

Product Authorization Codes Machine ID : 453305

PRODUCT SESSION AUTHORIZATION

SITE ID : 453305

Execute

```

Accept_Tokens (Product => [STRING-expression],
               Donation => [POSITIVE-expression],
               Resulting_Count => [POSITIVE-expression],
               Code => "",
               Authorization => "");

```

using the following values:

Product	Donation	Resulting_Count	Code	Authorization
X Interface	8	8	113FFD650F9DD931	1C6F16CE88
Full Session	8	8	7D2DC226ABA29DE	1FCDE53B88

PRODUCT AUTHORIZATION

Execute

```

"!Implementation".Product_Authorization.Register
(Product_Name => ">>Product Name<<",
 Authorization_Code => "",
 Expiration_Date => "");

```

using the following values:

Product_Name	Authorization_Code	Expiration_Date
Cmvc	7414361F81A5C3A	
Cmvc.Source_Control	16ADFB91718C0F27	
Ftp	17FD5B97339584B	
Rpc	27D02E52CF25C15	
Tcp/Ip	10D7CC642DB010F2	
Telnet	10C09E370B2B1B07	
Work_Orders	14B7CEB91C3F5656	
Lrm_Interface	2AA27A3F2323A29	

SO => S010148-9
ORIGINATOR => RANDY
COMMENT => TERMA ELEKTRONIK AS

Product Authorization Codes

Machine ID : 453305

PRODUCT SESSION AUTHORIZATION

SITE ID : 453305

Execute

```
Accept_Tokens (Product => [STRING-expression],
               Duration => [POSITIVE-expression],
               Resulting_Count => [POSITIVE-expression],
               Code => "",
               Authorization => "");
```

using the following values:

Product	Duration	Resulting_Count	Code	Authorization
Mc68020_Os2000	8	8	76A9898C4738A34	59CFB885188

PRODUCT AUTHORIZATION

Execute

```
"!Implementation".Product_Authorization.Register
(Product_Name => ">>Product Name<<",
 Authorization_Code => "",
 Expiration_Date => "");
```

using the following values:

Product_Name	Authorization_Code	Expiration_Date
Motorola_68k	13950C7F2D002399	
Mc68020_Os2000	15B605F5763108BA	

SO => SO10148-9

ORIGINATOR => RANDY

COMMENT => TERMA ELEKTRONIK AS

Software Installation Checklist

All products listed below have been installed at Rational prior to shipment (unless otherwise noted). Some products require steps that are not done at Rational (for example, downloading software to a workstation, optional steps such as installing an example project). Please take the time to review all of the documents included in this package. Steps done at Rational will be checked off in the installation note. Complete each product installation by executing the steps that have not been checked off in the product installation note.

Date : 8/20/92
 Sales Order : 5010198-9
 Machine Id : 453305
 Site ID : 453305
 Customer : TERMA ELEKTRONIK AS

Product #	Product Name	Sessions	Release #	Notes
3000-00009	ENV-400EXP	S	D-12.6-5	INSTALLED
3000-00373	MC68020-OS2000	S		"

Environment Disk Usage

Volume	Used (kb)	DFS (kb)	Total (kb)
1	165472	30,000	
2	25948	0	
3	60261	0	
4	19618	0	

SMSE Checklist SERIES_400

PRODUCT REQUIREMENTS

<input checked="" type="checkbox"/> <u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input checked="" type="checkbox"/> 508-003234-004	Exabyte Operations & Ordering Information
<input checked="" type="checkbox"/> 506-003274-005	SMSE Checklist Disk_400
<input checked="" type="checkbox"/>	<u>TAPE MEDIA</u>
<input checked="" type="checkbox"/> 541-002873-001	8mm
<input checked="" type="checkbox"/> 670-003271-001	Environment Bundled SW Backup Tape
<input checked="" type="checkbox"/> 680-003209-017	Environment D_12_6_5 DFS Tape
<input checked="" type="checkbox"/> 505-003249-001	Support Activity Report (SAR)

SMSE Checklist Environment_400EXP

SESSION AUTHORIZATION

<input checked="" type="checkbox"/> <u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input checked="" type="checkbox"/> 509-003267-001	Environment_400EXP Authorization Codes _____ Session Authorizations
<input checked="" type="checkbox"/> 505-003249-001	Support Activity Report (SAR)

PRODUCT REQUIREMENTS

<input checked="" type="checkbox"/> <u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input checked="" type="checkbox"/> 508-003207-006	ENVIRONMENT D_12_6_5 Release Note
<input checked="" type="checkbox"/> 507-003219-010	X_Interface Release10_9_0 Installation Procedure
<input checked="" type="checkbox"/> 508-003219-005	X_Interface Release10_9_0 Release Note
<input checked="" type="checkbox"/> 507-003275-002	RWI Release10_1_1 Installation Procedure
<input checked="" type="checkbox"/> 508-003275-003	RWI Release10_1_1 Release Note
	<u>PC MEDIA</u>
<input checked="" type="checkbox"/> 541-003258-001	3.5 Diskette
<input checked="" type="checkbox"/> 541-003282-001	5.25 Diskette
<input checked="" type="checkbox"/> 690-003275-005	RWI Release10_1_1D Diskette
<input checked="" type="checkbox"/> 507-003265-002	TRAINING 1_1_0 Installation Procedures
<input type="checkbox"/>	<u>INSTALLATION ACTIVITY</u>
<input type="checkbox"/>	Full Installation in Field Required.
<input type="checkbox"/>	Partial installation at Factory performed, Field installation of unchecked steps required, starting with step _____ of <i>Installation Procedure for Environment_400EXP</i> .
<input type="checkbox"/>	Complete installation performed at Factory.

- OVER -

FIELD INSTALLATION REQUIREMENTS

<u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input type="checkbox"/>	<u>TAPE MEDIA</u>
<input type="checkbox"/> 541-002873-001	8mm
<input type="checkbox"/> 670-003207-015	Environment Backup Tape
<input type="checkbox"/> 680-003264-002	DFS_400 Backup Tape

Exabyte Operations and Ordering Information

1. User/Operator Maintenance

In new system shipments or upgrades containing an Exabyte tape drive, Rational includes a blank 8mm tape cartridge and a 3-pass, approved cleaning kit.

The Exabyte unit requires low maintenance, however, it is dependent on the proper care and handling of the transport and magnetic tape. The cleaning procedures outlined in the Exabyte Cleaning Kit are brief and require only minutes of the operator's time, but cleaning must be done as explained in order to achieve continued reliability and low maintenance.

The User/Operator of the Exabyte tape drive is responsible for performing a periodic tape head/path cleaning. The cleaning of the tape head/path is accomplished with the Exabyte Cleaning Cartridge Kit. The cleaning kit contains the instructions and all items required to perform the tape head/path cleaning operation. The kit which Rational has included in your shipment is good for 3 passes. After the 3rd cleaning, the tape should be discarded. The recommended frequency for the cleaning is after 30 Gigabytes of data transfer or monthly, whichever occurs first.

Reliability

To maximize reliability, please observe the points outlined in this section. Please note that failure to follow these recommendations or requirements, may result in voiding manufacturing and/or support warranties.

- Use only Exabyte's brand 8mm tape cartridge or the Sony, video 8, metal particle, data grade tape cartridge. (recommendation)
- Maintain proper environmental conditions for the tape drive and the media. The drive or media should not be operated or stored in areas where there is excessive heat or cold or in areas where temperatures may change at a rate of more than 10% (Fahrenheit) per hour. Also, the relative humidity should be between 20% and 80% non-condensing. (recommendation)
- Use only approved cleaning tapes. (Exabyte part no. 727113 or 727114) Unapproved video store cleaning tapes may damage the heads or transport. (requirement)
- Avoid using faulty or damaged cartridges/media. (requirement)

Ordering Information

Orders for Exabyte-brand tapes and approved cleaning kits may be placed by calling Exabyte's sales outlet at 1-800-767-8273. (For outside U.S., 1-303-442-4333 or 1-303-447-7613)

Part Number	Description
180179	Exatape, 15 Meter
180180	Exatape, 54 Meter
180181	Exatape, 112 Meter
727113	Exabyte Cleaning Tape, 3-pass
727114	Exabyte Cleaning Tape, 12-pass


SMSE Checklist MEMORY_32_MB

FIELD INSTALLATION REQUIREMENTS

<u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input type="checkbox"/> 504-003207-007	SMSE Checklist for Environment Release 12_1_1 <input checked="" type="checkbox"/> Not Required with D_10_20_2 or later
<input type="checkbox"/> 507-003237-004	MEMORY_32_MB Installation Procedure
<input type="checkbox"/> 547-003230-001	2 Gold Wipes <input checked="" type="checkbox"/> Not Required
<input type="checkbox"/>	2 backplane jumpers <input checked="" type="checkbox"/> Not Required
<input type="checkbox"/> 505-003249-001	Support Activity Report (SAR)

SMSE Checklist DISK_400

PRODUCT REQUIREMENTS

 <u>PART NUMBER</u>	<u>DESCRIPTION</u>
507-003274-004	Installation Procedure for Series 400S

SERIES 400 SYSTEM

Installation Procedure

-
- System Installation
 - Disk Upgrade
-

RATIONAL

Copyright © 1991 by Rational

PN 507-003274-004

This document is subject to change without notice.

Rational and R1000 are registered trademarks and Rational Environment is a trademark of Rational.

Rational, 3320 Scott Boulevard, Santa Clara, California 95054-3197

Table Of Contents

1 OVERVIEW	1
2 PRE-INSTALLATION	3
Prerequisites For Installation	3
3 INSTALLING THE SYSTEM	5
Series 400S System Configuration	5
Unpacking and Inspection	7
Final Installation Checklist and Precautions	8
Checking for Proper Routing and Connection of Cables	9
Preparing to Apply System Power	11
Power Checkout	13
Installing the Operator Console	15
4 SYSTEM TESTING AND OPERATIONS	17
System Checkout Procedures	17
Modem and Networking Connections	18
Configuring the System for Remote Diagnostics (optional)	18
System Peripheral Exercisers	18
Installing the Environment Terminal (optional)	21
Configuring the Rational Environment	22
Overall System Operation	23
5 OTHER SUPPORT	25
Appendix A R1000 SERIES 400S SPECIFICATIONS	27
Appendix B RATIONAL INSTALLATION CHECKLIST	29
Appendix C UPGRADE 1.2 GB DISK DRIVE	35
Appendix C Preparation	35
Appendix C Installation	35

1

OVERVIEW

These installation instructions describe the installation procedures for a Series 400 System as shipped from the factory. In addition, an appendix to these instructions describes the installation procedure to follow when upgrading a previously installed Series 400 System with additional disk drives.

2

PRE-INSTALLATION

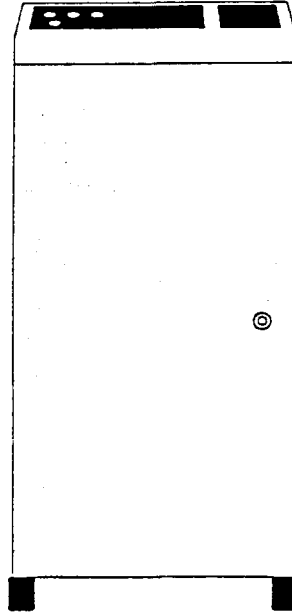


Figure 1.1 - Series 400 System

Prerequisites For Installation

Pre-installation planning is covered in the Rational "Site Planning Guide for the R1000 Series 400 Development System."

Pre-installation planning must be completed and an installation plan established prior to the system being delivered to the installation site. Pre-installation planning covers the following areas:

- Planning responsibilities and scheduling
- Site selection
- R1000 components with physical specifications and service clearances
- Environmental and electrical requirements

- Data communications line requirements

- Delivery preparation

The Implementation Plan should be completed at least *30 days* before the R1000 system delivery. Refer to the section "Implementation" in the *Field Operations Handbook* for a sample of information to be included in this implementation plan.

The implementation plan should be distributed to the following Rational representatives involved in planning, installation, project guidance, training, and post-installation support of the R1000 system:

- Field Technical Representative or Response Center support personnel responsible for the software and hardware installation, checkout, and initial training of the customer's System Manager and operations personnel.

- Marketing Representative responsible for assisting with coordination of system configuration, delivery dates and other sales-related issues.

- Field Technical Representative or home office Technical Consultant responsible for providing product training, special technical consulting in the areas of software engineering, the customer's program applications, and other specialized training to meet the customer's need.

3

INSTALLING THE SYSTEM

Having completed all of the necessary prerequisites, the Series 400S system is now ready to be installed. Refer to the diagrams as you perform the installation steps.

Table 1.1 below summarizes the activities that must be performed for a successful installation of the series 400. The activities that contain * are covered in much greater detail in the following sections.

Table 1.1 - Rational Site Preparation and Installation

Timetable	Installation Activity
Day 1	Unpack, take inventory, and perform an inspection for visible damage that may have occurred during shipment* Position the 400S cabinet over floor tiles* Install the cabinet's stabilizing brackets and lock wheels in place Install the disk drives as specified on the disk housing cover Connect the necessary terminal and network cables Perform pre-power checkout at customer power receptacle* Perform power-on and off-line checks* Execute first level diagnostics Boot system and ensure proper version of Environment software has been installed. Ensure that all Rational software products ordered, are installed. Begin overnight run of system diagnostics
Day 2	Analyze the overnight run of system testing Perform remote diagnostic link check to the Response Center (if required) Begin start-up training for customer's System Manager and operations personnel—1/2 day. (The System Manager receives two additional days of training, following three days of Rational Product training) Turn system over to customer and submit the completed copy of the support activity report to the Rational Customer Support Response Center

As you are installing the R1000, you should complete the installation checklist in the appendix on page 29, **Rational Installation Checklist** of this guide.

Series 400S System Configuration

As shown in Figure 1.1, a Series 400S consists of a single cabinet.

The 400S cabinet contains (See Figure 1.2 and Figure 1.3):

- R1000 Processor and Memory Boards
- RESHA I/O Controller board
- DC Power Supply
- System Control Panel
- 8mm Tape Drive
- Disk Housing containing 1 (std) to 4 (opt) Disk Drives
- Console and Single RS232 Comm Port
- AC Power Distribution Unit

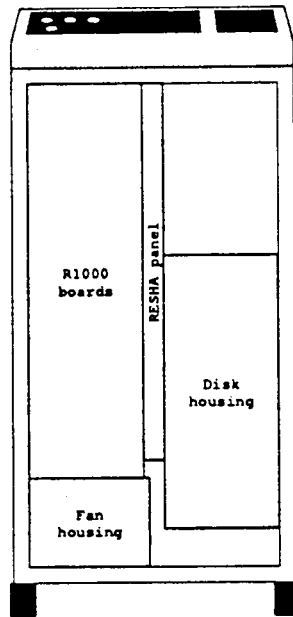


Figure 1.2 - Series 400S, Front View

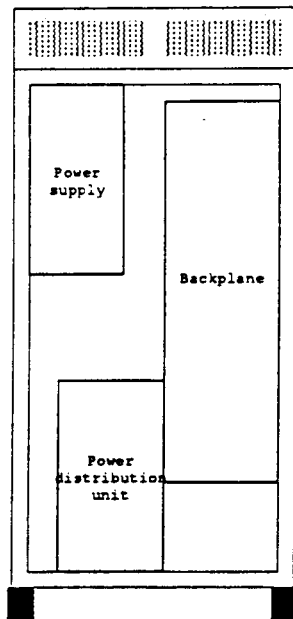


Figure 1.3 - Series 400S, Rear View

Unpacking and Inspection

Upon arrival of the R1000 system at the customer's site, the customer is responsible for moving the system from the shipping area to the installation area. The Rational representative is responsible for installing the system and should perform the following steps:

- 1. Remove all protective-packing material.
- 2. Check the shipment against the packaging list.
- 3. Check all boxes and system components for external damages.
- 4. Record discrepancies observed while checking the shipment and report them to the customer and to Production Control at Rational's home office. Production Control will help coordinate the effort to resolve discrepancies.
- 5. Document in the installation report observations relating to damaged items. The customer should be notified and Production Control contacted for instructions on how to handle a damaged item.

Final Installation Checklist and Precautions

Check through the following list of final installation preparations and precautions:

- 1. Make complete visual checks of cables, harnesses, backplane, foreplane, fans, filter pads, and all cable connectors for crushed wires, cut wires, bent or smashed pins, or any other abnormality.
- 2. Make sure you have the minimum tools required for installing the Series 400S system. These tools are contained in the basic tool kit which is issued to Rational support personnel.
- 3. If there is no raised floor, cables routed externally to the system's cabinet must be protected from the walking path of personnel. To protect these cables, the customer should use cable through or route them in overhead channels.
- 4. Make sure you have the stabilizing brackets that are required to be installed on the cabinet's wheels. The brackets must be installed to satisfy the anti-tip requirement specified by the Underwriters Laboratory. The brackets will be included with each system shipment.
- 5. Make sure that the disk drives are installed as specified on the back of the disk housing cover. Unlike previous R1000 systems, the disk drives in the series 400 are SLOT-DEPENDENT, i.e., the drive for unit 0 must be in the top slot of the disk housing, unit 1 must be in the 2nd slot from the top, etc.

Prior to shipment, the disk drive configuration is recorded on the back of the disk housing cover, using the serial number of each drive. The serial number of each drive appears on the front and back of the drive.

Check that the serial numbers specified on the list on the back of the disk housing cover match the serial numbers of the drives delivered with the system

and install them in the order specified on the list. Report any discrepancies to Production Control at the home office before proceeding.

Checking for Proper Routing and Connection of Cables

- 1. Position the 400S system cabinet in the computer area according to the layout plan established during the pre-site planning stage. (Refer to Figure 1.4.)

Note: Once the cabinet is in place, be sure to install the stabilizing brackets over each wheel as specified in the diagram that is included with the bracket package. The brackets are shipped in the crate with the system.

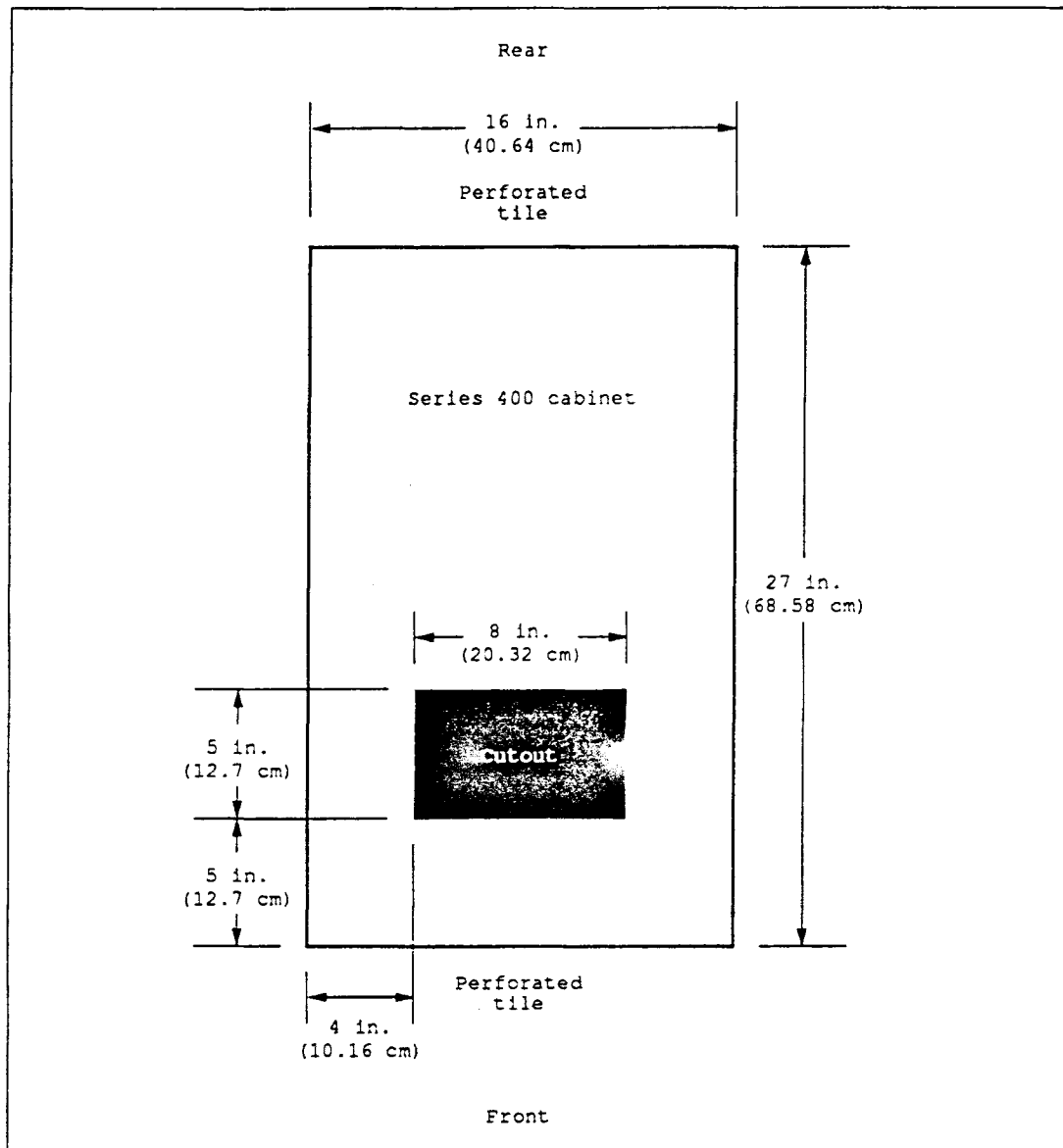


Figure 1.4 - Floor Tile Requirements for R1000 Series 400S Installation (Top View)

- 2. From the front of the 400S cabinet, ensure that the following cables are connected at the RESHA board face plate: Refer to Figure 1.5 for connector location.

- Cartridge Tape Drive
- Control Panel
- Disk Drives

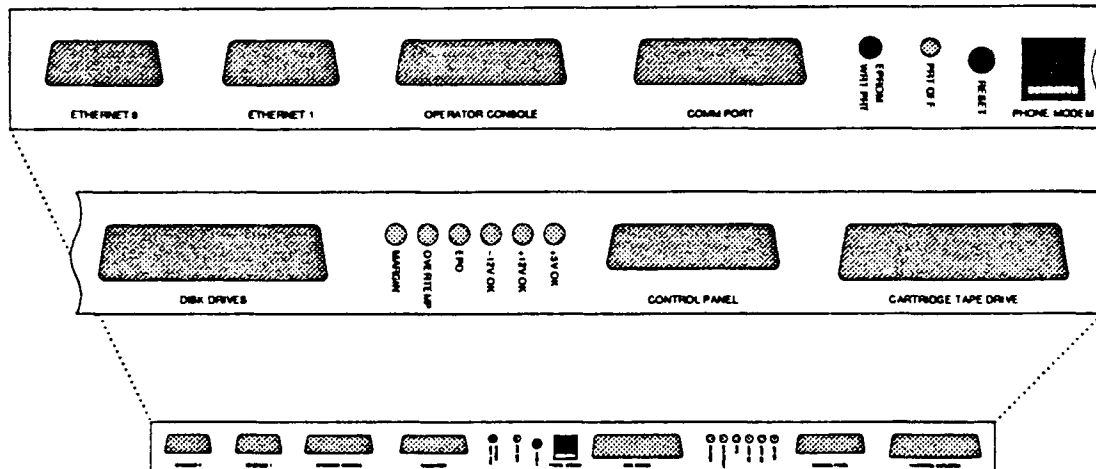


Figure 1.5 - RESHA Board Face Plate

- 3. From the rear of the 400S cabinet, check for cable connections on the DC power supply and card cage backplane:
- power cable (part no. 125-003168) on the end of power supply to PS0 power jack on the Power Distribution Unit (PDU)
 - P.S. Control cable (part no. 125-003180) from connector J9 on the upper left corner of the card cage backplane to the two molex connectors located underneath the left side of the power supply.
 - cable (part no. 125-003472) from terminal strip located on the power supply to the card cage backplane connector (J12) labeled +/- 12V IN.
 - cable (part no. 125-003473) from the molex connector on the rear of the disk cage to the card cage backplane connector (J13) labeled Disk Power.
 - cable from the card cage backplane connector labeled Exabyte Power (part no. 125-003175) and routed over the top of the card cage toward the front where the Exabyte Tape Drive is installed.
 - cable from the card cage backplane connector labeled +12 Fans (part no. 125-003174) and routed down the edge of the card cage and underneath to the fan cage located in the front of the 400S cabinet, underneath the cardcage.

Preparing to Apply System Power

Perform the following steps prior to applying AC power to the system:

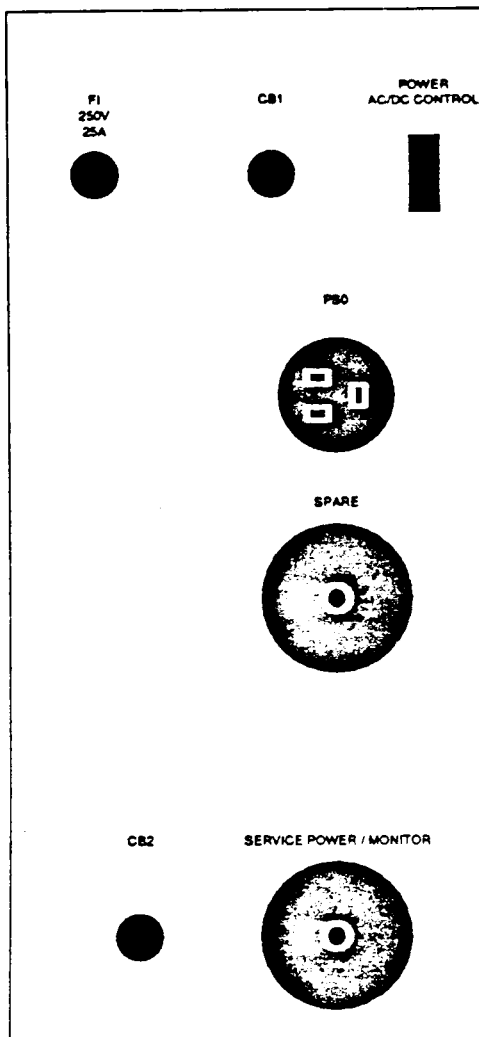


Figure 1.7 - Series 400S, Power Distribution Unit

Power Checkout

Check the manufacturer's name plate for the power requirements of the system being installed (Located at the bottom, right-hand corner of the rear door). Verify the requirements before proceeding with the following steps, referring to Figure 1.7:

- ❑ 1. Ensure that the circuit breakers and power key switch are in the *OFF* or *Standby* position for the following:
 - The Power Distribution Unit—PDU (lower rear of cabinet)
 - The main system power (key-switch on System Control Panel)
- ❑ 2. Verify the customer's power source circuit breakers are *ON*. Using a voltmeter set to AC volts, measure the AC power at the receptacle. The voltage measured

between the two hot leads should measure between a minimum value of $208 \pm 5\%$ VAC to a maximum of $240 \pm 5\%$. If these voltages are not correct, do not proceed with the installation until the facility's electrician has corrected the problem.

- 3. Connect the power cord to the mating connector originating from the customer's power source.
- 4. Turn on circuit-breaker CB2. Using a voltmeter, check the AC power for the proper input voltage. The voltage should be between a minimum value of $208 \pm 5\%$ VAC to a maximum value of $240 \pm 5\%$. Take this measurement at the power jack labeled "Service Power/Monitor" located near the lower, right-hand edge of the PDU. (Refer to Figure 1.8 for pin assignment).

Note: In Europe there is 1 phase, 1 neutral, and one ground pin.

The black cap unscrews from the power jack, providing access to the pins (see Figure

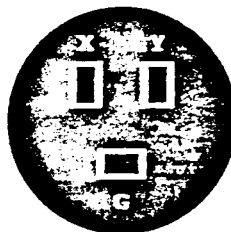


Figure 1.8 - Series 400S, PDU Plug Pinout

- 5. Turn on circuit-breaker CB1. Switch circuit-breaker CB2 to *OFF*. (CB2 should remain off during normal system operation).
- 6. Switch the DISK WRITE keyswitch on the System Control Panel to the "PROTECT" position.
- 7. Switch the OPERATOR MODE keyswitch on the System Control Panel to the "AUTO" position.
- 8. Switch the POWER keyswitch on the System Control Panel to the "ON" position—this turns on the AC power. All fans should start blowing. This also allows the DC power supplies to apply DC voltage to the CPU, controller boards and disk drives.
- 9. Switch the DISK WRITE keyswitch on the System Control Panel to the "ENABLED" position. Allow drives to run for a period of time to adjust to the room temperature.

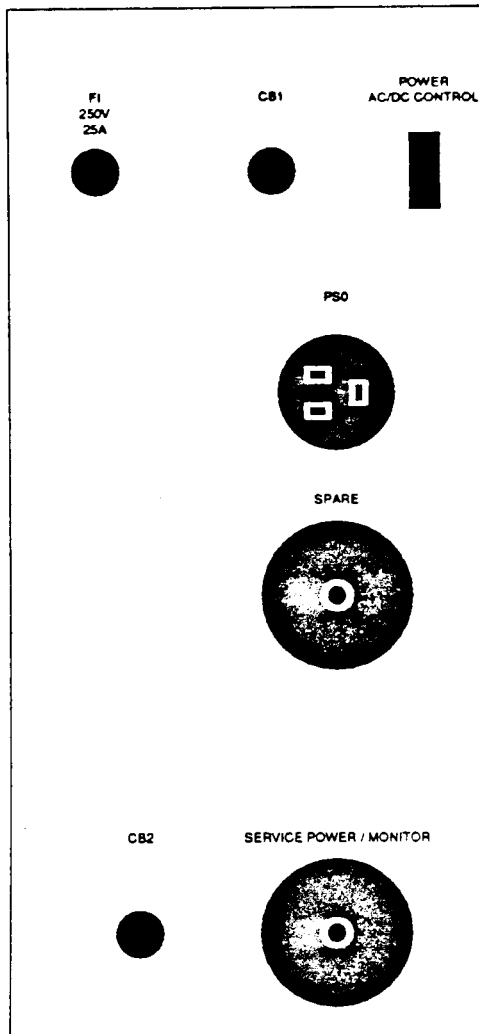


Figure 1.7 - Series 400S, Power Distribution Unit

Power Checkout

Check the manufacturer's name plate for the power requirements of the system being installed (Located at the bottom, right-hand corner of the rear door). Verify the requirements before proceeding with the following steps, referring to Figure 1.7:

- 1. Ensure that the circuit breakers and power key switch are in the *OFF* or *Standby* position for the following:
 - The Power Distribution Unit—PDU (lower rear of cabinet)
 - The main system power (key-switch on System Control Panel)
- 2. Verify the customer's power source circuit breakers are *ON*. Using a voltmeter set to AC volts, measure the AC power at the receptacle. The voltage measured

between the two hot leads should measure between a minimum value of 208 $\pm 5\%$ VAC to a maximum of 240 $\pm 5\%$. If these voltages are not correct, do not proceed with the installation until the facility's electrician has corrected the problem.

- 3. Connect the power cord to the mating connector originating from the customer's power source.
- 4. Turn on circuit-breaker CB2. Using a voltmeter, check the AC power for the proper input voltage. The voltage should be between a minimum value of 208 $\pm 5\%$ VAC to a maximum value of 240 $\pm 5\%$. Take this measurement at the power jack labeled "Service Power/Monitor" located near the lower, right-hand edge of the PDU. (Refer to Figure 1.8 for pin assignment).

Note: In Europe there is 1 phase, 1 neutral, and one ground pin.

The black cap unscrews from the power jack, providing access to the pins (see Figure

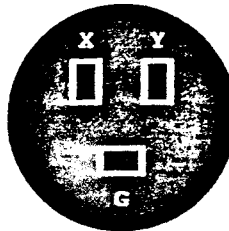


Figure 1.8 - Series 400S, PDU Plug Pinout

- 5. Turn on circuit-breaker CB1. Switch circuit-breaker CB2 to *OFF*. (CB2 should remain off during normal system operation).
- 6. Switch the DISK WRITE keyswitch on the System Control Panel to the "PROTECT" position.
- 7. Switch the OPERATOR MODE keyswitch on the System Control Panel to the "AUTO" position.
- 8. Switch the POWER keyswitch on the System Control Panel to the "ON" position—this turns on the AC power. All fans should start blowing. This also allows the DC power supplies to apply DC voltage to the CPU, controller boards and disk drives.
- 9. Switch the DISK WRITE keyswitch on the System Control Panel to the "ENABLED" position. Allow drives to run for a period of time to adjust to the room temperature.

- ❑ 10. Measure DC voltage for the DC power supply (located above the PDU) for the following values: (the measurements may be taken at the top side of the fuses located near the outer edge of the backplane)
 - a. +5 VDC
 - b. +12 VDC
 - c. -12 VDC
- ❑ 11. Verify that all fans are operational. (1 on the disk housing, 2 on the cardcage)

Installing the Operator Console

Note: At least one terminal is required to boot the Series 400S. The terminal must be an ANSI standard VT100 console and links to the Operator Console port (Refer to Figure 1.5). As Rational does not supply Rational terminals with the Series 400S, the customer should make sure an ANSI standard terminal is available at installation time.

- ❑ 1. Place the operator console in the location designated on the installation floor plan. Be sure there is a small cutout in the tile near the terminal. This cutout is for the power cord and the RS232 data cable for this terminal. The terminal requires 3-prong, 115 VAC power receptacles (domestic).
- ❑ 2. Install the operator console cable in the connector marked "Operator Console". The connector is located on the RESHA face plate. (Refer to Figure 1.5). This cable should be routed down the outside edge of the disk housing, out the bottom of the cabinet, and fed through the raised-floor tile that has been cut out for terminal and network cables. Route the cable as directly as possible to where the operator console is to be located. Feed the end of this cable back up through the tile cutout nearest the operator console and connect it to the terminal.
- ❑ 3. Set up the console's operating parameters:
 - Transmit Baud Rate: 9600
 - Receive Baud Rate : 9600
 - Parity : SPACE
 - Check Parity : OFF
 - Duplex : FULL
 - DTR Handshake : ON

After the installation of the Series 400S the operator console will be used to allow the system manager to communicate with the layers of software that support the Rational Environment™, the principal software component of the Series 400S.

4

SYSTEM TESTING AND OPERATIONS

Testing the system should begin with running the individual FRU tests and the functional diagnostics which are executed at the DFS level. Once this has been completed, the system may be booted and additional tests will be executed at the system's EEDB level.

System Checkout Procedures

Initiate the standard boot process. If the Operator Mode keyswitch is in the AUTO position, you will be asked if you want to boot the [STANDARD] configuration. Answer this question by depressing the "control" and "c" keys simultaneously. This will take you to the CLI> prompt.

At the CLI> prompt enter **x FRU** to get to the FRU Main Menu. At this point, select option 3 (Execute diagnostics) and answer the rest of the questions to execute the phase-2 tests for each board type:

- 1. IOC
- 2. VAL
- 3. TYP
- 4. SEQ
- 5. FIU
- 6. MEM 0
- 7. MEM 2 (if present)

Run the FRU confidence test (microdiagnostic) after the phase-2 tests if the phase-2 tests do not catch anything. Note that this can now be run from the FRU main menu (option 2).

Modem and Networking Connections

You may need to make final hardware connections in the following areas:

- 1. For customers with standard support, connect the customer-supplied modem to the "Comm Port"---if the port is currently being used to access the Environment, disconnect the Environment terminal and reconnect it to the "Operator Console" port (see the section "Installing the Operator Console").
- 2. If the customer has already installed networking, connect the public Ethernet cable to the R1000 at the Ethernet 0 receptacle located as the second connector from the bottom of the RESHA board face plate. (Refer to Figures 1.2 and 1.5)

Configuring the System for Remote Diagnostics (optional)

The proper use of the Comm Port is determined by the INITIOA program in the DFS. This program establishes the remote diagnostics connection to be used in case such a connection is allowed and needed. A remote diagnostic connection can be made via the modem on the RESHA board. If the system is configured to use the Modem on the RESHA board then the Comm Port port is automatically configured to the Environment port 16.

X INITIOA is used to set the Cluster Id for the machine and as the data needed for remote diagnostics. These instructions assume that the RESHA board modem will be used and the Comm Port will be used as an Environment port.

Perform the following steps:

- 1. Contact the Rational Response Center and inform them that you plan to test remote diagnostics. Verify with them that the diagnostic phone number is: 1-800-841-6400. This number has already been set in the novram of the IOC board and the number is only valid for UNITED STATES and CANADA.
- 2. Run `x INITIOA` from the `CLI>` prompt.
- 3. Verify the Cluster Id and modify it if necessary. (check the manufacturer's tag at the bottom of the rear door)
- 4. Verify the type to be used for Remote diagnostics and correct it if necessary. If no remote diagnostic connections will be allowed, make sure that the type is set to "M" for modem. This will allow the Comm Port to be used as Environment port 16.
- 5. Verify the data needed for the remote diagnostics connection and correct it if necessary.

System Peripheral Exercisers

Run the following peripheral exercisers from the CLI> prompt:

□ 1. Tape Exerciser

Enter **x tapex** at the CLI> prompt. After confirming the unit number of the tape drive to exercise, it will start running and continue to run until you explicitly stop it with a CTRL/G. Pressing any other key will give you the current status: how many bytes have been transferred, and the numbers of soft, hard, and data errors.

Let the program run for about 5 minutes or until 1 megabyte of data has been transferred. Check the status every minute or so. There should be no errors reported. Sample output:

```
CLI> x tapex
Exercise unit 0 [Y] ? y
DFS based tape exerciser, started at =>10:49:56 28-FEB-91
Type ^G to exit, any other character for status
[space bar]
Status at 10:50:08 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 23064      , soft => 0      , hard => 0      , data => 0
[space bar]
Status at 10:51:14 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 40388      , soft => 0      , hard => 0      , data => 0
[space bar]
Status at 10:52:12 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 270560     , soft => 0      , hard => 0      , data => 0
[space bar]
Status at 10:53:10 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 442468     , soft => 0      , hard => 0      , data => 0
[space bar]
Status at 10:54:10 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 648084     , soft => 0      , hard => 0      , data => 0
[space bar]
Status at 10:55:20 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 976868     , soft => 0      , hard => 0      , data => 0
[space bar]
Status at 10:56:14 28-FEB-91, test started at 10:49:56 28-FEB-91
Unit : 0, Bytes => 1253844    , soft => 0      , hard => 0      , data => 0
[CTRL/G] (takes several seconds to exit)
CLI>
```

□ 2. Ethernet Controller Exerciser

Enter **x netx** at the CLI> prompt. At the NETX> prompt, run all tests using the **test_all** command.

The program runs for about 5 minutes, and before it terminates, you will have to restart the machine. There should be no errors reported. Sample output:

```

CLI> x netx
NETX> test_all
  Testing RESHA STANDARD and SHORT ADDRESS registers
    Phase 1 - Short holding register wrap test
    Phase 2 - Standard holding register wrap test
  ADDRESS REGISTERS test ok
  Testing VME CONTROLS
    CONTROL test ok
  [Delaying for VME_SYSFAIL assertion]
  [Delaying for controller to complete self test diagnostics]
  Testing RAM
    Testing sliding 1
    Testing sliding 0
    Testing all 0
    Testing all 1
    Testing mapping (location := address)
      Initializing ram
      Verifying ram
  Memory tested successfully

CONTROL TESTS assert UNIBUS ACLO which reset the I/O controllers.
You must now crash the IOP with white button or
  BREAK KEY / option 3 / "res"
[press BREAK key]

Please enter
  0 => Restart system
  1 => Ignore break key
  2 => Redisplay recent console output
  3 => Enter debugger

Enter option : 3

Trapped into debugger
RD0 00000000 RD1 00002CA2 RD2 0000000C RD3 0000003B
RD4 00000013 RD5 00000002 RD6 00000001 RD7 00000008
RA0 0000FC40 RA1 00022105 RA2 0003FF64 RA3 00036141
RA4 00033AF0 RA5 000371F2 RA6 0003FF58 ISP 000CFAEC
PC DCFEFFFF USP DCFEFFFF ISP DCFEFFFF MSP DCFEFFFF SR 2100
VBR DCFEFFFF ICCR DCFEFFFF ICAR DCFEFFFF XSFC 7 XDPC 7
@res
R1000-400 IOC SELFTEST 1.0.0 ...
  512 KB memory ... [OK]
  Memory parity ... [OK]
  I/O bus control ... [OK]
  I/O bus map ... [OK]
  I/O bus map parity ... [OK]
  I/O bus transactions ... [OK]
  PIT ... [OK]
  Modem DUART channel ... [OK]
  Diagnostic DUART channel ... [OK]
  Clock / Calendar ... Warning: Calendar crystal out of spec! ... [OK]
Checking for RESHA board
  RESHA EEPROM Interface ... [OK]
Downloading RESHA EEPROM 0 - TEST
Downloading RESHA EEPROM 1 - LANCE
Downloading RESHA EEPROM 2 - DISK
Downloading RESHA EEPROM 3 - TAPE
  DIAGNOSTIC MODEM ... [OK]
  RESHA VME sub-tests ... [OK]
  LANCE chip Selftest ... [OK]
  RESHA DISK SCSI sub-tests ... [OK]
  RESHA TAPE SCSI sub-tests ... [OK]
  Local interrupts ... [OK]
  Illegal reference protection ... [OK]
  I/O bus parity ... [OK]
  I/O bus spurious interrupts ... [OK]
  Temperature sensors ... [OK]
  IOC diagnostic processor ... [OK]
  Power margining ... [OK]
  Clock margining ... [OK]
Selftest passed

```

Restarting R1000-4005 February 28th, 1991 at 11:51:24
 . . .

Boot the system and perform the Disk Exerciser procedure described below. This procedure is executed at the EEDB prompt from the operator console:

□ 1. Disk Exerciser (run for 5 minutes)

a. Elaborate the Disk Exerciser procedure:

```
EEDB: e disk_exerciser
      DISK_EXERCISER.x.x.x ...
```

b. Specify number of minutes to execute the test:

```
---->> Disk Exerciser <<----
For how many minutes do you want this test to run? 5
Test will end at : ....

---->> Disk Tester <<----

Test choices are:
TRIVIAL_READS_WRITES
.
.
The chosen test is run concurrently on all specified disks.
.
.
```

c. Execute the test:

```
Do you have a hardcopy terminal attached? [n] n
do you want to specify test options? [n] n

Unit 0 is mapped to volume 1, HDA number ..., size = ( .... )
Unit 1 is mapped to volume 2, HDA number ..., size = ( .... )
Unit 2 is mapped to volume 3, HDA number ..., size = ( .... )
.
.
Running test SEEK_PROFILE
End of pass 1, total I/Os = ...

---->> Disk Exerciser <<----
Disk Test Done
EEDB:
```

d. Unelaborate the Disk Exerciser procedure:

```
EEDB: u disk_exerciser
      Subsystem:
      Unelaborated DISK_EXERCISER.x.x.x
EEDB:
```

Installing the Environment Terminal (optional)

The operator console can be disconnected from the "Operator Console" port and reconfigured as an Environment terminal using the port marked "Comm Port" (Refer to Figure 1.5).

The Environment terminal can be used to access the Environment during installation, providing a better interface for configuring the 400S for the public network. The Environment terminal also simplifies the configuration of the port to allow dial-out capability for standard support.

Note: It is also possible to configure the system from the Operator Console. However, this interface has limited editing capabilities. Contact the Rational Response Center for further details about this alternate procedure.

To configure the Environment terminal:

- 1. Disconnect the operator console from the "Operator Console" port.
- 2. Install terminal cable to the connector marked "Comm Port" (Refer to Figure 1.5).
- 3. Install the power cords and apply power to the terminal.

Configuring the Rational Environment

Follow the procedures below to configure and check such areas as network addresses, machine names, and product authorization. (see the *System Manager's Guide for the R1000 Development System* for details).

- 1. Boot the [STANDARD] configuration.
- 2. Log into the R1000 using the Operator account. This can be done by placing a terminal on the Comm Port or by logging into the console via the Command Interpreter program.
- 3. Configure the system. From the !Users.Operator world, execute:

```
Do_Step ("Setup");
```

You will be asked to enter the machine name and an IP address. For example:

```
Enter Machine Name : Bluto  
Enter IP Address : 89.44.5.1
```

When this step has executed the machine should now be on the network, enabling you to log in via telnet.

- ❑ 4. If you are planning to configure the port marked "Comm Port," for standard support (optional) execute the `!Machine.Initialize_Terminals` command to set up the proper baud rate for the modem you plan to install (see the section "Configuring the Comm Port (optional)").
- ❑ 5. Run `Product_Authorization.Register` for all products to be registered. This will include the session limits. Purchased products and session limits are usually registered before the system is shipped. A list of Authorization Codes for purchased products and session limits should have been mailed to you. Verify that these have been registered, and register any products or session limits which have not already been registered. Call the Rational Response Center to verify or request the appropriate codes.

Verify the products that have been registered by using the `Product_Authorization.Show_Registration` command in the library `!Implementation` (as described above). Use `Show_Tokens` to display the number of `Full_Session` or `Fundamental_Session` tokens (login tokens) and product tokens (RDF, RXI, etc.).

- ❑ 6. Complete installation of products which could not be fully installed at the factory. (See the installation procedures, included with each product shipment, for more details.)
- ❑ 7. Shut the system down and reboot.

Note: Shutting the system down and rebooting by using the `Schedule_Shutdown` command takes a snapshot automatically. If you use the `[Break]` key make sure to take a snapshot before you take down the system— otherwise the changes above will not be saved.

Overall System Operation

Complete the following steps:

- ❑ 1. Perform system tailoring (for example, creating user accounts, access control groups, and so forth).
- ❑ 2. Check the login process.
- ❑ 3. Execute an overnight run of the `Disk_Exerciser` (so it runs through daemon cycle). Run this command from the EEDB.

5

OTHER SUPPORT

Complete the following steps:

- 1. The Technical Representative should review the operation of the Response Center with the customer.
- 2. The Technical Representative should provide an overview of system operations with the customer. For example, shutting the system down, booting the system, taking system backups, and handling emergencies.
- 3. The Technical Representative should review the completed "Rational Installation Checklist" with the customer.
- 4. The Technical Representative should complete a Support Activity Report (SAR) and distribute copies appropriately. Information provided should include such items as names and phone numbers of the System Manager and phone numbers for diagnostic lines, operations room, etc.

A

R1000 SERIES 400S SPECIFICATIONS

Physical Specification	Series 400S
CPU	Rational
Memory Capacity	32 or 64 Mbyte
Memory Transfer Bandwidth	80Mbyte/second
Logical Address Space	67 bits
Data Paths	Dual 64-bit ALUs
Disk Capacity	up to 4x1.2 Gbyte
Tape Drive, standard	8mm Cartridge
Height	38in(97cm)
Width	16in(40.6cm)
Depth	28in(71.1cm)
Weight	220 lbs. + 10 lbs./disk unit
Operating Temp	60 to 85 degrees F (15 to 29 degrees C)
Operating Humidity	20 to 80% non-condensing
Cooling	9,000 BTU's per hour (2000 watts)
Voltage	208 to 240 VAC +5%
Frequency	60 Hz (dom), 50 Hz (int)
Power Connection	3 wire
Amperage	12A
Max Service	30A
AC Power Cord	12awg 3 wire
Length	15 feet (4.57m)
Plug	NEMA L6-30P (domestic)
Receptacle	NEMA L6-30R (domestic)
Transceivers (Ethernet)	See Rational price list
Cables (tranceiver)	See Rational price list
Operator Console Connector	RS-232
Ethernet Connector	Ethernet
Telephone Jack	RJ45

B

RATIONAL INSTALLATION CHECKLIST

A Rational Technical Representative should check-off each item and show completed list to the customer.

Installation Address:

Customer Telephone _____

Customer Contact _____

Date (system arrival) _____

Date (start install) _____

R1000 ID Number: _____

Installation Task:

1. Unpacking and Inspection

- a. _____ Remove all protective-packing material.
- b. _____ Check the shipment against the packaging list.
- c. _____ Check all boxes and system components for external damages.
- d. _____ Record any discrepancies observed while checking the shipment and report them to the Response Center (document in the report any damaged items).
- e. _____ Document in the installation report any observations relating to damaged items, and notify the Response Center.

2. Final Installation Set-up Procedures

- a. _____ Position the system in the computer area according to the layout plan.
- b. _____ Make complete visual checks of cables, harnesses, backplane, foreplane, fans, filter pads and all cable connectors.
- c. _____ Make sure minimum tools required for installation are at the site.

- d. _____ Locate the cabinet stabilizing brackets.
- e. _____ Route cables externally to system's cabinet (if there is no raised floor).
- f. _____ Make sure disk drives are installed in the slots specified on the back of the disk housing cover.

3. Hardware Installation, Testing and Configuration

- a. _____ Install the cabinet stabilizing brackets over each wheel.
- b. _____ Check for proper routing and connection of cables.
- c. _____ Verify circuit breakers CB1 and CB2 are in the OFF position on the PDU.
- d. _____ Verify the power keyswitch is in the STANDBY position, disk write keyswitch is in PROTECTED and operator mode keyswitch is in AUTO position at the main system control panel.
- e. _____ Verify the RESHA board is seated properly.
- f. _____ Verify the R1000 processor and memory boards are seated properly.
- g. _____ Verify the disks are seated properly and match the configuration listed on the back of the disk housing cover.
- h. _____ Connect the AC power cord plug to the mating connector on the PDU.

4. Power Checkout.

- a. _____ Ensure the circuit breakers (located on the PDU) are in the "OFF" position and that the power keyswitch (on the system control panel) is in the "STANDBY" position.
- b. _____ Verify the customer's power source circuit breakers are ON.
- c. _____ Measure the AC power at the receptacle (using a voltmeter between the two hot leads). The voltage should measure between a minimum value of 208 VAC ($\pm 5\%$) to a maximum value of 240 VAC ($\pm 5\%$).

- d. _____ Connect the power cord to the mating connector originating from the customer's power source.
- e. _____ Place circuit breaker CB2 to the "ON" position.
- f. _____ Using a voltmeter, measure the voltage at the power jack labeled "Service Power Monitor" (located at the lower right-hand edge of the PDU). The voltage should measure between a minimum value of 208 VAC ($\pm 5\%$) to a maximum value of 240 VAC ($\pm 5\%$).
- g. _____ Place circuit breaker CB1 to the "ON" position and CB2 to the "OFF" position.
- h. _____ Switch the POWER keyswitch on the System Control Panel to the halfway (Power On) position.
- i. _____ Turn the Operator Mode Key switch to "INTERACT."
- j. _____ Check to see if any disk unit's FAULT indicator is lit on the status control panel.
- k. _____ Switch the Disk Write key switch to "ENABLED."
- l. _____ Measure DC voltage at the CPU backplane for +5 VDC ($\pm 5\%$), +12 VDC ($\pm 5\%$), and -12 VDC ($\pm 5\%$).
- m. _____ Verify all air filters are in place and clean.
- n. _____ Verify all fans are blowing and are operational.

5. Installing the Operator Console

- a. _____ Place the operator console in the location designated by the installation floor plan.
- b. _____ Verify that the operator console cable is installed in the connector marked "Operator Console" on the RESHA board face plate.
- c. _____ Set up the operator console's operating parameters.

6. System Checkout Procedures—Execute phase two FRU diagnostics for each board type:

- a. _____ IOC
- b. _____ VAL
- c. _____ TYP
- d. _____ SEQ
- e. _____ FIU
- f. _____ MEM 0 and 2
- g. _____ Run confidence test (option 2 on the FRU main menu).

7. Configuring the System for Remote Diagnostics (optional)

- a. _____ Contact the Rational Response Center and inform them that you plan to test remote diagnostics.
- b. _____ Run INITIOA from the CLI> prompt.
- c. _____ Verify the cluster ID and modify it if necessary.
- d. _____ Verify the type to be used for remote diagnostics and correct it if necessary.
- e. _____ Verify the data needed for the remote diagnostics connection and correct it if necessary.

8. System Peripheral Exercisers:

- a. _____ Tape drive
- b. _____ Ethernet controller
- c. _____ Disk drives

9. Installing the Environment Terminal (optional).

- a. _____ Place the terminal at the location designated on the installation floor plan.
- b. _____ Verify that the terminal cable is installed in the connector marked "Comm Port."

c. _____ Install the power cords and apply power to the terminal.

10. Configuring the Rational Environment

a. _____ Boot the standard configuration.

b. _____ Configure the System.

c. _____ Run Product_Authorization.Register to register any additional products or sessions.

d. _____ Shut down the system and reboot.

e. _____ Load any additional software that the customer has purchased, performing additional system configuration as needed.

11. Overall System Operation:

a. _____ Perform system tailoring (e.g. user accounts, access lists, etc.).

b. _____ Check Login process.

c. _____ Execute overnight run of the Disk_Exerciser (at EEDB).

12. Other Support:

a. _____ Customer Support Response Center: Review operation of the Customer Support Response Center with the customer's system manager.

b. _____ System operations: Provide overview of system operations.

c. _____ Installation Checklist: Review Completed Installation Checklist with System Manager.

d. _____ Support Activity Report (SAR): Complete SAR and distribute copies appropriately.

C

UPGRADE 1.2 GB DISK DRIVE

This appendix presents instructions for installing an additional 1.2 GB drive in Series 400 processor.

Preparation

- 1. Saving system data (This is performed by the customer)
The customer must take a backup of the system before you start this installation. This should be done with no users on the system. (Note: Do not proceed until the backup has been taken.)

- a. Have all users log off the system.
- b. Log in to the system as a user who is a member of group **Privileged**.
- c. Disable all terminal ports for login to ensure that no one logs into the system.

```
Op.Limit_Login (1);
```

- d. Mount a tape to be use for an environment backup.

- e. Take a Full backup of the system.

```
Full_backup;
```

- f. Verify that the tapes are readable.

```
Verify_Backup;
```

Installation

- 1. Shutting down the system
 - a. Enter the privileged Kernel (*executing this commands will trash all the environment*)

```
Kernel: enable_priv_cmds  
Proceed [False]: true  
Password: secret
```

b. Issue the **Go_Back_In_Time** command from the privileged kernel

```
*Kernel: go_back_in_time
The purpose of this command is to trash the current state
of this machine. When the system is next booted, it will
require that you build a new virtual memory system and
recover from backup tape.
Proceed [FALSE]: true
SNAPSHOT_NUMBER [0]: <cr>
```

2. Take a DFS backup from the CLI (you should not need to use this backup; it is merely taken as a precaution):

a. After you had issued the `go_back_in_time` command, the system shutdown, and will be asked if you want to boot the [STANDARD] configuration. Answer this question by pressing "CONTROL and G" keys at the same time. This will take you to the `CLI>` prompt.

b. Mount the tape that is to be used for the DFS backup.

c. Perform the backup

```
CLI> backup/v
```

3. Powering Off the System

a. On the System Control Panel turn **DISK WRITE** keyswitch to **PROTECTED** position.

b. Turn **OPERATOR MODE** keyswitch to the **AUTO** position.

c. Turn **POWER** keyswitch to the **STANDBY** position.

4. Installing the Disk Drives

a. Remove the disk housing cover by turning the fastener screw counter clockwise.

b. Slide the new 1.2 GB drive in the next available slot in the disk housing, make sure that the disk drive is seated properly. Do not move the other drive(s). Unlike previous R1000 systems, the disk drives in the series 400 are **SLOT-DEPENDENT**, i.e., the drive for unit 0 must be in the top slot of the disk housing, unit 1 must be in the 2nd slot from the top, etc.

5. Powering On the System

a. Switch the **POWER** keyswitch on the System Control Panel to the "ON" position—this turns on the AC power. All fans should start blowing. This also allows the DC power supplies to apply DC voltage to the CPU,

controller boards and disk drives.

- b. Unprotect the disk drives by switching the DISK WRITE keyswitch on System Control Panel to the "ENABLED" position.
- c. Boot the STANDARD configuration. When the kernel finds out that there is no Environment, the Environment Recovery program is invoked. The environment backup tape that you had taken earlier should be mounted on the 8mm tape drive to restore the data back in the disks. The following is an example of the output from the environment recovery program.

```

Enter name of configuration to boot [STANDARD] :

.....

---->> Kernel.11.5.0 <<----
Kernel: CHANGE_GHOST_LOGGING
WANT TRACING: FALSE
WANT LOGGING: FALSE
Kernel: START_VIRTUAL_MEMORY
ALLOW PAGE FAULTS: YES

---->> ERROR_LOG <<----
13:03:02 --- TCP_IP_Driver.Worker.Finalized

---->> CONFIGURATOR <<----
starting diagnosis of configuration

**** VOLUME CONFIGURATION ****

UNIT          VOLUME      VOL      DATE OF ROOT
              NAME       NUM      WHEN ADMITTED
-----
0 volume 1    1          06-AUG-91 12:45:56
1

**** PROBLEMS WITH CURRENT VOLUME CONFIGURATION ****

unit 1 state = FORMATTED
missing volume 2
WANT TO BUILD NEW SYSTEM (YES/NO): yes
starting creation of new system
VOLUME NAME FOR unit 0: volume 1
Setting free blocks to gc footprint takes about 15 minutes per drive.
Want to set free blocks to gc footprint? no
VOLUME NAME FOR unit 1: volume 2
creating root volume: 1
adding volume 2 to virtual memory system
creation of new system is complete
starting virtual memory system
the virtual memory system is up

---->> Kernel.11.5.0 <<----
Kernel: START_NETWORK_IO
Kernel: START_ENVIRONMENT
TRACE LEVEL: INFORMATIVE
Kernel:
---->> Environment Elaborator <<----
Elaborating subsystem: ENVIRONMENT_DEBUGGER
Elaborating subsystem: ABSTRACT_TYPES
Elaborating subsystem: MISCELLANEOUS
Elaborating subsystem: OS_UTILITIES

---->> Recovery <<----
Recovery Is Needed, Should I fake it? no

```


Please tell me Volume Id for the Backup Index Tape: `spacebar <cr>`

---->> SYSTEM & RECOVERY <<----

Please Load Tape

(Kind of Tape => CHAINED_ANSI,
Direction => READING)

Is Tape mounted and ready to read labels? `yes`

Info on tape mounted on drive 0 is:

(Kind of Tape => CHAINED_ANSI,
Writeable => FALSE,
Volume Id => D220S2,
Volume Set Name => BACKUP, 05-AUG-91 12:52:05 3)

OK to read volume? [YES] `<cr>`

---->> Kernel.11.5.0 <<----

Kernel:

---->> Recovery <<----

Positioning tape to Backup Index

Processing Backup Index

Processing tape to Backup Data

Processing Backup Data

---->> Kernel.11.5.0 <<----

Starting snapshot

Snapshot is finished

Starting snapshot

Snapshot is finished

---->> Recovery <<----

Recovery Is Complete

Garbage Collection can't run until the machine is rebooted

Reboot before elaborating the Environment

After recovering from the backup tapes, the system will have to be shutdown in order to enable disk collection to run. Reboot the system. (Note: In the future, when D2.2 is release the system will automatically shutdown when the environment recovery is finished.)

- 6. Fill out the SAR and return it to Rational.

Rational Environment Release Information

Release D_12_6_5

Copyright © 1992 by Rational

Part Number: 508-003207-006

May 1992 (Software Release D_12_6_5)

IBM is a registered trademark and RISC System/6000 is a trademark of International Business Machines Corporation.

PostScript is a registered trademark of Adobe Systems Incorporated.

Rational and R1000 are registered trademarks and Rational Environment is a trademark of Rational.

Sun Workstation is a registered trademark of Sun Microsystems, Inc.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

Rational
3320 Scott Boulevard
Santa Clara, California 95054-3197

Contents

1. Overview	1
2. Supported Configurations and Upgrades	1
3. Compatibility	2
4. Upgrade Impact	3
4.1. Impact of Specification Changes	4
4.1.1. Upgrading from D_12_5_0	4
4.1.2. Upgrading from D_12_2_4	4
4.1.3. Upgrading from D_12_1_1	5
4.2. Impact of Implementation Changes	5
5. Known Problems and Limitations	5
5.1. Problem Installing the Cross-Development Facility	5
5.2. Errors Reported When Archiving Switch Files to Non-D_12_6_5 Machines	7
5.3. Limitation in Archiving between Machines Running Different Releases	7
5.4. Problem with Record Representation Clauses and Long_Integer Type	8
5.5. Problem Using DFS Tools with Coprocessors	8
5.6. Problems with Documentation Tools	8
5.7. Problem When Initializing Session Switch Files	9
6. New Environment Interfaces	9
6.1. New Declarations in Package Access_List_Tools	9
6.1.1. Function Grants	9
6.1.2. Function Is_In_Group	10
6.1.3. Function Is_Valid_Group	10
6.1.4. Procedure Remove	10
6.1.5. Procedure Set	10
6.1.6. Procedure Set_Default	11
6.2. New Procedures in System_Maintenance Subsystem	11
6.2.1. Procedure Last_Gasp_Destroy	11
6.2.2. Procedure Release_Tape_User	11
6.2.3. Procedure Show_Load_Proc_Unit_Names	12
6.2.4. Procedure Show_Tape_Users	13
6.3. New Features in Package Dfs	13
6.3.1. Support Tools	13
6.3.2. Dfs_Release File	13
6.4. New Login_Policy World	14
6.4.1. Establishing a Login Policy	14
6.4.2. Specifying the Login Message	16
6.5. New Telnet_Tools Subsystem	16
6.6. Products Now Delivered with the Environment	17
6.6.1. Rational X-Interface and Windows-Interface	17
6.6.2. Documentation Tools	17
7. Changes from D_12_5_0	17
7.1. Changes to Package Common	18
7.2. Changes to Editing Ada Units	18
7.3. Changes to Debugging	19
7.3.1. Quitting While the Debugger is Running	19
7.4. Changes to Session and Job Management	19
7.5. Archive Changes	20
7.5.1. Copying Ada Units with Subobjects	20
7.5.2. Archiving Library Switch Files	20
7.6. Compilation Changes	21

7.6.1. Installing Units	21
7.6.2. Coding Units	23
7.6.3. Loading and Executing Units	23
7.6.4. Program Libraries	24
7.7. Changes to Library Switches	24
7.7.1. New Switch: Semantics.Limit_Semantic_Messages	24
7.7.2. New Switch: Semantics.Reject_Statement_Prompts	24
7.7.3. Change to Default Value of R1000_Cg.Retain_Delta1_Compatibility Switch	25
7.8. Changes to Other Library Management Facilities	25
7.9. Changes to Device Input and Output	26
7.10. Changes to Printing	26
7.11. Changes to System Backups	26
7.12. Changes to the Diagnostic File System (DFS)	27
7.12.1. Renamed Procedures	27
7.12.2. Analyze_Crashdump Prompts for Input	27
7.13. Changes to Crash Analysis	27
7.13.1. Changes to the Crash Analysis Procedures	27
7.13.2. Change to Crashdumps and Tombstones	28
7.14. Changes to Token Policies	28
7.14.1. Transferring Tokens between Series 400 and 300	28
7.14.2. Token Changes to RXI	29
7.15. Miscellaneous Changes to System Management	30
7.16. Changes to CMVC	31
7.17. Update to !Tools.Math_Support	31
7.18. Miscellaneous Changes	32
7.18.1. Change to Library_Object_Editor	32
7.18.2. Change to Code_Segment_Object_Editor	32
8. Documentation	32
8.1. New Printed Documentation	32
8.2. New Online Documentation	33
8.3. New Help for Switches	33
8.3.1. Session Switches	34
8.3.2. Library Switches	34
A Guide to Machine Initialization	35
A.1. Overview	35
A.1.1. World !Machine.Initialization.Rational	36
A.1.2. Worlds !Machine.Initialization.[Site,Local]	37
A.2. Setting Up the Site and Local Worlds	38
A.3. Hints for Implementing System Customizations	39
A.3.1. Writing Customized Initialization Procedures	39
A.3.2. Using “_Start” Files to Reference Initialization Procedures	40
A.3.3. Controlling the Order of Execution	41
A.3.4. Customizing Disk-Collection Thresholds	42
A.4. Enabling and Configuring Login Ports	42
A.4.1. Enabling Ports for Login	42
A.4.2. Customizing Port Characteristics	44
A.4.3. A Sample Terminal_Configuration File	45
A.4.4. Terminal-Configuration Options	46
A.5. Configuring Printers	47
A.5.1. Where to Specify Printer Information	48
A.5.2. Adding Entries to a Printer_Configuration File	49
A.5.3. Specifying a Directly Connected Printer	50
A.5.4. Specifying a Networked Printer	51
A.5.5. Specifying an Environment File	51

A.5.6. Specifying a Workstation Directory	52
A.5.7. Associating Default Printers with Individual Users	55
B Diagnostic Crash Procedures	57
B.1. Introduction	57
B.2. When Your System Crashes	57
B.3. Using These Diagnostic Crash Procedures	57
B.4. Saving Information about the System State	58
B.5. Crash due to Power Failure	59
B.5.1. Indication of Power Failure	59
B.5.2. Recovering from Power Failure	60
B.6. Crash due to Overheating	61
B.6.1. Indications of Overheating	61
B.6.2. Recovering from Overheating	61
B.7. Displays "Connect Modem" Message and Hangs	61
B.7.1. Indications of "Failure to Connect Modem"	61
B.8. Other System Failures	62
B.8.1. Redisplaying the Crash Reason	63
B.9. Common Crash Reasons and Responses	64
B.9.1. Halt => System Error	64
B.9.2. Halt => I/O Processor Hardware Error	64
B.9.3. Halt => I/O Processor Software Error	64
B.9.4. Halt => Processor Hardware Error	64
B.9.5. Halt => Processor Multi Bit Memory Error	65
B.9.6. Halt => Processor Sysbus Hardware Error	65
B.9.7. Halt => Processor Microcode Error	66
B.9.8. Halt => Processor Software Error	66
B.9.9. Halt => Shutdown from Environment	66
B.9.10. Halt => Processor Crash Error	67
B.9.11. Machine Check (Parity Error)	69
B.9.12. Any Other Reason	69
B.10. If You Suspect Disk Errors	69
B.10.1. Running Diskx	69
B.10.2. Running Checkdisk	70
B.11. Recovering Crash Information from the Environment	70
B.12. Contacting the Rational Response Center	72
C Problem Reports Closed in D_12_6_5	73
C.1. Software Problem Reports Closed	73
C.2. Documentation Problem Reports Closed	80

1. Overview

The D_12_6_5 release of the Rational Environment™ is primarily a maintenance release that:

- Fixes a large number of problems from previous releases, enabling existing commands to work correctly. Some of these fixes introduce new features (see section 6); other fixes involve changed features (see section 7).
Efforts to fix problems have been focused in:
 - System reliability.
Several problems that could cause the Environment to crash have been fixed; and the Environment has been made more robust in many (previously fatal) error conditions.
 - The Ada compilation system.
In select cases, some changes may cause code that used to compile to no longer compile. For details, see section 7.6.
- Changes the default value of the R1000_Cg.Retain_Delta1_Compatibility library switch (from True to False) so that, by default, the R1000 code generator takes advantage of numerous features and bug fixes introduced in the D_12_1_1 compiler. These features and bug fixes may cause the code generator to emit code that is not compatible with releases prior to D_12_1_1; see section 7.7.
- Introduces various new Environment interfaces, including:
 - New procedures and functions in packages Access_List_Tools, System_Maintenance, and Dfs; see sections 6.1, 6.2, and 6.3.
 - A new world (!Machine.Login_Policy), in which you can place subprograms and files that control login policies and messages; see section 6.4.
 - A new subsystem (!Tools.Telnet_Tools), which provides tools for writing programs that use the Environment's networking facilities; see section 6.5.
- Delivers the Documentation Tools, Rational X Interface (RXI), and Rational Windows Interface (RWI) products with the standard Environment. Product authorization tokens are no longer required to use these products. See sections 3 and 7.14.
- Introduces three expanded and up-to-date volumes of the *Rational Environment Reference Manual*:
 - Volume 4: Session and Job Management (SJM)
 - Volume 5: Library Management (LM)
 - Volume 10: System Management Utilities (SMU)
 New online help is also provided for each of the packages described in these books; see section 8.

Note: This release information can be found in the !Machine.Release.Release_Notes world in line-printer (Environment_Release12_6_5_Lpt) and PostScript® (Environment_Release12_6_4_Ps) formats.

2. Supported Configurations and Upgrades

D_12_6_5 supports the following configurations of the R1000:

- Series 100

- Series 200 (Models 10, 20, and 40)
- Series 300 System (300S)
- Series 300 Coprocessor (300C) for the IBM® RISC System/6000™ and Sun Workstation® servers
- Series 400 System (400S)
- Series 400 Coprocessor (400C) for the IBM RISC System/6000 and Sun Workstation servers

D_12_6_5 supports two kinds of tape drive:

- The 9-track tape drive, which is standard on the Series 100 and 200, and an optional upgrade to the Series 300S
- The 8-millimeter tape drive, which is standard on the Series 300S, 300C, 400S, and 400C, and an optional upgrade to the Series 200

D_12_6_5 also supports the optional expansion of memory from 32 megabytes to 64 megabytes to improve system performance. This upgrade applies to all series except the Series 100. The combinations of configurations and upgrades supported by D_12_6_5 are shown in Table 1.

Table 1 Configurations and Upgrades Supported by D_12_6_5

Configuration	8-mm Tape Drive	9-Track Tape Drive	32-Mb Memory	64-Mb Memory
Series 100	N/A	Standard	Standard	N/A
Series 200	Upgrade	Standard	Standard	Upgrade
Series 300S	Standard	Upgrade	Standard	Upgrade
Series 300C	Standard	N/A	Standard	Upgrade
Series 400S	Standard	N/A	Standard	Upgrade
Series 400C	Standard	N/A	Standard	Upgrade

3. Compatibility

D_12_6_5 is fully compatible with all production versions of Rational layered-software products (Table 2).

Table 2 Compatibility with D_12_6_5

Rational Product	Compatible Release
CDF: Mc68020_Bare	5_1_2 or later*
CDF: Mc68020_Os2000	6_1_3 or later
CDF: Mc68020_Hp_Unix	6_2_4 or later
Design Facility: 2167	6_0_7 or later
Design Facility: 2167A	6_2_5 or later
Design Tools	10_2_9 or later
Documentation Tools	10_2_9 or later†
Mail	11_4_5 or later
Rational Publishing Interface	1_0_2 or later
Rational Teamwork Interface	2_1_2 or later
Rational X Interface	10_5_2 or later†
Rational Windows Interface	10_1_1 or later†
Remote Compilation Facility	1_1_1 or later
RCF: Ibm_Rs6000_Aix	1_1_0 or later
RPC	1_0_2 or later
Software Analysis Workstation	6_0_0 or later
Target Build Utility	10_0_3 or later

* Before attempting to install new or changed CDF products or customizations, see section 5.1.

† Beginning with D_12_6_5, the most recent releases of Documentation Tools, RXI, and RWI are delivered with the Environment. Thus, compatibility with these products is no longer an issue. For more information, see sections 6.6 and 7.14.

4. Upgrade Impact

The Environment can be upgraded from to D_12_6_5 without forcing you to Archive.Save and Archive.Restore your applications. You will not have to modify or recompile any of your own tools, with the possible exception of:

- Tools written against the unit specifications listed in "Impact of Specification Changes," below.
- Customizations of unit bodies listed in "Impact of Implementation Changes," below.

The new declarations listed in section 6 are all installed upward-compatibly and therefore have no impact on user-written tools.

Rational does not provide tools for reverting to previous releases of the Environment. You will be able to revert to the previous release, however, if you make a complete backup of the Environment, including the Diagnostic File System (DFS), before you upgrade.

4.1. Impact of Specification Changes

The installation process for D_12_6_5 changes several Environment specifications. The installation process attempts to make all changes incrementally; however, if incremental changes fail, the installation process overwrites the necessary specifications. Overwriting these specifications will cause the demotion of any customer-created tools written against them. The installation process tries to recompile such tools automatically; however, depending on the nature of the tools, some may require modification before they can be recompiled. Units that cannot be recompiled during installation are listed in the installation log.

The unit specifications that are overwritten depend on the release from which you are upgrading.

4.1.1. Upgrading from D_12_5_0

The following unit specifications are overwritten when you upgrade from D_12_5_0 to D_12_6_5:

- !Commands.Library'Spec
- !Commands.System_Backup'Spec
- !Tools.Access_List_Tools'Spec
- !Tools.Code_Segment_Object_Editor'Spec
- !Tools.Library_Object_Editor'Spec

4.1.2. Upgrading from D_12_2_4

In addition to the unit specifications overwritten when upgrading from D_12_5_0, upgrading from D_12_2_4 overwrites the following unit specification:

- !Commands.Abbreviations.Print'Spec
- !Commands.Access_List'Spec
- !Commands.Archive'Spec
- !Commands.Queue'Spec
- !Commands.Remote_Passwords'Spec
- !Tools.Networking.Tcp_Ip_Boot'Spec
- !Tools.Networking.Transport'Spec
- !Tools.Networking.Transport_Defs'Spec
- !Tools.Networking.Transport_Name'Spec

Furthermore, the units in !Machine.Initialize@ are either demoted or moved to other locations to accommodate the D_12_5_0 mechanisms for initializing an R1000; for complete details, see

Appendix A, "Guide to Machine Initialization," or the online information in !Machine.Initialization.Guide_To_Machine_Initialization.

If you are upgrading from D_12_2_4, you should also see the release information for D_12_5_0; release information is located online in !Machine.Release.Release_Notes.

4.1.3. Upgrading from D_12_1_1

In addition to the unit specifications overwritten when upgrading from D_12_5_0 and D_12_2_4, upgrading from D_12_1_1 overwrites the following unit specification:

- !Implementation.Work_Order_Implementation'Spec

If you are upgrading from D_12_1_1, you should also see the release information for D_12_2_4 and D_12_5_0; release information is located online in !Machine.Release.Release_Notes.

4.2. Impact of Implementation Changes

The installation process for D_12_6_5 overwrites one Environment unit body (regardless of the release from which you are upgrading):

- !Commands.Abbreviations.Print'Body

Because this body may contain user customizations, its contents are saved in a text file in the same library as the overwritten body. The name of the file is of the form *Unit_Name_Vnn*, where *nn* is the unit's default version number. These customizations then can be transferred to the new implementation.

5. Known Problems and Limitations

5.1. Problem Installing the Cross-Development Facility

The D_12_6_5 release of the Environment affects the installation of any of Rational's family of Cross-Development Facility (CDF) products. Specifically, you will not be able to install new or changed CDF products or customizations without first making a minor change to the following package specification:

!Targets.Target_Key.Io.Floating_Point.M68k_Floats'Spec

As delivered, the CDF products release prior to the D_12_6_5 Environment cannot be promoted to the coded state on a D_12_6_5 system because of changes made to the Environment's compilation system that enable it to comply with validation. Existing CDFs are not affected by this problem unless package M68k_Floats is demoted and an attempt is made to repromote it.

The change you need to make to M68k_Floats'Spec follows.

Note: Changing M68k_Floats'Spec will cause the demotion of M68k_Floats'Body, M68k_Float_Conversions'Body, and M68k_Float_Conversions.Based_Value'Body. Demoting these units will also demote any coded main programs that were coded with the CDF.

1. Locate the following two representation clauses. They can be found at the beginning of the private part (line 164 of the unit).

```

for Single_Format use
  record at mod 4;
    Negative at 0 range 0 .. 0;
    Exponent at 0 range 1 .. 8;
    Fraction at 0 range 9 .. 31;
  end record;

for Double_Format use
  record at mod 4;
    Negative      at 0 range 0 .. 0;
    Exponent      at 0 range 1 .. 11;
    Fraction_Hi   at 0 range 12 .. 31;
    Fraction_Lo   at 0 range 32 .. 63;
  end record;

```

2. Move the representation clause for Single_Format so that it immediately follows the declaration of its type (line 34 of the unit).

```

type Single_Format is          -- line 34
  record
    Negative : Boolean;
    Exponent : Single_Exponent;
    Fraction  : Single_Fraction;
  end record;

for Single_Format use          -- move rep clause to here
  record at mod 4;
    Negative at 0 range 0 .. 0;
    Exponent at 0 range 1 .. 8;
    Fraction at 0 range 9 .. 31;
  end record;

function Convert_Out is new Unchecked_Conversion
  (Source => Single_Float,
   Target => Single_Format);

function Convert_In  is new Unchecked_Conversion
  (Source => Single_Format,
   Target => Single_Float);

```

3. Move the representation clause for Double_Format so that it immediately follows the declaration of its type (now line 60 of the unit).

```

type Double_Format is          -- line 60
  record
    Negative      : Boolean;
    Exponent      : Double_Exponent;
    Fraction_Hi   : Double_Fraction_Hi;
    Fraction_Lo   : Double_Fraction_Lo;
  end record;

for Double_Format use          -- move rep clause to here
  record at mod 4;
    Negative      at 0 range 0 .. 0;
    Exponent      at 0 range 1 .. 11;

```

```

    Fraction_Hi at 0 range 12 .. 31;
    Fraction_Lo at 0 range 32 .. 63;
end record;

```

```

function Convert_Out is new Unchecked_Conversion
    (Source => Double_Float,
     Target => Double_Format);

```

```

function Convert_In is new Unchecked_Conversion
    (Source => Double_Format,
     Target => Double_Float);

```

4. Repromote M68k_Floats'Spec.
5. Repromote M68k_Floats'Body and the other two units that have been demoted, M68k_Float_Conversions'Body and M68k_Float_Conversions.Based_Value'Body.
6. Repromote any coded main programs that were coded with the CDF.

This problem will be fixed in Release6_3_3 of the Bare compiler, which should be available soon. Rational is also in the process of creating a new version of the OS2000 CDF product with this fix in place.

5.2. Errors Reported When Archiving Switch Files to Non-D_12_6_5 Machines

The D_12_6_5 release of the Environment makes several changes to switches. When combined, two of these changes result in error messages when archiving switch files from D_12_6_5 systems to previous releases:

- The D_12_6_5 release of the Environment changes the way in which switch files are copied. When used to copy a switch file in previous releases, the Archive.Copy procedure copied only switches that had nondefault values. In D_12_6_5, the Archive.Copy procedure copies the entire switch file, including switches that have default values.
- D_12_6_5 also introduces a new library switch (see section 7.7).

As a result of these changes, if you use Archive.Copy to copy library switches from D_12_6_5 to a system running a previous release of the Environment, Archive.Copy may generate errors indicating that certain switches do not exist. These errors can be ignored; the copy operation will complete successfully.

Note that these errors are not considered a bug and will not be fixed in future releases.

5.3. Limitation in Archiving between Machines Running Different Releases

There is a potential problem when performing Archive.Resore operations between two systems that are running different releases of the Environment.

Because of changes made in D_12_5_0 (and perpetuated in D_12_6_5) in the way Archive saves subsystem compatibility databases (CDBs), you may experience problems when trying to restore a D_12_5_0 or D_12_6_5 archive on a pre-D_12_5_0 system. For example, the Archive operation may generate internal errors such as:

```
INTERNAL ERROR (declaration mapping): ARCHIVE_IS_BAD restoring
```

```
!Subsystem_Name.State.Compatibility.Declaration_Numbers.Unit_Name
```

A simple workaround exists. When saving objects to pre-D_12_5_0 systems, specify `Version=527` in the the Options parameter of `Archive.Save`. For example, the following operation successfully saves `My_View`:

```
Archive.Save (Objects => "My_View",  
             Options  => "Version=527",  
             Device   => "Machine.Devices.Tape_0",  
             Response => "<PROFILE>");
```

When an attempt is made to restore `My_View` to a `D_12_5_0` or `D_12_6_5` system, `Archive.Restore` checks to see under what version of `Archive` the object was saved and adjusts accordingly.

Note that this problem does *not* affect:

- `Archive.Copy` operations, which copy over the network. In this case, `Archive` checks the version number on each end and uses the lower version number of the two.
- `Archive` operations on text files, Ada units, or other non-CDB objects.
- `Archive` operations between systems running `D_12_6_5` and `D_12_5_0`.

Note also that this is a compatibility issue with how CDBs are saved and not a bug.

5.4. Problem with Record Representation Clauses and Long_Integer Type

A record representation clause may not include a component clause corresponding to a component of type `Standard.Long_Integer` (or a type derived, directly or indirectly, from type `Long_Integer`) unless an explicit range constraint has been given.

An update that fixes this problem is already being developed and will be available shortly.

(PRS number 8391844-Gato-Gbd)

5.5. Problem Using DFS Tools with Coprocessors

A problem has been found with the new Diagnostic File System (DFS) tools, located in `!Machine.Dfs`. Specifically, the `Fs_Download` procedure does not work correctly when attempting to load a new DFS onto either a series 300C or 400C. This is considered a bug and will be fixed in a future release.

Furthermore, the `Fs_Download` procedure cannot be used to load a new DFS onto a 100. This is a permanent limitation and will not be fixed.

5.6. Problems with Documentation Tools

The Documentation Tools delivered with the `D_12_6_5` release of the Environment include only the Document Formatter; they do not include the Spelling Checker. If you would like to obtain the Spelling Checker, contact your Rational representative.

For information about the Document Formatter, see section 6.6.2.

5.7. Problem When Initializing Session Switch Files

A problem has been found in the way in which the Environment initializes session switch files for sessions other than the default session. This problem is not introduced by the D_12_6_5 release; however, it has become more obvious with the introduction of new documentation for session switches in the SJM book.

Each user's default session (S_1) obtains its initial session-switch values from system-defined default values. If a switch file does not exist for session S_1, the session uses the same system-defined default values.

Subsequent sessions for which switch files have not been created use the switch values set for the user's default session, S_1. If a user creates a switch file for any of these subsequent sessions, the initial values are copied from the system-defined default values. This is a bug. Subsequent sessions should copy their initial switch values from the default session, S_1, as described in the new SJM book.

6. New Environment Interfaces

D_12_6_5 provides a number of new interfaces, including:

- New procedures and functions in:
 - Package Access_List_Tools; see section 6.1
 - Package System_Maintenance; see section 6.2
 - Package Dfs; see section 6.3
- New login policy features; see section 6.4.
- A new subsystem that provides networking tools; see section 6.5.
- New interface and documentation facilities that were previously delivered separately; see section 6.6.

6.1. New Declarations in Package Access_List_Tools

Package !Tools.Access_List_Tools contains the new procedures and functions described in the following paragraphs.

For complete information about this subprograms, see the new documentation for package Access_List_Tools in the Library Management (LM) book or in online help.

6.1.1. Function Grants

```
function Grants (For_List : Acl;
                Desired   : Access_Class;
                User_Name  : String := "") return Boolean;
```


Returns a Boolean value indicating whether the specified access-control list grants the specified username the desired access rights. This function is useful for restricting jobs that normally have unlimited access, such as servers and other *SYSTEM jobs. For example, you can use this function to cause a job or program to operate on objects only if members of the group Network_Public are permitted the appropriate access to those objects.

6.1.2. Function Is_In_Group

```
function Is_In_Group (User_Name : String;  
                    Group_Name : String) return Boolean;
```

Returns a Boolean value indicating whether the specified username is a member of the specified access-control group. This function returns True if the specified user is a member of the specified access-control group. This function returns False if an error is detected during the test or if the user is not a member of the specified access-control group.

(PRS numbers 0-0301-9 and 9723630-0011-5)

6.1.3. Function Is_Valid_Group

```
function Is_Valid_Group (Name : String) return Boolean;
```

Returns a Boolean value indicating whether the specified name is a valid access-control group.

6.1.4. Procedure Remove

```
procedure Remove (Group      : String;  
                Initial_Acl : Bounded_String.Variable_String;  
                New_Acl     : out Bounded_String.Variable_String;  
                Group_Found : out Boolean);
```

Removes the entry for the specified group from the specified access-control list.

6.1.5. Procedure Set

```
procedure Set (For_Object : Directory.Version;  
             To_List     : Acl;  
             Status      : in out Simple_Status.Condition;  
             Action_Id   : Action.Id);
```

Setting an object's ACL requires owner access to the nearest world enclosing the object.

Note that setting a world's ACL does not require owner access to the world itself, provided that you have owner access to its parent world. This allows you to change the ACL of a world when no one has owner access to it.

This procedure is the same as other Set procedures, with the addition of an Action_Id parameter.

6.1.6. Procedure Set_Default

```

procedure Set_Default (For_World :      Name;
                      To_List   :      Acl;
                      Status    : in out Simple_Status.Condition;
                      Action_Id  :      Action.Id);
```

Sets the default access-control list (ACL) for the specified world or worlds.

Setting a world's ACL requires owner access to that world. A world's default ACL is used to initialize the ACLs of new Ada units and files created in the world.

This procedure does not modify the ACLs of already existing objects whose ACLs were set by the previous default ACL.

This procedure is the same as other Set_Default procedures, with the addition of an Action_Id parameter.

6.2. New Procedures in System_Maintenance Subsystem

!Commands.System_Maintenance'Spec_View contains the new procedures described in the following paragraphs.

6.2.1. Procedure Last_Gasp_Destroy

```

procedure Last_Gasp_Destroy (Name : String);
```

Caution: Use this procedure only as a last resort.

Forcibly removes an object from a directory. Use this procedure only when you've tried every other means of deleting an object or set of objects. Normally:

-

This procedure should be used only if you are consistently getting Version_Error on a particular object.

- This procedure can be used instead of elaborating Dir_Tester.
- This procedure should *not* be used if you are getting "Key not found" errors.

This procedure should be used only by Rational personnel or at their discretion. For a detailed description, including a complete list of steps you should try before using this procedure, see this procedure's specification:

```

!Commands.System_Maintenance'Spec_View.Units.Last_Gasp_Destroy'Spec
```

Note that you must be operating in privileged mode to execute this procedure.

6.2.2. Procedure Release_Tape_User

```

procedure Release_Tape_User (Drive : Natural);
```

Resets the specified tape drive so that it is free, thereby allowing tape-mount requests to proceed without "Drive is in use" errors. This procedure also attempts to rewind the tape.

Note that this procedure does not physically reset the drive.

Note also that you must be operating in privileged mode to execute this procedure.

The fully qualified name of this procedure is:

```
!Commands.System_Maintenance' Spec_View.Units.Release_Tape_User
```

(PRS number 9123690-0160-1 and CSR number 4788)

6.2.3. Procedure Show_Load_Proc_Unit_Names

```
procedure Show_Load_Proc_Unit_Names (Load_Proc : String := "<SELECTION>";  
                                     Response  : String := "<PROFILE>");
```

Obtains information that allows you to find the objects from which a loaded main program (Load_Proc or Load_Func) was created. Specifically, this procedure obtains the Ada object instance ID numbers for the Ada units that make up the source code for the loaded main program specified in the Load_Proc parameter.

The display produced by this procedure includes lines of the form:

```
Library.Resolve ("<[ADA, 12345,1]>");
```

These lines indicate the Ada object instance ID of one or more of the original Ada source units used to create the loaded main program. If those Ada source units still exist, and if they exist on the current machine, then the Environment pathnames of those units are also displayed.

If the original source units do not exist (because they have been edited, moved, renamed, or deleted), or if the loaded main program was copied from another machine, then no pathname is printed. To attempt to locate the source units for a loaded main program that was copied from another machine:

1. Take the Library.Resolve lines that were printed by the Show_Load_Proc_Unit_Names procedure (as shown above).
2. Log into each R1000 where the source units might still exist.
3. Execute the Library.Resolve commands.

If the source unit exists on any of these machines, the Library.Resolve command returns the fully qualified pathname of that unit.

The fully qualified name of this procedure is:

```
!Commands.System_Maintenance' Spec_View.Units.Show_Load_Proc_Unit_Names
```

(PRS number 9123690-0148-4)

6.2.4. Procedure Show_Tape_Users

```
procedure Show_Tape_Users;
```

Displays the current user and session for each tape drive that is defined in the IOP configuration.

The fully qualified name of this procedure is:

```
!Commands.System_Maintenance' Spec_View.Units.Show_Tape_Users
```

6.3. New Features in Package Dfs

6.3.1. Support Tools

The access routines for the Environment's Diagnostic File System (DFS) have been enhanced to make it easier to upgrade the DFS and to aid in problem resolution at the DFS (and lower) level. These changes allow an operator to:

- Read a DFS tape into the Environment.
- Create an Environment MT Image file object, containing the contents of that tape. (An MT Image object contains an ordered collection of DFS files. Creation of an MT Image object makes it possible to handle a set of DFS files as a single object.)
- Download to the DFS from the Environment (either from an arbitrary Environment file object or from an Environment MT Image object created from a DFS tape).

Support tools to create DFS tapes and move DFS files within the DFS are also provided. These support tools take the form of:

- Several new commands in package !Machine.Dfs' Spec_View.Units.Dfs
- A new package, !Machine.Dfs' Spec_View.Units.Install_Tools

For complete details about the new commands in package Dfs and the new package Install_Tools, see the exported specifications online.

For information about changes to existing commands in package Dfs, see section 7.12.

6.3.2. Dfs_Release File

To aid in tracking updates to the DFS, D_12_6_5 also includes a new Dfs_Release file, which contains the release date, number, and notes for the current DFS release. To read Dfs_Release from the Environment, execute the following command:

```
!"!Machine.Dfs' Spec_View.Units".Dfs.Copy
  (Dfs_Filename    => "Dfs_Release",
   Env_Filename    => "Environment filename for output",
   Allow_Overwrite => False);
```

Then traverse to the file specified in the Env_Filename parameter. The file should contain DFS release information such as the following:

DFS D3.0

April 14, 1992

DFS release D3.0

User interface is now menu driven

New program TOMBSTONE for viewing tombstone files from the DFS invoked by using "x tombstone" from the "CLI>" prompt.

New program REWRITE for refreshing all data on a disk from the DFS invoked by using "x rewrite" from the "CLI>" prompt. For details and a copy of the standard help dialogue, see below.

New IOP Kernels for 300C, 300S, 400C and 400S are provided. The following

bug(s) was(were) fixed:

PRS => 9123690-0165-8/2,
9123690-0164-4

CSR => 5559

Software flow control on port 16 has been repaired.

This file was originally introduced in release D_12_5_0A.

6.4. New Login_Policy World

D_12_6_5 includes a new world, !Machine.Login_Policy, in which system managers can create subprograms and files that:

- Establish a policy monitoring and responding to successful and failed login attempts.
- Specify the login message that is displayed when users connect to the R1000.

For security purposes, this world is delivered with Operator access only. As delivered, the Login_Policy contains no procedures; sites that do not take advantage of this feature will notice no change in the system behavior.

These features are described in more detail in the following paragraphs.

6.4.1. Establishing a Login Policy

The login manager now contains a call to operator-defined procedures that determine site-specific login policies. Although primarily intended as a mechanism for system managers to react promptly to invalid login attempts, these facilities also provide the basis for additional site-specific features.

When a login attempt is made, the Environment's login manager now checks for the existence of operator-defined procedures Successful_Login and Failed_Login in world !Machine.Login_Policy. If these procedures exist, then:

- If the login attempt was successful, the login manager calls `Successful_Login`.
- If the login attempt was not successful, the login manager calls `Failed_Login`.

The user-defined procedures must be placed in `!Machine.Login_Policy` and have the following specifications:

```

procedure Successful_Login (Username      : String;
                             Session      : String;
                             Time         : String;
                             Port        : Natural;
                             Network_Name : Transport_Defs.Network_Name;
                             Host_Id     : Transport_Defs.Host_Id;
                             Socket_Id   : Transport_Defs.Socket_Id;
                             Error_Status : Natural);

procedure Failed_Login (Username      : String;
                          Session      : String;
                          Time         : String;
                          Port        : Natural;
                          Network_Name : Transport_Defs.Network_Name;
                          Host_Id     : Transport_Defs.Host_Id;
                          Socket_Id   : Transport_Defs.Socket_Id;
                          Error_Status : Natural);

```

The parameters of these procedures are described below:

- **Username**: the name supplied at the `Enter username` prompt during the attempted login or the null string if this information has not yet been entered.
- **Session**: the session specified at the `Enter session name` prompt during the attempted login or the null string if this information has not yet been entered. If a [Return] is supplied at the `Enter session name` prompt, session `S_1` is assumed.
- **Time**: the date and time at which the login attempt occurred. The format for this string is the date, one blank space, and then the time. The date is of the form *month/day/year*, and the time is expressed in military format. Procedures from package `!Tools.Time_Uutilities` can be used to process this information.
- **Port**: the number of the R1000 port on which the login attempt is being made.
- **Network_Name**: identifies one of several available protocol stacks for the current connection. For example, the value `TCP_IP` is passed for a Telnet connection, and the null string is passed for a direct connection to the R1000. See package `Transport_Defs` and subsystem `Telnet_Tools` in `!Tools.Networking` for procedures to process this variable.
- **Host_Id**: identifies the machine within a network making the connection. See package `Transport_Defs` and subsystem `Telnet_Tools` in `!Tools.Networking` for procedures to process this variable.
- **Socket_Id**: identifies a program within a machine. See package `Transport_Defs` and subsystem `Telnet_Tools` in `!Tools.Networking` for procedures to process this variable.
- **Error_Status**: a status code that indicates either success or the reason that the login attempt failed. The status codes are:

- | | |
|--------------------------|------------------------------|
| 0 - Successful | 1 - Bad_Username_Or_Password |
| 2 - Expired_Password | 3 - Abandoned_Login |
| 4 - Read_Timed_Out | 5 - Read_Error |
| 6 - Input_Unreadable | 7 - Terminal_Type_Unreadable |
| 8 - Resource_Restriction | 9 - Product_Not_Authorized |
| 10 - Internal_Error | |

*Note: The Successful_Login and Failed_Login procedures are run under the identity of *SYSTEM to allow privileged operations to be performed. For this reason, system managers may want to use existing access-control mechanisms to control access to these procedures and ensure system security. As it is delivered, the Login_Policy world is accessible only to the members of group Operator.*

If the procedures exist, and if an attempt to run either of these procedures is unsuccessful, this is noted in the system error log. If the procedures do not exist, then the login process runs as usual, and nothing is recorded in the system log.

This change does not affect the disconnect and login behavior options for a port. These options are still set separately through !Machine.Initialization or through commands in package !Commands.Terminal.

(PRS number 9723630-0074-7)

6.4.2. Specifying the Login Message

The D_12_6_5 release provides the ability for system managers to specify the message users see when they connect to an R1000.

The system's standard message indicates the release number for the current Environment, a name identifying the machine, the port number through which the connection has been made, and the current date and time:

```
Rational Environment D_12_5_0; Roget - Ada Tech Pubs; Port 232.  
January 16, 1992 at 2:07:34 PM
```

In D_12_6_5, system managers can place other messages in a text file of the name !Machine.Login_Policy.Login_Text. If this file exists, its contents are displayed when users connect to the R1000. Placing the string <Standard_Message> on its own line in this file causes the standard login message to be displayed (in addition to any other text in the file).

(PRS number 9723630-0025-8)

6.5. New Telnet_Tools Subsystem

This new subsystem provides programmatic interfaces to Rational Networking as well as some new utility software. It includes three new packages:

- !Tools.Networking.Telnet_Tools'Spec_View.Units.Telnet_Port
- !Tools.Networking.Telnet_Tools'Spec_View.Units.Telnet_Protocol
- !Tools.Networking.Telnet_Tools'Spec_View.Units.Telnet_Tools

For more information, see the exported specifications online.

6.6. Products Now Delivered with the Environment

6.6.1. Rational X-Interface and Windows-Interface

Rational's X-Interface (RXI) and Windows Interface (RWI) have become the primary access mechanisms to the Environment. In response to this evolution, the most recent releases of RXI (10_9_0) and RWI (10_1_1) are delivered with the D_12_6_5 release of the Environment.

Note that RXI and RWI still must be installed using a separate installation process; they are not installed during Environment installation:

- If you already have RXI and/or RWI installed on your system, you do not need to install them again. D_12_6_5 does not include any changes to these products.
- If you do not have RXI and/or RWI installed on your system and would like to install one or the other, contact your Rational representative for installation instructions, keyboard overlays, and documentation. To install RWI, you will also need to obtain a diskette that contains software to be installed on the PC.
- If you have purchased a new system, the installation instructions for RXI and RWI are included with that system. Contact your Rational representative to obtain keyboard overlays and documentation.

For further information about RXI token issues, see section 7.14. If problem reports concerning the improper consumption of tokens by RXI

6.6.2. Documentation Tools

D_12_6_5 includes Rational's Documentation Tools as part of the standard Environment. As part of the D_12_6_5 release, a new release of the Document Formatter has been introduced (Rev10_7_12) and the system default activity has been updated to reverence this release. To ensure backward compatibility, the previous version of the Document Formatter (Rev10_7_7) has been preserved and can be used simply by changing your activity file.

Although the Spelling Checker is part of Rational's Documentation Tools, it has not been included in the D_12_6_5 Environment release (see section 5.6). If you would like to obtain the Spelling Checker, contact your Rational representative.

Documentation is available for both the Document Formatter and the Spelling Checker. For information, contact your Rational representative.

7. Changes from D_12_5_0

This section describes the changes, enhancements, and user-visible problem fixes that D_12_6_5 makes to existing features of the Environment. The information in this section is presented in roughly the same order in which it would be found in the *Rational Environment Reference Manual* (ERM):

- Information on editing specific types of objects such as Ada units (EST) (sections 7.1 and 7.2)

- Information on session and job management (SJM)—specifically, packages Job, Remote_Passwords, Operator, and Search_List (section 7.4)
- Library-management information (LM), including changes to archive (section 7.5), compilation (7.6), library switches (7.7), and other library-management facilities (7.8)
- Device input and output (DIO) changes (section 7.9)
- System-management information (SMU), including changes to system backup (section 7.11), the Diagnostic File System (DFS) (7.12), crash analysis (7.13), token policies (7.14), and miscellaneous system-management facilities (7.15)
- Information about subsystems and CMVC (PM) (section 7.16)
- Information about changes to the Math_Support subsystem (section 7.17)
- Information about other miscellaneous changes (section 7.18)

7.1. Changes to Package Common

Performing Common.Definition on an object in a generic instantiation now takes you to the declaration in the generic. (PRS numbers 9123690-0163-9, 9123690-0165-3, 9123690-0169-0, and 9723630-0122-0 and CSR numbers 5501, 5454, and 5707)

The Common.Format procedure has been fixed so that it no longer destroys the contents of an Ada unit when an attempt to format that unit fails. (PRS numbers 1704151-Shei-Jst, 2437110-Sier-Geb, 5172201-Cook-Swb, 7039962-Sier-Geb, 7030779-Zebr-Lore, 8068463-Mago-Sdj, 9123690-0156-7, 9123690-0175-9, and 9123690-0181-8 and CSR numbers 4786 and 6261)

7.2. Changes to Editing Ada Units

The Ada.Make_Inline procedure now properly maintains Sm_Subunits. Sm_Subunits is an attribute on the DIANA nodes for compilation units. Specifically, Sm_Subunits is a sequence of the subunit Stub_Ids that exists for that unit. (PRS number 6452721-Shei-Jst)

The Ada.Show_Unused procedure no longer erroneously flags an enumeration-type declaration as unused when the enumeration type's literals are referenced elsewhere. (PRS number 9123690-0134-5)

Procedures Ada.Withdraw, Install_Unit, Install_Stub, and Make_Separate now require only that the executing user have Create access to the enclosing world or view to enter the unit or stub into the library listing. This change fixes a problem introduced in D_12_5_0 in which Owner access was required. (PRS number 9123690-0171-3 and CSR number 5841)

Procedure !Commands.Dependents.Find now releases actions that are no longer needed. (PRS number 9723630-0120-7)

A problem has been fixed so that an attempt to incrementally edit a statement that depends on another statement that has been incrementally demoted itself no longer fails. (PRS number 6265678-Sier-Geb)

The Environment now demotes units in the correct order when:

- An attempt is made to incrementally demote a type

- That type is used in a procedure's parameter list
- Both that procedure and the original type are used as parameters in a generic instantiation

(PRS number 9123690-0161-1)

A problem has been fixed so that the following valid incremental insertions now are accepted:

```

package P is
  type T is private;
  type A is array (1 .. 10) of T;  -- insert 1
private
  type T is new Integer;
  X, Y : A;                       -- insert 2
  B : Boolean := X < Y;           -- insert 3
end P;

```

(PRS number 9924694-Mago-Sdj)

7.3. Changes to Debugging

Function `Debug_Tools.Get_Exception_Name` now correctly returns the fully qualified name of the exception, regardless of the setting of the `R1000_Cg.Retain_Delta1_Compatibility` library switch. In previous releases, if this switch was set to `False`, the function returned a string such as `<Exception: Unit =-35983540864679680, Ord = 0>`. (PRS numbers 12729-Star and 9123690-0160-7 and CSR number 4626)

7.3.1. Quitting While the Debugger is Running

In previous releases of the Environment, if your session was terminated—for example by `Quit` or `Operator.Force_Logoff`—while a job under the control of the debugger was running, a deadlock could occur. The symptoms of this deadlock could be any combination of the following:

- Job 4 stuck in `Awaiting_Entry`
- Jobs stuck in `In_Fa_Queue`
- Jobs hanging on `Quit`

Most of these problems have been fixed. If, however, a job hangs during a `Quit`, it is now always possible to use `Job.Kill` to kill all debugger jobs, causing the system to resume normal operation.

(PRS numbers 9123690-0159-0, 9123690-0159-1, 9123690-0172-6, and 9123690-0183-3 and CSR numbers 4740, 5142, 5917, and 6091)

7.4. Changes to Session and Job Management

`Job.Kill` has been modified so that it can kill editor jobs—even unallocated jobs—and find any tasks holding actions and abort them. The user must have operator capability to perform these tasks. (PRS numbers 5360143-Rati-Jls and 9123690-0166-0)

The remote-passwords facility has been changed so that *SYSTEM jobs, such as machine initialization and spoolers, now use the Operator password as the key for decrypting remote-passwords files. To instruct a *SYSTEM job to use a remote-passwords file, set the remote-passwords file for the job by calling Profile.Set_Remote_Passwords from within the job. For additional information, see package !Commands.Remote_Passwords in the new Session and Job Management (SJM) book or in online help. (PRS number 9123690-0166-5)

Special names such as "<IMAGE>" can no longer be inserted either directly into searchlists or in indirect files named in searchlists. Be aware, however, that if you have a special name already in a searchlist and if you attempt to delete that searchlist or searchlist entry, the system may crash, as it did in previous releases. (PRS number 9123690-0138-2)

The Program.Change_Identity command now delays when it encounters an illegal username and password pair. The length of the delay increases with each illegal encounter. (PRS number 3289131-Gato-Bas)

When logging on to the system in previous releases, if you entered a valid username and an invalid password and if the correct password associated with that username had expired, the password-expiration warning was emitted. The password-expiration warning implies that the username was correct. This situation has been fixed so that no information is given when either the username or password is invalid. (PRS number 4289272-Voya-Cmd)

7.5. Archive Changes

The comments in the specification for package Archive have been improved. In particular, the comments now include all supported options for the Archive.Copy, Archive.Restore, and Archive.Save procedures. (PRS number 9123690-0131-8)

7.5.1. Copying Ada Units with Subobjects

The Archive.Copy, Archive.Save, and Library.Copy procedures now copy non-Ada subobjects of Ada units that are not managed by one of Rational's layered products as if they were Ada subunits. These subobjects are sometimes called *associated files* and indicated in library images by brackets (<>). A recursive copy of a parent unit now will pick up the subobject, but a nonrecursive one will not. In previous releases of the Environment, non-Ada subobjects of Ada objects were not copied as part of a recursive copy operation.

7.5.2. Archiving Library Switch Files

The Archive.Copy and Archive.Save procedures have been changed so that archiving a library switch file now archives the complete switch file. (In previous releases, only nondefault values were archived.) As a result, restoring a D_12_6_5 switch file to a system running a previous release will result in messages of the form: **Switch does not exist**. These messages result from the fact that new switches exist in the D_12_6_5 switch file that do not exist in switch files from the previous release. The messages are for your information only; the Archive operation will complete successfully. For information about the new switches, see section 7.7.

7.6. Compilation Changes

Procedures in package `!Commands.Compilation` now print correct symbols for log messages. In particular, commands no longer use the note symbol (`--`) for messages that should be auxiliary (`:::`). For information about log-message symbols and their meanings, see the documentation for `!Tools.Profile.Msg_Kind`. (PRS number 1000002-Sier-Geb)

A problem has been fixed in which compilation operations sometimes would hang—without generating an error message—if the program library was locked. In particular, if you executed `Definition ("<*Program_Library>")` to display a world's program library, the operation would lock the program library. If you attempted to perform a compilation operation while the program library was locked, the compilation operation would hang. In D_12_6_5, compilation operations now time out and produce an error message indicating that the program library is locked. (PRS number 2949267-Etoi-Hts)

7.6.1. Installing Units

Note: The D_12_6_5 release of the Environment contains a number of changes to adhere more closely to the standards defined in the Reference Manual For the Ada Programming Language (LRM). In certain cases, these changes can cause source code that installed in the past to install no longer. Existing installed or coded units are not affected by these changes.

Promoting a unit to the installed state for the first time now requires only that the executing user have `Create` access to the enclosing world or view. This change fixes a problem introduced in D_12_5_0 in which `Owner` access was required. (PRS number 9123690-0171-3 and CSR number 5841)

Semantic checking has been fixed so that the Environment no longer allows a private type to be used as a generic formal parameter before the type's completion in the private part; this way of using a private type is not legal in Ada (see LRM 7.4.1 (4)). (PRS number 9123690-0128-9)

In accordance with LRM 13.4.(8), semantic checking now allows implementation-dependent attributes to be used in record representation clauses. (PRS number 1000000-Sier-Geb)

Semantic checking has been improved so that an extra nonterminal (`[expression]` prompt) used as a record aggregate is flagged now as an error. In previous releases, an error was not generated until code was generated. (PRS number 9123690-0128-6)

Semantic checking now properly supports the renaming of subprograms. In previous releases, problems were encountered if the renamed subprogram had a parameter for which the default value was an expression that could be misinterpreted as a predefined type. (PRS number 9123690-0156-3 and CSR number 4845)

In accordance with LRM 13.1 (7), semantic checking no longer allows the following source code to install (PRS number 1140702-Gato-Gbd):

```
package Csr5310 is
  type Type_1 is
    record
      Q_Addr_2 : Integer;
    end record;
```

```

Forcing_Occurrence : constant Type_1 := (Q_Addr_2 => 0);
-- The compiler detects the violation against LRM 13.1 (7) if the
-- object is not constant.

for Type_1 use
  record at mod 2;
    Q_Addr_2 at 2 range 0 .. 4 * 8 - 1;
  end record;

end Csr5310;

```

A problem has been fixed in which performing semantic checking more than once on a single unit results in the disappearance of the error underlines. This problem has been fixed so that the same error underlines reappear each time the unit is semanticized. (PRS number 9123690-0157-9 and CSR number 5005)

Both the R1000 compiler and the Mc68020_Bare Cross-Development Facility (CDF) (Rev5_1_1) now flag as an error any renaming of a record component in which the record object is returned as the result of a function call (see LRM 8.5 (4)). Attempts to install subprograms containing such renames now fail with the message **Expanded name name is not a renamable object.** (PRS number 9123690-0169-8)

Semantic checking now allows the following to be applied to an enumerated type:

- An enumeration representation clause that starts with negative values
- A size clause that comes before the enumeration representation clause

(PRS number 9123690-0180-7 and CSR number 4844)

In accordance with LRM 3.2 (9) and 3.3.2 (2), semantic checking no longer allows you reference a type mark that is not a simple name. (PRS number 9123690-0152-8 and CSR number 4735)

Messages have been improved so that if you reach the architectural limit for the size of a DIANA node (2**23 bytes), the coding attempt now fails gracefully with the message **Promote Failed - Other Error** in the message window and with the message **!!! Action.Acquire Invalid_Class nnnnnnnnnn for instance nnnnnn machine id 1** on the console error log. (PRS number 8277786-Voya-Bas)

Error messages have been improved so that if you attempt to declare a record with more than 256 fields, the compiler now generates a message such as **Field F in record type R exceeds the number of fields implementation limit of 256**, and the offending unit fails to install. (PRS number 9123690-0142-3)

Messages have been improved so that if the physical representation of an enumeration type does not have a range of Integer'Range (as specified in LRM 3.4.1), semantic checking fails with a message such as ***UNIVERSAL_INTEGER* ... Constant expression not in Integer'Range.** (PRS number 5700190-Etoi-Tnu)

The error message generated when a body fails to install has been punctuated for easier readability. Specifically, quotation marks have been added around the declaration of the body. The message now appears similar to the following: **1: ERROR The spec for "package body FOO is " could not be installed.** (PRS number 10270-Shei-Swb)

7.6.2. Coding Units

The R1000 code generator now successfully codes any package that declares a constant of a subtype such that the subtype is:

- Externally declared
- Of an integer type that uses an explicit conversion for a range constraint

In previous releases, attempts to code these packages failed with `Constraint_Error`. (PRS numbers 9123690-0145-5 and 306859-Cook-Swb)

Messages have been improved so that if you attempt to code a unit that instantiates a generic whose original declaration has been corrupted, you no longer get an internal error. The primary cause of this kind of corruption is using `Program_Library_Maintenance.Build` on Delta-1-style code views. (CSR number 5215)

As documented in section 1.2.10 of Appendix F for the R1000 Target of the LRM, the R1000 code generator now ignores `pragma Private_Eyes_Only` for generic units. (PRS number 9123690-0151-1 and CSR number 4723)

A problem has been fixed in which the R1000 code generator failed with a `Constraint_Error` when coding a very large task-entry family. The code generator now codes the task-entry family successfully and generates the following warning: **1: WARNING Declaring a very large entry family (may cause STORAGE_ERROR): E1.** (PRS number 4967545-Voya-Phil)

You now can control whether the Environment allows you to promote to the coded state Ada units that contain `[statement]` prompts. For details, see the description of the new `Semantics.Reject_Statement_Prompts` library switches in section 7.7. (PRS number 9123690-0143-2)

The compiler now handles sparse types of discriminants properly. In previous releases, when sparse types were used as discriminant types, the compiler would not work correctly in assigning the variant part of the record, and the incorrect *case* alternative would be chosen. (PRS number 9123690-0166-6 and CSR number 5666)

7.6.3. Loading and Executing Units

The `Compilation.Load` procedure has been changed so that it always builds the specification of the loaded main program (`Load_Proc` or `Load_Func`) from the specification of the unit specified to the `Load` procedure. If a unit body is specified, the unit specification associated with that body is used. This practice ensures that the loaded main program obtains its context clause from the unit specification. If the `Load` procedure used the context clause from the unit body, and that unit body depended on the specification's context clause, the procedure would fail with semantic errors. (PRS number 6821610-Stan-Deg)

Attempting to execute a derived procedure inherited by instantiation in a generic that is itself instantiated no longer generates an `Internal_Error` (unsatisfiable import requirement) if the generics are in different subsystems. (PRS number 9123690-0171-8 and CSR number 5650)

Declaring a private derived type of `!Tools.Unbounded_String.Variable_String` no longer causes an unhandled exception at runtime when the unit that declared the derived type is coded with the `R1000_Cg.Retain_Delta1_Compatibility` library switch set to `True`. (PRS number

9123690-0168-1 and CSR number 5723)

If an unhandled exception is encountered at the outermost level, programs coded with the R1000_Cg.Retain_Delta1_Compatibility switch set to False no longer generate an Internal_Error. (PRS number 12729-Star)

7.6.4. Program Libraries

The Program_Library_Maintenance.Verify procedure and function have no valid options. The comments in the specification for these subprograms now state this explicitly. The full pathname for these subprograms is:

```
!Tools.Program_Library' Spec_View.Units.Program_Library_Maintenance.Verify
```

(PRS number 9123690-0131-7)

7.7. Changes to Library Switches

D_12_6_5 introduces two new library switches and changes the default value of one.

Because of the addition of these two switches, copying library switch files from D_12_6_5 to previous releases generates error messages indicating that certain switches do not exist. These errors can be ignored; the copy operation will complete successfully. This behavior is also mentioned in section 5.

D_12_6_5 also includes new online help for several library switches. See section 8.3 for a list of these switches.

7.7.1. New Switch: Semantics.Limit_Semantic_Messages

This switch controls the number of semantic messages (errors and warnings combined) that can be attached to an Ada image. Those messages with numbers greater than this limit will be discarded. Discarded messages can be regenerated by increasing the limit specified in this switch and resemanticizing the image. The default is 200 messages.

Note that generation of an extremely large number of messages for a single unit can cause semanticizing to take an excessive amount of time. Setting this switch to a lower number may speed semantic checking.

(PRS number 9123690-0136-8 and CSR number 3956)

7.7.2. New Switch: Semantics.Reject_Statement_Prompts

This switch controls whether the Environment allows you to promote to the coded state Ada units that contain `[statement]` prompts. When False (the default), units with `[statement]` prompts can be coded; an error is generated only if an attempt is made to execute the prompt. When True, units that contain `[statement]` prompts cannot be coded. Setting this switch to True may cause code that was able to code in the past to code no longer but will not affect units that are coded already.

(PRS number 3336165-Sher-Jst and 9123690-0143-2)

7.7.3. Change to Default Value of R1000_Cg.Retain_Delta1_Compatibility Switch

The D_12_1_1 release of the Environment contained a major revision of the R1000 compiler. The D_12_1_1 version of the compiler has been validated under releases 1.10 and 1.11 of the Ada validation suite (ACVC 1.10 and 1.11) and includes significant new features, improved generated-code quality, and numerous bug fixes.

Certain of the compiler's features and bug fixes are available only when explicitly enabled by the Retain_Delta1_Compatibility library switch. This is because enabling these features and bug fixes causes the compiler to produce code that is incompatible with the code generated by versions of the compiler prior to D_12_1_1.

D_12_6_5 changes the default value of this switch to False, allowing the compiler to take advantage of the features of the D_12_1_1 compiler unless otherwise specified. This change affects all new switch files created using Switches.Create or Switches.Edit. It also affects all new systems. Installation of D_12_6_5 preserves the current settings in switch files so that work in progress should *not* be significantly affected unless new switch files are created.

Note that units coded with one setting of the switch may reference units coded with the switch set the other way. However, there is one restriction: when a main program is prelinked, a check is made to ensure that each unit in a spec view was coded with the same switch setting as the corresponding unit in the load view. The compatibility checker also makes this check when checking load view/spec view compatibility.

To avoid confusion, it is recommended that you use the same switch setting for all units in a given view. Furthermore, to prevent problems with spec view/load view compatibility, it is recommended that you use the same switch setting for all views in a given subsystem.

There is no need for you to convert existing code that was compiled with the Retain_Delta1_Compatibility switch set to True. However, if you choose to convert existing code, you must set the switch to False in each of the affected libraries or views, demote the units to the installed state, and then repromote them to the coded state. Code views cannot be recompiled and therefore should be destroyed.

For more information about the effects of this switch, see the new Library Management (LM) book and the D_12_1_1 release information. (Release information can be found online in !Machine.Release.Release_Notes.)

7.8. Changes to Other Library Management Facilities

Commands in package !Commands.Access_List now print log messages indicating when they have completed and whether they completed successfully. Commands in package Access_List now also print the correct symbols for log messages. For information about log-message symbols and their meanings, see the documentation for !Tools.Profile.Msg_Kind. (PRS number 1000001-Sier-Geb)

Comments have been added to the specification for the Library.Copy procedure to more accurately describe the default value for the Volume parameter. For additional information, see the documentation for Library.Nil in the new LM book and in online help. (PRS number

9123690-0150-4)

The `Library.Move` procedure no longer moves any part of an Ada unit—either the specification or body—without sufficient access to the destination library. In previous releases, if you attempted to move a unit specification and body pair in a single operation, the `Move` command would fail to copy the unit specification but would successfully copy the unit body. (PRS number 2679572-Roge-Srp)

The `Library.Create@` procedures now require that the executing user have only `Create` access to the enclosing world or view. This change fixes a problem introduced in `D_12_5_0` in which `Owner` access was required. (PRS number 9123690-0171-3 and CSR number 5841)

`Procedures !Commands.Links.Edit` and `Links.Display` now require `Owner` access to the world for which the links are being edited or displayed. In previous releases, only `Read` access was required. (PRS number 5367576-Blut-Smp)

7.9. Changes to Device Input and Output

The `Sequential_Io.Read` procedure and the `Polymorphic_Sequential_Io.Read` procedure have been fixed so that they no longer raise `!Lrm.System.Operand_Class_Error` or `Nonexistent_Page_Error` when an attempt is made to read the first character in a file. (CSR number 6016)

7.10. Changes to Printing

The `Abbreviations.Print` command now handles print requests that specify multiple Ada units in the same way that it handles a print request for a single Ada unit. That is, when multiple objects are specified to the `Object_Or_Image` parameter and when *all* of those objects are Ada units, the `Print` command now processes the `Highlight_Reserved_Words_For_Ada` parameter. If any of the objects specified are not Ada units, this parameter is ignored. (PRS number 9123690-0168-2 and CSR number 5773)

The `Queue.Print` command now forms valid VMS filenames for print requests. In previous releases, if you printed a file to a queue that was physically located on another machine and if that queue was FTP-based, then the filename formed for FTP transfer was invalid for VMS. When the print spooler tried to dequeue the request, several errors were logged and the queue was disabled. (PRS number 9123690-0168-8 and CSR number 5710)

7.11. Changes to System Backups

The default parameter for the `System_Backup.Backup` procedure has been corrected so that the procedure now semanticizes successfully in command windows. Specifically, the value of the `Variety` parameter has been changed from `Full` to `System_Backup.Full`. See the new `System Management Utilities (SMU)` book or online help for additional information about this command. (PRS numbers 1711335-Gato-Gbd and 9123690-0135-8)

The process of recovering a system backup has been changed so that it always prints some form of diagnostic message whenever it detects a situation in which it cannot complete the restoration. Furthermore, additional log messages indicate the beginning and completion of major stages in the restoration process. In previous releases, when a problem occurred while a

backup was being restored, the process sometimes would hang without generating any messages. (PRS number 9123690-0133-9)

The Verify_Backup procedure has a new parameter, Wait_Until, that allows you to specify the amount of time the procedure must wait before starting. The tape should be mounted immediately, but Verify_Backup does not start reading the tape until Wait_Until has expired. The fully qualified pathname for Verify_Backup is:

```
!Commands.System_Maintenance' Spec_View.Units.Verify_Backup
```

(PRS number 9123690-0158-6)

7.12. Changes to the Diagnostic File System (DFS)

7.12.1. Renamed Procedures

D_12_6_5 changes the specification of package !Machine.Dfs'Spec_View.Dfs. Specifically, the Dfs_ prefix has been removed from the names of the procedures introduced in D_12_5_0:

- Procedure Dfs_Copy is now procedure Copy
- Procedure Dfs_Directory is now procedure Directory
- Procedure Dfs_Display is now procedure Display
- Procedure Dfs_Make_Copy is now procedure Make_Copy
- Procedure Dfs_Open is now procedure Open
- Procedure Dfs_Read is now procedure Read

Several new procedures also have been added to package Dfs; see section 6.3.

7.12.2. Analyze_Crashdump Prompts for Input

Procedure Analyze_Crashdump by default now queries for input in an Environment I/O window, rather than from within the log file. The full pathname for Analyze_Crashdump is !Machine.Dfs'Spec_View.Units.Analyze_Crashdump. (PRS number 9123690-0165-6 and CSR number 5588)

7.13. Changes to Crash Analysis

7.13.1. Changes to the Crash Analysis Procedures

The D_12_6_5 release includes enhancements to the process of analyzing R1000 crashes. In particular:

- Much of the process has been automated so that the system will capture as much information as possible and often will reboot automatically.
- A series of questions and menus have been added to the crash analysis process that:
 - Capture system-crash state

- Analyze system hardware

For complete details, see Appendix B, "Diagnostic Crash Procedures."

(PRS numbers 7483032-Gato-Jls and 7733953-Gato-Vnv)

7.13.2. Change to Crashdumps and Tombstones

A problem with incomplete crashdump and tombstone files that was introduced in D_12_5_0 has been fixed. Crashdumps and tombstones now work correctly, and the trace and crash reason appear correctly in the error logs.

This fix was released originally in DFS release D_12_5_0A (D2.2A).

7.14. Changes to Token Policies

To simplify token administration, two changes have been made to token management for D_12_6_5:

- Environment-session tokens now can be transferred between Series 400 and 300 machines.
- The Rational X Interface (RXI) and Windows Interface (RWI) no longer require special product tokens. These products now are bundled with the Environment and are restricted only by Environment session tokens.

Rational's other token-management facilities have been tightened to more rigorously account for other product tokens. Although the number of tokens has not been changed for any particular site, system, or product, you may discover that you are running out of tokens more frequently than in the past.

7.14.1. Transferring Tokens between Series 400 and 300

Beginning with D_12_6_5, Environment tokens may be transferred between Series 400 and Series 300 R1000s. That is, Full Session tokens and Login tokens now can be transferred between machines of different series. This change is intended to make it easier to maintain sites that have both Series 400 and 300 machines and to enable those sites to move tokens in response to project needs.

To make this token transfer possible, the !Commands.System_Maintenance'Spec_View.Units.Accept_Tokens and Donate_Tokens procedures have been changed.

For example, assume that you have a Series 400 with twelve Full Session tokens and a Series 300 with six Login tokens. To transfer two tokens from the Series 400 to the Series 300, perform the following steps:

- On the donating Series 400, execute the Donate_Tokens command, specifying Full Session for the product:

```
Donate_Tokens (Product          => "Full Session",
               Donation          => 2, -- number to donate
               Resulting_Remote_Count => 8, -- original 6 plus 2 new
```

```

Remote_Machine_Id    => Id of the accepting 300,
Remote_Site          => ">>SITE ID<<",
Response             => "<PROFILE>";

```

The output generated by this command indicates that two tokens have been donated, resulting in ten tokens remaining on the current Series 400. It also supplies the completed Accept_Tokens command to be run on the accepting system:

```

92/05/18 12:47:03 ::: [Donate_Tokens ("Full Session", 2, 8, 345678,
92/05/18 12:47:03 ... "XYZ000001", "<PROFILE>")].
92/05/18 12:47:03 +++ Donation of 2 tokens for Full Session has resulted
92/05/18 12:47:03 ... in 10 tokens.
92/05/18 12:47:03 +++ Running Snapshot daemon.

```

Run the following on the remote machine:

```

Accept_Tokens (Product      => "Full Session",
                Donation     => 2,
                Resulting_Count => 8,
                Code         => "16A8AE0E1E838FB5");

```

```

92/05/18 12:47:15 ::: [end of Donate_Tokens].

```

- On the receiving Series 300, execute the Accept_Tokens command generated in the previous step:

```

Accept_Tokens (Product      => "Full Session",
                Donation     => 2,
                Resulting_Count => 8,
                Code         => "16A8AE0E1E838FB5",
                Authorization => "",
                Response     => "<PROFILE>");

```

Upon acceptance of the token(s), the Environment automatically changes the token kind to Login.

Note that the site ID must be the same on both the donating and accepting machines. (You can use Show_Site to display the site ID for a machine.) Note also that the Donate_Tokens and Accept_Tokens commands must be run on the same day.

7.14.2. Token Changes to RXI

Beginning with D_12_6_5, the most recent release of RXI (10_9_0) is delivered with the Environment. Although RXI must still be separately installed, it is now authorized on all D_12_6_5 systems. Thus, product tokens are no longer required. This change is in response to the evolution of RXI as one of the primary access mechanisms to the Environment.

All problem reports concerning the improper consumption of tokens by RXI now are considered closed. (PRS numbers 2604297-Rati-Pbk, 4123589-Blut-Jgp, 6398339-Blut-Med, and others)

For more information about RXI and D_12_6_5, see section 6.6.1.

7.15. Miscellaneous Changes to System Management

You now can set the timeout value for logging in through the console command interpreter as part of machine initialization. The default setting is 60.0 (1 minute). To change the setting, edit the specification for `!Machine.Initialization.Rational.Servers` in either of the following ways:

- Change the value for the `Console_Timeout` parameter directly
- Insert a notation such as the following at the end of the specification:

```
--| Parameter Console_Timeout => "2 * 60.0"
```

The new value takes effect the next time you reboot your machine.

The `!Machine.Initization.Rational.Terminals` procedure now generates messages that properly reflect the parameters of the `Terminal.Set_Login_Disabled` procedure, which is called by `Rational.Terminals`. (PRS number 6242437-Blut-Vo)

A problem has been fixed in which the software flow control on port 16 failed for Series 300C, 300S, 400C, and 400S systems. (PRS numbers 9123690-0164-4 and 9123690-0165-8 and CSR number 5559)

Procedure `Log_Reader.Load_Logs` has been fixed so that it no longer loads the error logs for the current day unless the current day is between `Start_Time` and `Stop_Time`. The full pathname for `Load_Logs` is `!Tools.System_Availability'Spec_View.Units.Log_Reader.Load_Logs`. (PRS number 9123690-0173-8 and CSR number 5990)

The Action Manager now detects and recovers gracefully from certain types of invalid data. (PRS number 9123690-0152-5 and CSR number 4741)

The comments in the specification for the `System_Uilities.Get_Board_Info` function have been improved. In particular, they now explain that board number zero (0) is IOA ONLY in a Series 100 and that the number one (1) board is SYS/IOC, which is the SYSBUS board for a Series 100 and the IOC board for a Series 200, 300, or 400. For additional information, see package `System_Uilities` in the updated System Management Utilities book (Volume 10 of the ERM) or in online help. (PRS numbers 9123690-0155-0 and 9123690-0155-1)

The `Check_Universe_Acls` procedure has been fixed so that it now reports the correct access information for `!Machine.Initialization`. The full pathname for `Check_Universe_Acls` is `!Commands.System_Maintenance'Spec_View.Units.Check_Universe_Acls`. (PRS number 9176782-Gato-Gbd)

The directory daemon has been enhanced to perform more consistency checking. Whenever a potential problem is detected, the directory daemon now attempts to log a meaningful error message and to proceed with the collection.

A problem has been fixed in which the directory manager sometimes would detect an inconsistency erroneously when the version of an object was being changed (that is, when an existing object was being edited). This error would cause the following to be logged in the system's error log:

```
Directory inconsistency exception in Object_Perform
Calling task (16#xxxxxxx#) will be stopped in wait service
```

(CSR numbers 5260 and 5261)

7.16. Changes to CMVC

The `Cmvc.Initial` procedure has been fixed so that when it is used to create a new path in an existing subsystem, it no longer looks at the `System_Object_Type` and `Create_Load_View` parameters unless they are needed. Now `Cmvc.Initial` will ignore the default value of the `System_Object_Type` parameter if the subsystem already exists. If the subsystem exists and is either a `Combined` or `System` subsystem, `Cmvc.Initial` will ignore the `Create_Load_View` parameter because there is no choice of which kind of view to create. (PRS number 7311575-Wash-Jim)

`Cmvc.Copy`, `Make_Path`, `Make_Subpath`, and `Make_Spec_View` now accept a single space as the null string in the `View_To_Import` parameter. Furthermore, log messages have been corrected so that if an invalid value is entered for the `View_To_Import`, the resulting message now includes the closing parenthesis. (PRS Number 2809652-Clem-Jim)

`Cmvc.Copy`, `Make_Path`, `Make_Subpath`, `Release`, and `Make_Spec_View` now properly copy Ada units from any directory in the subsystem, not just those units in the `Units` directory. In particular for `Make_Spec_View`, these sibling directories are treated like the `Units` directory and only those items specified in the `State.Exports` file are copied. This file is resolved in the context of the `Units` directory; to include a unit outside this directory you must provide an appropriate relative pathname—for example, `^.Subdirectory_Name.Simple_Name`—or a fully qualified pathname. (PRS numbers 0-0358-2 and 9123690-0156-6 and CSR number 4971)

To create objects in a world or subsystem view, you now need only `Create` access to that world or view. This change fixes a problem introduced in `D_12_5_0` in which `Owner` access was required. (PRS numbers 9123690-0171-3 and CSR number 5841)

7.17. Update to !Tools.Math_Support

An update to the `!Tools.Math_Support` subsystem is included in the `D_12_6_5` release.

- In package `Generic_Primitive_Functions`:
 - The `Scale` function has been corrected for very large and very small scales.
 - The `Remainder` function has been improved in cases where arguments are several orders of magnitude apart.
 - The `Truncate` function has been simplified.
- The implementation of package `Generic_Elementary_Functions` has been replaced by a package with higher accuracy. The new implementation was developed by Peter Tang and Kenneth Dritz, from the Argonne National Laboratory, with minor adjustments for the R1000. This change fixes several errors in domain checking and a major error in the code for `Tangent`.

For more details, see the online information in:

`!Tools.Math_Support- .R1000_Code1_1_0.Units.Release_Note`

7.18. Miscellaneous Changes

7.18.1. Change to Library_Object_Editor

The specification for procedure !Tools.Library_Object_Editor.Display has been changed to match the Environment's implementation. In particular, the In_Place parameter has been added to the procedure's specification:

```

procedure Display (Library      : Directory.Object;
                   Selection    : Directory.Object := Directory.Nil;
                   In_Place    : Boolean          := False;
                   Force_Redraw : Boolean          := False);

```

The In_Place parameter specifies whether the current window frame should be used. The default is False.

7.18.2. Change to Code_Segment_Object_Editor

The specification to procedure !Tools.Code_Segment_Object_Editor.Edit has been changed to match the Environment's implementation. In particular, a parameter, The_Offset, has been added:

```

procedure Edit (The_Segment : Integer;
                The_Offset  : Integer := 0;
                In_Place   : Boolean := False);

```

Treating a segment as a file, The_Offset allows "editing" from a place other than the beginning of the segment.

8. Documentation

8.1. New Printed Documentation

The D_12_6_5 release of the Environment is accompanied by three updated volumes of the *Rational Environment Reference Manual* (ERM):

- Volume 4: Session and Job Management (SJM)
- Volume 5: Library Management (LM)
- Volume 10: System Management Utilities (SMU)

These books have been significantly rewritten and feature:

- A new, easier-to-read format
- Greatly expanded "how to use" information, in the form of:
 - New Key Concepts sections for each book
 - Extensive introductions to major packages

- A quick reference for parameter-value conventions (naming conventions) at the end of each book
- Two previously undocumented packages, Remote and Remote_Passwords (see SJM)
- Corrections and clarifications to existing information

Each customer site should receive one copy of each volume for every system under an active support contract. If you have not received your new documentation or if you would like to order additional documentation, please contact your Rational representative.

8.2. New Online Documentation

The D_12_6_5 release of the Environment includes new online help for all the packages documented in the SJM, LM, and SMU books:

- Session and Job Management (SJM):

Job	Log
Profile	Program
Remote	Remote_Passwords
Search_List	What
- Library Management (LM):

Access_List	Access_List_Tools
Archive	Compilation
File_Uilities	Library
Links	Switches
Xref	
- System Management Utilities (SMU):

Daemon	Message
Operator	Queue
Scheduler	System_Backup
System_Uilities	Tape
Terminal	

D_12_6_5 also includes a revision of the online information about machine initialization in !Machine.Initialization.Guide_To_Machine_Initialization. You can obtain this information by requesting online help for !Machine.Initialization.

To obtain online help for a particular package or command, use the What.Does command, supplying all or part of the name of the package or command in question.

8.3. New Help for Switches

D_12_6_5 includes new online help for many switches. To obtain help for a particular switch, place the cursor on the switch in question and execute !Commands.Common.Explain (generally available through the [Explain] key or the [Object] - [?] key combination).

8.3.1. Session Switches

The following session switches have new help:

Command_Cg.Check_Compatibility	Command_Cg.Elaboration_Order_Listing
Command_Cg.Page_Limit	Command_Cg.Retain_Delta1_Compatibility
Command_Cg.Terminal_Echo	Profile.Log_File
Profile.Reaction	Session_Ftp.Remote_Directory
Session_Ftp.Remote_Machine	Session_Ftp.Remote_Root
Session_Ftp.Remote_Type	

8.3.2. Library Switches

The following library switches have new help:

Directory.Create_Internal_Links	Ftp.Remote_Directory
Ftp.Remote_Machine	Ftp.Remote_Root
Ftp.Remote_Type	R1000_Cg.Delta1_Code_View_Compatibility
R1000_Cg.Elaboration_Order_Listing	R1000_Cg.Page_Limit
R1000_Cg.Retain_Delta1_Compatibility	R1000_Cg.Terminal_Echo
Semantics.Reject_Inevitable_Exceptions	Semantics.Reject_Undefined_Pragmas

Appendix A

Guide to Machine Initialization

D_12_5_0 changed the software that is automatically executed by the Environment during the boot process. These changes apply to D_12_5_0 and all subsequent releases (including D_12_6_5). In particular, the software was reorganized to make it easier to:

- Install layered products
- Distinguish Rational-specific, site-specific, and machine-specific customizations
- Configure a network of printers for large sites

The following subsections describe the D_12_5_0 mechanisms for initializing an R1000. Change bars indicate sections that have been added to this document since D_12_5_0; the initialization mechanisms themselves have not changed. You can read this document online by traversing to `!Machine.Initialization.Guide_To_Machine_Initialization` or by executing `What .Does ("!Machine.Initialization")`.

See also the "Installation Procedure" for specific steps to convert any existing initialization software to the D_12_5_0 mechanisms.

A.1. Overview

Each time an R1000 is booted, software is executed that initializes layered products, sets various system parameters (for example, disk-collection thresholds and snapshot intervals), starts servers, enables terminals, and so on.

In Environment releases prior to D_12_5_0, the boot process automatically executed `!Machine.Initialize`, which in turn executed a family of procedures (with names of the form `!Machine.Initialize_@`).

In D_12_5_0 and subsequent releases, `!Machine.Initialize` is no longer used. Instead, all system-initialization software resides in the world `!Machine.Initialization`, which is structured as shown:

```
!Machine.Initialization : Library (World);
  Local      : Library (World);
  Rational   : Library (World);
  Site       : Library (World);
  Start      : Ada (Load_Proc);
```

The D_12_5_0 (or later) boot process automatically executes the Start procedure, which in turn executes all the procedures that reside in (or are referenced in) the Local, Rational, and Site worlds:

- The world Rational contains or references software that initializes Rational products and provides standard settings for many system parameters. Users should not modify the objects in this world.
- The worlds Site and Local provide a place for system managers to put objects that customize the initialization process. Such objects can be used to override various standard system parameter settings, to initialize customer-written applications, and to specify terminal and printer configurations:

- The world Site is intended for customer-written objects that are common to two or more machines at a given site.
- The world Local is intended for customer-written objects that apply only to the current R1000.

The following subsections give more detail about these worlds.

A.1.1. World !Machine.Initialization.Rational

The Rational world contains objects supplied by Rational that perform basic initialization services for the current R1000. These objects include:

- Loaded main procedures that are executed by !Machine.Initialization.Start whenever the system is booted.
- “_Start” files that reference procedures located elsewhere in the Environment. These are text files whose names end with `_start`; the procedures they reference are executed by !Machine.Initialization.Start.

On a typical system, this world contains objects such as the following:

```
!Machine.Initialization.Rational : Library (World);
Clean_Machine_Temporary      : C Load_Proc;
Cross_Compilers              : C Load_Proc;
Design_Facilities            : C Load_Proc;
Dtia                          : C Load_Proc;
Finish_Install               : C Load_Proc;
Log_Previous_Outage_Start    : Text;
Mail_Start                    : Text;
Network                      : C Load_Proc;
Parameters                   : C Load_Proc;
Printers                     : C Load_Proc;
Servers                      : C Load_Proc;
Teamwork_Interface           : C Load_Proc;
Terminals                    : C Load_Proc;
```

The procedures that are supplied or referenced in this world:

- Perform cleanup and compaction
- Initialize the installed Rational products, such as CDFs, RDF, Rational Networking, and so on
- Initialize servers, including the archive and FTP servers
- Set standard values for various system parameters, such as the medium-term scheduler, snapshot intervals and warnings, disk-collection thresholds, and so on
- Initialize terminals and printers according to user-specified requirements (given in files in the Site and Local worlds)

For more specific information, you can browse the comments in each object in this world.

A.1.2. Worlds !Machine.Initialization.[Site,Local]

The Site and Local worlds are where system managers can create objects to control sitewide or machine-specific initialization and configuration. These objects may include:

- Ada procedures that supplement or override the basic initialization services performed by objects in the Rational world. All procedures in the Site and Local worlds are executed by !Machine.Initialization.Start each time the system is booted.
- “_Start” files that reference procedures located elsewhere in the Environment. These are text files whose names end with `_start`; the procedures they reference are executed by !Machine.Initialization.Start. Using a “_Start file” is equivalent to calling Program.Run or Program.Run_Job to execute the referenced procedure. (See section A.3.2.)
- *Configuration files* that tell the Environment how to enable and configure ports for login and ports for printing. These are text files that are read by two of the procedures executed by !Machine.Initialization.Start. A default file for enabling login ports is created in the Local world during installation. (See sections A.4 and A.5.)
- Text files that tell the Environment how to initialize layered products such as the Cross-Development Facility or the Rational Design Facility. (See the comments in the specifications of the Cross_Compilers and Design_Facilities procedures in world !Machine.Initialization.Rational.)

At most sites, system managers will use the Site and/or Local worlds as a place for procedures (or “_Start” files) that:

- Set password policy
- Set login limits
- Start server programs for site-specific networking, databases, and applications (for example, a login monitor or network security server)

At some sites, system managers may need to use these worlds for procedures that:

- Provide nonstandard daemon settings—for example:
 - The time at which daily and weekly clients run. (The standard times are 3:00 a.m. for daily clients and 2:30 a.m. for weekly clients.)
 - How often snapshots are taken. (The standard snapshot interval is every 30 minutes.)
 - Whether (and when) to send a snapshot warning, snapshot start messages, and snapshot finish messages. (The standard is to send a warning 20 seconds before the next snapshot and to notify users only when the snapshot has finished.)
 - The interval for daemon warnings. (The standard is to send a warning 2 minutes before the daily clients begin.)
 - Whether to send disk-collection threshold warnings. (The standard is to warn users when collection thresholds have been passed.)
 - What kinds of system log messages appear on the operator console. (The standard is to route only warning, problem, and fatal messages to the operator console.)
 - Whether clients should perform access-list compaction. (The standard is for all relevant clients to perform access-list compaction.)

- Provide customized disk-collection thresholds (see also section A.3.4). Note, however, that values set by the procedure in world Rational are calculated based on your disk configuration and should be sufficient; see your Rational technical representative if you want to use different thresholds.
- Provide nonstandard medium-term scheduler settings. (Use the Scheduler.Display command to see the standard settings.) Note, however, that values set by the procedure in world Rational should be sufficient; see your Rational technical representative if you want to use different settings.

A.2. Setting Up the Site and Local Worlds

If you have a new R1000, you can use the following guidelines to set up your Site and Local worlds:

1. Decide whether you need to provide any system customizations such as those listed in section A.1.2. Create the appropriate procedures and/or “_Start” files, using the hints given in section A.3.
2. Inspect the default Terminal_Configuration file that was created in the Local world during installation. This file enables ports 235 through 249 for login. Edit this file if you want to enable a different set of ports and/or specify further connection or communication information; see section A.4.
3. Decide whether to implement a printer-configuration mechanism to enable users to use !Commands.Abbreviations.Print and to facilitate the use of networked printers. Create the appropriate files; see section A.5.
4. If you have layered products such as the CDF or RDF, inspect the comments in the specifications of the Cross_Compilers and Design_Facilities procedures in world !Machine.Initialization.Rational. Create any files required for initializing these products (for example, a text file that registers an RDF customization).

If you are upgrading from a release prior to D_12_5_0, the Local world will already contain several objects that contain customizations:

- A Terminal_Configuration file is created automatically in the Local world, enabling the same set of ports that were enabled at the time of installation. This file also preserves any nondefault communication characteristics that were in effect for RS232 ports.
- The !Machine.Initialize_Site procedure is automatically moved into the Local world, as described in the “Installation Procedure.”

After your R1000 has been upgraded to D_12_5_0 or a more recent release, you can use the following guidelines to set up your Site and Local worlds:

1. Inspect the Initialize_Site procedure that has been copied into the Local world during installation. Edit this procedure to preserve any system customizations that are still appropriate. You may want to split this procedure into separate procedures and move appropriate procedures into the Site world. See the “Installation Procedure” for specific recommendations for handling this procedure. See also section A.3 for information about initialization procedures and “_Start” files.
2. Inspect the default Terminal_Configuration file that was created in the Local world during installation. Edit this file if you want to enable a different set of ports and/or specify

further connection or communication information; see section A.4.

3. Decide whether to implement a printer-configuration mechanism to enable users to use !Commands.Abbreviations.Print and to facilitate the use of networked printers. Create the appropriate files; see section A.5.
4. If you have layered products such as the CDF or RDF, inspect the comments in the specifications of the Cross_Compilers and Design_Facilities procedures in world !Machine.Initialization.Rational. Create any files required for initializing these products (for example, a text file that registers an RDF customization).

A.3. Hints for Implementing System Customizations

This section provides information about:

- Writing customized initialization procedures in the Site and Local worlds; see section A.3.1.
- Writing “_Start” files that reference procedures that reside in other Environment libraries; see section A.3.2.
- Controlling the order in which the customized initialization procedures and the “_Start” files are processed by the Start procedure; see section A.3.3.
- Customizing disk-collection thresholds; see section A.3.4.

A.3.1. Writing Customized Initialization Procedures

You can write procedures in the Site or Local worlds to implement system customizations such as password policies, system daemon settings, and so on. All procedures that appear in these worlds are executed by the Start procedure each time the R1000 boots.

Customized initialization procedures can contain calls to procedures in various standard Environment packages. Some useful packages include:

- Package Daemon, which contains procedures that schedule clients and warnings
- Package Operator, which contains procedures that set password policy and login limits
- Package Scheduler, which contains procedures that control the medium-term scheduler
- Package Program, which contains procedures that execute other programs

Note that:

- You do not have to provide initialization procedures for configuring login ports and printer ports; it is recommended that you use configuration files instead (see sections A.4 and A.5).
- You do not have to write an initialization procedure “from scratch” for customizing disk-collection thresholds; if you must customize these thresholds, it is recommended that you edit the sample initialization procedure provided in the Local world (see section A.3.4).

When writing customized initialization procedures:

- You can create separate procedures or put all calls in a single procedure. Separate procedures take longer to execute but make it easier to see what operations are being performed.
- You must not duplicate procedure names across the Rational, Site, and Local worlds.
- You can specify the relative execution order of procedures using annotations (see section A.3.3).

A.3.2. Using “_Start” Files to Reference Initialization Procedures

As an alternative to writing procedures directly in the Site and Local worlds, you can create “_Start” files in one or both worlds to reference customized initialization procedures that reside elsewhere in the Environment. “_Start” files are processed by the Start procedure each time the R1000 boots, and the procedures they reference are executed.

When writing “_Start” files:

- You must choose filenames that use the `_Start` suffix.
- You must not duplicate filenames across the Rational, Site, and Local worlds.
- You must use annotations to reference the procedure to be executed, as illustrated below.
- You may use annotations to control the order in which the Start procedure executes the referenced procedures (see section A.3.3).

Referencing a procedure in a world or directory: A “_Start” file with the following contents illustrates the basic set of annotations required to reference a procedure that resides in an Environment world or directory:

```
--|Procedure_Name Initialize_Routine
--|Procedure_Context !Commands.Example
--|Parameters Notify => "manager",
--|Parameters Effort_Only => False
```

- The `--|Procedure_Name` annotation specifies the name of the referenced procedure. If the procedure resides directly in a library, you supply the procedure’s simple name; if it resides in a package, supply the name in the form `Package_Name.Procedure_Name`.
- The `--|Procedure_Context` annotation specifies the Environment world or directory that contains the referenced procedure.
- Each of the `--|Parameters` annotations specifies the value to be used for one of the procedure’s parameters. Note that string values must be enclosed in quotation marks, and commas must be included to separate multiple parameters.

Referencing a procedure in a subsystem: A “_Start” file with the following contents illustrates how to reference a procedure that resides in an Environment subsystem. Note that you must replace the `--|Procedure_Context` annotation with the `--|Subsystem` and (optional) `--|Activity` annotations:

```
--|Procedure_Name Excelan_Boot_Server
--|Subsystem !Targets.Implementation.Motorola_68k_Download
--|Activity !Machine.Release.Current.Activity
--|No_Wait
```

- The `--|Subsystem` annotation specifies the name of the subsystem that contains the procedure. (When this annotation is specified, the `--|Procedure_Context` annotation is ignored.)
- The `--|Activity` annotation specifies the activity to be used to obtain the view name and construct the full pathname of the procedure. If you omit this annotation, `!Machine.Release.Current.Activity` is used.
- Note that `Excelan_Boot_Server` is a parameterless procedure; otherwise, one or more `--|Parameters` annotations would be present.
- The `--|No_Wait` annotation permits concurrent execution (see section A.3.3). This annotation is present because this procedure starts a server.

Specifying further information: “_Start” files may also contain annotations that control the order in which the Start procedure will execute the referenced procedures. See section A.3.3.

Using a “_Start” file is equivalent to executing the referenced procedure via `Program.Run_Job` or `Program.Run`. See the comments in the specification of `!Machine.Initialization.Start` for additional annotations that specify the equivalent of the `Options` parameter in the `Program.Run_Job` procedure and the `Context` parameter in the `Program.Run` and `Program.Run_Job` procedures.

A.3.3. Controlling the Order of Execution

You can specify the relative execution order of all initialization procedures (including those referenced in “_Start” files). To do this, you include annotations in the appropriate procedure specifications or “_Start” files:

- To specify that procedure A cannot run until procedure B has finished, you include the annotation `--|Prerequisite B` in the specification of procedure A (or in the “_Start” file that references procedure A).

If procedure B is referenced in a “_Start” file, you specify the filename as the annotation’s argument: `--|Prerequisite B_Start`.

Note that the argument of this annotation is a simple name and that all three worlds (Rational, Local, and Site) are searched for that simple name. Therefore, simple names must be unique across these three worlds if you want to use this annotation.

- To specify that procedure A must finish before any other procedure can start executing, you include the annotation `--|Wait` in the specification of procedure A (or in the “_Start” file that references procedure A).

Using this annotation is equivalent to executing procedure A using `Program.Run`.

- To specify that procedure A is to execute as a separate job concurrent with other procedures, you include the annotation `--|No_Wait` in the specification of procedure A (or in the “_Start” file that references procedure A).

Using this annotation is equivalent to executing procedure A using `Program.Run_Job`.

If none of the annotations listed above are present in a given procedure or “_Start” file, the `--|Wait` annotation is assumed. That is, procedures are executed sequentially unless told otherwise.

If a circular dependency results from a combination of annotations, it will be reported and ignored, so that each procedure will run.

Note that you can execute the Start command with the Effort_Only parameter set to True to test the execution order that results from your annotations.

See the comments in the specification of the !Machine.Initialization.Start procedure for a complete description of annotation usage, along with examples.

A.3.4. Customizing Disk-Collection Thresholds

You can customize the disk-collection thresholds for the particular needs of your site. To implement a change in your disk-collection thresholds:

1. Create an empty Ada unit in the Local world.
2. Copy the contents of the !Machine.Initialization.Local.Local_Gc_Thresholds_Sample file into the empty Ada unit.
3. In the Ada unit, edit the Thresholds1 and Thresholds2 arrays to specify the desired thresholds.
4. Promote the Ada unit.

Note, however, that values set by the procedure in world Rational are calculated based on your disk configuration and should be sufficient; see your Rational technical representative if you want to use different thresholds.

A.4. Enabling and Configuring Login Ports

D_12_5_0 and subsequent releases provide a file-driven mechanism in !Machine.Initialization for enabling and configuring ports for login.

At the very least, you must ensure that this mechanism has enough information to enable the desired login ports (see section A.4.1). In addition, you may optionally use this mechanism to specify:

- Connection and terminal-type characteristics for Telnet and RS232 ports, such as logoff on disconnect, disconnect on logoff, and so on
- Communication characteristics for RS232 ports, such as flow control, parity, and so on

Such information is specified using the options described in section A.4.4.

A.4.1. Enabling Ports for Login

Ports for terminal devices must be enabled for login each time an R1000 boots. Accordingly, the !Machine.Initialization.Start procedure calls a procedure called Terminals in the Rational world. This procedure in turn consults a file called Terminal_Configuration in the Local world to determine which ports to enable for login. This file-driven mechanism takes the place of a procedure such as !Machine.Initialize_Terminals, which enables terminals through calls to the Operator.Enable_Terminal procedure.

The Terminal_Configuration file is created automatically in the Local world during installation:

- On a new machine, a Terminal_Configuration file is created that enables ports 235 through 249 for login.
- On an R1000 that is being upgraded from a previous Environment release, the Terminal_Configuration file contains entries for the same set of ports that were enabled at the time of installation. (This file also preserves any nondefault communication characteristics that were in effect for RS232 ports; see section A.4.4.)

You can edit the Terminal_Configuration file at any time to change which ports are enabled. The changes take effect the next time you boot the R1000. Alternatively, you can execute the Rational.Terminals procedure to make the changes take effect without booting the R1000. Note that you must keep the Terminal_Configuration file in the Local world, even if you want to enable the same ports on all machines at a given site.

Following is a sample Terminal_Configuration file containing basic enabling information:

```
16 => (Enable)
224 .. 249 => (Enable)
-- Ports 250 and 251 are for printers; disable them for login
250..251 => (Login_Disabled)
```

As shown, the Terminal_Configuration file consists of:

- Comments preceded with Ada comment notation (--)
- Entries of the general form: *Port_Range* => (*Options*), where:
 - *Port_Range* can be a single port number or a range of port numbers
 - *Options* must be enclosed in parentheses

The options that pertain to enabling and disabling ports are summarized in Table A-1.

Table A-1 Options for Enabling and Disabling Ports

Option	Description
Enable	When specified for a given port, enables the port for login. Note that the port cannot subsequently be enabled for any other device, such as a printer.
Login_Disabled	When specified for a given port, prevents the port from being enabled for login—for example, by subsequent usage of the Operator.Enable_Terminal command. Note that the port can subsequently be enabled for other devices, such as printers.
Disable	When specified for a given port, disables the port for all devices. Note that the port can subsequently be enabled for any device, including login. Specifying this option is equivalent to having no entry for the port in the file.

Port 16 is always enabled for login, regardless of whether an entry exists for it. An entry for port 16 is included in the automatically created Terminal_Configuration file for explicitness.

Do not assign the Enable option to any port that you plan to enable for a printer or other device (such as a CDF). Instead, you can assign the Login_Disabled option or the Disable option to those ports, or you can simply omit entries for them from the file. Assigning the Login_Disabled option is recommended if you want to ensure that printer ports cannot be enabled for login even if the print spooler is killed.

A.4.2. Customizing Port Characteristics

You can add information to the Terminal_Configuration file to specify connection characteristics for RS232 ports and communication characteristics for RS232 and Telnet ports. Such information is specified through the options listed in section A.4.4.

The simplest way to specify multiple options is to assign them directly to a port or range of ports:

- Multiple options in a single entry must be enclosed in parentheses.
- Multiple options must be separated by commas.
- The options can extend over several lines, although the entry itself must start on a new line.

For example, the following entry assigns several connection characteristics to ports 224..249 and then enables those ports:

```
224..249 => (Logoff_On_Disconnect,
             Disconnect_On_Logoff,
             Enable)
```

You can organize recurrent sets of options and improve readability in the Terminal_Configuration file by defining an abbreviation for each set of options and then assigning each abbreviation to a port or range of ports:

- Abbreviation entries are of the general form **Abbreviation = Options**. Note that the equals sign (=) is used to define abbreviations; the => symbol is used for port assignment.
- Existing abbreviations can be nested in the definition of new abbreviations.

For example, the following entries create the abbreviations User_Ports and Telnet_Ports, assigning the Telnet_Ports abbreviation to ports 224..249:

```
-- Port settings for user login ports
User_Ports = (Logoff_On_Disconnect, Disconnect_On_Logoff)

-- Port settings for Telnet ports
Telnet_Ports = (Terminal_Type => Xrterm, User_Ports)

224..249    => (Telnet_Ports, Enable)
```

When adding entries to a Terminal_Configuration file, bear in mind that:

- Nondefault communication characteristics for RS232 ports must be set each time an R1000 boots. Consequently, if a port is to have nondefault values for any of the options listed in Table A-4, you must include these options in the entry for that port. Omitting an option causes its default value to be set.

- Connection and terminal-type characteristics persist across boots, retaining the last values that were set for them. Thus, in principle, the options listed in Tables A-2 and A-3 need to be set only once and then can be omitted from the Terminal_Configuration file. However, you may choose to include values for these options in the file to ensure that booting the system resets them to the proper values in case they had been changed.
- The options for each port are set in the order in which they are assigned in the Terminal_Configuration file. Similarly, the options in an abbreviation are set in the order in which they are declared. If a single port number is included in the ranges of more than one entry, that port takes the options of the last entry in which it appears.

A.4.3. A Sample Terminal_Configuration File

The following sample file shows how a system manager can use abbreviations to organize port information meaningfully. Note that a number of connection options have been explicitly set to ensure that booting the system sets them to a known value. Note also that specifying the Disable option for the printer ports is not absolutely necessary; however, specifying this option ensures that no previous entry in the file had inadvertently enabled these ports.

```
-- Operator line 16 settings
Operator_Port = (~Logoff_On_Disconnect,
                ~Disconnect_On_Logoff,
                ~Login_Disabled)

-- User login port settings
User_Ports = (Logoff_On_Disconnect, Disconnect_On_Logoff,
              ~Login_Disabled, ~Log_Failed_Logins,
              ~Disconnect_On_Failed_Login, ~Disconnect_On_Disconnect)

-- Dial-in port connection settings
Dialin_Ports = (Terminal_Type => VT100,
                Input_Rate => Baud_2400, Output_Rate => Baud_2400,
                Parity => None, Bits_Per_Char => Char_8, Stop_Bits => 1,
                User_Ports)

-- Telnet port settings
Telnet_Ports = (Terminal_Type => Xrterm, User_Ports)

-- Printer port settings
Printer_Ports = (Login_Disabled)

-- Ports not in use
Unused = (Login_Disabled)

16 => (Operator_Port, Enable)
17..31 => (Dialin_Ports, Enable)
224..249 => (Telnet_Ports, Enable)
250..251 => (Disable, Printer_Ports)
252..255 => (Disable, Unused)
```

A.4.4. Terminal-Configuration Options

Tables A-2, A-3, and A-4 summarize the connection, terminal type, and RS232 communication options you can specify in the Terminal_Configuration file. These options invoke corresponding procedures in package Terminal.

Table A-2 Boolean Options for Connection Characteristics

Option	Description
Disconnect_On_Disconnect	When specified for a given port, causes the Environment to respond to an incoming disconnect signal received on that port by initiating an outgoing disconnect signal on that port.
Disconnect_On_Failed_Login	When specified for a given port, causes the Environment to initiate an outgoing disconnect signal on that port when a user repeatedly fails to log in on that port (for example, by repeatedly entering an incorrect password or unrecognized username).
Disconnect_On_Logoff	When specified for a given port, causes the Environment to initiate an outgoing disconnect signal on that port when a user logs off a session running on that port.
Log_Failed_Logins	When specified for a given port, causes the Environment to write an entry to the system error log when a user fails repeatedly to log in on that port (for example, by repeatedly entering an incorrect password or unrecognized username).
Logoff_On_Disconnect	When specified for a given port, causes the Environment to respond to a disconnect received on that port by logging off that port's session.

Table A-3 Enumeration Option for Specifying Terminal Type

Option => Value	Description
Terminal_Type	Specifies the output driver type for a given port. <i>Value</i> can be any valid terminal type name, including (but not limited to): Cit500r Facit Rational Vt100 Xrterm

Table A-4 Enumeration Options for RS232 Communication Characteristics

Option => Value	Default Value	Description
Bits_Per_Char	Char_8	Specifies the number of data bits per character. <i>Value</i> can be: Char_5 Char_6 Char_7 Char_8
Flow_Control	None	Specifies software flow control for data transmitted by the R1000 on the specified port. <i>Value</i> can be: None Xon_Xoff Dtr Rts
Input_Rate	Baud_9600	Sets the incoming data rate for a given port. <i>Value</i> can be: Baud_50 Baud_75 Baud_110 Baud_134_5 Baud_150 Baud_200 Baud_300 Baud_600 Baud_1200 Baud_1800 Baud_2400 Baud_9600 Baud_19200 Disabled Ext_Rec_Clk
Output_Rate	Baud_9600	Sets the outgoing data rate for a given port. Values are the same as for Input_Rate .
Parity	None	Sets the parity for transmitted and received data on a given port. <i>Value</i> can be: None Odd Even
Stop_Bits	2	Sets the number of stop bits for a given port. <i>Value</i> is a natural number in the range 1..2.
Receive_Flow_Control	None	Specifies flow control of data received by the R1000 on the specified port. <i>Value</i> can be: None Xon_Xoff Dtr Rts
Receive_Xon_Xoff_Bytes	(17, 19)	Specifies flow-control bytes so that the R1000 can regulate the data it receives on the specified port. <i>Value</i> is (n, m) , where n and m are natural numbers in the range 0..255.
Xon_Xoff_Bytes	(17, 19)	Specifies the flow-control bytes that the R1000 recognizes for the specified port. <i>Value</i> is: (n, m) , where n and m are natural numbers in the range 0..255.

A.5. Configuring Printers

D_12_5_0 and subsequent releases provide a file-driven mechanism in !Machine.Initialization for configuring a group of networked and/or local printers. This mechanism allows you to define a *printer name* for each printer on the network and to specify how each printer is to be accessed. Furthermore, you can also associate a printer name with each user, so that when a given user enters the Print command (that is, !Commands.Abbreviations.Print), the print job will be sent by default to the device that is defined by the associated printer name.

This file-driven mechanism automatically adds the specified devices to the appropriate R1000s, creates the necessary print classes on the appropriate R1000s, and associates each class with the specified device, thereby creating print queues. Thus, when you use the file-driven mechanism,

you do not need to use procedures from package Queue (such as Add, Create, Enable, and Register) to do these things.

Existing sites can choose whether to use this file-driven mechanism or to continue using procedures from package Queue to configure printers. However, because the file-driven mechanism combines class and machine information, it is recommended that large sites with multiple networked printers use the file-driven !Machine.Initialization mechanism. Small sites with few printers connected directly to R1000s may want to continue using package Queue. Sites that want to create a single class with multiple devices should also use package Queue.

Note that the printer configuration (set either through commands in package Queue or through the file-driven mechanism) applies to both the Queue.Print and Abbreviations.Print commands. The ability to associate printer names with usernames is specific to the Abbreviations.Print command.

A.5.1. Where to Specify Printer Information

Each time an R1000 boots, the !Machine.Initialization.Start procedure calls a procedure called Printers in the Rational world. This procedure initializes the print spooler on that R1000 based on the information in the following user-created files:

- !Machine.Initialization.Site.Printer_Configuration, which defines a printer name for each of the devices available on the network and specifies how each device is to be accessed. A copy of this file must exist on all R1000s from which users will enter the Print command and on all R1000s that will handle print requests for the specified devices.
- !Machine.Initialization.Local.Printer_Configuration, which defines additional printer names for additional devices intended only for users of the current R1000.
- !Machine.Initialization.Local.User_Printer_Map, which associates a default printer name with individual users on the current R1000. When a user executes the Print command with the default Printer parameter, the command looks up the user's name and sends the print request to the corresponding printer.

At a minimum, the Print command requires that one Printer_Configuration file exist in either the Site or Local world. If no User_Printer_Map file exists (or if the file exists but contains no entry for a particular user), the Print command uses the first printer name defined in the Site.Printer_Configuration file. If this file doesn't exist, the first printer name defined in the Local.Printer_Configuration file is used.

The classes and devices specified in the Printer_Configuration files are created when the Printers procedure (that is, !Machine.Initialization.Rational.Printers) is executed (normally, during machine initialization). The Printers procedure also associates usernames with the appropriate printer information. Be aware that whenever information is changed in any of the files listed above, the Printers procedure must be run in order to update the information.

If you choose not to create any of these files, you will have to:

- Create an initialization procedure (in either the Site or Local world) that uses package Queue to create print queues each time the system boots.
- Either:

- Use the `Queue.Print` command instead of the `!Commands.Abbreviations.Print` command.
- Use the `!Commands.Abbreviations.Print` command, but explicitly specify the class name for the `Printer` parameter.

A.5.2. Adding Entries to a `Printer_Configuration` File

During installation, an empty `Printer_Configuration` file is created in the Local world. After installation, you can:

- Add entries to this file to enable the use of the `Print` command.
- Move this file (or create an additional file called `Printer_Configuration`) in the Site world, if you need to define sitewide printer information.

Each entry in a `Printer_Configuration` file defines a printer name and specifies the characteristics of the device it represents.

Each entry must start on a new line, but the information can extend over several lines and can include single and in-line comments. For readability, the entries are often formatted like command parameters.

Each entry has the general form:

```
Printer_Name => (Device_Type => Device_Info,
Other_Device_Characteristics,
                Laser_Comm           => Boolean,
                Reverse_Output_Pages => Boolean,
                On_Node              => R1000_Name)
```

where:

- *Printer_Name* is the name by which you want to refer to a given device in a `Print` command. *Printer_Name* must be a legal Ada simple name.
- *Device_Type* is one of the following four kinds of printer devices:
 - *Direct*, which specifies a printer connected to an R1000 via direct line.
 - *Telnet*, which specifies a printer connected to an R1000 via Telnet.
 - *File*, which specifies a file on an R1000 in which print-spooler output is collected. Subsequent processing is required to get this output printed.
 - *Workstation*, which specifies a directory on a remote workstation to which print requests are sent. Such requests are sent via FTP as individual files; from the remote directory, they can be printed using the workstation's print tools.
- *Device_Info* is further information about the device you specified. The information depends on the type of device specified (see the paragraphs below).
- *Other_Device_Characteristics* are additional entry elements, separated by commas, that give further information about the chosen device (see the paragraphs below).
- The *Laser_Comm* option, when True, specifies that printing will be done on a laser printer. Omitting this option implies that printing will be done on a line printer.

For Direct and Telnet devices, setting `Laser_Comm` to `True` specifies that a laser printer is connected; for File and Workstation devices, setting `Laser_Comm` to `True` specifies that the collected print requests will eventually be printed on a laser printer.

- The `Reverse_Output_Pages` option allows you to adjust the order in which pages are spooled to accommodate the way your printer stacks pages in its output tray. This option applies only if the `Laser_Comm` option is set to `True`.
 - Setting `Reverse_Output_Pages` to `True` causes the print spooler to reverse the order of output pages, so that the last logical page is printed first. Omitting this option is equivalent to specifying `True`.
 - Setting `Reverse_Output_Pages` to `False` causes the print spooler to keep the pages of output in the order in which they appear in the source file.
- The `On_Node` option specifies the network name of the R1000 that contains the print spooler for the device. Omitting this option is equivalent to specifying the name of the current R1000.

The following paragraphs describe printer-configuration file entries for each of the four kinds of devices. (See also the comments in the specification of `!Machine.Initialization.Rational.Printers`.)

A.5.3. Specifying a Directly Connected Printer

To specify a printer connected to an R1000 via direct line, you specify an entry of the following general form:

```
Printer_Name => (Direct           => Protocol,
                 Device           => Terminal_N,
                 Laser_Comm       => Boolean,
                 Reverse_Output_Pages => Boolean,
                 On_Node          => R1000_Name)
```

where:

- `Protocol` specifies the printer flow control and can be either `Xon_Xoff` or `Dtr`. See the printer manual for details.
- `Terminal_N` represents the RS232C port to which the printer is connected (`N` is the port number). The specified port must not be enabled for login in the `Local.Terminal_Configuration` file.
- The `Laser_Comm` option, when `True`, specifies that a laser printer is connected and enables a two-way printer-communication protocol. Omitting the option is equivalent to specifying `False`, which means that a line printer is connected.
- The `Reverse_Output_Pages` option allows you to adjust the order in which pages are spooled to accommodate the way your printer stacks pages in its output tray, as described in section A.5.2. This option applies only if `Laser_Comm` is set to `True`.
- The `On_Node` option specifies the network name of the R1000 to which the printer is directly connected. Omitting this option is equivalent to specifying the name of the current R1000.

The following entry creates a printer name called `Lp`, which represents a line printer that is directly connected to port 30 of an R1000 called `Jazmo`:

```
-- Line printer connected to Jazmo
Lp => (Direct => Xon_Xoff,
      Device => Terminal_30,
      On_Node => Jazmo)
```

A.5.4. Specifying a Networked Printer

To specify a printer connected to an R1000 via Telnet, you specify an entry of the following general form:

```
Printer_Name => (Telnet           => Host_Name,
                 Device           => Terminal_N,
                 Laser_Comm       => Boolean,
                 Reverse_Output_Pages => Boolean,
                 On_Node          => R1000_Name)
```

where:

- *Host_Name* is the network name of the printer.
- *Terminal_N* represents the Telnet port to which the printer is connected (*N* is the port number). The specified port must not be enabled for login in the Local.Terminal_Configuration file.
- The *Laser_Comm* option, when True, specifies that a laser printer is connected and enables a two-way printer-communication protocol. Omitting the option is equivalent to specifying False, which means that a line printer is connected.
- The *Reverse_Output_Pages* option allows you to adjust the order in which pages are spooled to accommodate the way your printer stacks pages in its output tray, as described in section A.5.2. This option applies only if *Laser_Comm* is set to True.
- The *On_Node* option specifies the network name of the R1000 to which the printer is connected via Telnet. Omitting this option is equivalent to specifying the name of the current R1000.

The following entry creates the printer name Dlaser, which represents a laser printer that is connected via Telnet to port 226 of an R1000 called Roget. Because of the way this printer stacks its output, print requests are spooled to this device with their pages in ascending (rather than reversed) order:

```
-- Documentation's laser printer
Dlaser => (Telnet           => Doc_Laser,
          Device           => Terminal_226,
          Laser_Comm,
          Reverse_Output_Pages => False,
          On_Node          => Roget)
```

A.5.5. Specifying an Environment File

To specify a file in which to collect print-spooler output, you specify an entry of the following general form:

```
Printer_Name => (File           => Environment_Pathname,
```

```

Laser_Comm           => Boolean,
Reverse_Output_Pages => Boolean,
On_Node              => R1000_Name)

```

where:

- **Environment_Pathname** specifies the file to which output is written. The pathname must name a file that exists on the R1000 named by **On_Node**. Note that the group Spooler must have access to the specified file.
- The **Laser_Comm** option, when True, specifies that the collected print requests will eventually be printed on a laser printer. Omitting the option is equivalent to specifying False, which means that a line printer will be used.
- The **Reverse_Output_Pages** option allows you to adjust the order in which pages are spooled to accommodate the way your printer stacks pages in its output tray, as described in section A.5.2. This option applies only if **Laser_Comm** is set to True.
- The **On_Node** option specifies the network name of the R1000 on which the file is located. Omitting this option is equivalent to specifying the name of the current R1000.

The following entry creates the printer name Hold, which represents a file on an R1000 called Logo. Low-priority print requests are sent to this file, where they are held until someone prints the file using the Print command (specifying a printer name that represents a connected printer):

```

-- Place to hold large requests until printers are free in the evening
Hold => (File      => !Machine.Queues.Local.Held_Print_Requests,
        On_Node   => Logo)

```

A.5.6. Specifying a Workstation Directory

To specify a directory on a workstation to which print requests are sent, you specify an entry of the following general form:

```

Printer_Name => (Workstation           => Host_Name,
                  Path                   => Directory_Name,
                  Laser_Comm            => Boolean,
                  Suffix                => String,
                  Reverse_Output_Pages => Boolean,
                  On_Node               => R1000_Name)

```

where:

- **Host_Name** is the network name of the workstation to which the files will be transferred.
- **Directory_Name** is the pathname of the workstation directory into which the files will be transferred. The directory pathname must have syntax appropriate to the workstation and must have trailing punctuation that permits the name of the transferred print-request file to be appended.
- The **Laser_Comm** option, when True, specifies that the collected print requests will eventually be printed on a laser printer. Omitting the option is equivalent to specifying False, which means that a line printer will be used.

- The **Suffix** option allows you to specify a string to be appended to the filenames that are created on the workstation. Omitting this option causes no suffix to be appended to the filenames. The specified suffix can be used by print tools as a way of identifying which files to print. This is useful when several printer names send files to the same directory.
- The **Reverse_Output_Pages** option allows you to adjust the order in which pages are spooled to accommodate the way your printer stacks pages in its output tray, as described in section A.5.2. This option applies only if **Laser_Comm** is set to True.
- The **On_Node** option specifies the network name of the R1000 whose print spooler handles the print requests. Omitting this option is equivalent to specifying the name of the current R1000.

In addition to creating an entry for each workstation in the Printer_Configuration files, you must also create a remote-passwords file, containing the username and password to be used for accessing each workstation. This remote-passwords file must be named Remote_Access and exist in the library !Machine.Queues.Ftp. (For information about remote-passwords files, see package Remote_Passwords in the Session and Job Management book of the *Rational Environment Reference Manual*.)

When print requests are sent as files to a workstation, any FTP messages are directed to a log file that is created in !Machine.Queues.Ftp. This log file is automatically cleared after each 100 print requests. Messages pertaining to creating print classes and enabling devices are directed to the system error log.

The following two entries create printer names Dc_Laser and Dc_Lineprinter, both of which direct print requests to a directory on a UNIX® workstation called Enterprise. These print requests, which are routed through the print spooler on an R1000 called Capitol, are sent as files with different suffixes, depending on the printer name (**_Lsr** and **_Lpt**, respectively):

```
-- Laser printer attached to workstation in Washington, D.C., office;
-- spooled on R1000 called Capitol.
Dc_Laser => (Workstation => Enterprise,
             Path          => /usr/spool/ratqueue/,
             Laser_Comm,
             Suffix        => _Lsr,
             On_Node       => Capitol)

-- Line printer attached to workstation in Washington, D.C., office;
-- spooled on R1000 called Capitol.
Dc_Lineprinter => (Workstation => Enterprise,
                  Path          => /usr/spool/ratqueue/,
                  Suffix        => _Lpt,
                  On_Node       => Capitol)
```

Print tools on the workstation, such as the following sample, are required to actually print the requests:

```
# This program spools print requests placed in /usr/spool/ratqueue to
# the appropriate printer based on the suffix of the spooled file.
# It checks the spool directory at 5-minute intervals to see if any
# new files need printing. This runs as a C-shell script. To execute,
# type csh and the name of this file.
```

```
:
```

```

set xFTPxDIR=/usr/spool/ratqueue          #spool directory
#
set xFTPXSUFFIXxLSRx=_Lsr                # Laser printer suffix
set xFTPXSUFFIXxLPTx=_Lpt                # Line printer suffix
#
set xPRINTxLISTxLSRx=/tmp/rat_print_lsr.$$ # Laser printer list file
set xPRINTxLISTxLPTx=/tmp/rat_print_lpt.$$ # Line printer list file
#
set xPRINTxCOMMANDx=/tmp/rat_command.$$   # temporary print command
file
#
set xPRINTxDELAYx=120                     # interval to wait before printing
set xRECHECKxTIMEx=180                   # interval for checking print requests
#
cd $xFTPxDIRx
while (1 == 1)
#
# Creates a list of files to print for each printer.
#
ls $xFTPxDIRx | grep $xFTPXSUFFIXxLSRx > $xPRINTxLISTxLSRx
ls $xFTPxDIRx | grep $xFTPXSUFFIXxLPTx > $xPRINTxLISTxLPTx
#
# Adds the laser- and line-printer files to the print-command file
# (the device name of each type of printer should be provided after the
-P).
#
cat $xPRINTxLISTxLSRx | sed -e 's^.^lpr -r -Plaser &^' >
$xPRINTxCOMMANDx
cat $xPRINTxLISTxLPTx | sed -e 's^.^lpr -r -Pline &^' >>
$xPRINTxCOMMANDx
#
# Waits the specified amount of time before printing the requests.
# (This delay allows time for the FTP operation to complete and should
# be adjusted based on the average length of files being printed.)
#
sleep $xPRINTxDELAYx
#
# Prints the files if there are any to print.
#
set LCNT=`cat $xPRINTxCOMMANDx | wc -l `
if ( $LCNT != 0 ) then
    chmod +x $xPRINTxCOMMANDx
    $xPRINTxCOMMANDx
endif
#
# Waits the specified amount of time.
#
sleep $xRECHECKxTIMEx
#
# Removes the old temporary files.
#
rm $xPRINTxLISTxLSRx
rm $xPRINTxLISTxLPTx
rm $xPRINTxCOMMANDx
#
end

```

A.5.7. Associating Default Printers with Individual Users

To associate a default printer with each user, you must create a file called `User_Printer_Map` in the Local world and add entries of the following form:

```
Username Printer_Name
```

where:

- **Username** is either:
 - The username for an R1000 user
 - The string **others**, which represents all users not explicitly listed by name.
- **Printer_Name** is defined in a `Printer_Configuration` file.

Following is a sample `User_Printer_Map` file:

```
phil dc_laser  
sue dlaser  
others lp
```

See also the comments in the specification of `!Machine.Initialization.Rational.Printers`.

Appendix B

Diagnostic Crash Procedures

B.1. Introduction

The D_12_6_5 release includes enhancements to the process of analyzing R1000 crashes. In particular:

- Much of the process has been automated so that the system will capture as much information as possible and often reboot automatically.
- A series of questions and menus have been added to the diagnostic crash procedures, which:
 - Capture system crash state
 - Analyze system hardware

The instructions in this appendix supersede those in your *System Manager's Guide*. For convenience, you may want to place a copy of these procedures with your *System Manager's Guide*.

B.2. When Your System Crashes

Call the Rational Response Center whenever your R1000 fails. You will be guided through the most appropriate parts of the procedures below—in some cases through other undocumented steps as well—to reach an effective and timely diagnosis of the problem. For information about contacting the Rational Response Center, see section B.12.

If your system crashes outside Rational's normal support hours, follow one of the three options described below. These options are listed in order, from most desirable to least desirable (in terms of gathering information needed to determine why your machine crashed and to prevent additional crashes):

- Wait and contact the Rational Response Center during normal support hours (see section B.12).
- Gather diagnostic information about your system crash (by following the diagnostic crash procedures in this appendix) before you try to reboot the machine.
- Try to reboot the machine immediately without gathering diagnostic information. Be aware that doing so may lengthen the time it takes to isolate and resolve the problem.

B.3. Using These Diagnostic Crash Procedures

The instructions in this appendix are designed to help you collect as much diagnostic information as possible about the state of your machine at the time of a system crash. By following these procedures, you will help shorten the time to diagnose and, if necessary, repair your system.

These procedures are organized into sections roughly according to the order in which you may need to perform them.

No matter why the system crashed, you should always capture system-state information, as described in section B.4.

Then, if you experienced or are experiencing:

- A power failure, see section B.5.
- Overheating, see section B.6.
- The "connect modem" message, begin with section B.7.
- Any other failure message, begin with section B.8.
- Disk errors, begin with section B.8 and continue through section B.10.

After executing the procedures in the following sections, fax or e-mail the information you have gathered to the Rational Response Center (see section B.12).

Note that when the R1000 CPU stops running, the I/O processor queries all the boards to determine what caused it to stop. The message given by the R1000 when it crashes names the board that detected the specific check; therefore, the named board is not necessarily the faulty part.

B.4. Saving Information about the System State

When your system fails, you should begin by saving information about the system state.

1. Write down the crash information displayed on the operator console. If this information has already scrolled off the screen, go to step 2, below. In particular, write down the machine check message (MC), crash reason (CR), trace (TR), and any other messages (OM) displayed. A sample crash output is shown below:

```
MC=> Sequencer has detected a machine check.
*****
CR=> Booting R1000 IOP after halt of machine crash detected
CR=> Boot reason code = 0C, from PC 0001A094
Restarting R1000-200 February 14th, 1992 at 14:14:35

-- you may see the Interactive menu here
-- if so, press [Return]

Logical tape drive 0 is a 8MM cartridge tape drive.
Logical tape drive 1 is declared non-existent.
Logical tape drive 2 is declared non-existent.
Logical tape drive 3 is a 1/2 inch 9 track tape drive at physical
unit 0.
Booting I/O Processor
IOC Bootstrap Version 5.7

Initializing I/O Processor Kernel 0_8_7
Disk Controller 0, Disk 0 is ONLINE and WRITE ENABLED
Disk Controller 0, Disk 1 is ONLINE and WRITE ENABLED
Disk Controller 0, Disk 2 is ONLINE and WRITE ENABLED
Disk Controller 0, Disk 3 is ONLINE and WRITE ENABLED

IOP Kernel is initialized
```

```

Initializing diagnostic file system ... [OK]
  FREEZE_WORLD.FIU -- several messages like these overwrite
  MF.IOC           -- each other
=====
CR=> Restarting system after R1000 Processor Halt => Processor multi-bit
CR=>   memory error
      -- previous two lines indicate the crash reason
      -- and appear in reverse video

Running post-mortem diagnostics.
TR=> uaddr trace: 4=2A92 3=018D 2=020E 1=0204 -- ucode M207_36
CR=> Microcode HALT
CR=> The processor got a(n) Halt.
OM=> Post-mortem diagnostics do not apply to this kind of failure

Saving state of: board JKLMFQTVI, Special Registers, Trace Rams [OK]
R1000 Crash Save done - tombstone file R1000_DUMP1 created.

```

2. Write down the context of the crash.

- How long had the R1000 been running since the last boot?
Was the R1000 still booting when it crashed?
Why was it booting?
- Was the Environment up at the time?
Was there anything abnormal about the way the Environment was functioning?
What were users doing? Any unusual activities?
Were any daemons (snapshot, daily, or weekly) running?
Was a tape drive in use?
- Have there been any hardware, firmware, or software changes to this R1000 within the past week?
- Have there been any problems with or modifications of the network in the past week?
- What was the temperature in the computer room at the time of the crash?
- Have there been any problems with the air conditioning recently?
- Are there other computers (R1000s or not) in the same building?
Did any of the other computers crash or have power or environmental problems at the same time?

3. Use the crash reason message (displayed in reverse video below the line of equal signs) to locate the appropriate recovery process in section B.9.

B.5. Crash due to Power Failure

Note: A crash due to a power failure does not cause the system to dial Rational through the system's internal modem, even when the system is configured to do so for other types of crashes.

B.5.1. Indication of Power Failure

In most cases, a crash due to power failure is reported by the following message in reverse video on the operator console:

```
Restarting system after loss of AC power
```

B.5.2. Recovering from Power Failure

When your system crashes because of a power failure:

1. If power to the system is still off, turn the system's power keyswitch to **OFF**.
Turning the power keyswitch to **OFF** reduces the chance of accidental damage to the R1000 when the source of power is restored.

2. *Before trying to restart the system, try to determine the cause and extent of the power failure and to correct the problem.*

Determining the cause and extent of the power failure commonly requires the aid of your site facilities personnel. If the power failure affected your entire site, your site facilities personnel should contact your local power company for further information.

3. Restore power to the R1000 and turn the power keyswitch to **ON**.
4. After power has been restored, the system will do one of three things, depending on its configuration:

- Boot automatically
- Begin booting and stop at the Interactive menu:

Options are:

- 1 => Change **BOOT/CRASH/MAINTENANCE** options
- 2 => Change **IOP ENVIRONMENT** configuration
- 3 => Enable manual crash debugging (**experts only**)
- 4 => Boot **IOP**, prompting for tape or disk
- 5 => Boot **system**

Enter option [Boot System]:

If the system stops at the Interactive menu, press [Return]. The system will display the Crash menu.

- Begin booting and stop at the Crash menu:

Options are:

- 1 => Enter **CLI**
- 2 => Make a **CRASHDUMP**
- 3 => Run the **FRU tests**
- 4 => Boot **DDC** configuration
- 5 => Boot **EEEDB** configuration
- 6 => Boot **STANDARD** configuration

Enter option [Boot STANDARD] :

If the system stops at the Crash menu, press [Return]. The system will continue booting.

5. If the system does not reboot properly, then:
 - A. Examine the operator console to see what messages are displayed.
 - B. Contact the Rational Response Center (see section B.12).

B.6. Crash due to Overheating

Note: A crash due to overheating does not cause the system to dial Rational through the system's internal modem (mounted on the PCM card in the PCM unit), even when the system is configured to do so for other types of crashes.

B.6.1. Indications of Overheating

In most cases, a crash due to overheating causes all of the front panel lights to be off although the power switch is on.

You can determine if the failure is due to overheating by examining the state of the LED overtemp indicator. On Series 200 and 300 systems, the overtemp indicator is on the power control module (PCM) and can be viewed by opening the rear door of the CPU cabinet. On a Series 400 system, the indicator is on the I/O panel, visible by opening the front door of the cabinet. (See your *System Manager's Guide* if you are unsure of the overtemp indicator's location.)

Warning: Do not reach inside the CPU bay. Electric-shock hazards are present inside the R1000 cabinets.

If the overtemp indicator is lit, this means that sensors in the R1000 had detected that the temperature inside the cabinet was too high and powered the system off.

B.6.2. Recovering from Overheating

1. Check the room temperature. Note that the room air temperature must not exceed 85°F (29°C) while the R1000 is operating. (For detailed environmental requirements, refer to the *Rational R1000 Development System: Site-Planning Guide* for your system.)
2. If the room temperature is correct, the problem might be a fan failure inside the cabinet, clogged filters, or some other problem. In this case:
 - A. Verify that the power keyswitch on the control panel is in the **off** position.
 - B. Contact the Rational Response Center for assistance (see section B.12).
3. After the overheating condition has been corrected, power off and restart the system by following the instructions in steps 3 through 5 of section B.5.2, above.
4. If the system does not reboot properly:
 - A. Examine the operator console to see what messages are displayed.
 - B. Contact the Rational Response Center (see section B.12).

B.7. Displays "Connect Modem" Message and Hangs

B.7.1. Indications of "Failure to Connect Modem"

The system crashes, displays a reverse-video error message on the operator console, and then starts to reboot. Midway through the boot process, the system pauses and dials the Rational Response Center, displaying:

Connecting modem

Wait at least three minutes for the call to complete. If you see the message:

```
Rational Response Center has instructed this cluster to wait for call
back
```

and do not want to wait, you can proceed with booting the system as follows:

1. Press [Break] on the operator console.
2. When the **Enter option** prompt is displayed, press [0] and [Return], then [Y] and [Return].
The modem connection is broken and the boot process begins.
3. Continue with section B.8, below.

Note: Systems with restricted-access support contracts may not be configured to use this dial-out capability.

B.8. Other System Failures

When your system crashes, it may display one or two menus or continue to boot, depending on the setting of the Operator Mode key, the setting of the boot/crash/maintenance options, and the type of crash:

- If the Operator Mode key is set to **Interactive**, the system displays the Interactive Key menu:

Options are:

- 1 => Change BOOT/CRASH/MAINTENANCE options
- 2 => Change IOP ENVIRONMENT configuration
- 3 => Enable manual crash debugging (experts only)
- 4 => Boot IOP, prompting for tape or disk
- 5 => Boot system

Enter option [Boot System]:

Press [Return]. This will take you to the Crash menu.

- If either the Auto Crash Recovery or the Auto Boot option is set to No, or if the Operator Mode key is set to **Interactive** and you have pressed [Return], then the system displays the Crash menu:

Options are:

- 1 => Enter CLI
- 2 => Make a CRASHDUMP
- 3 => Run the FRU tests
- 4 => Boot DDC configuration
- 5 => Boot EEEDB configuration
- 6 => Boot STANDARD configuration

Enter option [Boot STANDARD] :

The Crash menu provides access to the Command Line Interpreter (CLI), various diagnostic tests, and rebooting options.

The crash reason is displayed above in reverse video. Use the crash reason to find the relevant subsection of B.9, below.

B.8.1. Redisplaying the Crash Reason

If the crash reason is not visible as captured or if you can print from the console, then follow the procedures in this section to redisplay the crash reason and message. The crash reasons and recovery procedures for each are described in section B.9.

1. At the Crash menu, enter option 1 (Enter CLI):

```
Enter option [Boot STANDARD]: 1
```

Option 1 takes you to the CLI.

2. Enter `x tombstone`:

```
CLI> x tombstone
```

The console displays the Display Tombstone File menu:

Options are:

```
0 => Exit
1 => Display tombstone file 1
2 => Display tombstone file 2
3 => Display tombstone file 3
4 => Display tombstone file 4
```

3. Enter option 1 (Display tombstone file 1).

The four tombstone files act as a ring buffer with information about the most recent crash always in file 1. After entering option 1, you will see a message and menu like the one below:

```
Analysis of tombstone 1 dated - 12:44:52 24-FEB-92
```

Options are:

```
0 => DONE
1 => Show all
2 => Show last console output
3 => Show crash classifications
4 => Show restart output
5 => Show trace
6 => Show Cpu State
7 => Show queues
```

4. Enter option 4 (Show restart output).

The system displays the text shown in section B.4.

5. Make a note of the crash reason, which is indicated in reverse video under the line of equal signs. These messages are described in section B.9, below.

6. If your console is set up in such a way that you can easily send its output to a file or printer and e-mail or fax the output to the Rational Response Center, then:

- Enter option 1 (Show all).
- Capture the output in a file or send it to the printer.

If your console is not set up so that you can capture its output, you can capture this information from the Environment when your system is running again; see section B.11.

7. Enter option 0 (DONE).

The system returns you to the Display Tombstone File menu.

8. Enter option 0 (**Exit**).

The system returns you to the **CLI>** prompt.

9. Enter **bye**:

```
CLI> bye
```

The system returns you to the Crash menu.

10. Continue with section B.9, below. If you suspect disk problems or encountered disk errors, also see section B.10.
11. Send the information you have gathered to the Rational Response Center (see section B.12).

B.9. Common Crash Reasons and Responses

Once you have determined the crash reason (using the procedure described in section B.8, above), you can continue recovering from the crash.

Below are the common crash reasons, listed by number and message, and the recovery actions for each.

B.9.1. Halt => System Error

From the Crash menu, enter option 6 (**Boot STANDARD configuration**):

```
Enter option [Boot STANDARD] : 6
```

The system attempts to boot.

If your system will not boot, contact the Rational Response Center for further instructions (see section B.12).

B.9.2. Halt => I/O Processor Hardware Error

Press the white button to reboot the machine. Choose the **Boot system** or **Boot STANDARD configuration** options in any menus that you encounter.

If the system still fails to boot or crashes again with the same reason, call the Rational Response Center (see section B.12).

B.9.3. Halt => I/O Processor Software Error

Press the white button to reboot the machine. Choose the **Boot system** or **Boot STANDARD configuration** options in any menus that you encounter.

If the system still fails to boot or crashes again with the same reason, call the Rational Response Center (see section B.12).

B.9.4. Halt => Processor Hardware Error

See section B.9.6, below.

B.9.5. Halt => Processor Multi Bit Memory Error

See section B.9.6, below.

B.9.6. Halt => Processor Sysbus Hardware Error

Run the confidence (microdiagnostic) and field-replaceable unit (FRU) tests, by following the procedure below:

1. From the Crash menu, enter option 3 (Run the FRU tests):

```
Enter [Boot STANDARD] : 3
```

The system displays the FRU Main menu:

```
Rational R1000 FRU diagnostic driver
```

```
Initializing ...
```

```
Main menu
```

- ```

1 => Display cluster information
2 => Execute confidence test (microdiagnostic)
3 => Execute diagnostics
4 => Execute PM tests (not implemented yet)
5 => Margin cluster clocks/power
6 => Specify test options
7 => Repair assistance (model 100 CPU power control)
8 => Initialize processor state
0 => Exit

```

2. Enter option 2 (Execute confidence test (microdiagnostic)):

The system displays:

```
Preparing to run the microdiagnostic (uDIAG)
```

```
The long version stress tests the DRAMs but runs 2 minutes longer
```

3. When prompted to run the long version, enter y:

```
Do you want to run the long version [N] ? y
```

The system displays several messages and returns you to the FRU Main menu:

```
LOAD_HRAM_32_0.FIU ...
```

```
File : DIAG.M200_UCODE
```

```
Bound : November 15, 1989 13:02:00
```

```
Delta : RM:<MICROCODE.DIAG.IOC_DELTA>UCODE.DB.1
```

```
Mom : RM:<MICROCODE.DIAG.IOC_MOM>UCODE.DB.4
```

```
Loading Register Files and Dispatch Rams [OK]
```

```
Loading Control Store [OK]
```

```
PREP_WRITE_REG.TYP...
```

```
Main menu
```

- ```

1 => Display cluster information
2 => Execute confidence test (microdiagnostic)
3 => Execute diagnostics
4 => Execute PM tests (not implemented yet)
5 => Margin cluster clocks/power
6 => Specify test options
7 => Repair assistance (model 100 CPU power control)
8 => Initialize processor state
0 => Exit

```


4. Enter option 3 (**Execute diagnostics**).

The system displays the Diagnostic execution menu:

Diagnostic execution menu

- 1 => **Test the foreplane**
- 2 => **Run all tests**
- 3 => **Run all tests applicable to a given FRU (board)**
- 4 => **Run a specific test**
- 0 => **Return to main menu**

5. Enter option 2 (**Run all tests**).
6. When prompted for the maximum test phase, enter 2:

Enter maximum test phase (1-3) : 2

The system displays a long series of messages, such as those shown here:

Running FRU P1DCOMM...

...

Phase 1 passed

...

Phase 2 passed

After about 20 minutes, the system then returns you to the Diagnostic execution menu.

If the system halts with an error, the test has failed; skip to step 9.

7. Enter option 0 (**Return to main menu**).

The system returns you to the FRU Main menu.

8. Enter option 0 (**Exit**).

The system returns you to the Crash menu.

9. If either the confidence or the FRU tests failed, call the Rational Response Center before attempting to reboot the system (see section B.12).

If the confidence and FRU tests passed, enter option 6 (**Boot STANDARD configuration**):

Enter [Boot STANDARD] : 6

The system attempts to continue booting.

B.9.7. Halt => Processor Microcode Error

See section B.9.10, below.

B.9.8. Halt => Processor Software Error

See section B.9.10, below.

B.9.9. Halt => Shutdown from Environment

This message indicates that the system was deliberately shutdown with the Operator.Shutdown command. If the system was shutdown because of a problem, then see section B.9.10, below.

B.9.10. Halt => Processor Crash Error

1. Make a crash dump tape. *Note: Even if your site does not allow crash dump tapes to be sent to Rational, you should still make the tape because it might be possible for someone to analyze it at your site.*
 - A. From the Crash menu, enter option 2 (**Make a CRASHDUMP**):
Enter option [Boot STANDARD] : 2
 - B. Mount a tape.
 - If you have a 9-track tape drive, mount a 2,400-foot 9-track tape.
 - If you have an 8-mm tape drive, mount an 8-mm tape cartridge.
 - C. Make sure the tape is write-enabled.
 - If you are using a 9-track tape, insert the write-enable ring into the hub of the tape reel.
 - If you are using an 8-mm tape, slide the write-protect tab away from the center of the cartridge.
 - D. If your tape drive has a door, close it.
The system responds by displaying:
Tape unit number (0..3) [0] ?
 - E. Enter the unit number of your tape drive and press [Return].
 - If you are making the crash dump tape on a 9-track tape drive, the tape drive is probably configured as Unit 0.
 - If you are making a crash dump tape on an 8-mm tape drive, the tape drive is probably configured as Unit 0 on a series 300 or 400 system or Unit 3 on a series 200 system.
 - F. The system may respond by displaying:
**Tape density is currently 6250 BPI,
do you want to change it [N] ?**
If so, verify the density setting.
 - If you are using a 9-track tape drive, verify that its density is set to 6,250 BPI.
 - If you are using an 8-mm tape drive, the density is not selectable.
 - G. Press [Return] to continue. The system responds by displaying:
Volume Id, (1..6 characters) ?
 - H. Enter a six-character tape name (volume identification). For example:
910704
 - I. When prompted for comments with a) prompt, enter comments that could help someone isolate or reproduce this problem. Include the following in your comments:
 - The machine's name and cluster ID.
 - The date and time, using this format: 31 Mar 92 22:17.
 - The reason the system started to reboot (found in the reverse-video message and in the **Restarting system after . . .** message).
 - The name of the company or organization that owns/operates the R1000.

- The location of the R1000 installation.
 - Your name.
 - The information about the state of the machine and what caused the problem that you have gathered so far (except for the trace information).
- J. End your comments by entering, on a new line, a second right parenthesis next to the) comment prompt.

For example:

```
) Mariner, 902469
) 31 Mar 92 22:17
) WCS TAGSTORE PARITY ERROR
) ACME Widgets, Inc.
) Rockville, MD
) J. Smith
)
) There were no users on the machine.
) A backup was in progress.
) There had been a power failure two hours earlier.
))
```

Several messages are displayed while the crash dump is made. Wait for the crash dump to complete (about 15 minutes). The system automatically unloads your tape when it is done.

After the system finishes making the tape, it returns you to the Crash menu.

- K. Prepare a label for the tape with the words "Crash Dump" and the following items:
- The machine's name and cluster ID.
 - The date and time, using this format: 31 Mar 92 22:17.
 - Your site's configuration. This can be determined later by executing the Show_Default command from the **EEDB**: prompt.
 - A brief, one-line summary of why the system started to reboot.
 - The CSR number, if you have one.
- L. Dismount and store the tape as follows:
- i. Remove the tape from the tape drive and put the label on the 9-track tape reel (or on the 8-mm tape cartridge).
 - ii. Remove the write-enable ring from the 9-track tape reel (or slide the write-protect tab toward the center of the 8-mm tape cartridge), thereby write-protecting the tape.
 - iii. Put the 9-track tape reel or (8-mm cartridge) in a safe storage container.

At this point you have saved the volatile system state, both in your observations and on the crash dump tape.

2. Enter option 6 (Boot **STANDARD** configuration):

```
Enter option [Boot STANDARD] : 6
```

The system attempts to continue booting.

3. Contact the Rational Response Center for instructions on having your crash dump tape analyzed (see section B.12).

B.9.11. Machine Check (Parity Error)

This message indicates that there was a problem with the CPU hardware. The confidence test may automatically run or the Rational Response Center may be called via the modem. If the test passes, the system may automatically reboot or display the Crash menu, in which case you should enter 6 (Boot **STANDARD** configuration) and press [Return]. If the confidence tests fail, see section B.9.6.

B.9.12. Any Other Reason

Call the Rational Response Center (see section B.12).

If you are unable to reach someone, then:

- From the Crash menu, enter option 2 (**Make a CRASHDUMP**):
Enter option [Boot STANDARD] : 2
 The system requests that you load a tape, enter some information, and then returns to the Crash menu (for more details, see section B.9.10, above).
- Run the confidence (microdiagnostic) and field-replaceable unit (FRU) tests by following the instructions for "0205 Have_Sysbus_Hardware_Error" in section B.9.6, above.
- If the confidence and FRU tests fail, call the Rational Response Center (see section B.12). Otherwise, attempt to reboot the system.
- Fax or e-mail the information you have gathered to the Rational Response Center (see section B.12).

B.10. If You Suspect Disk Errors

If there were any disk errors, or if you suspect disk problems, run the disk exerciser (Diskx) and Checkdisk from the CLI.

B.10.1. Running Diskx

You should run the disk exerciser (Diskx) on all disks for at least 5 minutes. Diskx is a read-and-write test, but it uses only a small test area of each disk for writes, so it will not harm user data on the disks.

To run Diskx:

- Enter the CLI. To enter the CLI from the Crash menu, enter option 1 (**Enter CLI**):
Enter option [Boot STANDARD] : 1
- At the CLI> prompt, enter **x diskx**:
CLI> x diskx
- Press the space bar to cause the latest output to be displayed. Output similar to the following will appear:

```

u0 bytes=> 1246208   Soft=> 0   Hard=> 0   C=> 705   T=> 25   S=> 6
u1 bytes=> 1203200   soft=> 0   hard=> 0   C=> 73    T=> 14   S=> 8
u2 bytes=> 1206272   soft=> 0   hard=> 0   C=> 611   T=> 3    S=> 24

```

```
u3 bytes=> 1243208    soft=> 0    hard=> 0    C=> 634    T=> 11    S=> 10
```

4. Press [Control][G] to stop Diskx.
The system returns you to the CLI> prompt.
5. Write down the results and any messages where soft or hard does not equal zero (0).
6. Enter **bye** to return to the Crash menu:
CLI> **bye**

B.10.2. Running Checkdisk

You should run Checkdisk on all disks or at least on each disk you suspect.

Note: Checkdisk may take up to 30 minutes per disk per pass.

To run Checkdisk:

1. Enter the CLI. To enter the CLI from the Crash menu, enter option 1 (Enter CLI):
Enter option [Boot STANDARD] : 1
2. At the CLI> prompt, enter **x checkdisk**:
CLI> **x checkdisk**
3. When prompted, request that only new errors be reported.
4. Run at least one pass. The system returns you to the CLI> prompt.
5. Write down the HDA numbers.
6. Write down any other results.
7. Enter **bye** to return to the Crash menu:
CLI> **bye**

B.11. Recovering Crash Information from the Environment

If your console is *not* set up in such a way that you can capture its output, you can obtain the crash information displayed in section B.8 from the system error logs when the system boots.

To obtain the crash information from system error logs:

1. From an Environment command window, execute the following command:

```
Operator.Internal_System_Diagnosis
```

The system opens an I/O window containing an **EEDB**: prompt.

2. Enter **kernel** and press [Promote]:
EEDB: **kernel** [Promote]
The system displays a **Kernel**: prompt.
3. Enter **show_error_log** and press [Promote]:
Kernel: **show_error_log** [Promote]
The system displays a range such as:
first entry => 1; last entry => 440

4. Enter the first entry of 1 and press [Promote]:

```
FIRST [-10]: 1 [Promote]
```

5. Enter the number of the last entry (here 440) and press [Promote]:

```
LAST [-9]: 440 [Promote]
```

The system displays a message such as the following and returns you to the **Kernel**: prompt:

```
11:59:29 --- CRASH_ANALYZER INFO Analysis for Tombstone dated
11:59:29 ... CRASH_ANALYZER INFO 11:27:52 22-JAN-1992
11:59:30 --- CRASH_ANALYZER INFO
11:59:30 --- CRASH_ANALYZER INFO CRASH REASON (based on last micro-PC)
11:59:30 --- CRASH_ANALYZER INFO -----
11:59:30 --- CRASH_ANALYZER INFO 0206: cpu microcode error; refer to
11:59:30 --- CRASH_ANALYZER INFO
11:59:30 --- CRASH_ANALYZER INFO LAST CONSOLE OUTPUT
11:59:31 --- CRASH_ANALYZER INFO -----
11:59:31 --- CRASH_ANALYZER INFO 2KREAD_NOVRAM_DATA.VAL
...
11:59:38 --- CRASH_ANALYZER INFO MACHINE CRASH, I/O ADAPTOR
11:59:38 --- CRASH_ANALYZER INFO Restarting system after operations
console
11:59:39 --- CRASH_ANALYZER INFO
11:59:39 --- CRASH_ANALYZER INFO R1000 cpu analysis
11:59:39 --- CRASH_ANALYZER INFO -----
11:59:39 --- CRASH_ANALYZER INFO UCODE TRACE (last 20 entries)
11:59:39 --- CRASH_ANALYZER INFO -----
11:59:40 --- CRASH_ANALYZER INFO Microcode trace for Wcs version M207_17
11:59:40 --- CRASH_ANALYZER INFO 8A1
...
11:59:43 --- CRASH_ANALYZER INFO 34CF
11:59:43 --- CRASH_ANALYZER INFO 34D0
11:59:44 --- CRASH_ANALYZER INFO C 210
11:59:44 --- CRASH_ANALYZER INFO 206
11:59:44 --- CRASH_ANALYZER INFO
11:59:44 --- CRASH_ANALYZER INFO Micro Stack contents :
11:59:45 --- CRASH_ANALYZER INFO 0 - 34D0
...
11:59:47 --- CRASH_ANALYZER INFO 12 - 51F
11:59:48 --- CRASH_ANALYZER INFO 13 - 51F
11:59:48 --- CRASH_ANALYZER INFO 14 - 51F
11:59:48 --- CRASH_ANALYZER INFO 15 - 51F
11:59:48 --- CRASH_ANALYZER INFO
11:59:49 --- CRASH_ANALYZER INFO UCODE STATE
11:59:49 --- CRASH_ANALYZER INFO -----
11:59:49 --- CRASH_ANALYZER INFO Top => 04350C04 00000A80 (15)
...
12:00:01 --- CRASH_ANALYZER INFO
Kernel:
```

6. Enter quit and press [Promote]:

```
Kernel: quit [Promote]
```

The system returns you to the **EEDB**: prompt.

7. Enter quit and press [Promote]:

REDB: quit [Promote]

The system returns you to the Environment level.

8. Send to the Rational Response Center all the lines of the system error log that have the words "Crash_Analyzer Info" in them.

B.12. Contacting the Rational Response Center

Send the information you have gathered above to the Rational Response Center in Santa Clara, California.

- Via e-mail (TCP/IP Internet):
support@rational.com
- Via fax (addressed to the "Response Center"):
+1-408-496-3636 or
+1-408-496-3637 (use this second number only if the first line is down, not just busy)
Standard CCITT Group III telephone facsimile
9600 bps, automatic fall-back to 7200/4800/2400 bps
Unattended, 24-hour answering
- Via postal or other delivery service:
RATIONAL
Attention: Response Center
3320 Scott Boulevard
Santa Clara, California 95054-3197
USA
- Via telephone:
+1-408-496-3610 From anywhere
1-800-433-5444 From U.S. except California, Alaska, and Hawaii
1-800-533-5444 From California

Appendix C

Problem Reports Closed in D_12_6_5

D_12_6_5 closes many problem reports and customer-service requests. The following two subsections list:

- Software problem reports closed
- Documentation problem reports closed

C.1. Software Problem Reports Closed

Table C-1 lists the problem reports closed by D_12_6_5. It includes the PRS number of the problem, a brief description of the problem, and the section of this release information in which the problem is discussed in more detail (when applicable).

Table C-1 Software Problems Fixed by D_12_6_5

PRS Number	PRS Summary	Section
none	Csr5215: Internal_Error elaborating generics	7.6.2
none	Csr5260: Directory manager crash	7.15
none	Csr5261: Directory manager crash	7.15
none	Csr6016: Sequential_Io.Read does not work	7.9
12729-Star	Kernel elaborator assumes exceptions fit in 48 bits	7.6.3
0-0301-9	Add function to return if user is in a specified ACL group	6.1.2
0-0358-2	Goal parameter for Cmvc.Make_Spec_View is ignored for other directories	7.16
10270-Shei-Swb	Text of error message needs improvement	7.6.1
1000000-Sier-Geb	Problem with attribute in record clause	7.6.1
1000001-Sier-Geb	Access_List commands print incorrect symbols in logs	7.8
1000002-Sier-Geb	Compilation used ::: for messages that should have --	7.6
1140702-Gato-Gbd	Csr5310: Representation clause allowed after forcing occurrence	7.6.1
1711335-Gato-Gbd	Default expression on System_Backup.Backup	7.11
1704151-Shei-Jst	Error message: Reformat failed - please abandon image	7.1
1739656-Gato-Gbd	Csr6645: R1000 code generator fails with !Lrm.System.Operand_Class_Error	
2437110-Sier-Geb	Parser/Pretty_Printer destroys an image	7.1
2604297-Rati-Pbk	RXI tokens transfer during failed login	7.14.2
2617411-Gato-Gbd	Bad online documentation of Reject_Undefined_Pragmas switch	8.3
2679572-Roge-Srp	Ada bodies copied despite ACL restrictions	7.8
2809652-Clem-Jim	Cmvc.Make_Path is inflexible in its View_To_Import parameter	7.16
2949267-Etoi-Hts	Definition on <*Program_Library> locks the program library	7.6

Table C-1 Software Problems Fixed by D_12_6_5 (continued)

PRS Number	PRS Summary	Section
3068569-Cook-Swb	Semantics bug	7.6.2
3289131-Gato-Bas	Security problem: A way to find out user passwords	7.4
3336165-Shei-Jst	Treatment of expression prompts in semantics could be better	7.7.2
4123589-Blut-Jgp	RXI tokens are not consumed	7.14.2
4289272-Voya-Cmd	Login displays password-expiration warning with wrong password	7.4
4967545-Voya-Phil	Code generator fails when task entry is promoted to coded	7.6.2
5172201-Cook-Swb	Parser problem	7.1
5360143-Rati-Jls	Job.Kill is ignored by Common.Definition	7.4
5367576-Blut-Smp	Links.Edit and Display should require Owner access	7.8
5700190-Etoi-Tnu	Compiler bugs: anomalies found when using representation clauses	7.6.1
6242437-Blut-Vo	Typo in !Machine.Initialization.Rational.Terminals	7.15
6265678-Sier-Geb	Incremental edit fails invalidly	7.2
6398339-Blut-Med	RXI tokens are consumed erroneously at login	7.14.2
6452721-Shei-Jst	Problem with Ada.Make_Inline and Sm_Subunits attributes	7.2
6821610-Stan-Deg	Compilation.Load generates semantically incorrect spec	7.6.3
7030779-Zebr-Lore	Format causes most of the image to be deleted	7.1
7039962-Sier-Geb	Error message: Reformat failed - please abandon image	7.1
7135597-Gato-Gbd	Unsuccessful login message should not include user name	6.4
7311575-Wash-Jim	Cmvc.Initial is unfriendly	7.16
7483032-Gato-Jls	Field needs crash-dump tape tools in default Environment	7.13.1, B
7733953-Gato-Vnv	System-boot prompts and questions need to be simplified	7.13.1, B
8068463-Mago-Sdj	Pretty printer bug	7.1
8277786-Voya-Bas	Csr6102: System hangs; task In_Wait_Service	7.6.1
9123690-0128-6	Code generator fails on unit that should never have been installed	7.6.1
9123690-0128-9	Private types and generic formals (LRM 7.4.1 (4))	7.6.1
9123690-0131-7	Options parameter for Program_Library.Verify not documented	7.6.4
9123690-0131-8	New Archive options not documented in the specification	7.5
9123690-0133-9	Problem in restoring a system backup	7.11
9123690-0134-5	Problem with Ada.Show_Unused and enumeration literals	7.2
9123690-0135-8	Bad default parameter on System_Backup.Backup	7.11

Table C-1 Software Problems Fixed by D_12_6_5 (continued)

PRS Number	PRS Summary	Section
9123690-0136-8	Csr3956: Performance issue	7.7.1
9123690-0138-2	"<IMAGE>" in a searchlist can crash the Environment	7.4
9123690-0142-3	Problem in installing a unit under Delta2	7.6.1
9123690-0143-2	Suggested enhancements for coded units with statement prompts	7.7.2
9123690-0145-5	R1000 code generator fails with Constraint_Error (Variant)	7.6.2
9123690-0148-4	A way to find the sources that made Load_Proc is needed	6.2.3
9123690-0150-4	Bug in Library.Create	7.8
9123690-0151-1	Csr4723: Private parts required to be open for generics	7.6.2
9123690-0152-5	Csr4741: System hangs	7.15
9123690-0152-8	Csr4735: Bad compiler-error message	7.6.1
9123690-0153-1	Csr4836: Compilation.Parse fails because of excessive nesting	
9123690-0155-0	Information for IOC NOVRAM is incorrect	7.15
9123690-0155-1	Pm_Report calls fails to get IOC NOVRAM information	7.15
9123690-0155-9	Load_Proc Compute_Recoding does not behave properly	
9123690-0156-3	Csr4845: Problem in installing package with renames	7.6.1
9123690-0156-6	Csr4971: Cmvc.Make_Spec_View has problems	7.16
9123690-0156-7	Csr4786: All work lost after doing Common.Format	7.1
9123690-0157-9	Csr5005: Problem with error marks disappearing	7.6.1
9123690-0158-6	Suggested enhancement: Improved operations for verifying system backup	7.11
9123690-0159-0	Csr4740: Runaway jobs and disk space problems	7.3.1
9123690-0159-1	Csr5142: Runaway jobs and disk space problems	7.3.1
9123690-0160-1	Csr4788: 8-mm tape will not unload after aborted backup	6.2.2
9123690-0160-3	Csr4168: Attempt to reference offset above TOS	
9123690-0160-7	Csr4626: Debug_Tools.Get_Exception_Name fails if Retain_Delta1_Compatibility is False	7.3
9123690-0161-1	Dependency Object Editor demotes in incorrect order	7.2
9123690-0163-9	Csr5454: Definition fails on generic instantiations	7.1
9123690-0164-4	Port 16 dies after five minutes of use	7.15
9123690-0165-3	Csr5501: Definition on exception takes you nowhere	7.1
9123690-0165-6	Csr5588: Analyze_Crashdump runs all night	7.12.2
9123690-0165-8	Csr5559: Xon/Xoff port 16 does not work for Series 300s or 400s	7.15
9123690-0166-0	Editor jobs cannot be killed	7.4

Table C-1 Software Problems Fixed by D_12_6_5 (continued)

PRS Number	PRS Summary	Section
9123690-0166-5	Remote passwords should be able to be used from *SYSTEM jobs	7.4
9123690-0166-6	Csr5666: Unhandled-exception program error	7.6.2
9123690-0167-7	Csr5538: Assertion_Error from locally renamed private type	
9123690-0168-1	Csr5723: Elaboration of Unbounded_String bombs in D_12_5_0	7.6.3
9123690-0168-2	Csr5773: Print command ignores option for sets of Ada units	7.10
9123690-0168-8	Csr5710: Problem exists with FTP queues	7.10
9123690-0169-0	Csr5707: Definition problem with exception defined in generic	7.1
9123690-0169-8	Invalid rename statement not detected by compiler	7.6.1
9123690-0171-3	Csr5841: CMVC problems with D_12_5_0	7.16
9123690-0171-8	Csr5650: Unsatisfiable import requirements	7.6.3
9123690-0172-6	Csr5917: Jobs hang on quit (IN FA QUEUE); debugger hangs	7.3.1
9123690-0173-8	Csr5990: Log_Reader.Load_Logs always adds today's log	7.15
9123690-0175-9	Code vanishes when [Format] is pressed	7.1
9123690-0180-7	Csr4844: Semantic errors with valid representation clause	7.6.1
9123690-0181-8	Csr6261: Semanticizing problem; image destroyed	7.1
9213690-0183-3	Csr6091: System hangs, idle; Job 4 stuck attempting entry	7.3.1
9176782-Gato-Gbd	Problem with Check_Universe_Acls and system access to !Machine.Initialization	7.15
9723630-0011-5	No function exists to check if User_Id is in a group	6.1.2
9723630-0025-8	Login sequence: Welcome output before username prompt	6.4.2
9723630-0074-7	A way to modify the login banner should be added	6.4.1
9723630-0120-7	!Commands.Dependents.Find consumes actions	7.2
9723630-0122-0	Cannot browse to generic type declarations	7.1
9924694-Mago-Sdj	Bug in incremental semantics	7.2

Table C-2 lists the problem reports that have been investigated and closed because of one of the following reasons: the problem could not be reproduced in D_12_6_5; the problem has been fixed in a previous Environment release; there is not enough information about the problem to proceed.

Table C-2 Not Reproducible in D_12_6_5

PRS Number	PRS Summary
6583-Star	Unhandled exception in R1000 code generator
6705-Star	Anomaly in Search_List and name resolution
12676-Star	Unhandled exception during code generation of variant records
0-0004-1	Interface crash
0-0174-0	Problem with access controls on release tapes of the Environment
0-0192-3	Incremental promote fails for function
0-0245-0	Tape automatically unloaded
0-0245-3	Add_To_Acl interface needed
0-0245-8	Problem with verification of backup tapes
0-0264-4	Error promoting an anonymous Ada object from archived state
0-0297-1	Use clause removed
0-0298-3	Several Archive.Copy procedures in one job often fail
0-0311-2	Illegal links are not detected
1013207-Zebr-Jim	Load_Procs do not archive well
1460343-Sier-Geb	Promoting modified activity results in Lock_Error
1516292-Wash-Jim	Inability to find cause of Storage_Error
1773909-Shei-Jst	New Print command dies on Ada unit
1934610-Wash-Jim	Networking will talk but will not listen
2300105-Shei-Jst	Problem with Archive.Copy and session switches
2465646-Mago-Sdj	Unhandled exception in code generation
2474799-Wash-Jim	Exception encountered during R1000_Code_Generator.Initialize
2606070-Rati-Jls	Links.Delete does not accept wildcards
2785456-Wood-Peg	Unhandled exception from the Core Editor
2823763-Jazm-Smc	Windows larger than 80 columns wide occur in RXI
29236-Capi-Ray	Problem with activity when spec/load views are on different lines
3036900-Wash-Jim	Inconsistency in Common.Promote and Ada.Code_Unit
3330141-Clem-Jmk	Problem with Archive.Copy and !Machine.Tcp_Ip_Name_Server
345551-Zebr-Jim	Editor.Char.Tab_Backward does not work on empty space
3560244-Pica-Jim	Debugger cannot show locations of breakpoints
3758781-Wood-Jim	Nonexistent_Page_Error after user interrupt
3780027-Rati-Jls	Window commands do not accept numeric arguments
4452630-Calv-Jim	Changed option of Archive no longer seems to work
4489106-Shei-Jst	Problem with Activity Editor selections
4697876-Gato-Gbd	Csr5082: Index error for Name_Gen.Importing.Note_Used_Imports

Table C-2 Not Reproducible in D_12_6_5 (continued)

PRS Number	PRS Summary
4761811-Roge-Smc	RXI seems to hang/break R1000 connection
4971889-Clem-Jim	Common.Promote only does Format
5250100-Zebr-Jim	Assertion_Error deletes Load_Proc
5384534-Smau-Phil	Queue.Display does not end its line in the message window
5390266-Blut-Smp	Problem with Time_Uilities and Log.Put_Line around midnight
6123643-Rang-Oper	Generic instantiation fails (error in checking array bounds)
6194736-Roge-Smc	RWI cursor gets confused by a command window attached to a message window
6517153-Flor-Jst	Problem using Set_Target_Key with '?'
6801458-Gato-Jls	Editor error after [Object] - [G] performed on message window
6862896-Blut-Smp	Output to message window is lost for most commands
7067816-Wash-Jim	Disappearing procedure and Ada.Install_Stub
7102190-Zebr-Jim	Problem in compilation
7245446-Voya-Jim	Editor exception is using screen cursors
7677278-Cook-Jst	File_Uilities.Find/Text_Coupler (?): Bad error message
7746569-Pica-Jim	Console_lo does not appear
778583-Voya_Drl	RWI does not always redraw correctly
7821978-Clem-Marl	Resolving "<TEXT>" fails
7872950-Shei-Swb	Ada front-end bug
8181636-Shei-Jst	Archive option Changed_Objects not working
9123690-0126-0	Csr2952: System_Report.Generate (Tape_Mounts) crossing 12 P.M.
9123690-0128-7	Crashes that look like Operator shutdown of system occur
9123690-0132-1	Csr3614: Primary backup hangs after recovery
9123690-0132-2	Problem with remote-password prompt window
9123690-0138-1	System crashes when rebooting
9123690-0138-5	Delta2 Cmvc.Initial complains about ACLs
9123690-0140-0	Csr3985: RXI does not pick up defaults file for menu fonts
9123690-0140-6	*Kernel Shutdown should be removed from Kernel CI
9123690-0140-8	Csr4067: Duplicate error logs created by reboot server
9123690-0143-3	Csr4075: System hung
9123690-0144-8	Csr4327: Error message no tokens available is incorrect
9123690-0146-2	Cmvc.Show, Check_In, and Check_Out will not raise exceptions
9123690-0146-5	Csr4470: User Rational is using layered-product tokens

Table C-2 Not Reproducible in D_12_6_5 (continued)

PRS Number	PRS Summary
9123690-0149-1	Cannot demote or promote unit(s) from installed state
9123690-0150-9	Csr4572: Bug in Transport_Interchange.Convert
9123690-0151-5	Reject_Inevitable_Exceptions rejects correct code
9123690-0154-1	Machine appears stuck (IN_FA_QUEUE)
9123690-0154-3	Csr4934: Delayed secondary backup does not delay
9123690-0155-5	Compiler: Type Identity has an index with different bounds
9123690-0156-4	Csr4655: Problem with PM report
9123690-0156-5	Unexpected exception R1000_Code_Generator.Initialize during boot
9123690-0157-3	Csr5058: Internal error: Unsatisfiable import requirement
9123690-0158-0	Csr5077: Check_Universe_Acls results in unhandled exception
9123690-0159-2	Problem with derived type of String
9123690-0159-7	Csr5174: Wrong profile used
9123690-0161-8	Csr5042: Installing a unit returns Spec of with ...
9123690-0165-5	Csr5540: Missing Controlled_Operations in D_12_5_0
9123690-0174-6	Csr2242: Cmvc.Make_Path does not work with archived units
9123690-0176-9	Bug in package Window_Io
9123690-0178-2	Archive fails to restore because of bad Tool_State switches
9123690-0178-3	Diff never completes in Cmvc.Check_In
9123690-0178-4	!Lrm.System.Operand_Class_Error and Illegal_Reference raised
9123690-0178-8	Code-generation error
9123690-0178-9	Semantically incorrect Ada unit executable on Rational
9123690-0179-0	Library.Create_World can create an incorrect world
9123690-0180-1	Error during incremental changes
9123690-0180-3	Disable_Job does not work from Kernel prompt
9445015-Shei-Dian	Editing switch files without proper ACLs locks switch file
9723630-0009-2	Ugly message logged on demote
9723630-0021-9	Problem with accepting changes between different subpaths
9723630-0023-8	Cmvc.Accept_Changes bug
9723630-0090-5	Erasedisk should be modified to allow different number of passes
9723630-0096-2	Crash during D_12_1_1 installation with message kernel assert failure
9723630-0114-1	Archive.Restore giving layout error

Table C-2 Not Reproducible in D_12_6_5 (continued)

PRS Number	PRS Summary
9723630-0123-0	Csr2410: Editor job hogs CPU

C.2. Documentation Problem Reports Closed

Table C-3 lists the problems reports that have been closed by the release of new online and printed documentation for the Session and Job Management (SJM), Library Management (LM), and System Management Utilities (SMU) books of the *Rational Environment Reference Manual* (ERM). The table includes the PRS number of the problem, a brief description of the problem, and the document in which the problem has been fixed. All problems indicated as fixed in a specific document also have been fixed in online help.

Table C-3 Documentation Problems Fixed by D_12_6_5

PRS Number	PRS Summary	Book
5653-Star	Problem with !Tools.Debug_Tools.Get_Exception_Name	DEB
5742-Star	Debugger documentation for Put is inaccurate	DEB
6559-Star	Debugger documentation has an error	DEB
7025-Star	Newly created session defaults need documentation	SJM
7074-Star	Package descriptions lack an overview of package contents	SJM, LM, SMU
9759-Star	Maximum for Job_Page_Limit is not documented	SJM, LM
10138-Star	Default parameter value for Profile.Warn is incorrect	SJM
10240-Star	Description of Library.Space does not match the specification	LM
10522-Star	Documentation for Overwrite option of Archive.Restore is incorrect	LM
10557-Star	Documentation for Compilation.Load is insufficient	LM
11456-Star	Typo on page SMU-1	SMU
12059-Star	Documentation for pragma Main and pragma Loaded_Main is insufficient	LM
0-0244-9	Documentation for Overwrite option of Archive commands is incorrect	LM
0-0282-4	Misleading documentation on page LM-316	LM
0-0282-7	Documentation of the File, Directory, and Ada daemons is incorrect	SMU
0-0284-2	No error message is given when Archive.Copy is used on nonexistent objects	LM

Table C-3 Documentation Problems Fixed by D_12_6_5 (continued)

PRS Number	PRS Summary	Book
0-0317-9	Description of Format.Consistent_Breaking library switch has an error	LM
0-0334-5	Example on page SMU-38 is incorrect	SMU
0-0358-6	Online help for !Commands.Terminal.Set_Logoff_On_Disconnect is incorrect	SMU
1495461-Gato-Rjg	Online help for Mail session switches is missing	Online
2023294-Roge-Jaf	Reject_Undefined_Pragmas library switch displays incorrect help	Online
3201747-Voya-Phil	Documentation of interaction of Enable_Privileges and editors is inadequate	SMU
729632-Gato-Vnv	Documentation is incorrect for Overwrite option of Archive	LM
8466957-Rati-Jls	Suggested enhancement to documentation for package Archive	LM
8594091-Rati-Pbk	Incorrect explanation for Session.Prompt_Delimiters session switch	SJM
8692100-Gato-Mpr	Online help crossed for Session.Window_Frames and Window_Default_Sort_Order session switches	Online
9123690-0131-5	Documentation is insufficient for prefix expressions in Archive	LM
9123690-0131-9	Operator.Enable_Privileges does not work for members of group Privileged	SMU
9123690-0136-2	Csr3923: Documentation is insufficient for enabling a laser printer	Section A and SMU
9123690-0137-2	Csr3975: Session.Search_Regular_Expr session switch does not work	SJM
9123690-0166-4	Csr5617: Session switches do not apply when Ftp.Get is run under *SYSTEM	SJM
9723630-0021-2	Format.Alignment_Threshold switch description is insufficient	LM
9723630-0036-0	No documentation exists for Host parameter of Queue.Add	SMU
9723630-0058-1	Documentation is incorrect for Log.Set_Output and Log.Set_Error	SJM
9723630-0099-2	Spec and documentation for Tape.Write do not match	SMU

Installation Procedure for X_Interface (RXI) Release10_9_0

RXI gives users way to access the Rational Environment from a workstation using the X Window System or an X-based terminal.

Review the complete Installation Procedure before proceeding.

Warning: *Be sure to complete ALL of the phases of the installation exactly as written and follow the installation steps in each phase in a serial manner. Failure to adhere to these instructions will often result in the failure of the product.*

1. Impact of the Release

Installation of this product may require that the R1000 be rebooted in order for the new keymaps to take affect. Machines running the Delta 2 (D_11_4_1, or later) release of the Environment do not need to be rebooted.

2. Installation Checklist

Main Installation Checklist

✓	Do	Step
<input type="checkbox"/>	Read	"Do_Step Information", Section 3, page 3, if not familiar with the Do_Step command. This command is used throughout the installation process.
<input type="checkbox"/>	Read	"Packaging Information", Section 4, page 5 to verify that the release is complete. The installation will fail if any of the pieces are missing.
<input type="checkbox"/>	Read	"Installation Overview", Section 17, page 36 for an overview of the RXI installation process. Common installation problems and issues are covered in this section. IF YOU HAVE NOT INSTALLED RXI BEFORE, READ THIS SECTION!
<input type="checkbox"/>	Read	"Supported Configurations", Section 5, page 7 and pick the appropriate RXI Terminal-Type(s) to be installed. There may be more than one.
<input type="checkbox"/>	Record	Your choice of RXI Terminal-Type(s) here (from the "Supported Configurations" section) _____. (There may be more than one.)
<input type="checkbox"/>	Record	The workstation-specific installation chapter here (from the "Supported Configurations" section) _____. (There may be more than one.)
<input type="checkbox"/>	Read	"Supported Configurations", Section 5, page 7 to verify that installation prerequisites for are met. The installation, or the installed product, will fail if the prerequisites aren't met.
<input type="checkbox"/>	Install	The keymaps for your chosen RXI Terminal-Type(s) on the R1000 using "Installation on R1000", Section 6, page 9.
<input type="checkbox"/>	Install	RXI on the workstation(s) and/or filesaver(s) using the proper workstation-specific installation section(s) recorded above.
<input type="checkbox"/>	Test	The RXI Terminal-Type(s) to verify installation was successful. See "Test", Section 7, page 12.
<input type="checkbox"/>	Fill Out	Fill out the "Feedback" form at the end of this install note.
<input type="checkbox"/>	Clean up	RXI installation areas on the R1000 and workstation. See "Clean Up", Section 8, page 13.
<input type="checkbox"/>	Done	Installation complete.

3. Do_Step Information

The majority of the steps which must be performed for this installation utilize the procedure `Do_Step` (included as part of the release tape and located in `!Commands`) to execute the required sequence of commands which implement the step. This procedure **must always** be executed from a command window in the release library

```
!Machine.Release.Archive.X_Interface.Release10_9_0
```

Each step which utilizes procedure `Do_Step` will be of the form

```
<STEP_NAME>                [<TIME TO EXECUTE>]
                             <DESCRIPTION>
```

where `<STEP_NAME>` is passed as the parameter to procedure `Do_Step` which performs the necessary actions to complete the step, `<TIME TO EXECUTE>` is the amount of time the step takes to execute (this may be expressed as a range, in which case the lower values is typically the minimum, and the upper value is an estimate of the maximum), and `<DESCRIPTION>` is a description of what the step does, including any action which you will need to manually perform. For example:

```
FOO                          [5 Minutes]
                              This step is an example of the format used for
                              steps which are executed using procedure
                              Do_Step. To execute step FOO, you would go to
                              !Machine.Release.Archive.X_Interface.
                              Release10_9_0 and in a command window,
                              execute:
```

```
Do_Step ("FOO");
```

which would result in automatic execution of all commands required to implement this step. If you needed to take any manual action in addition to executing `Do_Step`, it would be noted as :

Perform some manual check

If any errors occur for this step, you should always fix the problem before proceeding on to the next step. Each step is defined by a fragment of Ada code which is executed by `Program.Run`. These fragments are stored in the `Steps` file located in the `Command_Data` library of the release. In the event you want to modify a step, you can use the "PROMPT => `<STEP>`" form when invoking `Do_Step`. See the spec of procedure `Do_Step` (located in `!Commands`) for more detailed information.

Warning: *If you interrupt the execution of Do_Step (by using Job.Interrupt such as CONTROL-G) it is possible that certain interactive commands executed by some steps, such as Common.Definition, may fail with an exception and display a message such as*

Unable to read file due to Constraint_Error (Null Access)

In general, this message has no negative impact on the execution or completion of the step, and can be ignored.

4. Packaging Information

The installation materials for this release are:

- This Installation Procedure
- X Interface Release10_9_0 Tape
- X Interface Release10_9_0 Release Note

See the X_Interface Release10_9_0 SMSE Checklist for the complete list of the items that make up this release. Contact Rational if any items on the SMSE Checklist are missing.

The following Product Numbers apply to this release:

3000-00033	3000-00449	3000-00539
3000-00034		
3000-00035	3000-00477	3000-00541
3000-00036		
3000-00037	3000-00491	3000-00610
3000-00312	3000-00501	3000-00641
3000-00313		3000-00642
3000-00314	3000-00518	3000-00643
3000-00315	3000-00519	3000-00644
3000-00316	3000-00520	3000-00645
	3000-00521	
	3000-00522	3000-00429
		3000-00432
3000-00384	3000-00523	3000-00433
3000-00385	3000-00524	3000-00434
3000-00386	3000-00525	3000-00435
3000-00387	3000-00526	
3000-00388	3000-00527	
3000-00440	3000-00528	
3000-00441	3000-00529	
3000-00442	3000-00530	
3000-00443	3000-00531	
3000-00444	3000-00532	
3000-00445	3000-00535	
3000-00446	3000-00536	
3000-00447	3000-00537	
3000-00448	3000-00538	

5. Supported Configurations

The customer site must have all of these before an RXI installation can proceed.

- A workstation/fileserver (or an X-based terminal with a separate host workstation), with:
 - one of the supported machine architectures,
 - running one of the supported operating system versions,
 - and, TCP/IP network connections allowing it to connect to the R1000(s).
- The workstation, operating system, and TCP/IP must be installed and operational before RXI can be installed.
- A workstation or an X-based terminal with:
 - A supported keyboard,
 - and, a supported version of the X Window System.
- The workstation/terminal and the X Window System must be installed and operational before RXI can be installed.
- R1000 with Environment release D_10_20_0 or later.
- A user community that has been notified of installation down time (if applicable).

RXI has been developed and tested on the following workstation configurations. This release of RXI may work with other versions of operating or window systems. However, as only these configurations have been tested, Rational can not support, or guarantee that other configurations work.

Note: These configurations can be used in any combination. That is, the RXI executable can be running on any supported binary host machine, while, the RXI window appears on any supported X server, allowing, the user to use any supported keyboard attached to that server, connected with, any R1000 with an installed keymap for that keyboard/X-server combination.

X_Interface Release10_9_0 Supported Workstation Configurations

Workstation	Operating System	Window/ Network System	Keyboard Type	RXI Terminal Type	Work-station Install Section
Apollo	Domain/OS SR10.2	Domain/X11 V1.2	Model 7121 (no Alt/Meta keys)	Xapollo3_Sr10_2_Unix	9, page 14
Apollo	Domain/OS SR10.2	Domain/X11 V1.2	Motif (Alt/Meta keys)	Xapollom_Sr10_2_Unix	9, page 14

X_Interface Release10_9_0 Supported Workstation Configurations (continued)

Workstation	Operating System	Window/ Network System	Keyboard Type	RXI Terminal Type	Workstation Install Section
DEC DECstation	Ultrix V4.1	DECwindows	LK201	Xultus_Ds_Ultrix4_1	12, page 20
DEC VAXstation	Ultrix V4.1	DECwindows	LK201	Xultus_Vs_Ultrix4_1	12, page 20
DEC VAXstation	VMS 5.1B	DECwindows, Multinet 2.0.3 Rev.D	LK201	Xdecus_Vms5_1_Multinet203	10, page 17
DEC VAXstation	VMS 5.1B	DECwindows, Wollongong 5.0.2	LK201	Xdecus_Vms5_1_Wollongong502	10, page 17
HP 9300	HP-UX 7.0	X Windows	46021A	Xhp46021a_Hpux7_0	13, page 22
IBM RS6000	AIX 3.1	AIX/R2 X-Windows 3.1	IBM U.S.	Xr6us_Aix3_1	14, page 24
Sun M680x0	SunOS 4.0	Sun X11/NeWS V2.0 (Openwindows)	Type 3 (F1..F9)	Xnews3_V2_0	15, page 27
Sun Sparc	SunOS 4.0	Sun X11/NeWS V2.0 (Openwindows)	Type 3 (F1..F9)	Xnews3_V2_0_Sparc	15, page 27
Sun M680x0	SunOS 4.0	Sun X11/NeWS V2.0 (Openwindows)	Type 4 (F1..F12)	Xnews4_V2_0	15, page 27
Sun Sparc	SunOS 4.0	Sun X11/NeWS V2.0 (Openwindows)	Type 4 (F1..F12)	Xnews4_V2_0_Sparc	15, page 27
Sun M680x0	SunOS 4.0	MIT X11.R3, X11.R4	Type 3 (F1..F9)	Xsun3_X11r3	16, page 31
Sun Sparc	SunOS 4.0	MIT X11.R3, X11.R4	Type 3 (F1..F9)	Xsun3_X11r3_Sparc	16, page 31
Sun M680x0	SunOS 4.0	MIT X11.R3, X11.R4	Type D43 (F1..F12)	Xsun4_X11r3	16, page 31
Sun Sparc	SunOS 4.0	MIT X11.R3, X11.R4	Type 4 (F1..F12)	Xsun3_X11r4_Sparc	16, page 31

6. Installation on an R1000

6.1. Log in as Rational

Log in as user `rational`, which must be a member of group `Privileged`. Do NOT use another account due to problems which might arise with product token usage.

6.2. Mount RXI Tape

Mount the RXI Release10_9_0 tape on the tape drive. It should be read-only.

6.3. Read RXI Archive

Restore the RXI Release10_9_0 tape on the system:

Create a command window and restore the tape onto the system. Some systems may have a predefined `Do_Step` to load the tape. Try `Do_Step (Load_Tape)`; to load the tape. If this fails load the tape manually with the command:

```
Operator.Enable_Privileges;
if Operator.Privileged_Mode then
    Archive.Restore (Options => "Replace, Promote");
else
    Io.Put_line ("User does not have privileges.");
end if;
```

Answer the appropriate tape mount requests on the system console.

This will restore `!Machine.Release.Archive.X_Interface.Release10_9_0` (after about 10 minutes).

6.4. Go To Archive Area

Go to the `!Machine.Release.Archive.X_Interface.Release10_9_0` library.

6.5. Check Authorizations

AUTHORIZATION_CHECK

[1 minute]

This step will verify that `x Interface` is authorized and `x Interface` tokens are available on this machine. If there are products which need to be authorized, this will be indicated and a command window created with a call to the procedure `Product_Authorization.Register`. Modify the appropriate parameters and authorize the products required.

6.6. Restore Release

RELEASE_RESTORE

[30 minutes]

This step restores the **Release** archive. When completed, a filtered error log is displayed. Examine this log for errors, ignore errors about switches and links.

6.7. Restore Release Notes

RESTORE_NOTES

[1 minute]

This step restores the **_PS** and **_LPT** versions of the **RXI** release note to **!Machine.Release.Release_Notes**.

6.8. Go To Release Area

Traverse to **!Machine.Release.X_Interface.Release10_9_0**.

6.9. Install Keymap Files

Install the keymap files and update the **Terminal_Type** and **Terminal_Recognition** files on the R1000. This step must be repeated for each **RXI** terminal type you have selected for use with this R1000. For example, to install the files for a DEC workstation under the VMS operating system, enter the following in a command window:

```
Rxi_Install.Machine_Editor_Data_Files
  (Keyboard => Rxi_Install.Xdecus_Vms5_1_Multinet203,
   Suppress_Rxi_files => False);
```

and press the [PROMOTE] key. If you are installing more than one terminal type on this R1000, you can save time by setting the **Suppress_Rxi_Files** parameter to **True** after this command has been executed once.

6.10. Enable New Keymaps

Enable the new keymaps. The new **RXI** keymaps must be enabled before they can be used. On machines running an Environment release prior to **D_11_4_1** (D2 Beta) the R1000 must be rebooted. On machines running Environment release **D_11_4_1** or later, the procedure **Refresh_Terminal_Information** may be used.

- On machines running an Environment release Prior to **D_11_4_1**, reboot the R100 for the new keymaps to take affect. The best time to reboot the R1000 is after you have transferred the **RXI** source files to the workstation(s) and/or fileservers(s), this way you can install **RXI** on the workstation while the R1000 is rebooting.

Use the instructions in the following sections to transfer and install **RXI** on your workstation(s) and/or fileservers(s). After the files have been transferred to the workstation(s) and/or fileservers(s) reboot the R1000. In a Command window, type:

```
Schedule_Shutdown
  (At_Time      => ">> Some Future Time <<",
   Reason       => "Release",
   Explanation  => "Install RXI Release10_9_0");
```

and press the [PROMOTE] key when you are ready to reboot the R1000.

- On machines running Environment release D_1_4_1 or later, create a Command window and type:

```
Refresh_Terminal_Information
```

and press the [PROMOTE] key to enable the keymaps.

7. Installation Testing

The RXI product is keyboard and X server specific. This means that it must be uniquely and specially configured for each and every keyboard and X server that will be used. In order for RXI to work correctly, a user must log into an R1000 using an RXI terminal type name that correctly, and uniquely, specifies the keyboard-layout, the X server, and usually the keyboard-vendor and X software-vendor that is being used.

For example, a Sun Microsystems(tm) workstation using the SunOS(tm) operating system can have a Sun type-4 keyboard. If a user will be using the MIT X Window System then his terminal type is **XSun4**. However, if a user will be using the Sun OpenLook (a.k.a. OpenWindows or NeWS) then his terminal type is **XNews4**. The keyboard is the same keyboard in either case; however, the software being used will cause the keyboard to appear different to RXI and thus to the R1000.

This means that, in order to properly test an RXI installation, it is necessary to log into an R1000 from each and every keyboard type that will be used. Most sites will only use one or two keyboard types so this is not usually difficult.

After the installation is complete, log-on to the R1000 from each variety of workstation or X-based terminal keyboard. If more than one vendor's X Window System software will be used, make sure to log in once from each keyboard while using each of the various systems. Try some function keys and menus to verify that the keyboard, X Window System software, and R1000 keymap combination is working correctly.

8. Installation Clean Up

After the installation is complete, you may delete some objects to save some disk space.

On The R1000:

- `!Machine.Release.Archive.X_Interface.Release10_9_0`. After these archives have been restored they are of no use and may be deleted.
- `!Machine.Release.X_Interface.Release10_9_0`. If you are never going to install additional terminal types this may be deleted. Do not delete this if you think additional workstations or terminal types will be required.

On The workstation(s) and/or fileserver(s):

- Everything in the temporary source directory may be deleted after the installation of RXI is complete.

9. Installation using Apollo Domain/OS and Domain/X11

The steps involved in getting the rxi program running on a workstation or fileserver are (NOTE: These instructions are for use with the Unix shell not the Domain/Aegis shell):

1. Create a temporary source directory for the rxi program on the workstation/fileserver. For example, if program sources are usually kept in a directory called `/src` then within this directory create a directory called `/src/rxi`.
2. Traverse to `!Machine.Release.X_Interface.Release10_9_0` on the R1000.
3. Run the `Rxi_Install.Workstaion_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. Use the `Rxi_Install.XApollo3_Sr10_2_Unix` or the `Rxi_Install.XApollom_Sr10_2_Unix` terminal types. For example:

```
Rxi_Install.Workstaion_Files
  (Keyboard =>
    Rxi_Install.XApollo3_Sr10_2_Unix,
  On_Machine => "My_Machine",
  Username   => "MyName",
  Password   => "MyPassword",
  Account    => "",
  Rxi_Source_Directory => "/src/rxi");
```

4. Connect to the source directory on the workstation and execute the command `make restore`. This unpacks rxi and makes it available for complete installation. For example:

```
cd /src/rxi
make restore
```

5. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the rxi Environment menu entries for that product.
6. Edit the file called `Makefile`. A particular site may wish to change the places where rxi, its fonts, and/or its man page are installed. These can be changed by changing the definition of:
 - BINDIR** - where the rxi program goes;
 - usually `/usr/bin/X11`
 - XBINDIR** - where the X Window System executables currently reside;
 - usually `/usr/bin/X11`
 - XAPFLOADDIR** - where the application defaults files reside;
 - usually `/usr/lib/X11/app-defaults`
 - FONTDIR** - where the compiled fonts live;
 - usually `/usr/lib/X11/fonts/misc`
 - MANDIR** - where the system-wide "new" man pages live;
 - usually `/usr/new/catn`

7. You now have three more choices. A) Install the Rational-supplied rxi as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate rxi using all local definitions. The supplied version of rxi should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist.
 - a. If you wish to just use rxi as supplied then skip this step.
 - b. If you wish to simply relink rxi so that it uses local libraries then execute the `make relink` command and proceed to the next step.
 - c. If you wish to recreate rxi from scratch then execute the `make clean` command followed by `make all`.
8. Switch users to "super-user" and execute `make install.all`; this will:
 - Install the rxi program, the RXI default resource file, and any support programs needed by RXI users. (The command `make install` performs just this step.)
 - Install the rxi man page. (The command `make install.man` performs just this step.)
 - Install the rxi terminfo description. This allows terminfo-using programs (such as vi) to operate within an rxi window. (The command `make install.terminfo` performs just this step.)
 - Compile the various fonts `fixed-*.bdf` into the `$(FONTDIR)` directory. These are the normal and bold-face fonts used by rxi. (The command `make install.fonts` performs just this step.)

Some sites may not want to use the Rational supplied fonts. They can skip the `install.man` and `install.fonts` steps if they wish. They will want to edit the `RXI.xdefaults` file to change the names of the default rxi fonts.

9. Run rxi and decide whether or not the default font is too small. Some Apollo displays have a very small pixel size. This can make the default font too small for many users. RXI comes with 8 different fonts. They are:
 - `fixed-screen-{r,b}-11` - default font - smallest "thin" font
 - `fixed-screen-{r,b}-12`
 - `fixed-screen-{r,b}-13`
 - `fixed-screen-{r,b}-14` - largest "thin" font - 1 pixel line thickness
 - `fixed-screen-{wr,wb}-15` - smallest "thick" font - > 1 pixel thickness
 - `fixed-screen-{wr,wb}-16`
 - `fixed-screen-{wr,wb}-17`
 - `fixed-screen-{wr,wb}-22` - largest "thick" font

You can try out the different fonts with command lines like these, restart your X server before trying this:

```
shell> xset fp rehash # Tell server to read new RXI fonts.
shell> rxi -fn fixed-screen-r-13 -fb fixed-screen-b-13 &
shell> rxi -fn fixed-screen-wr-15 -fb fixed-screen-wb-15 &
```

Always use matching normal (-fn) and bold (-fb) fonts or your rxi window may be unreadable when connected to an R1000.

If you decide to change the system-wide default font then edit the `RXI.Xdefaults` file. Change the `*vt100.font:` and `*vt100.boldFont:` lines to reference the new font. You may also want to change the `*vt100.menuFont:` line if you find the menus to be unreadable as well. Then execute `make install` to install the new system-wide defaults.

10. Installation using Vax VMS and DECwindows

The steps involved in getting the rxi program running on a DEC VAXstation workstation or fileserver are:

1. Create a temporary source directory for the rxi program on the workstation/file-server. For example, `User: [Rxi]`.
2. Traverse to `!Machine.Release.X_Interface.Release10_9_0`.
3. Run the `Rxi_Install.Workstaion_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. Specify the main directory (eg. `User: [Rxi]`) as the destination of the transfer. For example:

```
Rxi_Install.Workstaion_Files
  (Keyboard =>
    Rxi_Install.Xdecus_Vms5_1_Multinet203,
  On_Machine => "My_Machine",
  Username   => "MyName",
  Password   => "MyPassword",
  Account    => "",
  Rxi_Source_Directory => "User: [Rxi]");
```

4. Do a `SET DEFAULT` to the source directory on the workstation and execute the command `@makefile restore`. This unpacks rxi and makes it available for complete installation.
5. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the rxi Environment menu entries for that product.
6. Edit the `MAKEFILE.COM` file in the source area. Follow the editing instructions at the front of the file. You will be asked to specify the TCP/IP library to use (Excelan MultiNet or Wollongong WIN TCP/IP) and to specify the directories containing the library include files and the library `.OLB` file. This is also the place where you indicate the proper installation directories and protections for the various pieces of the RXI system.
7. You now have three more choices. A) Install the Rational-supplied rxi as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate rxi using all local definitions. The supplied version of rxi should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist. NOTE: The optional C compiler is required to relink or recompile rxi.
 - a. If you wish to just use rxi as supplied then skip this step.
 - b. If you wish to simply relink rxi so that it uses local libraries then execute `@makefile relink` command and proceed to the next step.

- c. If you wish to recreate rxi from scratch then execute `@makefile clean` command followed by `@makefile all`.
8. Enable your privileges; `BYPASS` should be sufficient. Execute `@makefile install.all`; this will:
 - Installs the rxi program, its helper program `rxi_detached`, and the RXI default resource file. (The command `@makefile install` performs just this step.)
 - Install the rxi help library. (The command `@makefile install.help` performs just this step.)
 - Compiles and installs the various fonts `fixed-screen-*.bdf` in the `FONTDIR` directory (defaults to `SYS$COMMON:[SYSFONT.DECW.USER_75DPI]`). These are the normal and bold-face fonts used by rxi. (The command `@makefile install.fonts` performs just this step.)
9. Edit the `SYS$MANAGER:SYLOGIN.COM` file and add a definition for a system-wide symbol named `RXI`. eg.

```
$ RXI := $ SYS$COMMON:[SYSEXE]RXI_DETACHED.EXE
```

This creates a system-wide foreign command that will run rxi as a detached process. You will have to logout and log back in to get it defined for yourself.

10. Reboot any and all DECwindows workstations (or just the servers) that need to use rxi. The reboot is necessary in order to cause the servers to recognize the new rxi fonts.
11. You may wish to do a `PURGE` on: `SYS$SYSTEM:`, `DECW$SYSTEM_DEFAULTS:`, and `SYS$COMMON:[SYSFONT...]`.

11. New or Changed System Files

All files are installed with `OWNER=SYSTEM` and a protection more or less of `/PROTECTION=(SYSTEM:RWED,OWNER:RWED,GROUP:RE,WORLD:RE)` as appropriate. See the `Makefile.Com` file for complete details on what protection is given to what class of file.

Symbol `FONTDIR` (usually `SYS$COMMON:[SYSFONT.DECW.USER_75DPI]`)

new fixed-screen-r-11.decw\$font (various fonts)
 new fixed-screen-b-11.decw\$font
 new fixed-screen-r-12.decw\$font
 new fixed-screen-b-12.decw\$font
 new fixed-screen-r-13.decw\$font
 new fixed-screen-b-13.decw\$font
 new fixed-screen-r-14.decw\$font
 new fixed-screen-b-14.decw\$font
 new fixed-screen-wr-15.decw\$font
 new fixed-screen-wb-15.decw\$font
 new fixed-screen-wr-16.decw\$font
 new fixed-screen-wb-16.decw\$font
 new fixed-screen-wr-17.decw\$font
 new fixed-screen-wb-17.decw\$font
 new fixed-screen-wr-22.decw\$font
 new fixed-screen-wb-22.decw\$font

changed none

Symbol BINDIR (usually SYS\$COMMON:[SYSEXE])

new rxi.exe (main executable)
 new rxi_detached.exe (utility program)

changed none

Symbol HELPLIB (usually SYS\$HELP:HELPLIB.HLB)

new none

changed HELPLIB.HLB (add RXI help data)

Symbol XAPPLOADDIR (usually SYS\$COMMON:[DECW\$DEFAULTS.SYSTEM])

new rxi.dat (system default file)
 new rxi_env_me: . . . (system menu file)

changed none

SYS\$MANAGER:SYLOGIN.COM is also changed, by whoever is performing the installation. A command of the general form 'RXI := BINDIR:RXI_DETACHED.EXE' is added in order to make RXI generally available to users.

The system must be rebooted to make all changes take complete effect.

12. Installation using DEC Ultrix with DECwindows

The steps involved in getting the rxi program running on a DEC workstation or fileserver running Ultrix are:

1. Create a temporary source directory for the rxi program on the workstation or fileserver. For example, if the X sources are located in a directory called `/src/X` then the suggested place for rxi would be `/src/X/rxi`. If you would prefer to put the sources into a different place, eg. `/vendor/src/rxi`, then that works as well.
2. Traverse to `!Machine.Release.X_Interface.Release10_9_0` on the R1000.
3. Run the `Rxi_Install.Workstation_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. For example:

```
Rxi_Install.Workstation_Files
  (Keyboard           => Rxi_Install.Xultus_Vs_Ultrix4_1,
   On_Machine         => "nova",
   Username           => "sagan",
   Password           => "verbose",
   Account            => "unlimited",
   Rxi_Source_Directory => "/src/X/rxi");
```

4. Connect to the source directory on the workstation and execute the command `make restore`. This unpacks rxi and makes it available for complete installation.
5. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the rxi Environment menu entries for that product.
6. At the top of the file called `makefile` is a list of the source, library, include, and font paths that will be used during installation. If you wish to change any of these default locations please do so now. On a DECstation, the defaults are:

```
BINDIR   - where rxi executables will reside;
          usually /usr/bin/X11
XBINDIR  - where X Window System executables reside;
          usually /usr/bin/X11
FONTDIR  - where the server fonts reside;
          usually /usr/lib/X11/fonts/apps/75dpi
MANDIR   - the man page directory for the rxi man page;
          usually /usr/man/mann
XAPPLLOADDIR - where application defaults files reside;
          usually /usr/lib/X11/app-defaults
```

On a VAXstation, the defaults are:

BINDIR - where rxi executables will reside;
 usually **/usr/bin**
XBINDIR - where X Window System executables reside;
 usually **/usr/bin**
FONTDIR - where the server fonts reside;
 usually **/usr/lib/dwf/75dpi**
MANDIR - the man page directory for the rxi man page;
 usually **/usr/man/man**
XAPPLLOADDIR - where application defaults files reside;
 usually **/usr/lib/X11/app-defaults**

7. You now have three choices. A) Install the Rational-supplied rxi as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate rxi using all local definitions. The supplied version of rxi should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist.
 - a. If you wish to just use rxi as supplied then skip this step.
 - b. If you wish to simply relink rxi so that it uses local libraries then execute the **make relink** command and proceed to the next step.
 - c. If you wish to recreate rxi from scratch then execute the **make clean** command followed by **make all**.
8. Switch users to "super-user" and execute **make install.all**; this will:
 - Install the rxi program, the RXI default resource file, and any support programs needed by RXI users. (The command **make install** performs just this step.)
 - Install the rxi man page. (The command **make install.man** performs just this step.)
 - Compiles the various fonts (**fixed-screen-*.bdf**) and places them into $\$(\text{FONTDIR})$ directory so that the X server can find them. These are the normal and bold-face fonts used by rxi. (The command **make install.fonts** performs just this step.)
9. On some VAXStations, the call to **xset** during the installation of rxi causes the x server to "loose" it's fonts. If this situation exists, you will get errors about loading fonts when you run rxi. Execute the command **xlsfonts** to list the fonts the x server knows about. If the message **xlsfonts: pattern "*" unmatched** is displayed, you must reload the fonts. To reload the fonts, execute **xset fp default**
10. Install the termcap definition for the rxi terminal type into your local termcap database. There is a file called **termcap** in the rxi source area that contains the termcap definition for rxi. (There is also a **terminfo** file if you are using terminfo on your system.) The **termcap** file is inserted into your local **/etc/termcap** file and the **terminfo** file is used by issuing the command **tic terminfo**.

13. Installation using HP-UX and the X Window System

The steps involved in getting the rxi program running on a HP-UX workstation or fileserver running the X Window System are:

1. Create a temporary source directory for the rxi program on the workstation or fileserver. For example, if program sources are usually located in a directory called `/src` then the suggested place for rxi would be `/src/rxi`.
2. Traverse to `!Machine.Release.X_Interface.Release10_9_0` on the R1000.
3. Run the `Rxi_Install.Workstation_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. For example:

```
Rxi_Install.Workstation_Files
  (Keyboard           => Rxi_Install.Xhp46021a_hpux7_0,
   On_Machine         => "my_hp",
   Username           => "my_username",
   Password           => "my_password",
   Account            => "",
   Rxi_Source_Directory => "/src/rxi");
```

4. Connect to the source directory on the workstation and execute the command `make restore`. This unpacks rxi and makes it available for complete installation.
5. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the rxi Environment menu entries for that product.
6. Edit the file called `Makefile`. A particular site may wish to change the places where rxi, its fonts, and/or its man page are installed. These can be changed by changing the definition of:

The defaults are:

```
BINDIR - where rxi executables will reside;
         usually /usr/bin/X11
HPBINDIR - where the HP X Window System executables reside;
         usually /usr/bin/X11
FONTDIR - where the compiled fonts reside;
         usually /usr/lib/X11/fonts/misc
MANDIR - the man page directory for the rxi man page;
         usually /usr/man/mann
XAPPLLOADDIR - where application defaults files reside;
         usually /usr/lib/X11/app-defaults
```

NOTE: The `XAPPLLOADDIR` directory may be changed but changes will only actually take effect when rxi is recompiled by performing `make clean` and `make all` steps.

7. You now have three more choices. A) Install the Rational-supplied `rx` as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate `rx` using all local definitions. The supplied version of `rx` should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist.
 - a. If you wish to just use `rx` as supplied then skip this step.
 - b. If you wish to simply relink `rx` so that it uses local libraries then execute `make relink` command and proceed to the next step.
 - c. If you wish to recreate `rx` from scratch then execute `make clean` command followed by `make all`.
8. Switch users to "super-user" and execute `make install.all`; this will:
 - Install the `rx` program, the RXI default resource file, and any support programs needed by RXI users. (The command `make install` performs just this step.)
 - Install the `rx` man page. (The command `make install.man` performs just this step.)
 - Compiles the various fonts (`fixed-screen-*.bdf`) and places them into `$(FONTDIR)` directory so that the X server can find them. These are the normal and bold-face fonts used by `rx`. (The command `make install.fonts` performs just this step.)

After installing `rx`, at any time:

If an X Server is already running then the `rx` program may be run directly.

If an X Server is not already running then the `rx` program may be used to run the X Server and then invoke a "console" window, a clock window, and one `rx` window on the workstation.

14. Installation using AIX and IBM X-Windows

The steps involved in getting the rxi program running on an IBM RS6000 workstation or fileserver are:

1. Create a temporary source directory for the rxi program on the workstation/fileserver. For example, if program sources are usually kept in a directory called `/src` then within this directory create a directory called `/src/rxi`.
2. Traverse to `!Machine.Release.X_Interface.Release10_9_0` on the R1000.
3. Run the `Rxi_Install.Workstaion_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. For example:

```
Rxi_Install.Workstaion_Files
      (Keyboard           => Rxi_Install.Xr6us_Aix3_1,
      On_Machine         => "My_Machine",
      Username           => "MyName",
      Password           => "MyPassword",
      Account            => "",
      Rxi_Source_Directory => "/src/rxi");
```

4. Connect to the source directory on the workstation and execute the command `make restore`. This unpacks rxi and makes it available for complete installation. For example:

```
cd /src/rxi
make restore
```

5. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the rxi Environment menu entries for that product.
6. Edit the file called `Makefile`. A particular site may wish to change the places where rxi, its fonts, and/or its man page are installed. These can be changed by changing the definition of:
 - `BINDIR` - where the rxi shell script goes; this is the twin to the aixterm script; usually `/usr/bin`
 - `AIXBINDIR` - where the AIX X Window System binaries currently reside; usually `/usr/bin`
 - `X11BINDIR` - where the real rxi goes; this is where most X11 executables go; usually `/usr/lpp/X11/bin`
 - `XAPPLLOADDIR` - where the default resource file goes; usually `/usr/lib/X11/app-defaults`
 - `FONTDIR` - where the system-wide X11 fonts live; usually `/usr/lpp/fonts`
 - `MANDIR` - where the system-wide "new" man pages live; usually `/usr/man/mann`
7. You now have three more choices. A) Install the Rational-supplied rxi as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate

rxl using all local definitions. The supplied version of rxl should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist.

- a. If you wish to just use rxl as supplied then skip this step.
 - b. If you wish to simply relink rxl so that it uses local libraries then execute the `make relink` command and proceed to the next step.
 - c. If you wish to recreate rxl from scratch then execute the `make clean` command followed by `make all`.
8. Switch users to "super-user" and execute `make install.all`; this will:
- Install the rxl program, the RXI default resource file, and any support programs needed by RXI users. (The command `make install` performs just this step.)
 - Install the rxl man page. (The command `make install.man` performs just this step.)
 - Install the rxl terminfo description. This allows terminfo-using programs (such as vi) to operate within an rxl window. (The command `make install.terminfo` performs just this step.)
 - Compile the various fonts `fixed-*.bdf` into the `$(FONTDIR)` directory. These are the normal and bold-face fonts used by rxl. (The command `make install.fonts` performs just this step.)

Some sites may not want to use the Rational supplied fonts. They can skip the `install.man` and `install.fonts` steps if they wish. They will want to edit the `RXI.xdefaults` file to change the names of the default rxl fonts.

Some sites may not have the optional licensed IBM "man" product and they can skip the `install.man` step.

9. Run rxl and decide whether or not the default font is too small. Some IBM displays have a very small pixel size. This can make the default font too small for many users. RXI comes with 8 different fonts. They are:
- `fixed-screen-{r,b}-11` - default font - smallest "thin" font
 - `fixed-screen-{r,b}-12`
 - `fixed-screen-{r,b}-13`
 - `fixed-screen-{r,b}-14` - largest "thin" font - 1 pixel line thickness
 - `fixed-screen-{wr,wb}-15` - smallest "thick" font - > 1 pixel thickness
 - `fixed-screen-{wr,wb}-16`
 - `fixed-screen-{wr,wb}-17`
 - `fixed-screen-{wr,wb}-22` - largest "thick" font

You can try out the different fonts with command lines like these, restart your X server before trying this:

```
aix> rxl -fn fixed-screen-r-13 -fb fixed-screen-b-13 &
aix> rxl -fn fixed-screen-wr-15 -fb fixed-screen-wb-15 &
```

Always use matching normal (-fn) and bold (-fb) fonts or your rxi window may be unreadable when connected to an R1000.

If you decide to change the system-wide default font then edit the `RXI.Xdefaults` file. Change the `*vt100.font:` and `*vt100.boldFont:` lines to reference the new font. You may also want to change then `*vt100.menuFont:` line if you find the menus to be unreadable as well. Then execute `make install` to install the new system-wide defaults.

15. Installation using SunOS with X11/NeWS

The steps involved in getting the rxi program running on a Sun workstation or fileserver running XNews are:

1. Pick an RXI terminal type to install.

If all of the Sun workstations that will be using RXI to connect to R1000's have the older Type 3 keyboard (no numeric pad on the right, no help key on the left) then go to the next step and install the `xNews3_v2_0` or `xNews3_v2_0_Sparc` terminal type.

If all of the Sun workstations that will be using RXI to connect to R1000's have the newer Type 4 keyboard (numeric pad on the right instead of the R1-R15 keys, help key on the left) then go to the next step and install the `xNews4_v2_0` or `xNews4_v2_0_Sparc` terminal type.

If you will have a mixture of Type 3 and Type 4 keyboards that will all be using the same installed copy of RXI then you will have to do one of two things. Either install RXI as an `xNews3` terminal type or as an `xNews4` terminal type.

If you install RXI as an `xNews3` then:

- Type 3 keyboard users do nothing extra.
- Type 4 keyboard users should edit their `~/.xdefaults` file to contain a line that looks like this: `RXI*recognition:xnews4`.

If you install RXI as an `xNews4` then:

- Type 3 keyboard users should edit their `~/.xdefaults` file to contain a line that looks like this: `RXI*recognition:xnews3`.
- Type 4 keyboard users do nothing extra.

The line `RXI*recognition:xnews3` or `RXI*recognition:xnews4` tells RXI that when an R1000 asks the question "What type of terminal are you?", it should answer "xnews3" or "xnews4". RXI is built with a default answer that is determined by the `Rxi_Install.xNews?_v2_0` terminal type that you chose to install. This is a way to override that default.

Note: If the Type 3 keyboards are using one fileserver and the Type 4 keyboards are using another then you can simply install RXI as an `xNews3` on the one server and as an `xNews4` on the other server. Treat the two servers as two separate RXI installations as shown above. The R1000's should all have both the `xNews3` and the `xNews4` Editor_Data files installed.

NOTE to R1000 users using the model 4 keyboard with X11/NeWS:

Sun has seen fit to make some keys on the keyboard "identical".

In particular, the keys "Stop" and "F11" as well as the keys "Again" and "F12" are defined, by X11/NeWS, to be the same X "key symbol". This means that the R1000 cannot tell the difference between a user typing what he thinks of as "Again" and what he thinks of as "F12".

The keys "Help" and "F1" are defined, by the default X11/NeWS startup script, to be the same X "key symbol". This means that the R1000 cannot tell the difference between a user typing what he thinks of as "Help" and what he thinks of as "F1". This can be fixed by removing the following command from your `.xinitrc` file:

```
xmodmap -e 'keysym F1 = Help'
```

If you have already started NeWS then you can repair this on-the-fly with this command:

```
xmodmap -e 'keycode 12 = F1'
```

In addition the Alternate key is defined as NoSymbol which means that it does nothing. In order to use the Alternate key with an R1000 you will need to issue the command:

```
xmodmap -e 'keycode 26 = Alt_R'
```

This will define the Alternate key and make it available for use. This will have to be done once each time you log into the Sun.

2. Create a temporary source directory for the rxi program on the workstation or fileserver. For example, if your various X11 application sources are located in `/src/x` then a likely place would be `/src/x/rxi`.
3. Traverse to `!Machine.Release.X_Interface.Release10_9_0` on the R1000.
4. Run the `Rxi_Install.Workstation_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. Use the `Rxi_Install.Xnews3_V2_0`, `Rxi_Install.Xnews3_V2_0_Sparc`, `Rxi_Install.Xnews4_V2_0_Sparc` or the `Rxi_Install.Xnews4_V2_0` terminal types. For example:

```
Rxi_Install.Workstation_Files
  (Keyboard           => Rxi_Install.Xnews3_V2_0,
   On_Machine         => "nova",
   Username           => "sagan",
   Password           => "verbose",
   Account             => "unlimited",
   Rxi_Source_Directory => "/src/x/clients/rxi");
```

5. Connect to the source directory on the workstation and do the command `make restore`. This unpacks rxi and makes it available for complete installation.
6. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the rxi Environment menu entries for that product.
7. Edit the file called `Makefile`. There are a series of "make" macros at the beginning of the file. These macros tell the installation script where the various pieces of RXI are to

be installed. The default values for these macros may be correct for your site but they should be checked. The macros are:

OPENWIN - the "home" directory for Open Windows (X/NeWS);
 eg. /openwin
BINDIR - where rxi executables will reside;
 eg. /openwin/bin
FONTDIR - where the server fonts reside;
 eg. /openwin/lib/fonts
MANDIR - the man page directory for the rxi man page;
 eg. /usr/man/man1
XAPPLLOADDIR - where application defaults files reside;
 eg. /openwin/etc

NOTE: The **XAPPLLOADDIR** directory may be changed but changes will only actually take effect when rxi is recompiled by performing **make clean** and **make all** steps.

8. FOR NCD INSTALLS ONLY: Edit the file **Makefile** and modify the pathname for the **FONTDIR** entry. On some systems, add /ncd to the end of the pathname. On other systems, use the pathname **/usr/lib/X11/ncd/fonts/misc**. See the **Installation Overview** section for more information about finding the correct location for the fonts.
9. You now have three more choices. A) Install the Rational-supplied rxi as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate rxi using all local definitions. The supplied version of rxi should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist.
 - a. If you wish to just use rxi as supplied then skip this step.
 - b. If you wish to simply relink rxi so that it uses local libraries then execute **make relink** command and proceed to the next step.
 - c. If you wish to recreate rxi from scratch then execute the **make clean** command followed by a **make all**.
10. Switch users to "super-user" and execute **make install.all**; this will:
 - Install the rxi program, the RXI default resource file, and any support programs needed by RXI users. (The command **make install** performs just this step.)
 - Install the rxi man page. (The command **make install.man** performs just this step.)
 - Compile the various fonts (**fixed-screen-*.bdf**) into the **\$(FONTDIR)** directory and then does a "bldfamily" there. These are the normal and bold-face fonts used by rxi. (The command **make install.fonts** performs just this step.)
11. Install the termcap definition for the rxi terminal type into your local termcap database. There is a file called **termcap** in the rxi source area that contains the termcap definition for rxi. (There is also a **terminfo** file if you are using terminfo on your system.) The **termcap** file is inserted into your local **/etc/termcap** file and the

`terminfo` file is used by issuing the command `tic terminfo`.

16. Installation using SunOS with MIT X11.R3 or X11.R4

The installation steps for a Sun Microsystems workstation or fileserver, either the older type-3 keyboard, or the newer type-4 keyboard, and the MIT X Window System (X11.R3 or X11.R4) are:

1. Pick an RXI terminal type to install.

If all of the Sun workstations that will be using RXI to connect to R1000's have the older Type 3 keyboard (no numeric pad on the right, no help key on the left) then go to the next step and install the `xSun3_X11R3` or the `xSun3_X11R3_Sparc` terminal type.

If all of the Sun workstations that will be using RXI to connect to R1000's have the newer Type 4 keyboard (numeric pad on the right instead of the R1-R15 keys, help key on the left) then go to the next step and install the `xSun4_X11R3` or the `xSun4_X11R3_Sparc` terminal type.

If you will have a mixture of Type 3 and Type 4 keyboards that will all be using the same installed copy of RXI then you will have to do one of two things. Either install RXI as an `xSun3` terminal type or as an `xSun4` terminal type.

If you install RXI as an `xSun3` then:

- Type 3 keyboard users do nothing extra.
- Type 4 keyboard users should put the `rxsun4` command into their personal `~/.rsrc` file (or they can run the command by hand in any shell window prior to the time they log into an R1000). They should also edit their `~/.Xdefaults` file to contain a line that looks like this: `RXI*recognition:xsun4`.

If you install RXI as an `xSun4` then:

- Type 3 keyboard users should edit their `~/.Xdefaults` file to contain a line that looks like this: `RXI*recognition:xsun3`.
- Type 4 keyboard users will have the `rxsun4` command performed for them by the default `rsrc` file in the `app-defaults` directory. If they have a personal `~/.rsrc` file then they will need to place the `rxsun4` command in that file as well.

The line `RXI*recognition:xsun3` or `RXI*recognition:xsun4` tells RXI that when an R1000 asks the question "What type of terminal are you?", it should answer "xsun3" or "xsun4". RXI is built with a default answer that is determined by the `Rxi_Install.XSun?_X11R?` terminal type that you chose to install. This is a way to override that default.

The `rxsun3` and `rxsun4` commands read the `xSun3-xmodmap` and `xSun4-xmodmap` files respectively. These files contain keyboard definitions for the Type 3 and Type 4 keyboards. These commands are intended only for the use of users with Type 4 keyboards.

Note: If the Type 3 keyboards are using one file server and the Type 4 keyboards are using another then you can simply install RXI as an `xSun3` on the one server and as an `xSun4` on the other server. Treat the two servers as two separate RXI installations as

shown above. The R1000's should all have both the `xsun3` and the `xsun4 Editor_Data` files installed.

X11 from MIT has a "problem" with Type 4 keyboards. SunOS 3.? does not know about Type 4 keyboards and so it always thinks that users have Type 3. SunOS 4.? knows about Type 4 keyboards and so the X server will realize that a Type 4 keyboard is in use, however, the MIT server for Suns has a default keyboard mapping that makes most of the keys in the rightmost keypad useless; they don't transmit anything when pressed unless the Shift key is also being held.

Since the R1000 enjoys the use of these keys this problem had to be addressed. The `rxsun4` command tells the X server to redefine the entire keyboard. Every key on the keyboard is assigned to be a key of some type. Typically the new type is the same as the old type; the differences lie in the area where the default keyboard mapping has various keys defined as NoSymbol (no-operation). This allows a Type 4 user to access all of his keys.

Please note that this is a "global" effect. It affects not just RXI but rather globally effects every application being run upon the workstation. At some point either MIT or Sun will come out with a proper default keyboard mapping that does not make the extra keys NoSymbol; at that time the `xsun4` terminal type will be changed to reflect the new defaults and the `rxsun4/rxsun3` commands will disappear.

The `rxsun3` command turns a Type 4 keyboard into a Type 3 keyboard. Any "new" keys (keys that are new with the Type 4 keyboard) are marked as NoSymbol (no-operation). This is another option for Type 4 keyboard users; they can turn their keyboard into a Type 3.

Special Installation Note for Type 4 keyboard installations:

The `XSun3-xmodmap` and `XSun4-xmodmap` files should be tested prior to installation by the `make install.all` step. While the X server is running, issue the following two commands:

if on a Type 4 keyboard :

```
xmodmap XSun3-xmodmap
xmodmap XSun4-xmodmap
```

if on a Type 3 keyboard :

```
xmodmap XSun4-xmodmap
xmodmap XSun3-xmodmap
```

If you get any messages of the general form "bad keycode value (out of range)" then you are using a version of the X server that contains a minor bug. The problem is that the server has some tables inside of it for describing the keyboard. The tables are 1 entry short. To fix the problem just edit the `XSun3-xmodmap` and `XSun4-xmodmap` files. Search for the line (there is one in each file) that has "keycode 132" on it. Comment out that line by placing a "!" at the front of the line. This will eliminate the problem and you can proceed with the next installation step. This bug means that your numeric keypad "+" key will not be available for use; the server does not realize that it

exists.

2. Create a temporary source directory for the RXI program on the workstation or fileserver. For example, if the X11 sources from MIT are located in a directory called `/src/x` then the suggested place for `rx` would be `/src/x/clients/rxi`. If you would prefer to put the sources into a different place, eg. `/vendor/src/rxi`, then that works as well.
3. Traverse to `!Machine.Release.X_Interface.Release10_9_0` on the R1000.
4. Run the `Rxi_Install.Workstation_Files` procedure; this gets the files copied to the workstation or fileserver from the R1000. Use the `Rxi_Install.XSun3_X11R3`, `Rxi_Install.XSun3_X11R3_Sparc`, `Rxi_Install.XSun4_X11R3` or `Rxi_Install.XSun4_X11R3_Sparc` terminal types. For example:

```
Rxi_Install.Workstation_Files
  (Keyboard           => Rxi_Install.Xsun3_X11r3,
   On_Machine         => "nova",
   Username           => "sagan",
   Password           => "verbose",
   Account            => "unlimited",
   Rxi_Source_Directory => "/src/x/clients/rxi");
```

5. Connect to the source directory on the workstation and execute the command `make restore`. This unpacks `rx` and makes it available for complete installation.
6. If this site has some or all of the optional licensed Rational products then edit the `RXI_Env_Menu` file. It's format is that of a C source file so it should look familiar. There is a `#define` line for each licensed product. For each licensed product at this site, set the corresponding `#define` to be equal to 1. This activates the `rx` Environment menu entries for that product.
7. At this point you have two choices. A) Handle `rx` in the typical MIT fashion using the "imake" and "makedepends" programs, or, B) handle `rx` as an independent stand-alone program.
 - a. If your site uses the imake/makedepends programs for X11.R3 (or X11.R4) based applications then execute:

if the sources are in `/src/x/clients/rxi` as suggested execute:

```
make -f Makefile.MIT Makefile
```

if the sources are in some other directory then do something like:

```
make -f Makefile.MIT Makefile TOP=/src/x
```

Substitute the appropriate main X11.R3 (or X11.R4) source directory for `/src/x`. This will regenerate the `Makefile` in the typical MIT fashion. Follow that with a `make depend` command.

Note: X11.R4 sites may have a command named "xmkmf". If that command is present then use it instead of this procedure. The `xmkmf` command requires no

arguments.

- b. If your site does not use the image/makedepends programs, or if you do not want to use them in this case then you may wish to hand-edit the Rational-supplied **Makefile**. At the top of the **Makefile** is a list of the source, library, include, and font paths that will be used during installation. If you wish to change any of these default locations please do so now. The defaults are:

```

BINDIR - where rxi executables will reside;
          usually /usr/bin/X11
XBINDIR - where X Window System executables reside;
          usually /usr/bin/X11
FONTDIR - where the server fonts reside;
          usually /usr/lib/X11/fonts/misc
MANDIR - the man page directory for the rxi man page;
          usually /usr/man/mann
XAPPLLOADDIR - where application defaults files reside;
          usually /usr/lib/X11/app-defaults

```

8. FOR NCD INSTALLS ONLY: Edit the file **Makefile** and modify the pathname for the **FONTDIR** entry. On some systems, add **/ncd** to the end of the pathname. On other systems, use the pathname **/usr/lib/X11/ncd/fonts/misc**. See the **Installation Overview** section for more information about finding the correct location for the fonts.
9. You now have three more choices. A) Install the Rational-supplied rxi binary executable as-is, or, B) relink to use local (possibly shared) libraries, or C) completely recompile and recreate rxi using all local definitions. The supplied version of rxi should be able to execute on your system. However, it will not use any shared libraries, nor will it incorporate any local library fixes, extensions, or modifications that may exist.
- a. If you wish to just use rxi as supplied then skip this step.
- b. If you wish to simply relink rxi so that it uses local libraries then execute the **make relink** command and proceed to the next step.
- c. If you wish to recreate rxi from scratch then execute the **make clean** command followed by **make all**.
10. Switch users to "super-user" and execute **make install.all**; this will:
- Install the rxi program, the RXI default resource file, and any support programs needed by RXI users. (The command **make install** performs just this step.)
 - Install the rxi man page. (The command **make install.man** performs just this step.)
 - Compiles the various fonts (**fixed-screen-*.bdf**) and places them into **\$(FONTDIR)/bdf/misc** directory so that the X server can find them. These are the normal and bold-face fonts used by rxi. (The command **make install.fonts** performs just this step.)

If you wish to add these fonts to the collection of fonts that came with X11.R3 (or X11.R4) from MIT (so that they will always be automatically regenerated whenever the MIT fonts are regenerated then issue the command `make install.fonts.source`.

Then connect to the `$(FONTSRC)/bdf/misc` directory and add the appropriate lines to the `Imakefile` there to cause compilation and installation of the new `fixed-screen-*-*.bdf` files. A `bdf.Imakefile` is supplied by Rational which shows the type of entries to be made. Execute `make all` and `make install` there to compile and install the new fonts on your system.

11. Install the termcap definition for the `rx` terminal type into your local termcap database. There is a file called `termcap` in the `rx` source area that contains the termcap definition for `rx`. (There is also a `terminfo` file if you are using `terminfo` on your system.) The `termcap` file is inserted into your local `/etc/termcap` file and the `terminfo` file is used by issuing the command `tic terminfo`.
12. If the `rx` files were installed in the `/src/x/clients/rx` hierarchy as suggested in step #2 then you will probably wish to edit the `Imakefile/Makefile` in the `/src/x/clients` directory. Add the `rx` directory to the `SUBDIRS=...` list. This will cause `rx` to be remade and reinstalled whenever all MIT X11 clients are remade.

17. Installation Overview

There are two general classes of RXI-based terminal types and the steps in their installation may differ somewhat.

- *Real* workstations. For example, you might have a DEC workstation running the VMS operating system on your desk and that is the workstation that you would use when working with an R1000. This means that you would install one of the "real" RXI terminal types such as `Xdecus_Vms5_1_Multinet203` or `Xdecus_Vms5_1_Wollongong502`.
- *Parasite* X-based terminals. For example, you might have an NCD X Terminal on your desk that keeps its font files and such on the file-system of a nearby DEC workstation running VMS. To use the NCD as your R1000 connection you would install one of the "parasite" terminal types for the NCD, such as `Xncd_Vms5_1_Multinet203` or `Xncd_Vms5_1_Wollongong502`.

Real RXI terminal types are installed using a set of terminal-type-specific keymap files on the R1000 and an RXI binary executable targeted to the real workstation.

Parasite RXI terminal types are installed using a set of terminal-type-specific keymap files on the R1000 and an RXI binary executable targeted to the *host* of the parasite terminal.

The R1000 side of things are specific to the target keyboard. The workstation/fileserver side of things are specific to the host machine of the target keyboard. Real workstations are their own hosts. Parasite terminals always have an external host machine. The external host is the machine that actually runs the RXI binary executable.

17.1. R1000 Installation Overview

The following is intended as an overview of the RXI installation process. Please read the install note for the actual instructions for installing RXI. The general steps for installing a mythical XYZ terminal type are:

1. Execute the `Rxi_Install.Machine_Editor_Data_Files` procedure to install the various XYZ keymap files on your R1000. This automatically enables any product-specific commands contained in the new keymap. Commands are enabled if the pertinent product is authorized for this R1000. The files would be:

```
!Machine.Editor_Data:
  Xyz_Commands      : I Proc_Spec;
  Xyz_Commands      : I Proc_Body;
  Xyz_Keys           : Text;
  Xyz_Key_Names     : I Pack_Spec;
  Xyz_User_Commands : Text;
```

This step also creates the appropriate XYZ entries in the `Terminal_Types` and `Terminal_Recognition` files in the same directory.

This step is performed for all real and parasite terminal types that will be used to talk to the R1000. It installs the keymaps for this terminal type and it tells the R1000 how to perform auto-recognition of the new terminal type when a user logs into the R1000.

This step is performed once for each RXI terminal type that your site will be using, real or parasite.

2. Enable the RXI keymaps (Reboot the R1000 or run the Refresh_Terminal_Information command).
3. Ship the workstation files to the workstation(s) and/or fileserver(s) where the workstation installation(s) will be performed. The Rxi_Install.Workstation_Files procedure is used to ship the files, using the R1000 FTP mechanisms.

17.2. Workstation/Fileserver Installation Overview

The following is intended as an overview of the RXI installation process. Please read the install note for the actual instructions for installing RXI. The general steps for installing a mythical XYZ terminal type are:

1. Create a temporary directory on the workstation or fileserver. This directory is used as a staging area during the installation. It can be deleted after the installation is complete. Typical installation space requirements are in the 10..20 Mb range (depending upon the target system).
2. Ship the workstation files from the R1000 to this temporary directory.
3. Log into the workstation/fileserver and connect to the temporary directory.
4. Restore the backup file that was shipped from the R1000. (**make restore** on UNIX systems and **@makefile restore** under VMS.)
5. Optionally change the default installation directories. RXI will be automatically installed in various system directories. These directories are appropriate for a large class of customers. However, some sites may have their own, different, desired configurations. See Section 17.4, page 40 for details.
6. Install RXI. As one step using **make install.all** on UNIX and **@makefile install.all** on VMS. Or, as several steps, **install** to install the binary executable and the system default resources file, **install.fonts** to install the RXI fonts, **install.man** to install the man page or help file, and **install.terminfo** to install a termcap description file for systems using the termcap library.
7. Restart the X server(s) that will be used with RXI. They must be restarted so that they will discover the existence of the new RXI fonts. (Note: Some servers have a utility program, often called **xset**, that can be used for this purpose. These servers would not need to be restarted.)

17.3. Workstation/Fileserver Components

There are four major components to the RXI product on the workstation, they are:

- The RXI executable(s)
- The RXI default resources file

- The RXI fonts
- The RXI man page or help file

These four components are installed, by default, on a single filesystem attached to a single workstation. However, some customers may have complex networks with scattered file servers and hundreds of workstations of different types.

Use the following to help determine where the various components must go.

17.3.1. RXI Executable

The executables are installed by the **make install** (UNIX) or the **@makefile install** (VMS) command. This is one of the sub-steps performed by the **make install.all** (UNIX) or the **@makefile install.all** (VMS) command. It can be performed separately.

The RXI binary executable (a.k.a. the RXI program) goes on the filesystem(s) that serves the machine(s) that will provide the binary execution environment for RXI. These are the *host* machines discussed above. Most workstations will serve as their own hosts. X-based terminals always have an external host machine.

Most of RXI's target environments have a single RXI executable. However, some targets, such as IBM's AIX operating system, have two. The main RXI executable is an ordinary C program. The other RXI executable is a shell script whose purpose is to a) know the exact physical location of the main RXI executable, and b) to run that executable in the background and then to exit. Under AIX, the main executable goes in one directory and the script file goes in another. See the individual target instructions for details.

The RXI executables must be placed where they are readily available to the host machines. If each host machine has its own separate and unique filesystem then, in the worst case, the RXI executables will need to be installed on each and every host machine's filesystem. However, most sites will be using one or more file servers. For those sites, the RXI executables would be installed on one or more of the file servers.

Only the customer has any way of knowing which file servers should be used for the installations. Carry out the workstation installation instructions once for each workstation or file server.

Note: It is much easier to repeat the installation instructions once for each file server than it is to attempt to install RXI once on one machine and then try to copy the installed files to each of the other file server. It is very tempting to try to do this. It appears to be very easy. Do not fall into this trap. Perform the installation instructions once for each file server.

The executables should be placed in a directory that is visible to each user that will be running RXI (for example, `/usr/bin/x11`). If the directory is not visible (because it is protected or because it is not mounted on the user's host machine) then that user will not be able to run RXI.

Also, the executables should be placed in a directory that is on the default search path for user commands. If the RXI executables are placed in a directory which is not in the default search path, it will be necessary to modify each user's search path to gain easy access to the RXI executable.

17.3.2. RXI Default Resources File

The default resources files are installed by the **make install** (UNIX) or the **@makefile install** (VMS) command. This is one of the sub-steps performed by the **make install.all** (UNIX) or the **@makefile install.all** (VMS) command. It can be performed separately.

The operation of the RXI program is governed to a large extent by the contents of the filesystem-wide *system defaults file*. This file is an X default resources file. Among other things, it tells RXI what its default terminal type is when logging into an R1000 and what its default window size should be. These default values govern the operation of all copies of RXI that are run using this particular filesystem. This may affect one workstation or it may affect hundreds; it depends upon how the customer network is arranged.

These files are read, by RXI, when RXI initializes itself. This means that these files must be present, on the host machine's filesystem, when RXI is run by the user. If these files are not present, RXI will still run, but it will probably not operate in quite the expected fashion. See "Common Installation Problems", Section 18, page 45.

The RXI default resources files must be placed in the same area that other X Window System programs will be looking for their default resources files. This location is hard-wired into the Xlib library that is supplied as part of an X Window System implementation. See your vendor's documentation to find out the name of this directory.

When RXI is installed, the **RXI.xdefaults** file in the RXI source area is installed in the X application defaults area under the name **RXI**. A user can also have a local **.xdefaults** (UNIX) or **DECW\$XDEFAULTS.DAT** (VMS) file in their home directory. This local file can be used to override any or all of the resource values specified by the system defaults file.

See the discussion of **XAPPOADDIR** for more information about the X application defaults and RXI.

17.3.3. RXI Fonts

The fonts are installed by the **make install.fonts** (UNIX) or the **@makefile install.fonts** (VMS) command. This is one of the sub-steps performed by the **make install.all** (UNIX) or the **@makefile install.all** (VMS) command. It can be performed separately.

The RXI fonts should be installed with the fonts that came with the customer's X Window System. These font files are read by the X server; they are not read by the RXI program.

- If a user is using a workstation directly (meaning that he will be sitting directly in front of the workstation console) then the fonts should be installed on that workstation's filesystem so that the workstation's X server can find them.
- If a user is using an X-based terminal, then the fonts are installed on the host machine, but they are installed in the font directory for the X-based terminal and not the fonts directory for the workstation.
- If a user is using one workstation (sitting in front of it), but is running RXI on some other machine on the network, then the user is using the first machine as if it were an X-based terminal and the second machine as the host. The fonts must be installed on the first machine because that machine is the one that is running the X server.

It is recommended that the RXI fonts be installed into the main font area for the X server(s). However, the RXI fonts may be installed in any location. In that case it will be necessary to inform the X server of the location of the RXI fonts before running RXI. See the discussion about `FONTDIR` below for more information on RXI font location.

17.3.4. RXI Man/Help File

The man page (UNIX) or help library (VMS) is installed by the **make install.man** (UNIX) or the **@makefile install.help** (VMS) command. This is one of the sub-steps performed by the **make install.all** (UNIX) or the **@makefile install.all** (VMS) command. It can be performed separately.

The RXI man page (UNIX) should be placed into the usual man page area for the user system. Similarly, the RXI help library (VMS) should be merged with the main system help library.

The man page (help library) should be placed on the same filesystem as is used for the RXI executable(s). If a user expects to use some machine ABC to run the RXI program, then that user will also expect that machine ABC to provide help.

If RXI is being run on one machine, and the user is actually sitting in front of some other machine, do not install the man/help files on the other machine, especially if the two machines are not of the same type. The man/help files are not the same for every type of machine.

17.4. Workstation/Fileserver Installation Directories

One step in the installation process is to review and possibly edit the **Makefile** (UNIX) or **Makefile.Com** (VMS) to change the default locations to be used when installing RXI. The Makefile lists the default names of the directories used to install RXI. If a customer is determined that RXI must be installed elsewhere, or if the X Window System software is not in the usual location, then:

- Determine where the binary executables, the resources files, the fonts, and the man/help files should be placed.
- Set `BINDIR` in the Makefile to indicate the place for the binaries.
- Set `XBINDIR` in the Makefile to indicate where the X Window System utility programs can be found (for use in installing RXI).
- Set `FONTDIR` to indicate the place for the fonts.
- Set `XAPPLOADDIR` and `XAPPLOADDIRstr` (if applicable) to be the same thing, the place for the resources files.

It is very important that you know the names of these directories before you attempt to install RXI. The following list describes the entries in the **Makefile** in greater detail. (All examples are "general case" and may not apply to your specific workstation type.)

17.4.1. BINDIR Directory

DESCRIPTION: Where the RXI executable will reside.

LOCATION: The RXI executable may be installed in any location. It is recommended that the RXI executable be placed in a location which is visible to users in their default search paths. For example, a good location on Unix machines would be `/usr/bin` or `/usr/bin/x11`.

ERRORS: If the RXI executable is installed in a location that is not visible in the users' paths, or is protected, a message something like:

```
rxl: Command not found.
```

or

```
Permission denied.
```

will be displayed when you try to run RXI.

17.4.2. XBINDIR Directory

DESCRIPTION: Where the various X Window System utility programs reside. The installation of the RXI fonts requires the use of various X Window System utility programs usually found in this directory. These utilities compile and register the RXI fonts as appropriate for the workstation.

LOCATION: The location of the X Window System utilities varies from vendor to vendor. The Makefile contains an entry for the default location of the X Window System utilities for the vendor. If that directory is not present on the workstation/fileserver, look for a directory containing the programs that compile the fonts and which add the fonts to the font directory. The names of these programs are:

Font Compiler Names

Workstation	Program Names
Sun (MIT X11)	<code>bdftosnf</code> and <code>mkfontdir</code>
Sun (XNeWS)	<code>dumpfont</code> and <code>bldfamily</code>
IBM RS6000	<code>bdftosnf</code> and <code>mkfontdir</code>
Apollo	<code>bdftosnf</code> and <code>mkfontdir</code>
HP	<code>xfc</code> and <code>mkfontdir</code>
Dec Ultrix	<code>bdftopcf</code> and <code>mkfontdir</code>
Dec VAX/VMS	<code>font</code>

ERRORS: An incorrect value for this entry will cause error messages to be generated during the installation. And, since the fonts will not be properly installed, this will also

prevent the X server from finding the RXI fonts when RXI is run. RXI will request that the X server use its fonts and that request will fail. RXI will then print a message of the form: "X Toolkit Warning: Can not convert...". This usually indicates that the fonts were not properly installed or that the X server has not been restarted.

17.4.3. FONTPATH Directory

DESCRIPTION: Where the RXI fonts will reside.

LOCATION: It is highly recommended that the RXI fonts be installed in one of same directories used for the X server fonts. Again, the location of the X server fonts varies from vendor to vendor. Also, the names of the files in the X server font directory vary with vendors. The directory containing the X server fonts will contain many files with similar names. For example, `fixed.snf`, `6x13.pcf` or even `Courier.Decw$Font`.

ERRORS: The installation procedure for RXI attempts to inform the current workstation's X server about the new RXI fonts. (Note: This may or may not be same the X server that normal users will be using.) If this fails, a message similar to: "X Toolkit Warning: can't convert" may be displayed when RXI is tested. If this happens, restart the X server to load the RXI fonts. On some systems you can shutdown then restart the X server, while on others you must log off to restart the X server. There may be other issues with restarting the X server running on a fileserver.

CUSTOMIZATION: It is possible to install the RXI fonts in a location other than that of the X server fonts. If the fonts are installed in another location, users must tell the X server where they were installed in order to use RXI. The method for locating the fonts in the X server varies from vendor to vendor. Refer to the reference manual for the customer's X Window System for details on how to do this. Some examples of how to add the new RXI fonts to the X server follow. Remember that these are examples only and may NOT work with your windowing system.

- Use the `xset` program to add the RXI fonts to the X server's font search path. For example, if the Fonts were installed in `/fonts/rxi`, then try the command `xset +fp /fonts/rxi`. The X server must be running for this to work. This will have to be used each time the X server is restarted.
- Put an `xset` command entry for the font directory in the user's `.xinitrc` file. (The name of this file also varies from vendor to vendor.) This file contains shell commands that are executed when the X server is started by this user.
- There may be other ways to add the RXI fonts to the X server. See the vendor's windowing system users guide for more information.

17.4.4. XAPPLLOADDIR Directory

DESCRIPTION: Where the X application default files reside. The files `RXI` and `RXI_Env_Menu` will be put here.

The `RXI` file contains the various default control values that govern RXI's default behavior. Eg. the default window size and the default terminal type to use when logging into an R1000.

The `RXI_Env_Menu` file contains the Environment menu specification for use when the middle mouse button is used while logged into an R1000.

Changing the contents of either of these files will affect all users of this filesystem.

LOCATION: The location of the X application defaults directory varies on different workstations. Depending on what X applications are on this system, there may or may not be any files in this directory before installing RXI. Some common X applications that would have something here would be `xclock` or `twm`.

This location is not controlled by RXI. This location is hard-wired into the Xlib library that was supplied by the windowing system vendor. RXI uses this library and inherits this hard-wired location.

ERRORS: If the correct pathname for this directory is not used, the RXI Environment menu will be empty except for an error message about the missing file. If the message "No RXI Env Menu specified" is displayed when the middle mouse button is pressed in an RXI window while logged into an R1000, then the correct pathname was not used, the file has not been installed, the file contains garbage, the file has been deleted, or the file is protected. It will not be possible to customize RXI on a system-wide basis without these files in the correct location.

17.4.5. MANDIR Directory

DESCRIPTION: Where the man pages (or help files) reside.

LOCATION: The location of the Man pages vary on different machines. The Makefile contains an entry for the default location of the Man pages on the system.

The man pages should be installed on the same machine/filesystem as the binary executable. If a user expects machine ABC to be able to run RXI then he will also expect that machine to provide help. RXI's man/help files differ depending upon the vendor.

ERRORS: If the RXI man pages were not installed in the proper location, the message "No manual entry for RXI" or "No help available for RXI" will be displayed when `man rxi` or `help rxi` is used.

17.4.6. OPENWIN Directory

DESCRIPTION: Sun Microsystem OpenLook (a.k.a. OpenWindows, NeWS) systems only. Where OpenWindows resides.

LOCATION: `$OPENWINHOME`

The `OPENWIN` macro will use the current definition of the `OPENWINHOME` environment variable and RXI will install its binary, fonts, man page, and resources files directly into the OpenLook directories.

ERRORS: The definitions in the Makefile for OpenWindows (X/NeWS) applications assume that the installer is actually operating under the OpenLook/OpenWindows/NeWS system. This implies that you have the `$OPENWINHOME` environment variable set. Do this command: `echo $OPENWINHOME`. If you get an error then you are not operating OpenLook.

If `FONTDIR` is not to be `$(OPENWIN)/lib/fonts` then each and every RXI user will have to tell the OpenLook server where the RXI fonts can be found. This will be done every time a user starts his OpenLook server. Read the OpenLook manuals to determine the command to use.

18. Common Installation Questions and Problems

18.1. Not using default location for RXI objects

RXI executables, RXI fonts, RXI resources, RXI man/help. See the "Workstation/Fileserver Components" section for a discussion of the location for RXI objects.

18.2. Hetrogenous networks

RXI physically runs on machine A, user is sitting at machine B with the RXI window, RXI executables, resources, and help files go on machine A, the fonts go on machine B.

If there are lots of machine A's then they all get the fonts. If they all share a fileserver then that is good enough, put the fonts there.

If there are lots of machine B's then they all get the binary/help/defaults. If they all share a fileserver then that is good enough, put the files there.

18.3. Using links for default resources files

Install the files wherever you want, then set up soft or hard links to those files in the app-defaults area.

18.4. Key Foobar doesn't work

Use the Prompt-For key, then press the key that doesn't work. If the R1000 doesn't "see" the key (nothing happens), or if the R1000 gives the "wrong" name for the key, then the wrong terminal type is probably being used.

- Use the command **Io.Echo (System_Uilities.Terminal_Type)**; Does it print the expected terminal type name?
- What keyboard is being used? Check the installation instructions for RXI, is this keyboard layout supported?
- What operating system (and what version) is being used? Check the installation instructions for RXI, is this OS/version supported?
- What windowing system (and what version) is being used? Check the installation instructions for RXI, is this system/version supported?
- Use the Control-Middle-Mouse-Button menu, at the bottom, is the Rational Mode entry checked? If not, then you aren't really logged in as an RXI terminal type. Was the R1000 rebooted (or was the Refresh_Terminal_Information command executed) as per the instructions?
- Is there an entry for this terminal type in the !Machine.Editor_Data.Terminal_Type file?

- Is there an entry for this terminal type in the `!Machine.Editor_Data.Terminal_Recognition` file?

18.5. Missing resources files

No menu, wrong default window size, wrong auto-recognition when logging in. See the "RXI Default Resources File" section and the "XAPPLOADDIR Directory" section for information about RXI default resources.

18.6. Questions and Answers

Q. The X-Client and X-Server must be able to communicate with each other. It is not enough to have Vaxstations (VMS) with DECwindows as X-Server and a Sun with MIT's X11.R3 that could host an X-Client. This is because the X-Server expects clients to issue X Windows requests using the LAT protocol whereas X-Clients hosted on the Sun issue requests using the TCP/IP protocol.

Is this true? Is the X Windows protocol based on for instance TCP/IP or LAT which make X-Clients and X-Windows invisible to each other if they do not communicate on the same network protocol?

A. The X protocol doesn't, itself, care about the transmission medium. It is a simple stream-based protocol. A simple bi-directional stream of bytes. Things often get confusing here for the field because they often don't really understand network communications.

An X client is working with two different, and separate, protocols. First, it is communicating with an X server via some medium. That medium, today, is usually TCP/IP. TCP/IP is a communications protocol that is used to allow two machines to communicate. Neither the X server nor the X client actually deal directly with the TCP/IP protocol. Rather, they ask the operating system to open a connection to another machine and to use TCP/IP as the protocol. After that they are both dealing, in their minds, simply with a stream of bytes. That stream of bytes has a structure, that is only known to them, that is called the X protocol.

The X protocol can be used across any communications medium that supports the transmission of bi-directional byte streams. This means that it can be sent across TCP/IP, SNA, X.25, and almost anything else you can think of.

However, the X server being used, and any X applications being used, must share a common communication protocol. Both must talk TCP/IP, or DECnet, or shared memory, or SNA, or dial-up modems, or something. Two programs that don't have a common communication protocol can't talk. This is true regardless of the type of the programs, X applications are just one example of this situation.

Folks seem to get the X protocol, and the underlying transmission method, or protocol, confused. An analogy that I sometimes use is that of two people talking on the phone. They don't care if the phones are analog or digital. They don't care if the phone company uses wire, glass fibers, or satellites. They just care that one person knows the phone number of the other and that they both talk the same language.

X programs are similar. They know a certain number of phone numbers (TCP/IP, SNA, DECnet, etc.) If they can call that number and get an answer then they can talk. (The names of the servers are like calling different area codes. You get directory assistance if you dial 555-1212 no matter what area code you use. But you get a different directory assistance operator.)

Q. The X-Server produces key events if the user uses the keyboard or mouse. These events are sent as numbers to RXI. The numbers depend on the used keyboard and hopefully not on the version of X Windows on the X-Server.

A. Sorry, they depend exactly on the X server. It is the X server that determines the symbolic name(s) of each key. Those symbolic names, encoded as 16-bit numbers, are what RXI uses to talk to R1000's. It is also those symbolic names that allow a user to rearrange his keyboard at will. He can do this slightly, such as eliminating an annoyingly placed Caps-Lock key, or he can do this radically, such as changing his keyboard into a Dvorak layout.

Q. X-Windows defines a big list of possible keys consisting of a logical name such as Escape, Execute, or Undo, and corresponding numbers.

A. They call them "symbols", or symbolic names. They are supposed to correspond to the names commonly found printed on the keys themselves. (There is very little logic in some of the names.)

Q. Different versions of X-Windows should send the same number to the XClient if someone presses the Undo button on a Sun4 keyboard for instance. In the other direction, Xrequests coming from the RXI X-Client are drawn on the screen in the RXI window by the X-Server.

A. This may or may not be true. "presses the Undo button" is ambiguous. The answer is one way if you mean "the key with UNDO printed on it" and the answer is another way if you mean "the key with the symbolic name UNDO". Different versions, or versions from different vendors, of the X server will typically send the same "key code", or numeric value to an X client if the same key is pressed.

However, whether or not a key is an "Undo" key is a function of the symbolic name that is currently attached to a key by the server. Since the user is free to (is actually encouraged to) change the symbolic names of keys, we must work with the symbolic names and not the physical key codes.

By the way, there is no guarantee that two different X servers will use the same numeric key codes for the same physical key. They will *typically* report to the X client, the physical hardware key number that was given to them by the hardware. This is not required or guaranteed. However, some keyboards do not report simple numbers to their hardware drivers when a key is pressed. So, for some keyboards, the so-called "physical key code" is a figment of the mind of the X server and so cannot be considered to be a constant common to all X servers for that machine.

· It's like trying to get a firm hold on a ghost. You see it, but there's nothing really there to grip.

Q. Can different versions of X Windows X-Servers send different key numbers for a key on the same physical keyboard?

A. Let's use the X terminology.

The so-called physical key code (a numeric value in the 8..255 range, (0..7 are reserved by the protocol)) is usually the same across X servers. But there is no guarantee and there are known machines where, if a 2nd vendor ever appeared, the key numbers could easily change. The DEC keyboard comes to mind here; the hardware reports keyboard X/Y coordinates for keys and not simple key numbers. The key code for the X server is something constructed from a table. Tables can change.

The very-non-physical key symbol (also a numeric value, but in the 0..2**16-1 range) is something that users can, and do, change at will. Key symbols are the defined way of making applications semi-independent of keyboards.

Q. If X-Server and X-Client have different versions of X Windows, which is relevant for the choice of the installation of the RXI application?

A. Both. (And isn't that annoying!)

The vendor, and version, of the X server is important because that is what determines:

- the RXI keymap that you install on the R1000
- which form of the fonts you have to install for consumption by the X server.

The vendor, and version, of the X libraries on the workstation that executes the RXI program is important because that is what will be used when RXI executes. If a vendor supplied shared-libraries then the customer will typically want to relink RXI to use those shared libraries in order to reap the runtime swap space benefits that result.

The vendor, and version, of the workstation operating system are also relevant here. RXI must be able to execute on the machine.

Also note, the version of the X libraries and the version of the operating system can be independent of each other. This means that if some X vendor has two different servers, and if they both run under two versions of some OS vendor's operating system, then potentially, we could have four different versions of RXI (!!!) that we have to support.

Q. The X-Client is the actual RXI application. For hosting the RXI application the following items are relevant:

- a version of X Windows
- an operating system
- the help facility
- in case of Vax VMS, the TCP/IP communication package for communication with the R1000

The RXI application is basically a terminal simulation. The user must start this RXI application on the machine on which it was installed. The usual way in which this happens is: The user uses some remote login to connect from the X-Server that he is using to the RXI host. There he starts RXI with the option to open the display on the X-Servers screen. The RXI application simulates a terminal in the RXI window and the User still has the prompt

of the RXI host in this window. To connect to the R1000, the user typically uses telnet from this RXI window.

The RXI application for Vax VMS is a bit different, it builds the connection to the R1000 itself since Vax VMS does not allow one to build a terminal simulation that sets itself between the user and the operating system. Hence the dependency on the communication package for the communication between the RXI application and the R1000.

The RXI application turns keypress events coming from the X-Server into escape sequences. Normal characters in the ascii range are passed along unchanged. keynumbers corresponding to other keys are transformed into escape sequences. It will do this for any valid X Windows keynumber even if it does not belong to the keyboard type for which the RXI application was installed. This allows the user to build a telnet connection to the R1000 from the RXI window, so the R1000 will get these escape sequences.

Does this mean that one could install an RXI application of the type Xsun4_X11r3 on a Sun and use for instance Apollo workstations as X-Servers, as long as one installs the correct keymaps on the R1000?

A. Yes. RXI cares about its own personal execution environment only to the extent that it must be compiled using the correct compiler and linked with the correct libraries. It must be able to execute before it can do anything else (like talk to an R1000 or open a window on a server).

After that, it can place its window on any X server that is willing to talk to it and it can connect to any R1000 that is willing to listen. RXI literally does not care one little bit about what kind of keyboard is being used, or what kind of mouse is being used, or how they are set up. Whatever it receives from the X server it will blindly pass along to the R1000.

RXI does its very best to act like a simple piece of wire between the user's keyboard/mouse and the R1000. At most, RXI could be considered to be a protocol translation mechanism between what the X server produces and what the R1000 consumes. The translation is very simple and totally unintelligent.

By the way, terminology again:

RXI receives events that contain "key codes". These codes are looked up in a table, provided by the X server, and they are translated into "key symbols". These symbols are 16-bit numeric values. They, and any associated Shift, Control, and Meta flags that may be present, are taken, as a 19-bit number. That 19-bit number is broken up into 2..4 pieces and those pieces are encoded as printable ASCII characters. Those characters are then sent to the R1000. The phrase "escape sequence" is typically reserved for ANSI ASCII standard terminal sequences that begin with the character ESCAPE. RXI does not use ESCAPE because it is not a printing character.

Q. The R1000 translates escape sequences into symbolic keynames using the files in !Machine.Editor_Data. It doesn't even know that certain installed keymaps correspond to physical terminals and other correspond to the X-Server keyboards whose keynumbers are translated to escape sequences by the RXI application. Since different keyboard cause the X-Server to send different keynumbers, a keymap must be installed for each X-Server keyboard that will be used using RXI.

Is this true? Is the combination of R1000 keymap and X-Server currently used actually independent of the RXI application and where it runs?

A. Yes, the keymap reflects the combination of a particular vendor/version X server and a particular vendor/version/OS keyboard. The version of RXI, and the vendor/version/OS that is being used to run it, are irrelevant. RXI tries to be just a wire between any two points.

(Note: If you get historical, then versions of RXI prior to rev10_4 used a different encoding scheme for mouse actions. However, I don't think anyone is using such an ancient RXI anymore. It was still independent of the keyboard, but the keymap files on the R1000 would be different than more recent RXI keymaps.)

Q. If this is not the case, you might be forced to install two RXI application on the RXI host to support different X-Servers.

A. This *should* not be the case. However, some vendors might have a very-old, or even a very-new, X server. In this case it is possible that one version of RXI would be required for one and another for the other. This would be due to protocol differences.

This has never quite happened. When MIT put out X11R4 it broke RXI. Old versions of RXI would not work with the new MIT server. However, the new RXI, that works with the new server, will work just fine with any old server. Fortunately, to date, all protocol changes have been backwards compatible. That is, a client that works with the new protocol will also work with the old protocol.

So, getting a second X server might cause a customer to request a new version of RXI, but, when the new one was installed, the old one would just go away.

Q. Where does the mouse come in here? Isn't it the case that

- the RXI application translates mouse events in escape sequences
- mouses can be different for different X-Servers (two or three buttons for instance)

hence the RXI application depends on the choice of the X-Server?

A. RXI turns mouse events into a series of keystrokes. When the user pushes a mouse button, the R1000 will see a series of keystrokes come in.

These keystrokes use character sequences that cannot be generated by any X protocol key event. These "mouse keys" were invented and defined in such a way that they are outside the set of character sequences generable by all possible X key events and do not conflict with them. They provide a means whereby RXI can generate fake keystrokes that are not generable by the user. This allows RXI to report mouse events to the R1000 without confusion or conflict.

The X protocol provides events that tell an X client that Button1 (or Button2, or Button3, or Button4, or Button5) has been pressed (or released). Like with keystrokes, RXI merely translates what it receives and fires this off to the R1000. RXI does not interpret.

So, if some mouse has only two keys, and if the X server does not provide any way to fake a third key, then a user with that mouse has only two keys. It is very much the same situation of you have one user with 10 function keys and another user with 9 function keys.

One user has one less function keys and that is that.

Most X servers that support 2-button mice will also provide a 3rd fake key. You press the fake 3rd key by simultaneously pressing both keys. In a case like this, the X client has no idea what is going on. It only knows that it got an event that said "Button3 goes down".

An X client has no way of knowing:

- what the keyboard looks like physically,
- how many keys the keyboard really has,
- how many buttons are on the mouse,
- whether the "mouse" is really a "light pen" or something else

All it knows, all it can know, is "key number 93 went down" and "Button2 went up" and things like that. It only knows what the X server is willing to tell it. There is no way to query about hardware configuration.

19. FEEDBACK

In order to help us improve our instructions and make installations easier to do, the following form is provided to allow you to give us feedback. Please complete and send along with a printout of the file `Do_Step_Execution_Time` located in `!Machine.Release.Archive.X_Interface.Release10_9_0.Logs` to:

Rational
Atten: SMSE
3320 Scott Blvd.
Santa Clara, CA 95054-3197
Re: X_Interface

1. How long did the installation take you?
2. Did the installation proceed as described in the instructions?
3. What could be done to improve these (or other) instructions/installations?
4. What did you like about this set of instructions/installation?

Rational X Interface Release Information

Release10_9_0 Release

Copyright © 1989-1990 by Rational

Part Number: 508-003219-005

03/29/91 (Software Rev. Release10_9_0)

Rational and R1000 are registered trademarks and Rational Environment and Rational Subsystems are trademarks of Rational.

UNIX is a registered trademark of AT&T in the USA and other countries.

Apple, A/UX, and Macintosh are registered trademarks of Apple Computer, Inc.

DEC, DECnet, DECstation, DECwindows, ULTRIX, VAX, VAXstation, and VMS are trademarks of Digital Equipment Corporation.

Excelan is a registered trademark of Excelan, Inc.

AIX, AIXwindows, IBM, and RISC System/6000 are trademarks of International Business Machines.

The X Window System is a registered trademark of the Massachusetts Institute of Technology.

NCD16 and NCD19 are trademarks of Network Computing Devices, Inc.

NeWS and Sun are registered trademarks, and, Sun Workstation, SunOS, Sun-3, and SPARC are trademarks of Sun Microsystems, Inc.

MultiNet is a registered trademark of TGV, Inc.

WIN/TCP is a trademark of The Wollongong Group, Inc.

Ethernet is a trademark of the Xerox Corporation.

Rational
3320 Scott Boulevard
Santa Clara, California 95054-3197

Contents

1. Overview	1
2. Configurations	1
2.1. Apollo	1
2.2. Digital Equipment Corporation	2
2.2.1. DEC VMS	2
2.2.2. DEC ULTRIX	2
2.2.3. Installation Options	3
2.3. Hewlett-Packard	3
2.4. IBM	4
2.5. NCD	4
2.6. Sun Microsystems	5
3. Compatibility	5
4. Upgrade_Impact	5
5. Known Problems	6
6. New Features	6
7. Changes	6
8. Documentation	6
9. Copyrights	7

1. Overview

The Rational X Interface (RXI) allows users to access the Rational Environment via their Ethernet from any of the supported workstation or X terminal platforms.

The Release10_9_0 release of the Rational X Interface supports the following platforms:

- Apollo - DN3000
- DEC - all VMS platforms using MultiNet or Wollongong TCP/IP software.
- DEC - ULTRIX 4.1 for the DECstation or the VAXstation architectures.
- Hewlett-Packard - 9000 series 300/800.
- IBM - RISC System/6000.
- NCD - NCD16 and NCD19 hosted by any other supported platform.
- Sun - workstation models: 3, 4 (68K and SPARC).

See the Configurations section below for a complete listing of supported workstation/X-terminal hardware configurations.

This is the first production release of the DEC ULTRIX support.

Installation instructions for RXI may be found in the Rational X Interface Release10_9_0 Installation Procedures document(s).

2. Configurations

Release10_9_0 is compatible with Rational Environment versions D_10_20_0 or later. However, a small number of key bindings will not be available under D_10_20_0; these pertain to Environment commands that did not exist prior to D_12_1_1.

2.1. Apollo

RXI was built and tested under Domain SR 10.2 using the BSD UNIX interfaces. It should operate correctly on any Domain platform. It requires:

- Domain SR 10.2.
- BSD interfaces.
- Standard type 3 keyboard, or the modified MOTIF compliant type 3 keyboard.

Domain, the BSD interfaces, and TCP/IP must be installed and running before RXI can be successfully installed.

DECwindows must be installed but need not be actually running before RXI can be successfully installed.

RXI can be installed in three ways under Domain.

A "binary" installation simply installs the Rational-supplied RXI executables on the workstation.

An "object module" installation relinks RXI with local shared libraries for increased runtime efficiency.

A "full source" installation recompiles RXI from the C sources.

2.2. Digital Equipment Corporation

2.2.1. DEC VMS

RXI for VMS is built and tested on a VAXstation 3100. It should operate correctly on any VMS platform. It requires:

- VMS 5.1B, 5.2 and 5.3.
- DECwindows 5.1, 5.2, and 5.3.
- DEC standard LK201 keyboard; other keyboards are not presently supported.
- Either of:
 - Excelan MultiNet version 2.0.3-D.
 - Wollongong WIN/TCP version 5.0.2.

VMS must be installed and running before RXI can be successfully installed.

MultiNet or WIN/TCP must be installed and running before RXI can be successfully installed.

DECwindows must be installed but need not be actually running before RXI can be successfully installed.

2.2.2. DEC ULTRIX

RXI for ULTRIX is built and tested on both a DECstation 3100 and a VAXstation 3100. It should operate correctly on any DECstation or Vax platform. It requires:

- ULTRIX 4.1.
- DECwindows for ULTRIX 4.1.

- DEC standard LK201 keyboard; other keyboards are not presently supported.

ULTRIX and TCP/IP must be installed and running before RXI can be successfully installed.

2.2.3. Installation Options

RXI can be installed in three ways on VMS or ULTRIX:

A "binary" installation simply installs the Rational-supplied RXI executables on the VMS platform.

An "object module" installation relinks RXI with local shared libraries for increased runtime efficiency.

A "full source" installation recompiles RXI from the C sources.

Both the "object module" and the "full source" installation require a C license from DEC. The "binary" installation may be performed without a C license. RXI can be installed as a "sharable image" using DEC's Install facility; this decreases the overall memory requirements when running multiple copies of RXI on one platform.

2.3. Hewlett-Packard

RXI is built and tested on an HP 9000 series 300. It should operate correctly on any HP 9000 platform. It requires:

- HP-UX 7.0.
- HP's X Window System release 11 version 2.
- Model 46021A keyboard; other keyboards are not presently supported.

HP-UX must be installed and running before RXI can be successfully installed.

The X Window System must be installed but need not be actually running before RXI can be successfully installed.

RXI can be installed in three ways on HP-UX.

A "binary" installation simply installs the Rational-supplied RXI executables on the 9000 platform.

An "object module" installation relinks RXI with local shared libraries for increased runtime efficiency.

A "full source" installation recompiles RXI from the C sources.

2.4. IBM

RXI is built and tested on a RISC System/6000 model 530. It should operate correctly on any RISC System/6000 platform. It requires:

- AIX 3.1.
- AIXwindows version 3.1.
- IBM standard PC keyboard; other keyboards are not presently supported.

AIX must be installed and running before RXI can be successfully installed.

AIXwindows must be installed but need not be actually running before RXI can be successfully installed.

RXI can be installed in three ways on AIX.

A "binary" installation simply installs the Rational-supplied RXI executables on the RISC platform.

An "object module" installation relinks RXI with local shared libraries for increased runtime efficiency.

A "full source" installation recompiles RXI from the C sources.

The IBM X-120 X terminal is also a supported platform. The X-120 keyboard is physically identical to the RS/6000 keyboard. In addition, the software keyboard translation tables within the X server for the RS/6000 and the X server for the X-120 are sufficiently close that the XR6US terminal type on the R1000 is used for both keyboards.

2.5. NCD

RXI supports the NCD as yet-another-keyboard attached to any supported workstation platform. RXI is installed for an NCD by installing the NCD-specific files on an R1000 and then by installing the appropriate workstation files on some host workstation. RXI actually runs on the host workstation and the RXI window and menus appear on the NCD screen.

The only difference between RXI on a platform and RXI on an NCD hosted by that same platform is the type and layout of the keyboard.

RXI supports the NCD16 and NCD19 terminals using the NCD101 keyboard.

The NCD terminal must be installed and fully operational before RXI can be successfully installed.

2.6. Sun Microsystems

RXI is built and tested on a Sun model 3/6 and on a Sun SPARC. It should operate correctly on any model 3, 4, or SPARC platform. It requires:

- SunOS 4.1.
- Either of:
 - X Window System from MIT, version 11, release 3 or 4.
 - X11/NeWS version 2.0.
- Model 3 or model 4 keyboard; other keyboards are not presently supported.

SunOS must be installed and running before RXI can be successfully installed.

The windowing system must be installed but need not be actually running before RXI can be successfully installed.

RXI can be installed in three ways on SunOS

A "binary" installation simply installs the Rational-supplied RXI executables on the Sun platform.

An "object module" installation relinks RXI with local shared libraries for increased runtime efficiency.

A "full source" installation recompiles RXI from the C sources.

3. Compatibility

Release10_9_0 is fully compatible with Rational Environment version D_12_1_1.

Release10_9_0 is fully compatible with Rational Environment version D_10_20_0 although the keybindings for optional products are only available if the optional products are available for D_10_20_0.

The IBM PC/RT running AIX 2.2 is no longer a supported platform.

4. Upgrade_Impact

Release10_9_0 of RXI completely replaces all previous versions for all platforms. Installation will replace all existing files and programs. Any local customizations that may have been performed will be overwritten by the new files and programs.

5. Known Problems

The Motif window manager requires most "meta" function keys for its own use. (A meta function key is any function key that is pressed while the "meta" key is also pressed.)

This means that user applications (RXI included) cannot use M_F1, M_F2, etc. The HP 9000 and the IBM RISC System/6000 versions of RXI are configured to be usable under Motif. This means that the R1000 keymaps for these configurations do not have any Rational Environment functions bound to the meta function keys.

On Apollo systems, if the X server is configured using a DM-owned root window, the server must be started using a -K switch that provides the X server will full visibility to all of the "gray" keys. The R1000 keymaps for the Apollo assume that these keys will be available for use with RXI. Without them the keymap's usefulness is reduced.

6. New Features

ULTRIX 4.1 on DECstation and VAXstation architectures is now supported. Earlier version of ULTRIX are no longer supported.

7. Changes

There are no changes in this release. See New Features.

8. Documentation

A version of the RXI User's Guide is available for these platforms:

- DEC (LK201 keyboard) (with an appendix for the NCD (NCD101 keyboard)) - Product Number 4000-00425
- Sun (model 3 keyboard) - Product Number 4000-00226

Keyboard overlays are available fore these keyboards:

- Apollo (type 3 keyboard) - Product Number 4000-00226 (both type 3 and Motif)
- DEC (LK201 keyboard) - Product Number 4000-00427 (both VMS and ULTRIX)
- IBM (RS/6000 or X-120 keyboard) - Product Number 4000-00479
- HP (46021A keyboard) - Product Number 4000-00484
- NCD (NCD101 keyboard) - Product Number 4000-00428

- Sun (model 3 keyboard, MIT X11) - Product Number 4000-00192
- Sun (model 4 keyboard, MIT X11) - Product Number 4000-00317
- Sun (model 3 keyboard, X11/NeWS) - Product Number <no longer available>
- Sun (model 4 keyboard, X11/NeWS) - Product Number 4000-00533

9. Copyrights

In addition to objects written by Rational, the RXI distribution includes some objects originally delivered with the X Window System from MIT. All objects which originated in the X Window System distribution contain the original copyright notice in their source file. This includes objects delivered with the X Window System and modified by Rational.

The following copyright notices pertain to those objects that have not been written wholly by Rational.

Copyright 1989 - 1990 by Rational, Santa Clara, California. Copyright 1988 by Wyse Technology, Inc., San Jose, California. Copyright 1987 - 1989 by Digital Equipment Corporation, Maynard, Massachusetts. Copyright 1987 - 1989 by Massachusetts Institute of Technology, Cambridge, Massachusetts.

All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice(s) appear in all copies and that both that copyright notice(s) and this permission notice appear in supporting documentation, and that the names of Digital, MIT, Wyse, or Rational not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

Digital, MIT, Wyse, and Rational disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall Digital, MIT, Wyse, or Rational be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Installation Procedure

Release10_1_1

Copyright © 1991 by Rational

PN 507-003275-002

This document is subject to change without notice.

Rational and R1000 are registered trademarks and Rational Environment is a trademark of Rational.

Rational, 3320 Scott Boulevard, Santa Clara, California 95054-3197

Table Of Contents

1 Installation	1
Overview	1
Assessing Impact	2
Verifying Prerequisites	2
Terminal Types	3
Installing The Release On The R1000	3
Installing The Release On The PC	3
Testing RWI	5
Cleaning Up	5
Appendix A Procedure Do_Step	7
Appendix B Install_RWI	9
Appendix C Customer Checklist	11
Appendix D Feedback	13

1

Installation

Overview

Estimated time for installation: 1 hour.

This package is composed of:

- 1. RWI Release10_1_1 tape for R1000
- 2. RWI Release10_1_1 diskette for the PC with the following files:
 - a. InstRwi.exe
 - b. RwiSer.exe
 - c. RwiNet.exe
 - d. Rwi.mnu
 - e. PcBasic.mnu
 - f. Fontres.fon
- 3. These installation instructions
- 4. RWI release note

When installing the RWI release, you can view the upgrade process as a series of phases. These phases must be performed in a serial manner.

- 1. Assessing the impact of the release on the user community
- 2. Verifying prerequisites to the installation
- 3. Verifying the terminal types supported

- 4. Installing the release on the R1000
- 5. Installing the release on the PC
- 6. Testing the new release
- 7. Cleaning up

The following sections take you through the installation process a phase at a time. Be sure to complete ALL of the phases of installation and to perform these phases in the order in which they appear here.

Warning: *Follow these installation steps in a serial manner. Failure to adhere to these instructions may result in the failure of portions of the installation.*

Assessing Impact

In this phase of the install, you will assess the impact of the release on the user community and the machine.

Keymaps are updated; therefore, previous keymaps are unrecoverable unless backups of the keymaps have been done before the installation.

Verifying Prerequisites

In this phase of the installation, you will verify that the prerequisites for installing this upgrade have been met. Do not proceed with the upgrade until all prerequisites have been fulfilled.

- 1. You have read the release note.
- 2. You have read this entire install note.
- 3. The R1000 is executing Environment release D_12_1_1 or later.
- 4. The PC is running MS-DOS or PC-DOS version 3.1 or later.
- 5. On PC's that will not use serial communication, networking must be installed and running on the PC. This release supports Novell Lan Workplace Rev4.0 with Microsoft Windows 3.0.
- 6. For network users the PC must be running Microsoft Windows 3.0, and Windows must be set up to use the network.

- ❑ 7. For serial users the PC must be running Microsoft Windows 3.0, must have a serial port or a modem depending on the remoteness of the machines.

Terminal Types

The following terminal type configurations are supported in this release.

- **Pc101_Ms_Windows_3_0** - 101/102 key keyboard. Numeric keypad on right, arrow and other keys between alpha keys and numeric keypad.
- **Pc91_Ms_Windows_3_0** - 91 key keyboard. Numeric keypad on right, no keys between alpha keys and numeric keypad.
- **Pc86_Ms_Windows_3_0** - 86 key keyboard. No numeric keypad on right.
- **PcBasic_Ms_Windows_3_0** - Compatible with the above three keyboards. Designed for training purposes.

Installing The Release On The R1000

In this phase of the release, you will restore the tape which contains a release world located in `!Machine.Release.Archive.RWI.Release10_1_1`. This release world contains nested archives which will then be restored to complete the installation.

- ❑ 1. Load the RWI Release10_1_1 tape onto the tape drive.
- ❑ 2. Log in as user `Rational`, which must be a member of group *Privileged*. Do NOT use another account due to problems which might arise with product token usage.
- ❑ 3. **LOAD_TAPE** *[3 minutes]*
This step restores RWI Release10_1_1 tape. This step can be executed from any command window. There should be no errors. If `Do_Step` is not available, use `Archive.Restore (Options => "Promote, Replace");`
- ❑ 4. Go to `!Machine.Release.Archive.RWI.Release10_1_1`.
- ❑ 5. **INSTALL_RWI** *[45 Minutes]*
This step will execute the steps necessary to do most of the installation on the R1000. The appendix `Install_RWI` contains more detailed information about the steps executed by this step and possible errors. This step may be run individually if desired.

Installing The Release On The PC

The software for the PC is delivered on a diskette. The following examples assume the diskette will be read using drive A.

Windows must be running in order to install RWI.

- 1. Insert the RWI diskette in drive A.
- 2. Execute "a:instrwi.exe" from Windows.
 - Pull down the **F**iles menu and select the **R**un command. The **R**un dialog box opens.
 - Enter **a:instrwi** in the **C**ommand **L**ine edit box.
 - Press [Enter].

Follow the instructions, and if you are not sure on any of the choices, use the default value provided. The default values will install RWI in the following configuration. These settings may not be correct for your PC/Modem. Verify they are correct for your application and change them as necessary.

Data bits: 8
Stop bits: 1
Parity bit: None
Flow Control: None

- 3. (optional) Add the RWI icon to a Program Manager group.
 - Select the Program Manager group in which you want to place the RWI icon by moving the mouse pointer over the group and double-clicking the left mouse button.
 - Pull down the **F**iles menu and select the **N**ew command.
 - Single-click on the **P**rogram **I**tem radio button if it is not already selected.
 - Press [Enter]. The **P**rogram **I**tem **P**roperties dialog box opens.
 - Enter **RWI** in the **D**escription edit box.
 - Enter **RWI** in the **C**ommand **L**ine edit box.

- Press [Enter]. The Program Manager adds RWI to the group.

You can now start RWI one of two ways; by using the Run command or Application-Group Icon from the Program Manager.

- Run command of Program Manager:
 - Pull down the **F**iles menu and select the **R**un command. The Run dialog box opens.
 - Enter **R**WI in the **C**ommand **L**ine edit box.
 - Press [Enter].
- Application-Group Icon in a Program Manager group:
 - Place the mouse pointer on the RWI icon.
 - Double-click the left mouse button.

See the Getting Started section in the RWI User's Guide for more information on setting-up and running RWI.

Testing RWI

After the installation is complete, log-on to the R1000 from the PC and try some function keys and menus to verify RWI is working correctly.

Cleaning Up

In this phase, you will perform some cleanup activities. The time required for this phase will vary widely, depending on how much cleanup is required to be done.

- 1. Go to `!Machine.Release.Archive.RWI.Release10_1_1.Logs` and print out the file `Do_Step_Execution_Time`. Return this printout to SMSE along with other feedback supplied on the last page of this note. Note that this step must be executed prior to performing the following step which will destroy the `Do_Step_Execution_Time` file.
- 2. **CLEANUP**

This step destroys the release and archive libraries. Only execute this step when you are confident that the release has been properly installed.
- 3. Have customer take a full Environment backup.

A

Procedure Do_Step

These phases must be performed in a serial manner, as must the steps in each of the phases. This installation note is organized by phase, with each phase comprising a section of this document. Once you have performed this installation, please fill out and return the feedback form that is the last page of these instructions along with a printout of the `Do_Step_Execution_Time` found in the `Logs` library of the release archive. Your feedback is very important in helping us to improve our installation instructions.

The majority of the steps which must be performed for this installation utilize the procedure `Do_Step` (included as part of the release tape and located in `!Commands`) to execute the required sequence of commands which implement the step. This procedure **must always** be executed from a command window in the release library

```
!Machine.Release.Archive.RWI.Release10_1_1
```

Each step which utilizes procedure `Do_Step` will be of the form

```
<STEP_NAME>                [<TIME TO EXECUTE>]  
                             <DESCRIPTION>
```

where `<STEP_NAME>` is passed as the parameter to procedure `Do_Step` which performs the necessary actions to complete the step, `<TIME TO EXECUTE>` is the amount of time the step takes to execute (this may be expressed as a range, in which case the lower value is typically the minimum, and the upper value is an estimate of the maximum), and `<DESCRIPTION>` is a description of what the step does, including any action which you will need to manually perform. For example:

```
FOO                            [5 Minutes]  
                                This step is an example of the format used for  
                                steps which are executed using procedure  
                                Do_Step. To execute step FOO, you would go to  
                                !Machine.Release.Archive.RWI.Release10_1_1  
                                and in a command window, execute:
```

```
Do_Step ("FOO");
```

which would result in automatic execution of all commands required to implement this step. If you needed to take any manual action in addition to executing `Do_Step`, it would be noted as :

□ Perform some manual check

If any errors occur for this step, you should always fix the problem before proceeding on to the next step. Each step is defined by a fragment of Ada code which is executed by `Program.Run`. These fragments are stored in the `Steps` file located in the `Command_Data` library of the release. In the event you want to modify a step, you can use the "PROMPT => <STEP>" form when invoking `Do_Step`. See the spec of procedure `Do_Step` (located in `!Commands`) for more detailed information.

Warning: *If you interrupt the execution of `Do_Step` (by using `Job.Interrupt` such as `CONTROL-G`) it is possible that certain interactive commands executed by some steps, such as `Common.Definition`, may fail with an exception and display a message such as*

Unable to read file due to Constraint_Error (Null Access)

In general, this message has no negative impact on the execution or completion of the step, and can be ignored.

B

Install_RWI

The step `INSTALL_RWI` runs the following steps in this order.

- ❑ 1. `RELEASE_RESTORE` *[10 Minutes]*

This step restores the `Release` archive. When completed, a filtered error log is displayed (`Restore_Release_Log_Summary` located in `!Machine.Release.Archive.RWI.Release10_1_1.Logs`) Examine this log for errors. Ignore the following types of errors (if present):

```
!!! can't restore link...

++* Can't resolve default
    switch file ...
```
- ❑ 2. `INSTALL_KEYMAPS` *[15 minutes]*

This step installs all the keymaps for PcBasic, 101/102, 91, and 86 key keyboards, and restores their respective keymap overlays. When completed, a filter error log will be displayed. Check this log for errors, there should be no errors.
- ❑ 3. `RESTORE_NOTES` *[1 minute]*

This step restores RWI Release Notes into `!Machine.Release.Release_Notes`.
- ❑ 4. `RECORD_INSTALLATION` This step records release information in `!Machine.Release.Current.Products` and sets the login limit back to unlimited logins. This is a required step and establishes that the release has successfully been installed.
- ❑ 5. The RWI Keymaps must be enabled before they can be used.
 - Create a Command window and type:

```
Refresh_Terminal_Information
```

and press the `[PROMOTE]` key to enable the keymaps.

C

Customer Checklist

- 1. Each user should merge the contents of the default activity into his or her own private activity file with the command `Activity.Merge`.

```
Activity.Merge (Source => "!Machine.Release.Current.Activity",  
              Mode => Activity.Differential,  
              Target => ">>User Activity<<");
```

Note that this command may have the undesirable effect of modifying spec/load view settings which were explicitly set in the user activity.

- 2. Distribute the release note to user community. A postscript and lineprinter copy of the release note can be found in `!Machine.Release.Release_Notes`.
- 3. Update the Daily Message indicating that RWI Release10_1_1 has been installed.
- 4. Set the ACL's on new specs/files restored as part of this release, if desired.
- 5. Make a Full backup of the Environment.

D

Feedback

In order to help us improve our instructions and make installations easier to do, the following form is provided to allow you to give us feedback. Please complete and send along with a printout of the file `Do_Step_Execution_Time` located in `!Machine.Release.Archive.RWI.Release10_1_1.Logs` to:

Rational
Atten: SMSE
3320 Scott Blvd.
Santa Clara, CA 95054-3197
Re: RWI

- 1. How long did the installation take you?
- 2. Did the installation proceed as described in the instructions?
- 3. What could be done to improve these (or other) instructions/installations?
- 4. What did you like about this set of instructions/installation?

Rational Windows Interface Release Information

Release 10_1_1

Copyright © 1991 by Rational

Part Number: 508-003275-003

December 1991 (Software Release 10_1_1)

Rational and R1000 are registered trademarks and Rational Environment is a trademark of Rational.

Microsoft and MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation.

Novell is a registered trademark and LAN WorkPlace is a trademark of Novell, Inc.

IBM is a registered trademark of International Business Machines Corporation.

PostScript is a registered trademark of Adobe Systems Incorporated.

Rational
3320 Scott Boulevard
Santa Clara, California 95054-3197

1. Overview

The Rational Windows Interface (RWI) provides access to the Rational Environment™ from IBM®-compatible personal computers running Microsoft® Windows™ 3.0 and connected to R1000® systems through TCP/IP network or RS232 serial connections, including modems. The RWI application is a terminal emulator.

This is the first production release.

See the “Configurations” section, below, for a complete listing of supported hardware configurations.

Installation instructions for RWI are found in the “Installation Procedure for RWI, Release 10_1_1.”

2. Configurations

Release 10_1_1 is compatible with Rational Environment versions D_12_1_1 or later.

2.1. PC Hardware Requirements

- PC/AT-compatible or better
- 640K of conventional memory, minimum for Microsoft Windows Real mode; additional memory required for Standard and Enhanced Windows modes
- A monitor that is compatible with Microsoft Windows 3.0
- A keyboard with 12 function keys (note that custom changes can be made by the user that may allow other keyboards)
- Approximately 256K disk space available, local or server
- A mouse that is compatible with Microsoft Windows is recommended
- One of the following communications hardware configurations:
 - RS232 serial port or modem
 - Network adaptor that is supported by an ODI (ODLI) driver, such as those from Novell, 3Com, or IBM token ring support

2.2. PC Software Requirements

- Microsoft Windows Graphical Environment, version 3.0, operating in 386 Enhanced, Standard, or Real mode
- MS-DOS® or PC-DOS, version 3.1 or later
- If using a network connection, Novell® LAN WorkPlace™ 4.0 for DOS
- If you want to run a network operating system, it must be compatible with the ODI support required by LWP4

2.3. Installation

RWI is installed according to the “Installation Procedure for RWI, Release 10_1_1.” The installation requires that Windows be installed and running on the PC. If a network connection

is used, the network software should already be correctly installed.

If you are using RS232 serial communications with either a direct serial connection or a modem, and no network, then select a serial installation during the installation procedure. The installation procedure will install an executable file named RWISER.EXE.

If you are using a network, or a network and serial ports, then select the network installation during the installation procedure. The installation procedure will install an executable file named RWINET.EXE, which includes code for serials ports so that RWISER.EXE is not needed.

2.4. Executables

There are two RWI executables: one for basic serial connections, and one that adds network capability. The executable for basic serial connections is named RWISER.EXE, and the executable that adds network capability is RWINET.EXE. Note both of the following:

- The network version, RWINET.EXE, requires that the network software be installed.
- The network version includes code for serial connections.

See the "Installation Procedure for RWI" for instructions on installing the executable that is required for your installation.

2.5. Terminal Types or Keyboard Variants

The different types of PC keyboards supported by RWI are identified by a terminal type, or keyboard variant. RWI currently supports four of these keyboard variants:

- The PC101 keyboard variant provides a standard set of Environment key bindings for the enhanced keyboard with 101 keys.
- The PC91 keyboard variant provides a standard set of Environment key bindings for laptops that have twelve function keys and a separate numeric keypad.
- The PC86 keyboard variant provides a standard set of Environment key bindings for laptops that have twelve function keys and no numeric keypad.
- The PCBASIC keyboard variant provides a basic set of Environment key bindings that is much smaller than the standard set. The PCBASIC keyboard variant is designed for training purposes, and it works with all three of the above keyboards.

These keyboard variants are selected during installation and may be changed later by the user. The R1000 needs to be informed of the chosen keyboard variant. This can be done automatically through a setting in the RWI Terminal Information dialog box or manually in response to the Environment's Enter Terminal Type prompt. For more information, see the "Installation Procedure for RWI" and the *RWI User's Guide*.

2.6. Pull-Down Menus

User-defined pull-down menus are available in RWI. These pull-down menus are defined in text files stored on the PC. The initial RWI profile is installed with one of two sample pull-down menus, depending on which keyboard variant is chosen:

- A menu that is compatible with the standard set of Environment key bindings (PC101, PC91, PC86)
- A menu that is compatible with the basic set of Environment key bindings (PCBASIC)

Note that both sample pull-down menus are installed on the PC hard disk, so that the user can incorporate either menu in subsequent user-created custom profiles.

3. Compatibility

First production release of RWI.

Compatible with Environment release D_12_1_1 or later.

4. Upgrade Impact

None.

5. Known Problems

5.1. Flow Control and Pasting from the ClipBoard

When a serial connection is used, the flow control should be set properly to avoid problems when pasting large buffers of data from the Windows Clipboard into RWI.

Pasting large data buffers is slow, and a file-transfer utility may be more appropriate. One such utility is LAN WorkPlace 4.0 File Express, available to network users.

5.2. Environment Window Style and Copy to Clipboard

If the Window_Have_Sides session switch is set to True, then Copy to Clipboard includes border characters with the text sent to the Windows Clipboard. Set the Window_Have_Sides switch to False to correct this problem.

5.3. Incorrect RWI Window Size

If the RWI window itself is not contained within the boundaries of the display monitor screen, reduce the lines or columns or both through the Terminal Information dialog box:

- Pull down the RWI Control menu and select the Terminal command. The Terminal Information dialog box opens.
- Enter a smaller value in the Lines or Columns edit box.
- Single-click on the Save button with the left mouse button. The RWI window size reduces. Repeat this procedure until you are satisfied with the window dimensions.

If, after logging into the Rational Environment, you find that the Environment windows are not properly contained in the RWI window, try resizing the RWI window slightly with the mouse.

If the Resize Host Window check box in the Terminal Information dialog box is selected, then resizing the RWI window with the mouse will inform the Environment of the actual RWI window size, and the Environment will adjust its windows to fit the RWI window. This problem usually results from logging in with a nonstandard window size (the standard window size is 66 lines by 80 columns) without specifying the nonstandard window dimensions.

To specify nonstandard window dimensions when logging in, do one of the following:

- Select the RWI profile's Send Terminal Type check box (Telnet connections only)
- Edit the Initial Connection String to include the desired window dimensions as part of the Terminal Type entry
- Enter the window dimensions manually along with the terminal type (see the *RWI User's Guide*)

5.4. Cursor is not in a Window

If the cursor is not within an Environment frame when the RWI window is resized, the Environment window will be cleared and the message "cursor is not in a window" will be displayed. To fix this, reposition the cursor in an Environment frame, and resize the window.

6. New Features

First release.

7. Changes

First release.

8. Documentation

8.1. User's Guide

A user's guide is available for RWI:

- *Rational Windows Interface (RWI) User's Guide*, product number 3000-00613

Note the following changes to the *RWI User's Guide*:

- It is now possible to select a 19.2K baud rate from the Communications Setup dialog box (see Chapter 4).
- Two default menu files are supplied with RWI: PCBASIC.MNU and RWI.MNU.

8.2. Keyboard Overlays

A plastic keyboard overlay with replaceable paper inserts is available for:

- The 101-key Enhanced keyboard using the PC101 key bindings, product number 3000-00614

Paper keyboard overlays are provided with this release information for:

- The 101-key Enhanced keyboard
- The 91-key laptop keyboard
- The 86-key laptop keyboard

The paper overlays are provided online in !Machine.Editor_Data.Keyboard_Overlays as PostScript® files that can be printed on PostScript printers:

- Keymap_Pc101_User_Defs_Ps
- Keymap_Pc86_User_Defs_Ps
- Keymap_Pc91_User_Defs_Ps
- Keymap_Pcbasic_User_Defs_Ps

The source for these files, written for the Rational Document Formatter (Compose), is also provided online in the following files:

- Keymap_Pc101_User_Defs
- Keymap_Pc86_User_Defs
- Keymap_Pc91_User_Defs
- Keymap_Pcbasic_User_Defs
- Keymap_Template_User_Defs

The source includes user-customizable templates with instructions. Note that the PCBASIC key bindings and overlay are applicable to all three keyboard types.

Rational Environment Keymap for PC101

	Help	Debug	Debug	Debug		Traverse	CMVC	Promote	Demote		Create	Analyze	Items	InfoJobs
CMS					CMS					CMS				
MS					MS	Enclosing (IP)				MS			Underlines Off	
CM		Task Display			CM	Home Library	Show Info	Code (AW)	Unicode (TW)	CM	Create Text	Show Unused	Show Errors	Job Connect
CS	Help Window	Step Returned	Show Breaks	Debugger Window	CS	Other Part (IP)	Accept Changes	Code (TW)	Source (TW)	CS	Create World	Show Usage	Previous Underline	Show Access List
M	Help On Help	Stop	Activate	Catch Exception	M	Enclosing	CMVC Notes	Install (TW)	Withdraw Unit	M	Create Directory	Show Unused (U)	Item Off	What Load
C	Help On Key	Step Local	Remove Break	Propagate Exceptn	C	Other Part	Check In	Code Unit	Source Unit	C	Create Ada	Show Usage (U)	Previous Item	What Locks
S	Help On Cmd	Go	Break	Stack	S	Definition (IP)	CMVC Edit	Install Stub	Demote	S	Create Body	Explain	Next Underline	What Object
	Prompt For	Step	Show Source	Put		Definition	Check Out	Install Unit	Edit		Create Cmd	Semanticize	Next Item	What Users

Auxiliary and Item Keys		
Object	Region	Window
Promote	Complete	Format

Numeric Keypad			
Num Lock	/ Line	* Word	- Mark
7 Begin Of	8 Cursor Up	9 Image Up	+ Image
4 Cursor Left	5	6 Cursor Right	
1 End Of	2 Cursor Down	3 Image Down	Enter
0		. EOI	

Rational Environment Keymap for PC91														
	Help	Debug	Debug	Debug		Traverse	CMVC	Promote	Demote		Create	Analyze	Items	Info/Jobs
CMS					CMS					CMS				
MS					MS	Enclosing (IP)				MS			Underlines Off	
CM		Task Display			CM	Home Library	Show Info	Code (AW)	Unicode (TW)	CM	Create Text	Show Unused	Show Errors	Job Connect
CS	Help Window	Step Returned	Show Breaks	Debugger Window	CS	Other Part (IP)	Accept Changes	Code (TW)	Source (TW)	CS	Create World	Show Usage	Previous Underline	Show Access List
M	Help On Help	Stop	Activate	Catch Exception	M	Enclosing	CMVC Notes	Install (TW)	Withdraw Unit	M	Create Directory	Show Unused (U)	Item Off	What Load
C	Help On Key	Step Local	Remove Break	Propagate Exceptn	C	Other Part	Check In	Code Unit	Source Unit	C	Create Ada	Show Usage (U)	Previous Item	What Locks
S	Help On Cmd	Go	Break	Stack	S	Defintion (IP)	CMVC Edit	Install Stub	Demote	S	Create Body	Explain	Next Underline	What Object
	Prompt For	Step	Show Source	Put		Defintion	Check Out	Install Unit	Edit		Create Cmd	Semanticize	Next Item	What Users

Numeric Keypad			
Num Lock On	/ Line Object	* Word Region	- Mark Window
7 Begin Of	8 Cursor Up	9 Image Up	+ Format Commit (Meta)
4 Cursor Left	5 Image	6 Cursor Right	
1 End Of	2 Cursor Down	3 Image Down	Enter
0 Promote		EOI/Complete	

Rational Environment Keymap for PC86

	Help	Debug	Debug	Debug		Traverse	CMVC	Promote	Demote		Create	Analyze	Complete	Format
CMS					CMS					CMS				
MS					MS	Enclosing (IP)				MS				
CM		Task Display			CM	Home Library	Show Info	Code (AW)	Unicode (TW)	CM	Create Text	Show Unused		
CS	Help Window	Step Returned	Show Breaks	Debugger Window	CS	Other Part (IP)	Accept Changes	Code (TW)	Source (TW)	CS	Create World	Show Usage		
M	Help On Help	Stop	Activate	Catch Exception	M	Enclosing	CMVC Notes	Install (TW)	Withdraw Unit	M	Create Directory	Show Unused (U)		
C	Help On Key	Step Local	Remove Break	Propagate Exceptin	C	Other Part	Check In	Code Unit	Source Unit	C	Create Ada	Show Usage (U)		
S	Help On Cmd	Go	Break	Stack	S	Definition (IP)	CMVC Edit	Install Stub	Demote	S	Create Body	Explain		
	Prompt For	Step	Show Source	Put		Definition	Check Out	Install Unit	Edit		Create Cmd	Semanticize	Complete	Format

Rational Environment Keymap for PCbasic

	F1	F2	F3	F4		F5	F6	F7	F8		F9	F10	F11	F12
	Definition	Enclosing	Edit	Create Command		Promote	Complete	Format	Semantics		Object	Window	Region	Line

Training

Installation Procedure

Release1_1_0

Copyright © 1991 by Rational

PN 507-003265-002

This document is subject to change without notice.

Rational and R1000 are registered trademarks and Rational Environment is a trademark of Rational.

Rational, 3320 Scott Boulevard, Santa Clara, California 95054-3197

Table Of Contents

1 Installation	1
Overview	1
VERIFYING PREREQUISITES	1
INSTALLING THE RELEASE	2
Appendix A Procedure Do_Step	5
Appendix B Feedback	7

1

Installation

Overview

When installing the Training release, you can view the process as a series of phases. These phases must be performed in a serial manner.

- Verifying prerequisites to the installation
- Installing the release

These phases must be performed in a serial manner, as must the steps in each of the phases. This installation note is organized by phase, with each phase comprising a section of this document. Once you have performed this installation, please fill out and return the feedback form that is the last page of these instructions along with a printout of the `Do_Step_Execution_Time` found in the `Logs` library of the release archive. Your feedback is very important in helping us to improve our installation instructions.

VERIFYING PREREQUISITES

In this phase of the installation, you will verify that the prerequisites for installing this upgrade have been met. Do not proceed with the upgrade until all prerequisites have been fulfilled.

- 1. You have read the release note.
- 2. You have read this entire install note.
- 3. The machine is executing Environment release D_12_1_1 or later.
- 4. The machine is running the necessary product(s) to support the functioning of the desired training course.
- 5. If desired, request the `Training_Authorization` code for use in inhibiting product token checking, allowing the maximum number of students to be trained without restriction to the number of tokens purchased by the customer. This must be done in advance of the training date.

Note that this installation allows selective restoration of Rational training courses.

INSTALLING THE RELEASE

In this phase of the release, you will restore the tools tape which contains a release world located in `!Machine.Release.Archive.Training.Release1_1_0`. This release world contains nested archives which will then be restored to complete the installation.

- 1. Load the Training Release1_1_0 tools tape onto the tape drive.
- 2. Log in as user `Rational`, which must be a member of group `Privileged`.
- 3. **LOAD_TAPE** *[5 minutes]*
This step restores the Training Release1_1_0 tape. This step can be executed from any command window. There should be no errors.
- 4. Go to `!Machine.Release.Archive.Training.Release1_1_0`.

The following steps utilize the procedure `Do_Step` as described previously.

- 5. **RESTORE_NOTES** *[1 Minute]*
This step restores the release note `Training_Release1_0_0_LPT` to `!Machine.Release.Release_Notes`.
- 6. **RECORD_INSTALLATION** This step records release information in `!Machine.Release.Current.Products`. This is a required step and establishes that the release has successfully been installed.
- 7. Only execute the step name(s) corresponding to the course(s) you wish to install on the machine (see release note). It is assumed that the product(s) required to support installation of a course are resident and operational on the machine.
Valid names are:

```
ENV_FUNDAMENTALS
ENV_ADVANCED_TOPICS
ENV_TOOLSMITHING
ENVIRONMENT
NETWORKING
PROJECT_DEVELOPMENT_METHODS
CMVC
SYSTEM_MANAGEMENT
MC68020_BARE
MC68020_OS2000
RCF_FUNDAMENTALS
RDF_2167A_FUNDAMENTALS
RDF_2167A_TOOLSMITHING
RDF_2167_FUNDAMENTALS
```

RDF_2167_TOOLSMITHING

Once a course has been restored, follow the instructions in the release note for creating the appropriate number of training user accounts.

SMSE Checklist MC68020_OS2000 Release6_1_3

SESSION AUTHORIZATION

<input checked="" type="checkbox"/>	<u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input checked="" type="checkbox"/>	509-003218-002	MC68020_OS2000 Release6_1_3 Authorization Codes
<input checked="" type="checkbox"/>	505-003249-001	Support Activity Report (SAR)

PRODUCT REQUIREMENTS

<input checked="" type="checkbox"/>	<u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input checked="" type="checkbox"/>	507-003218-003	Installation Procedure for MC68020_OS2000 Release6_1_3
<input checked="" type="checkbox"/>	508-003218-003	MC68020_OS2000 Release6_1_3 Release Note
<input type="checkbox"/>	504-003205-001	ENV Update Special CDF 11_0_0D Checklist <input type="checkbox"/> Not Required with Environment D_12_1_1 or Later
<input type="checkbox"/>	504-003206-005	DFS Update Special Microcode 6_219/6_419 Checklist <input type="checkbox"/> Not Required with Environment D_10_20_2 or Later
<input checked="" type="checkbox"/>		<u>INSTALLATION ACTIVITY</u>
<input type="checkbox"/>		Full Installation in Field Required.
<input checked="" type="checkbox"/>		Partial installation at Factory performed, Field installation of unchecked steps required, starting with step <u>12</u> of Installation Procedure for MC68020_OS2000 Release6_1_3.
<input type="checkbox"/>		Complete installation performed at Factory.

Page 6

- OVER -

□ FIELD INSTALLATION REQUIREMENTS

	<u>PART NUMBER</u>	<u>DESCRIPTION</u>
<input type="checkbox"/>		<u>TAPE MEDIA</u>
<input type="checkbox"/>	541-002872-001	9 Track
<input type="checkbox"/>	541-002873-001	8mm
<input type="checkbox"/>	690-003218-009	MC68020_OS2000 Release6_1_3 Tape

Installation Procedure for MC68020_OS2000 Release6_1_3

When installing the MC68020_OS2000 release, you can view the upgrade process as a series of phases. These phases must be performed in a serial manner.

- 1. Assessing the impact of the release on the user community
- 2. Verifying prerequisites to the installation
- 3. Preparing the user community and the machine
- 4. Loading the release
- 5. Testing the new release
- 6. Restoring customer state
- 7. Cleaning up

These phases must be performed in a serial manner, as must the steps in each of the phases. This installation note is organized by phase, with each phase comprising a subsection of this document. Once you have performed this installation, please fill out and return the feedback form that is the last page of these instructions along with a printout of the `Do_Step_Execution` found in the `Logs` library of the release archive. Your feedback is very important in helping us to improve our installation instructions.

The majority of the steps which must be performed for this installation utilize the procedure `Do_Step` (included as part of the release tape and located in `!Commands`) to execute the required sequence of commands which implement the step. This procedure **must always** be executed from a command window in the release library

```
!Machine.Release.Archive.MC68020_OS2000.Release6_1_3
```

Each step which utilizes procedure `Do_Step` will be of the form

```
<STEP_NAME>                                [<TIME TO EXECUTE>]  
                                             <DESCRIPTION>
```

where `<STEP_NAME>` is passed as the parameter to procedure `Do_Step` which performs the necessary actions to complete the step, `<TIME TO EXECUTE>` is the amount of time the step takes to execute (this may be expressed as a range, in which case the lower values is typically the minimum, and the upper value is an estimate of the maximum), and `<DESCRIPTION>` is a description of what the step does, including any action which you will need to manually perform. For example:

```
FOO                                         [5 Minutes]  
                                             This step is an example of the format used for
```

steps which are executed using procedure `Do_Step`. To execute step `FOO`, you would go to `!Machine.Release.Archive.MC68020_OS2000.Release6_1_3` and in a command window, execute:

```
Do_Step ("FOO");
```

which would result in automatic execution of all commands required to implement this step. If you needed to take any manual action in addition to executing `Do_Step`, it would be noted as :

Perform some manual check

If any errors occur for this step, you should always fix the problem before proceeding on to the next step. Each step is defined by a fragment of Ada code which is executed by `Program.Run`. These fragments are stored in the `Steps` file located in the `Command_Data` library of the release. In the event you want to modify a step, you can use the "PROMPT => <STEP>" form when invoking `Do_Step`. See the spec of procedure `Do_Step` (located in `!Commands`) for more detailed information.

Warning: *If you interrupt the execution of `Do_Step` (by using `Job.Interrupt` such as `CONTROL-G`) it is possible that certain interactive commands executed by some steps, such as `Common.Definition`, may fail with an exception and display a message such as*

```
Unable to read file due to Constraint_Error (Null Access)
```

In general, this message has no negative impact on the execution or completion of the step, and can be ignored.

1. INSTALLING THE RELEASE

The following subsections take you through the installation process a phase at a time. Be sure to complete ALL of the phases of installation and to perform these phases in the order in which they appear here.

Warning: *Follow these installation steps in a serial manner. Do not attempt any other work or even to look at objects on the system (such as `Environment specs`) while performing this install. All users must be logged off during the install. Failure to adhere to these instructions may result in the failure of portions of the installation.*

1.1. Assessing the Impact of the Release

In this phase of the install, you will assess the impact of the release on the user community and the machine.

- 1. This installation will take a minimum of 4 hours on a system which has never had the MC68020_OS2000 product installed. For systems which have a previous version of the MC68020_OS2000 product installed along with compiled customer units, this time will increase, dependent upon the amount of time required to demote and recompile all customer units.
- 2. When installing a new Environment release or a new layered product release, private activity files may become obsolesced (i.e., the subsystems they reference may not be compatible with the software being upgraded to). When installing several products at once, the Environment release should be installed first. Subsequently, each layered product can be installed. Following this process will result in the system default activity (`!Machine.Release.Current.Activity`) referencing a compatible set of subsystems.

After the upgrades have been completed, each user should merge the contents of the default activity into his or her own private activity file with the command `Activity.Merge`.

- 3. During this installation no users should be logged onto the system as there is potential they may cause portions of the installation to fail.
- 4. All `mc68020_os2000` target libraries will be demoted, requiring recompilation.

1.2. Verifying Prerequisites to the Installation

In this phase of the installation, you will verify that the prerequisites for installing this upgrade have been met.

- 1. You have read the release note.
- 2. The system Environment must be running :

```
D_10_20_0 (or later)
Microcode 6_219/419 (or later OR Environment 10_20_2 or later)
Cross_Development 11.0.0D (or later)
```

This can be determined by using the `EEDB` interface to display which configuration is executing and which subsystems are in that configuration. See the appropriate installation note for installing these releases.

Note: The `Cross_Development` release can be loaded as part of this upgrade during the reboot steps after the `MC68020_OS2000` release has been installed (see section on `Testing the Release`) and is required for remote target debugging. The `Microcode` release fixes a bug which occasionally resulted in a machine crash due to the `CDF`, and implements new instructions used during remote debugging. This should be installed prior to beginning this installation.

1.3. Preparing the User Community and the Machine

In this phase of the installation you will prepare both the user community and the machine for the installation.

- 1. All `MC68020_OS2000` libraries will be demoted. Request that the users demote to the Archived state and/or delete all unnecessary `MC68020_OS2000` Ada units.

1.4. Loading the Release

In this phase, you will load the tools required to install the user visible part of the release onto the machine. The release world located in

```
!Machine.Release.Archive.MC68020_OS2000.Release6_1_3
```

contains nested archives. This phase of the install will take a minimum of 3 hours on a machine which has not had the `MC68020_OS2000` installed before. For machines which have `MC68020_OS2000` code compiled, this time will increase dependent upon the amount of recompilation required.

- 1. Login as user `Rational`, which must be a member of group `Privileged`.
- 2. Mount and load the `MC68020_OS2000 Release6_1_3` tape onto the tape drive.
- 3. *[10 Minutes]*
Create a command window and restore the tape onto the system by executing:

```
Operator.Enable_Privileges;
if Operator.Privileged_Mode then
    Archive.Restore (Options => "Replace, Promote");
else
    Io.Put_line ("User does not have privileges.");
end if;
```

Answer the appropriate tape mount requests on the system console.

This will restore `!Machine.Release.Archive.MC68020_OS2000.Release6_1_3`.

- 4. Go to the `!Machine.Release.Archive.MC68020_OS2000.Release6_1_3` library. The following steps utilize the procedure `Do_Step` described earlier in this document.
- 5. **CONFIGURATION_CHECK** *[1 Minute]*

This step will display the currently executing configuration, and if applicable, the verbose information about the `Cross_Development` subsystem that is executing.

Verify that Environment `D_10_20_0` (or later) is executing.

Verify that Microcode `6_219` (M200/300) or `6_419` (M100) (or later) is running. Note that if Environment `D_10_20_2` is running, this check

should be ignored.

Verify that `Cross_Development.11.0.0D` (or later) is running.

6. AUTHORIZATION_CHECK

[1 Minute]

This step will verify that the appropriate products are authorized for this machine. These products are:

```
CMVC
CMVC.Source_Control
Motorola_68k
MC68020_OS2000
```

If there are products which need to be authorized, this will be indicated and a command window created with a call to procedure `Product_Authorization.Register`. Modify the appropriate parameters and authorize the products required.

7. USERS_CHECK

[1 Minute]

This step sets the login limit to 1 and verifies that there are no other users logged into the system. Users working in MC68020_OS2000 worlds or any of the CDF related subsystems may cause portions of this install to fail. Note that the login limit will be restored to its initial value when the machine is rebooted during the Release Testing phase.

8. RELEASE_RESTORE

[7 - 40 Minutes]

This step restores the `Release` archive. When completed, a filtered error log is displayed (`Restore_Release_Log_Summary` located in `!Machine.Release.Archive.MC68020_OS2000.Release6_1_3.Logs`). Examine this log for errors. Ignore the following types of errors (if present):

```
!!! can't restore link...
```

```
!!! Creating CMVC database ... no
    history is currently being
    saved.
```

```
+++ Can't resolve default switch
    file ...
```

9. INSTALL_PRODUCT

[1 ~ 8 Hours]

This step executes the `INSTALL_PRODUCT` procedure located in `!Targets.Implementation.Release_6_1_3`.

When completed, a filtered error log is displayed (`Install_Log_Summary` located in `!Targets.Implementation.Release_6_1_3`). Examine this log for errors. Ignore the following types of errors if they are present (Note: Errors which have a prefix of `***` will be followed by a `++` prefix which applies to the `***` prefixed message. Only the `++` prefix messages are shown below (`***` messages are not shown); ignore the associated `***` messages):

```

*** Can't resolve
    "_archive_freeze_file" ...

*** Had trouble finding units to
    demote...

*** Had trouble finding objects to
    delete...

!!! !TARGETS.IMPLEMENTATION.
    MC68020_OS2000_RUNTIMES.
    ..KERNEL.MESSAGE_QUEUES' SPEC
    should have a body.

!!! !TARGETS.IMPLEMENTATION.
    MC68020_OS2000_RUNTIMES.
    ..KERNEL.TIMER' SPEC
    should have a body.

!!! Creating CMVC database ...
    no history is currently
    being saved.

```

10. RESTART_COMPILER

[2 Minutes]

This step restarts the compiler (1 stream). Note that you can have from 1 to 3 streams, allowing from 1 to 3 CDF compilations to occur simultaneously.

11. RESTORE_CROSS_COMPILERS *[2 Minutes]*

This step saves the current body of `Initialize_Cross_Compilers` to `Initialize_Cross_Compilers_Save` and then restores the most recent body for this procedure.

12. *[2 Minute]*

All D85 target boxes which will use this release must have the new target components installed. Go to `Motorola_68k.Mc68020_Os2000` library located in `!Targets.Implementation.Release_6_1_3` and execute the `Install_Product_On_Target` procedure:

```
Install_Product_On_Target (Target_Machine =>
```

```

        ">>Target Machine Name<<",
Runtime_Module_Directory =>
        "/h0/cmds/bootobjs",
Runtime_Errors_Directory =>
        "/h0/sys",
Debugger_Module_Directory =>
        "/h0/cmds");

```

This will download to the following files on the D85 target:

```

/H0/CMD5/BOOTOBS/ADA_RUNTIME.
/H0/CMD5/BOOTOBS/ADA_TASK.
/H0/SYS/ART_ERRMSG.
/H0/CMD5/RATCOM.
/H0/CMD5/RATDBG.
/H0/CMD5/RATSEND.
/H0/CMD5/RATSHUT.
/H0/CMD5/DBGCOM.

```

Note that the startup command file on the D85 system may need to be modified to get the runtime modules loaded and initialized upon boot. An entry of the following form should be in the **Startup** file of the D85 system:

```

load /h0/cmds/bootobjs/ada_runtime; ada_runtime
load /h0/cmds/bootobjs/ada_task

```

Reboot the target box to effect loading of the new runtime modules.

13. RESTORE_NOTES

[1 Minute]

This step restores the online release note **_LPT** and **postscript** **_PS** file for **MC68020_OS2000_Release6_1_3** to **!Machine.Release.Release_Notes**.¹

14. RECORD_INSTALLATION

[1 Minute]

This step records release information for the MC68020_OS2000 product. This is a required step.

1.5. Testing The New Release

The following will verify operation of the compiler and target execution (if a D85 is available).

1. [10 Minutes]

Load on the **CROSS_DEVELOPMENT.11.0.0D** subsystem if applicable.

¹Inside the **Release_Notes** archive library, there is a **Source_Archive** library which contains an archive of the source for the release note. This is provided per request of the customer.

2. [10 Minutes]

Reboot the machine:

```
Schedule_Shutdown
  (At_Time => ">>A few minutes from now<<",
  Reason => "Release",
  Explanation =>
    "Installation of MC68020_OS2000 Release6_1_3");
```

Note: If the new microcode is to be loaded on as part of this release, make sure that the system will not auto-reboot, allowing you to get to the CLI to load the microcode tape.

 3. [5 Minutes]

Load the **Microcode 6_219/419** if applicable and boot the machine (note that this should have been done prior to installing the release).

 4. When the machine has completed booting, log back in and verify:

a. Execution of the **Motorola_68k** compiler. Execute **What.Users** and examine the output for the 3 compiler jobs. For example (output from **What.Users**):

User	Job	S	Time	Job Name
*SYSTEM	4	RUN	5/20:04	System
	5	RUN	5/20:04	Daemons
	.			
	.			
	227	IDLE	2/00:15	Cross Compiler Common Layer ...
	233	IDLE	1/23:53	Motorola_68k Compiler Control ...
	239	IDLE	1/23:52	Motorola_68k Stream_1 ...

Go back to **!Machine.Release.Archive.MC68020_OS2000.Release6_1_3** to execute the **Do_Step** procedure.

 5. TEST_RESTORE

[3 Minutes]

This step restores the test software to the test library, compiles the units, and converts the executable to OS2000 format. Then the summary error log of these operations is displayed. Examine this log for errors; there should be none.

 6. EDIT_SWITCHES

[3 Minutes]

This step displays the switch file associated with the test world for your editing.

Set the **FTP.Remote_Machine** switch to the name of the target machine.

Set the `FTP.Remote_Directory` switch to the name of the target test library.

Commit the changes made to the switch file.

7. Go to `!Tests.Staging.Install_Verifications.Mc68020_OS2000`.

8. [1 Minute]

Download the main program `Hello`:

```
Os2000_Put (From_Local_File => "Hello.<OS2000>",
           To_Remote_File => "hello",
           Remote_Machine => "<DEFAULT>",
           Remote_Directory => "<DEFAULT>",
           Transliterate => False,
           Response => "<PROFILE>");
```

9. Go to the target and execute the program.

10. [1 Minute]

Exercise the debugger (this requires that you start the `RATCOM` portion of the debugger communications executing on the target prior to executing this command; for example execute `ratcom >/nil; sleep 0` on a target terminal):

```
Debug.Invoke (Main_Unit => "hello",
             Options => "", Spawn_Job => True);
```

11. Set a breakpoint:

```
Debug.Break (Location => ".hello.4s",
            Stack_Frame => 0,
            Count => 1,
            In_Task => "",
            Default_Lifetime => False);
```

12. Execute to the breakpoint:

```
Debug.Execute;
```

13. Display the value of `N`:

```
Debug.Put ("%ROOT_TASK._1.N");
```

This should display the value of 5.

14. Execute until termination:

```
Debug.Execute;
```

1.6. Restoring Customer State

In this phase of the install you will restore to the original state customer units which were demoted by the installation. The time required for this phase will vary widely, depending on how much state is required to be restored.

Note: Before proceeding with repromotion of customer units it is recommended that you perform the Install Verification described in a previous section of this document.

- 1. To repromote customer units which were demoted by this installation, use the files `Install_File`, `Code_File`, and `Freeze_File`, located in `!Targets.Implementation.Release_6_1_3` as indirect files to repromote and refreeze units affected by the upgrade. The units listed in the files are in the correct compilation order. It is advised that these files are split up into smaller files, each containing 300 lines or less to allow more control over the recompilation and its affect on disk garbage generation/collection.

Use the following as a template to restore units to their original state:

```
Operator.Enable_Privileges;
if Operator.Privileged_Mode then
  Io.Set_Output ("Install_Code_Freeze_Log");
  Compilation.Promote (Unit => "_Install_File",
    Limit => "<ALL_WORLDS>",
    Response => "<PROGRESS>");
  Compilation.Make (Unit => "_Code_File",
    Limit => "<ALL_WORLDS>",
    Response => "<PROGRESS>");
  Library.Freeze (Existing => "_Freeze_File",
    Recursive => False,
    Response => "<PROGRESS>");
else
  Io.Put_line ("User does not have privileges.");
end if;
```

1.7. Cleaning up

In this phase, you will perform some cleanup activities. The time required for this phase will vary widely, depending on how much cleanup is required to be done.

- 1. *[1 Minute]*
Print out the file `Do_Step_Execution_Time` located in `!Machine.Release.Archives.MC68020_OS2000.Release6_1_3.Logs` and return to Rational along with the feedback form from this document.
- 2. **DESTROY_ARCHIVE** *[1 Minute]*
This step should only be executed after it has been established that the release has successfully been installed. It will destroy the release archive in `!Machine.Release.Archive.MC68020_OS2000.Release6_1_3`.
- 3. After verification that this release is working correctly, older views from the release subsystems (see release note for list of released subsystems) may be deleted using:

```
Operator.Enable_Privileges;
if Operator.Privileged_Mode then
  Cmvc.Destroy_View (What_View => ">>View Name<<",
```



```
Demote_Clients => False,  
Destroy_Configuration_Also => False);  
else  
  Io.Put_line ("User does not have privileges.");  
end if;
```

- 4. Inform the user community that this release has been installed, and of the location for the online release note.

Stop Time: _____
Total Time: _____

2. CUSTOMER CHECKLIST

- 1. Each user should merge the contents of the default activity into his or her own private activity file with the command **Activity.Merge**.

```
Activity.Merge (Source => "!Machine.Release.Current.Activity",
               Mode => Activity.Differential,
               Target => ">>User Activity<<");
```

Note that this may have the side effect of modifying spec/load view settings which were explicitly set differently by the user.

- 2. Distribute release note to user community.
- 3. Update Daily Message indicating that MC68020_OS2000 Release6_1_3 has been installed.
- 4. Update all D85 target boxes with new target components using procedure **Install_Product_On_Target** located in **!Targets.Implementation.Release_6_1_3**.

3. FEEDBACK

In order to help us improve our instructions and make installations easier to do, the following form is provided to allow you to give us feedback. Please complete and send along with a printout of the file `Do_Step_Execution_Time` located in `!Machine.Release.Archive.MC68020_OS2000.Release6_1_3.Logs` to:

Rational
Atten: SMSE
3320 Scott Blvd.
Santa Clara, CA 95054-3197
Re: MC68020_OS2000

- 1. How long did the installation take you?
- 2. Did the installation proceed as described in the instructions?
- 3. What could be done to improve these (or other) instructions/installations?
- 4. What did you like about this set of instructions/installation?

Motorola 68020/OS-2000 Cross Development Facility Release Note

Release 6.1.3

November 14, 1990

Rational Proprietary Information

CONFIDENTIAL

5. (9123690-0129-9) For-loop does not terminate properly when iteration rule is "reverse Boolean".
6. (Reported by mail 10/23/90) Simultaneous output from different tasks to the same file causes problems.

4. Other problems fixed by this release

1. It is now possible to set breakpoints in library unit instantiations.
2. Single stepping in the vicinity of accept statements now works much better (the source position no longer appears to bounce around).
3. Direct_Io.Reset no longer destroys the contents of the file being reset.

5. Other features

1. Session token checking has been enabled; this has effect only on Series 300 machines.

6. SPRs not closed by this release

Workarounds have been provided for each of these problems.

1. (2984167-Voya-Mv) Exception in Decl_Gen.Gen_Decl.
2. (2268739-Voya-Mv) Internal error during phase FIN caused by complicated construct in unreachable code.
3. (Reported by mail 10/23/90) Compiler problem with nested function calls (actually, caused by complicated dead code). Same as 2268739-Voya-Mv.
4. (4814499-Etoi-Kjm) Problem with packed array as field of packed record. .