
New Screens Tab .....	11	
<b>Distributing</b>		
completed application .....	24	
<b>FAQs .....</b>	<b>37</b>	
moving projects .....	38	
<b>Global Functions .....</b>	<b>29</b>	
"*" Prefix .....	31	
"." .....	31	
"?" .....	31	
checking in .....	31	
checking out .....	32	
creating .....	29	
deleting .....	32	
prefixes .....	31	
recreating in Master CSD .....	32	
viewing .....	33	
<b>Group Dialog .....</b>	<b>See Development Group Dialog</b>	
<b>Hardware requirements .....</b>	<b>3</b>	
<b>Installation .....</b>	<b>3</b>	
software .....	4	
<b>Introduction .....</b>	<b>1</b>	
<b>Local CSD</b>		
creating .....	17	
deleting a screen .....	23	
developing .....	18	
<b>Locking</b>		
component .....	5	
<b>Master CSD</b>		
adding screens .....	11, 22	
creating .....	15	
defined .....	5	
deleting a screen .....	23	
recreating Global Functions .....	32	
recreating Message Processors .....	28	
saving .....	15	
<b>MasterX.csg</b>		
defined .....	6	
<b>Message Processors .....</b>	<b>25</b>	
"*" .....	26	
"." .....	26	
"?" .....	26	
checking in .....	27	
checking out .....	27	
creating .....	25	

deleting .....	28	<b>Requirements</b>	
prefixes .....	26	hardware .....	3
recreating in Master CSD .....	28	host access .....	3
<b>Objects</b>		software.....	3
adding .....	22	system.....	3
check in.....	20	<b>Saving</b>	
check out.....	19	Master CSD.....	15
properties while checked out .....	20	<b>Setting up a Development Project.....</b>	<b>5</b>
<b>Project File</b>		<b>Software</b>	
creating .....	12	installation .....	4, 12
<b>Recommendations .....</b>	<b>37</b>	requirements .....	3
<b>Records .....</b>	<b>35</b>	<b>Tips .....</b>	<b>43</b>
characteristics .....	35	<b>Troubleshooting .....</b>	<b>43</b>
creating at same time as another		<b>Viewing</b>	
developer .....	35	Global Functions .....	33
Records (CB) .....	35		



