# Manage Data Storage

**L61522**
**LCN**

TotalPlant

# NOTICES AND TRADEMARKS



This module supports **TotalPlant** Solution (TPS) system network.

TPS is the evolution of TDC 3000$^{X}$.

Honeywell Inc.
Industrial Automation and Control
Automation College
2820 West Kelton Lane
Phoenix, AZ 85053-3028
**1-800 852-3211**

# TABLE OF CONTENTS

# FIGURES



# TABLES

# MODULE INTRODUCTION

In this module you will learn some key concepts associated with checkpointing and how to best back up your system.

The following topics are discussed:

- Automatic and demand checkpointing
- Adding new checkpoint volumes to the history module
- Backing up point data to EB files
- System backup guidelines

### Module Objective

After completing his module you will be able to adjust autocheckpointing intervals, interpret and recover from the checkpoint errors, and back up your points to EB files.

# WHAT IS A CHECKPOINT?

Briefly, a checkpoint is a database save. You can perform a “checkpoint” (database save) at any time.

Checkpoints are a form of data storage. The various files that contain this saved data will be discussed later in this module. For now, note that the checkpoint files are a form of data storage that give you the means to reload or "restore'" data to a device that has been reset and restarted or repaired. Other forms of data storage are available to you as well (including exception build files and intermediate data files); they are discussed in Appendix A. Review Appendix A at a future time if you feel a need to do so.

# HOW ARE CHECKPOINTS REQUESTED?

Checkpointing can be performed in one of two ways on the **TotalPlant** Solution (TPS) system:

- by manually requesting or "demanding" a checkpoint from a process network display, or
- by enabling periodic, automatic checkpoints, performed for you by the History Module.

In order for a checkpoint to be completed, the process-connected device you are checkpointing must be in its proper processing state. These states are summarized later in this module. After ensuring that the process-connected device is in the proper state, the checkpoint (database save) can be requested.

You should note that whenever a checkpoint is requested, the first device that is checkpointed is always the gateway to that device [that is, Hiway Gateway (HG), Programmable Logic Controller Gateway (PLCG), or Network Interface Module (NIM)], even if the gateway was not specifically requested by the user. The Gateway is checkpointed because part of the point definition resides in the gateway as well as in the process connected device; this ensures that you always have a complete copy of your database.

# HM CHECKPOINT VOLUMES

When an HM is configured for checkpointing, there must be a volume created for each applicable physical node type. If automatic checkpointing has been configured for a History Module under Volume Configuration in the NCF, the volumes and directories needed to support automatic checkpointing are created when the HM is initialized. These volumes and directories are shown in Table 1.

The size of the created volumes and the number of files are specified by the user at configuration time for the AM, CG, and NIM. The Configurator supplies fixed numbers for the HG and PLCG.

The directories that are created support the checkpointing of logical nodes within the physical nodes specified under Volume Configuration.

**Table 1 - Checkpoint Volumes and Directories**

| Volume | Description | Directories | Purpose |
|---|---|---|---|
| &5np | AM checkpoint volume | &Znn*<br>&Fuu | • One &Znn directory is created for each AM being checkpointed by this HM, where nn = LCN node address. This directory is called the Master Reference Directory, but currently contains no files.<br>• One &Fuu directory is created for each unit assigned (through LCN NODES) to the AM being checkpointed, where uu = Unit index number. The actual checkpoint files for each unit database are created here. |
| &6np | CG checkpoint volume | &Znn*<br>&Huu | • One &Znn directory is created for each CG being checkpointed by this HM, where nn = LCN node address. This directory is called the Master Reference Directory, but currently contains no files.<br>• One &Huu directory is created for each unit assigned (through LCN NODES) to the CG being checkpointed, where uu = Unit index number. The actual checkpoint files for each unit database are created here |
| &7np | HG and PLCG checkpoint volume | &Inn | One &Inn directory is created for each HG/PLCG being checkpointed by this HM, where nn = Hiway number (1-20) as assigned to the HG/PLCG under LCN NODES of the NCF. |
| &8np | NIM checkpoint volume | &Inn | One &Inn directory is created for each HG/NIM being checkpointed by this HM, where nn = UCN number (1-20) as assigned to the NIM under LCN NODES of the NCF. |
| np = node pair number<br>* = Required to save to HM. | | | |

# AUTO-CHECKPOINTING

This section discusses how the Honeywell **TotalPlant** Solution (TPS) system performs automatic checkpointing.

## Physical Node Types That Can Be Auto-Checkpointed

At configuration time, a History Module (HM) can be configured to trigger the checkpointing of one or more physical nodes. The physical nodes belong to one of the following physical node types:

- Application Module (AM),
- Computer Gateway (CG),
- Hiway Gateway (HG),
- Programmable Logic Controller Gateway (PLCG), or
- Network Interface Module (NIM).

Not all physical nodes checkpointed by one HM have to be of the same physical node type.

## Enabling/Disabling Auto-Checkpointing

Automatic checkpointing can be enabled/disabled on a physical node basis by changing the automatic checkpoint status of the physical node. The operator can change the enable/disable status from the Node Status display at a Universal Station or Global User Station. Demand checkpoints can still be requested even when automatic checkpointing is disabled.

The auto-checkpoint status information is part of the distributed Node Administrator's database. On a system restart, default values for these states will be set by the Node Administrator.

> **ATTENTION**
>
> It is recommended that automatic checkpointing be disabled during point loading operations. Because point data is loaded to both the HG/NIM and process box databases, it is possible that a checkpoint operation could occur while the two are in an inconsistent state (data is loaded in one of the databases only). This situation, although unlikely, results in inconsistent checkpoint files.

## Identifying Nodes To Be Auto-Checkpointed

Automatic checkpointing can be enabled on a physical node basis, but it is triggered on a logical node basis. For example, one physical AM with three units configured is considered three logical nodes. All three logical nodes would be individually triggered for checkpointing. The AM and CG can have one or more units. The HG and NIM have one process network. (Recall that the PLCG emulates the HG and all references to the HG in this module apply to the PLCG.)

All logical nodes that require automatic checkpointing in a physical node are separately triggered by a History Module subsystem called Checkpoint Control.

On an HM restart, Checkpoint Control receives the following configuration data:

1. The identity of the physical nodes that must be triggered for checkpointing (from volume configuration in the NCF).

2. Per physical node, the logical nodes that require checkpointing (from LCN node configuration in the NCF).

3. The automatic checkpointing status of the physical node. (Checkpoint Control will not trigger checkpointing of nodes that are disabled.)

Based on the above information, Checkpoint Control determines which logical nodes exist in the different physical nodes and which physical nodes must be triggered for checkpointing. This action is performed through logical node tracking by the Node Administrator.

All checkpoint files of a logical node are bound to the corresponding volume.

## Network Synchronization

Each HM configured for automatic checkpointing operates independently of all other HMs on the system.

## Automatic Checkpointing Start Times

The default automatic checkpoint duty cycle is 4 hours. It can be configured for less than 4 hours, but no longer than 12 hours. If the duty cycle is 4 hours, this means that every 4 hours, checkpoint control asks each logical node to checkpoint its database.

If checkpointing completes within 4 hours, checkpoint control delays until the start of the next 4-hour duty cycle.

If checkpointing does not complete within 4 hours, checkpoint control starts the next duty cycle as soon as the previous cycle completes.

The exact starting times of the checkpoint duty cycle are determined by an offset from midnight. The default offset time is 30 minutes; therefore, the wall clock starting times of the checkpoint duty cycle are 12:30 a.m., 4:30 a.m., 8:30 a.m., 12:30 p.m., 4:30 p.m., and 8:30 p.m.

When the HM starts up, checkpoint control looks at the current time and calculates a delay between it and the start of the next duty cycle.

#### Default Values

When an HM is initially loaded, this file is created that contains the default checkpoint interval and offset values:

```
NET>!9np>Cnp_CPNT.MM
```

The default automatic checkpoint values are:

Interval—four hours

Offset—30 minutes

If the checkpoint interval values are changed at the HM, the file is updated accordingly. The file is not accessed by the HM during normal checkpoint operations. It is accessed by the HM only on HM restart, at which time the HM reads the file to determine the configured checkpoint interval values before restarting.

If `Cnp_CPNT.MM` is not present in the local volume at restart, it is recreated with the default values.

# PSDP AUTO-CHECKPOINT PARAMETERS

The following are the History Module Processor Status Data Point (PSDP) parameters you can use to monitor auto-checkpointing (where nn is the History Module's physical node address):

| | |
|---|---|
| $PRSTSnn.TIMEBASE (l) | This HM's automatic-checkpoint interval. The default for this parameter is 4 hours. |
| $PRSTSnn.TIMEBASE (2) | This HM's automatic-checkpoint offset value. The default for this parameter is 30 minutes. |
| $PRSTSnn.TIMEBASE (3) | The time it took to complete the last automatic checkpointing cycle. This is a view-only parameter. |
| $PRSTSnn.TIMEBASE (4) | Start time of the current or next automatic checkpoint. This is a view-only parameter. |
| $PRSTSnn.NODENUM | The number of the node currently being processed. If no automatic checkpointing is in progress, this parameter is zero. |

The *Engineer's Reference Manual* provides more details about the PSDP parameters.

As shown in Figure 1, the parameters are accessible from the CHKPTIME display on PERFMENU in Release 400 and later.

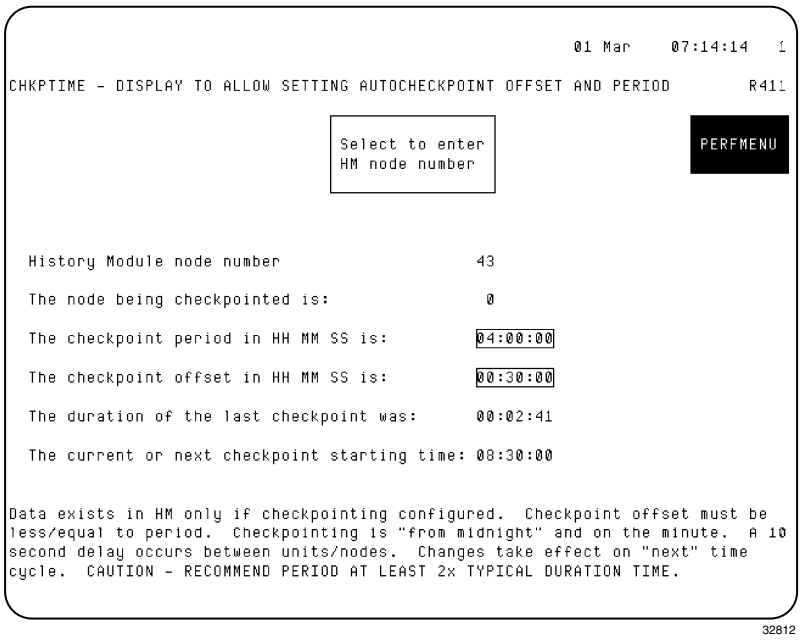



**Figure 1 - CHKPTIME Display**

## NOTE

It is recommended that the minimum automatic checkpoint interval be no less than twice the time it took to complete the last automatic checkpointing cycle.

$$\text{TIMEBASE(1)} \geq 2\ (\text{TIMEBASE(3)})$$

All nodes being checkpointed by a particular HM will be checkpointed on the same time interval.

## Checkpoint Request Processing

When a logical node receives a checkpoint request, it must

- checkpoint all the data that is required for a restart, then
- write its checkpoint files to the volume in the checkpoint request.

Checkpoint request processing must be finished within a predefined time period because Checkpoint Control will timeout on an unanswered checkpoint request. This timeout value is defined as 60 minutes for each physical node type.

If a timeout occurs, Checkpoint Control issues a timeout event message. Checkpoint Control then goes on to the next physical node, if one exists.

If the target node fails while executing a checkpoint, Checkpoint Control issues a failure event message and goes on to the next node.

## Retrieval of Checkpoint Files on Node Restart

On a normal node restart, the automatic checkpoint files are read by the logical node through the file manager. Only one HM may be responsible for the checkpointing of a particular physical node in order to avoid duplicates.

# CHECKPOINT ERROR MESSAGES AND RECOVERY

If a UCN node other than the NIM cannot be successfully checkpointed, checkpointing of the other nodes proceeds; however, if an error occurs during a checkpoint of the NIM database, no other UCN nodes are checkpointed.

## Data Hiway or UCN Status Display - Demand Checkpoint

The error messages listed below may appear on the process network Status display (Data Hiway or UCN) when you attempt to manually save HG/NIM and box checkpoint data, or load an HG/NIM or box.

| Message | Cause/Recovery |
|---|---|
| HG - `FILE ERROR`<br>NIM - `FILE ACC` | Caused by one of the following:<br>• An error occurred as access to the medium was attempted.<br>**Recovery:**<br>Ensure removable media is mounted properly.<br>• The checkpoint files were not the correct revision.<br>**Recovery:**<br>1. Use the backup checkpoint media.<br>2. Check the revision of the files. |
| HG - `NO ROOM`<br>UCN - `NO ROOM or FILE ACC` | A save was attempted, but the medium was full (too few files formatted) or the directory was too small to accept the data.<br>**Recovery:**<br>1. Change media.<br>2. Reformat media. |
| HG - `VOLUME NF`<br>NIM - `FILE ACC` | The required checkpoint directory is not present on the removable media mounted in the drive. |
| HG - `BAD REQUEST`<br>NIM - `NODE-ASN` | A request was made for a box logically assigned to another gateway. |
| HG - `CONFLICT`<br>NIM - `NODE BSY` | The HG or NIM is busy with another save or load operation.<br>**Recovery from CONFLICT**<br>1. Attempt checkpoint again.<br>2. If unsuccessful in second attempt, delete .CP files from media and try checkpoint again. |
| `NOT IDLE`<br>`BAD STATE` | Box was not in proper state to accept command.<br>**Recovery:**<br>Refer to Tables 2 and 3 for proper box states. |

**Table 2 - HIWAY States Required For Save/Load**

| Data Hiway or Box State | SAVE DATA | LOAD DATA |
|---|---|---|
| Hiway control state | Any | Any |
| Box control | Any but TEST | BASIC CTRL (CB,EC,PIU) |
| Box processing status | OK (including partial and soft failures) | OK (CB, EC, PIU)<br>RESET (PIU)<br>IDLE (A-MC, MC) |
| Box assignment | This HG | This HG |

**Table 3 - UCN States Required For Save/Load**

| UCN or Box State | SAVE DATA | LOAD DATA |
|---|---|---|
| UCN control state | Any | Any |
| Box control state | Any | Any |
| Box processing status | OK or IDLE | IDLE |
| Box assignment | This NIM | This NIM |

## System Status Messages – Automatic Checkpointing

If a node fails while processing an automatic checkpoint request, or if a requested checkpoint is not complete within 60 minutes (timed out), system status messages are printed in the System Status Journal and in the Real-time Journal (if the journal is active on a US/GUS with the Operator or Universal Personality).

The System Status messages are in this form:

```
Checkpoint of UNI uu in XX nd to HM nd Failed; Code nn
Checkpoint of UNI uu in XX nd to HM nd Timed Out
Checkpoint of HWY hh in HG nd to HM nd Failed; Code nn
Checkpoint of HWY hh in HG nd to HM nd Timed Out
Checkpoint of HWY hh in NM nd to HM nd Failed; Code nn
Checkpoint of HWY hh in NM nd to HM nd Timed Out
```

where:

- `XX` = AM or CG
- `nd` = Node number
- `uu` = Unit number
- `hh` = Hiway number
- `nn` = Error code number
  - `06` = File access error, such as trying to access a protected file

    AM:
    When the error code is 06, AM Processor Status Data Point parameter CPFMERR ($PRSTSnn.CPFMERR) contains a code that further defines the type of error. Commonly encountered codes in CPFMERR are
    - 03 = Physical error
    - 20 = Device not found
    - 26 = Insufficient storage space
    - 28 = Exceeded maximum number of files per volume
  - `07` = Auxiliary file access error
  - `14` = Target node failed
  - `15` = Target node had an unexpected state change, such as being reset during the checkpoint operation
  - `36` = Duplicate directory error

## System Error Messages—Hiway Auto-Checkpoint

Checkpoint errors may cause messages to be printed in the Real Time Journal and in the System Error Journal.

The messages are in the form:

```
D$ CP CHKPNT   AA BB CC DD
```

where:

- `AA` = Indicates the internal HG module name:
  - 14 = lost process event
  - 15 = lost hiway event
  - 16 = lost peripheral event
  - 178 = file manager error
  - 182 = checkpoint task
- `BB` = Provides a cause-of-error indication:
  - nn = number of node that lost event
  - 190 = Error against HM (bad drive, sector, or data)
  - 195 = Checkpoint volume full or not enough room to do checkpoint.
  - 199 = Checkpointing inhibited.
  - 205 = Box in TEST control state; must be FULL or BASIC.
  - 216 = Gateway cannot communicate with Process Network.
  - 220 = Hiway Error
  - 221 = Data Inconsistency (checkpoint file may be protected or corrupted)
- `CC` = 5001 through 5063 (the last two digits are the number of the box)
  - 60nn (nn is the HG file number)
  - 0 (the system crashed)
- `DD` = Event serial number

## System Error Messages—UCN Auto-Checkpoint

If an automatic checkpoint operation is not completely successful, a software error message will be reported to the System Error Journal and the Real Time Journal. The software task that reports these errors is the slow command handler, the NIM processing task responsible for checkpoint and restore operations:

```
UG$_CH_SLOW   28, 94, nn, xx
```

where:

`nn` = node's UCN node number, 1 through 64
`xx` = type of error as listed in Table 4

**Table 4 - UCN Auto-Checkpointing Errors**

| xx Value | Meaning |
|---|---|
| 05 | Checkpoint volume too small to create the checkpoint file. |
| 07 | &Inn checkpoint volume not found on an HM. |
| 41 | UCN communication problem—NIM cannot send checkpoint request to UCN node. |
| 55 | UCN node is in IDLE state and cannot be checkpointed. Database may not be valid. |
| 65 | I/O link resource error. Not enough resources on I/O link to checkpoint IOM |
| 66 | UCN node is in softfail state, so cannot respond to checkpoint request. |
| 68 | Checkpoint file on HM is wrong version or wrong revision. Proper upgrade of checkpoint files is needed. |
| 69 | Checkpoint request rejected by the UCN node. Contact Honeywell TAC |
| 73 | Communication problem on the I/O link. |

# CHECKPOINT FILENAMES

Table 5 lists the checkpoint filenames.

> **ATTENTION**
>
> It may be useful to check the time stamps or revision numbers that are in the catalog data to see if an incompatible time stamp or revision may be causing a problem. For example, box data should not be older than HG data.

**Table 5 - Checkpoint Filenames**

| HG Checkpoint Files | | NIM Checkpoint Files | |
|---|---|---|---|
| HG0hhHGr . CP | HG-resident data | NM0uuNMr . CP | NIM-resident data |
| HG0hhbbr . CP | Box-resident data | NM0uuMAS . CP | NIM master file |
| HG0hhbbL . CP | MC Logic Block data | PM0uummr . CP | PM-resident data |
| HG0hhMAS . CP | Master checkpoint file | PM0uummM . CP | PM master file |
| | | LM0uummr . CP | LM-resident data |
| | | LM0uummM . CP | LM master file |
| | | APMuummr . CP | APM-resident data |
| | | APMuummM . CP | APM master file |
| | | HPMuummr.CP | HPM-resident data |
| | | HPMuummM.CP | HPM Master file |
| | | SM0uummr.CP | Safety Manager-resident data |
| | | SM0uummM.CP | Safety Manager Master file |
| **AM Checkpoint files** | | **CG Checkpoint Files** | |
| AMuuuMAS . CP | Master checkpoint file | CIuuuMAS . CP | Master checkpoint file |
| AMuuuPDr . CP | Point data | CIuuuPDr . CP | Point data |
| AMuuuNTr . CP | Point names | CIuuuNT. CP | Point Names |
| AMuuuCDr . CP | Custom Data Segments | CIuuuCDr . CP | Custom Data Segments |
| AMuuuLAr . CP | CL | | |

where:

- bb = Box number
- hh = Hiway number (1 through 20)
- mm = UCN Node number (1 through 32)
- r = Revision level of this file (r = 1 or 2)
  - r = 1 the first time the file is saved,
  - r = 2 the second time,
  - then it alternates each time the file is saved.
- uu = UCN number (1 through 20)
  - uuu = Unit number (1 through 100)

```
LS NET>&F10>*.*

NET>&F10>*.* 07/18/96 10:18:43

 DEVICE  DIR  FILENAME EXT TYP P VER  TIME STAMP   # RECS RECSIZ #BLKS BLKSIZ ST
-------- ---- -------- --   -  - -- -------------- ------ ------ ------ ---- ---
NET      &F10 AM010NT2 CP   L     1 07/18/96 10:18      1 ------      1    9
NET      &F10 AM010CD2 CP   L    -- 07/18/96 10:18      0 ------      0   28
NET      &F10 AM010NT1 CP   L     1 04/22/96 14:21      1 ------      1    9
NET      &F10 AM010CD1 CP   L     2 04/22/96 14:21      1 ------      1   28
NET      &F10 AM010LA1 CP   L    -- 04/22/96 14:21      0  11520      0   45
NET      &F10 AM010PD1 CP   L    -- 04/22/96 14:21      1  11520      1   45
NET      &F10 AM010MAS CP   L    20 04/22/96 14:21      1 ------      1    4
FILES LISTED:     07


Catalog Complete
```

34473

**Figure 2 - AM Checkpoint File Listing (Checkpoint in Progress)**

```
LS NET>&F10>*.*

NET>&F10>*.* 07/18/96 10:18:43

 DEVICE  DIR  FILENAME EXT TYP P VER  TIME STAMP   # RECS RECSIZ #BLKS BLKSIZ ST
-------- ---- -------- --   -  - -- -------------- ------ ------ ------ ---- ---
NET      &F10 AM010NT2 CP   L     1 04/22/96 14:21      1 ------      1    9
NET      &F10 AM010CD2 CP   L     2 04/22/96 14:21      1 ------      1   28
NET      &F10 AM010LA2 CP   L    -- 04/22/96 14:21      0  11520      0   45
NET      &F10 AM010PD2 CP   L    -- 04/22/96 14:21      1  11520      1   45
NET      &F10 AM010MAS CP   L    20 04/22/96 14:21      1 ------      1    4
FILES LISTED:     07


Catalog Complete
```

34474

**Figure 3 - AM Checkpoint File Listing (Checkpoint Complete)**

When reloading a device, the Master checkpoint file for that device type points to the most current checkpoint file to be used for the reload. For example, if the revision 2 file is the most recent file (it has the most recent timestamp), then the Master file requires this file to be used on a reload. However, the Master file points to the filename, not the timestamp.

By understanding the relationship between the Master checkpoint file and the revision numbers, it may be possible to recover from a corrupted checkpoint by simply renaming the files. For example, if the Master file is pointing to revision 2 as the most current file, but you know that the revision 2 file is corrupted, you can simply rename the revision 1 file to revision 2 in order to "fool" the Master file into using it to reload a node or process control device.

# ADDING NEW CHECKPOINT VOLUMES TO THE HISTORY MODULE

What if an additional checkpoint directory is required on the History Module and the standard checkpoint volume in which that directory should be located is too small? This scenario could occur if you added another unit to the AM or CG or if you added a PLCG, HG, or NIM to your LCN. Your options are

Option 1 Perform an online volume configuration change and initialize the HM. Option 1 can be performed online, and if you are in the initial stages of system startup, this step is preferred.

Option 2 Move your checkpoint directories to a different volume that is large enough. You can then initialize the HM and perform volume configuration at a more convenient time. Option 2 is for users who have developed a larger database and do not wish to reconfigure the volume configuration.

**ATTENTION**

Note that if the checkpoint directories are not located in the required system volumes, as described in Option 2, autocheckpointing will not take place. Demand checkpoints can still be performed, however.

## Moving Checkpoint Files

Figure 4 shows an example of moving checkpoints to a different volume and adding an additional checkpoint directory.

**ATTENTION**

If you decide to move your checkpoint directories to a different volume, be sure to clearly document this step. This will help ensure that the directory is backed up during your next HM reconfiguration session.

**Example 1**

A user has added a Hiway Gateway to the LCN and a unit to the Application Module. The user determines that the present system checkpoint volumes do not have enough space. The user then proceeds to find volumes (any volume) that have sufficient space for the directory.

In this example in Figure 4, the user added the new HG checkpoint directory to volume &601. Since all AM directories must be located together in the same volume, the user decided to move the AM's directories to a user volume called FRED. The example is at best a *temporary* workaround; when HM initialization is required, it should not be delayed for a lengthy period of time.

For systems on R300 and later, it is especially important to not delay HM initialization, because any "offline" NCF change will "roll up" any previously made "online" NCF changes. If these changes affect the volume configuration information in the NCF, it is possible to have a mismatch occur between the information in the network's NCF file and the information in the NVCF file in the HM's local volume. The HM will then fail to autoboot if there is a mismatch; therefore, the above procedure is not a recommended procedure in all cases.

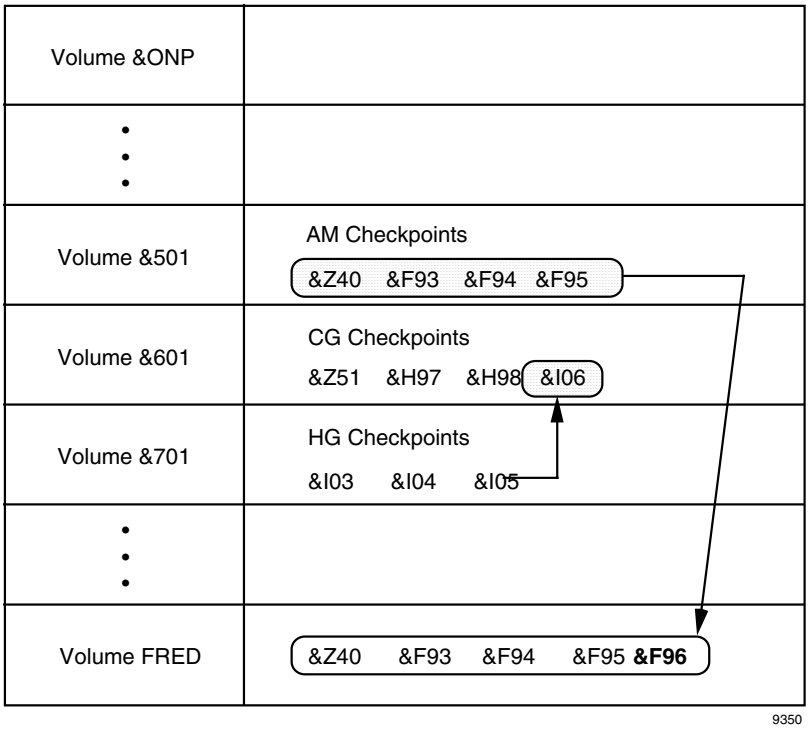



**Figure 4 - HM Directory Example**

## Rollup

When an HM is initialized

- It sets up its volumes and directories based on the HM image in the NCF (volume configuration information).
- It creates the following file from the HM Image information in the NCF. The file contains the HM's volume configuration information:

```
NET>!9np>Lnp_NVCF.MM
```

The HM's Lnp_NVCF.MM file information must exactly match the NCF's HM Image information; otherwise, the HM crashes upon autobooting.

As illustrated in Figure 5, in R300 and later, any offline NCF install

- rewrites the HM Image in the NCF, and
- includes ("rolls up") any previously made online changes when it rewrites the HM image.

Because of the "rollup and rewrite," if there was an online change made previously that affects volume configuration, the NCF's HM Image and the NVCF file will reflect different information until the HM is initialized.

A mismatch between the HM image and the NVCF file causes the HM to fail upon autoboot.

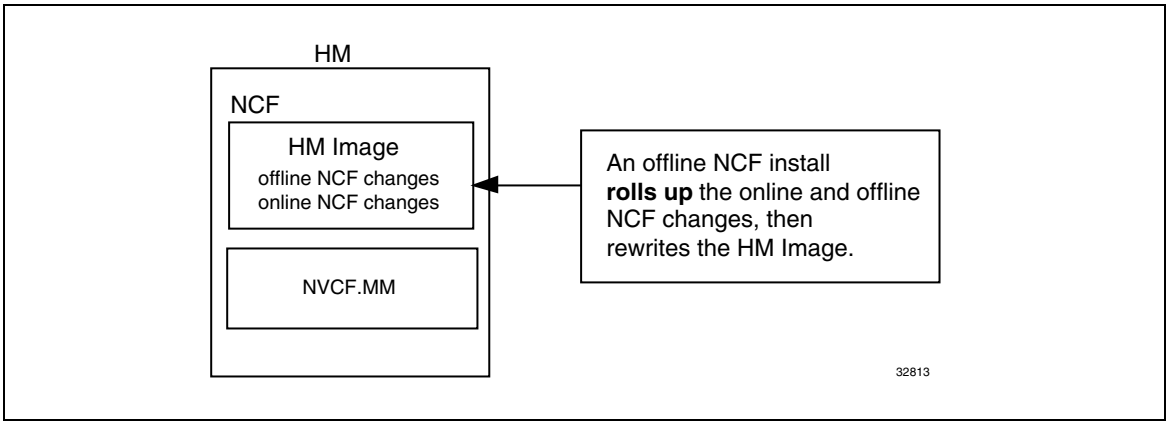



**Figure 5 - Rollup – R300 and Later**

**Example 2**

In example 1, the HM was configured for AM checkpointing during NCF volume configuration (defines the HM image). AM checkpoint configuration is indicated by the presence of volume &501 on the HM.

Because the HM was set up to checkpoint the AM in volume configuration, the addition of a new unit to the AM in LCN node configuration is linked to the HM Image information in the NCF file.

For this example, assume the new AM unit was added *online*; consequently,

- The change is made to the NCF.CF file.
- The HM Image is *not* rewritten because the change is made online (therefore, the NVCF file and the HM Image continue to reflect the same information).

Assume that, sometime later, an NCF change (such as to the Network Gateway under System Wide Values) is made *offline*:

- Because offline changes "rollup" previously made online changes *and* rewrite the HM Image, the HM Image is updated to reflect the online addition of the new AM unit.
- The NVCF file, however, still reflects the HM Image configuration that existed before the addition of the new unit and will continue to do so until the HM is initialized.
- Because the NVCF file and the HM Image reflect different information, any attempts to autoboot the HM prior to initialization will cause it to fail.

# BACKING UP POINT INFORMATION IN EB FILES

## What are EBs?

Exception Build (EB) files are text files that contain point configuration information.

## Why use EBs?

Honeywell recommends using Exception Build (EB) files to back up your point database for several reasons:

- EB files are transportable between software releases (Note: Some editing may be required when going from release-to-release to add new parameters to a point type.)
- EB files are transportable between sites and can be easily modified.
- EB files are ASCII files, which can be viewed from the Text Editor or can be accessed from a PC (after going through a text file conversion).
- EB files are a good final backup to your points. They do not contain internal identifier references.

## What Data To Back Up With EBs

Exception Build files should be created for the following types of data:

- configured control and DAS points
- configured continuous history groups
- area database entities, including group displays, and alarm units.

## How to Create Backup EB Files

One common way to create backup files is to do a Print System Entities to EB files. While this method is convenient, there is a scenario to consider on heavily loaded systems. On a heavily loaded system, the Data Entity Builder may timeout waiting for data and insert default values on your point types. The point types that are most susceptible to this problem are Application Module points with a large number of Custom Data Segments and HPM/APM/PM logic slot points.

To minimize the possibility that this will occur, you may want to consider reconstituting points (backbuilding) to Intermediate Data Files (IDFs). If some of the values are unavailable, a .UL file will be created upon reconstitution. This will alert you to a problem. Once you have successfully reconstituted all your points to IDFs, you can then build your EB files from IDF files.

**Table 6 - Preferred Procedure to Create EB Files from Builder Commands**

| Step | Action | Comment |
|---|---|---|
| **1** | Determine how many points you want to have in an IDF. | The maximum size of an IDF is 64 K. Although the maximum number of entities that can exist in an IDF varies with the size of the entity, as a rule of thumb, use 255 entities as the limit. |
| **2** | Build a selection list for that IDF. | A selection list allows you to have an IDF for each point type. This also helps reduce NIM loading when backbuilding points. |
| **3** | From Builder Commands, execute a Reconstitute Multiple to IDF. | You would have to repeat this step for each point type IDF you had defined. |
| **4** | If an error occurs, locate the -UL file. | The -UL file is the “Unsuccessful Entity List” of points that the system is not able to get.<br>The usual reason for this is the system was too busy to get the data for you. |
| **5** | Use the -UL file to try to get the points the system failed to get during the first attempt. | |
| **6** | After getting all your points, Print Entities from IDFs to an EB file. | In this step, you have retrieved the current point configurations into an EB file. |

# SYSTEM BACKUP

## Steps to Organize System Data

The following steps can help you establish your TPS database backup policies if none are in place.

**ATTENTION**

Your database backup procedure is often determined by company standards and operating policies. For this reason, each site should prepare site-specific backup procedures.

| Step | Action |
|---|---|
| 1 | **Organize data on all the HM(s):**<br>a. Put all files needed to load and run the system into convenient root volumes:<br>• System files, personalities, checkpoints, sequence objects, area databases, display objects, and history.<br>b. Put the files needed for database development into other well organized root volumes:<br>• CDS objects, parameter lists, CL source files, display source files, and point source files.<br>(Organization is half the battle.) |
| 2 | **Organize Backup Media**<br>a. Create a System Boot removable media, containing all of the system files and NCF, plus necessary personality files. (Follow directions in the *Command Processor* manual.)<br>b. Create another removable media that has all display materials on it:<br>• Area databases and display objects.<br>NOTE: These two removable media would be part of the backup scheme, but could also be used to bootload and run the system in the event of HM failure.<br>c. Through trial and error, find out how many additional removable media would be needed to backup the remaining software. Go by root volumes and leave plenty of space on each removable media for future growth. |
| 3 | **Write ECs**<br>Write .ECs to actually do the backup to the removable media, so that operators can do it with little or no assistance. |

## Create Volume with Extended Directories

In R510 and later releases, if more than 63 HM directories require backup, you can use removable media preinitialized with the extended directory option used with the Backup command for a backup of up to 2046 directories. The extended directory option typically allows an entire HM to be backed up on fewer disks reducing, if not eliminating, the need to manually change out removable media to complete the operation.

Procedure:

1. Enter the BACKUP command in the Command Processor.

2. Enter NO to the prompt, "Do you want to FORMAT/INIT this media?"

3. Enter YES to the prompt "Skip FORMAT/INIT function?"

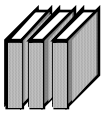
   Result: The backup proceeds normally and uses the extended directory support of the preinitialized removable media.

If you create a .EC file to backup the HM, edit the file by adding the -X option to the Create Volume command. On execution, the .EC file initializes the specified removable media before performing the backup.

**Create Volume Command**

```
CR $F1>VOLM> -MF 3000 -BS 1500 -X
```

Where:
-X = Extended Directory Option



REFERENCE—*Command Processor Operation*, Subsection 6.4, Create Volume.

## Additional Reference information

The *Engineer's Reference Manual*, Section 18 discusses additional software/database backup issues. Review the information on "Avoiding Mistakes and Errors," then return to this section.

## How many Backup Copies?

At least two copies of system backups should be stored by the customer at different locations, and ideally an extra copy kept by Honeywell. Keep a master offsite.

Honeywell recommends that three or four sets of removable media be rotated, so that if it is desired to return to an old configuration, it will be available.

## Database Change Guidelines

After the initial backup, the engineer working on the system should follow some guidelines when making changes to the system:

- Before making changes to the system, the engineer should make a backup before starting, in case something goes wrong. A backup includes a checkpoint if changes to configuration are needed. An EB file is the most useful as a reference during configuration.
- The engineer should make a backup of all daily work (such as IDFs, checkpoints, copies of the area database).
- When the work is complete and verified, the engineer should make a new set of EBs, and then update the backup set of removable media.

## Backing Up Using File Transfer

Another approach to backing up data is to use the file transfer feature either between the History Module and a GUS hard drive or the History Module and a host computer. File transfer between the History Module and a host computer (VAX) is supported by the combined use of a Computer Gateway at R400 or later (or Plant Network Module, PLNM) and a CM50S at Release 4.0. or later. The file transfer capability, along with event-initiated reporting to files, increases the interoperability between the host computer and the LCN system.

The types of files that can be transferred are

- variable length files (such as ASCII files),
- fixed length files (such as checkpoint files),
- contiguous binary files (such as personality image files and journals).

Neither the GUS nor the VAX can decode contiguous binary files into a readable format; however, binary files can be transferred for archival purposes. Refer to the publications for additional detail when performing any file transfer command.

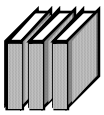
## System Backup Frequency

How often backups should occur depends on how often changes are made to the system. Obviously a full system backup is necessary after the factory acceptance test and after tuning is complete.

It is a good idea to make a full system backup at regular intervals. The interval depends on the number of changes that are occurring to the system. If the engineer can keep the backup set updated as changes are made, the full backup might be necessary once every six months. If more work is being done to the system, or it is hard to continually update the backup media with changes, a full system backup is warranted once a month or so.

## Backing Up By Timestamp

When you are backing up files from your system to removable media, there may be occasions when you want to back up only newly created or modified files.



REFERENCE—The *Command Processor Operation* manual, Section 8.4, shows the options to sort by date.

Section 10.7 shows a sample executable command file that may be used to backup files within certain data range.

Appendix C of this course module contains an excerpt from the manual.

**LAB TIME** <30 Minutes

Use your US and your assigned prebuilt partition.

Take with you:
- This course module

# LAB EXERCISE 1

## Manage Data Storage

1. List the HM checkpoint files and their timestamps for a controller the instructor has assigned to you:

   _____

2. List the Master checkpoint file and its timestamp for your assigned device.

   _____

3. If your assigned device is a PM or a Data Hiway box, rename the most current checkpoint file. Example:

   ```
   RN NET>&I02>HG002211.CP HG002213
   ```

   If your assigned device is an AM, rename the most current checkpoint file for point data of one unit only to the other revision. Example:

   ```
   RN NET>&F24>AM024PD2.CP AM024PD1
   ```

4. Attempt a Restore (UCN) or Load (Hiway or AM) command.

   NOTE: The AM must be shutdown before it can be reloaded.

   You should get a file error. Check the Event History journals for error messages.

5. Recover from the file error by reestablishing (renaming) the file from step #3.

6. Try the Load Data command again. Your device should be restored.

7. Try saving data from the wrong logical network. You should receive a Node Assignment error.

8. Look at the CHKPTIME display on PERFMENU.

### End of Lab Exercise 1

# LAB EXERCISE 2—USE EXCEPTION BUILD

**Introduction**

One common way to create backup files is to do a Print System Entities to EB files. This method works well in most cases; however, on heavily loaded systems, you can avoid timeouts by reconstituting points (backbuilding) to Intermediate Data Files (IDFs), then building your EB files from IDF files. The lab uses that procedure.

**Lab requirements**

This lab requires a process-connected device with points built. To minimize lab time, it is necessary to back up only the regulatory control point types. At your plant, back up all point types.

**Lab prerequisites**

This lab assumes familiarity with Documentation Tool and Data Entity Builder commands. Note: In the lab procedure, replace the characters ### with your assigned student partition number.

**Backup procedure**

| Step | Action |
|---|---|
| **1** | Create a selection list for the Regulatory Control points in your assigned process-connected device. Use Documentation Tool (Doc Tool) or Data Entity Builder (DEB) commands to generate the lists. Both methods are documented. Select one to use.<br>• **To use Doc Tool**:<br>a) From the Engineering Main Menu, select Documentation Tool<br>b) From the top line of the Doc Tool display, select QUERY<br>c) Select “Build”<br>d) Select “UCN” or “HWY”<br>e) Enter data for the following:<br>- **Proc Net List?** (enter your UCN or Hiway number)<br>- **Device List?** (enter the process device number)<br>f) For “Resource/Entity Types:” select REG CTL (for UCN) or REG (for Hiway).<br>g) Press [ENTER]<br>h) After data appears on the screen, select OUTPUT from the top line of the Doc Tool display<br>i) Select “To a File”<br>j) For “Pathname” enter: NET>S###>REGCTL.EL<br>k) Select “Overwrite Without Field Definitions”<br>l) Press [ENTER]<br>m) After “OPERATION COMPLETE” at the bottom of the screen, press the [CTL] and [HELP] keys to return to the Main Menu<br>• **To use DEB commands:**<br>a) From the Engineering Main Menu, select BUILDER COMMANDS<br>b) From the DEB COMMAND DISPLAY, select LIST ENTITIES IN MODULE<br>c) Enter the following data:<br>**pathname for SELECTION LIST**<br>NET>S###>REGCTL.EL<br>**MODULE or BOX number** (your assigned device)<br>**HIWAY/UCN number** (your assigned UCN or Hiway)<br>d) For “BUILD TYPES of Entities to List:” select NIM or HG<br>e) Select the REG CTL (for UCN) or REGLATRY (for Hiway) option<br>f) Press [ENTER]<br>g) The selection list created will be displayed on the screen. Press the [CTL] and [HELP] keys to return to the COMMAND DISPLAY<br>h) Either method creates a selection list, named REGCTL.EL, of the regulatory control points in your assigned device. The file is created in your student directory. |

**Backup procedure**

| Step | Action |
|---|---|
| **2** | From the DEB COMMAND DISPLAY, select RECONSTITUTE MULTIPLE. Enter the following data :<br><br>**pathname for IDF**<br>NET>S###>REGCTL.DB<br><br>**pathname for SELECTION LIST**<br>NET>S###>REGCTL.EL<br><br>Press [ENTER] |
| **3** | If there are no errors, continue with the next step.<br><br>If an error occurs, locate the -UL file. Use the UL file as the selection list to try to get the points it failed to get. If successful, continue to the next step. If not, consult your course manager. |
| **4** | After successfully reconstituting all your Regulatory Control points into an IDF, select PRINT ENTITIES. Then select PRINT IDF Entities from the choices that appear. Enter the following data:<br><br>**pathname for IDF**<br>NET>S###>REGCTL.DB<br><br>**PRINT device ID or DESTINATION pathname**<br>NET>S###>REGCTL.EB<br><br>Press [ENTER] |

# LAB EXERCISE 3-USING EC FILE TO GENERATE EBS (OPTIONAL)

**Introduction**

Builder Commands Executable command files can be used to expedite the process of backing up point data. Selection lists (.EL files) are used to reconstitute entities to IDFs. The command file then prints the IDF entities to exception build files.

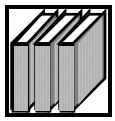
The selection lists can be created by the EC file by using the "list entities in module" command. For UCN devices, the EC “list entities in module” command must specify both point realm and point type to ensure that the selection lists generated contain less than the maximum number of points (usually 255). See Table 7.

**Table 7 - Point Realms and their Point Types**

| Point Realm | Point Types | | | |
|---|---|---|---|---|
| AM | REGLATRY | NUMERIC | COUNTER | CUSTOM |
| | FLAG | TIMER | SWITCH | |
| HG | COUNTER | DIGIN | DIGOUT | DIGCMP |
| | ANALOGIN | ANALOGOUT | ANALOGCOMP | FLAG |
| | TIMER | NUMERIC | LOGBLK | REGLATRY |
| | CNTRLCNTR | BATCHASSN | PROCESSMOD | |
| CIU | ACIDP | CRDP | | |
| NIM | ANALOGIN | ANALOGOUT | REGPV | REGCTL |
| | DIGIN | DIGOUT | DIGCOMP | LOGIC |
| | PROCESSMOD | FLAG | TIMER | NUMERIC |
| | ARRAY | DEVICECONTROL | CMNIM | |

**Instructions**

Create an EC file similar to either the UCN or the hiway version shown on the following pages.



Refer to Section 8 in the *Date Entity Builder* manual for descriptions of the DEB commands that can be used in .EC files.

## HIWAY EC FILE

```
{SOURCE FILE:     filename
{AUTHOR:          your name
{CREATED:         date
{MODIFICATIONS:

{PURPOSE:         THIS COMMAND FILE CREATES A SELECTION LIST
{                 OF REGULATORY CONTROL POINTS, {THEN USES IT
{                 TO PERFORM A RECONSTITUTE MULTIPLE.
{                 IT THEN CREATES EB FILES USING THE PRINT
{                 IDF ENTITIES COMMAND.

V1=NET>S###       {SET VARIABLE 1 = DEVICE/VOLUME
                  {REPLACE ### WITH YOUR PARTITION NUMBER

LS V1>REGCTL.EL ## nn HG REGLATRY
                  {LS=CREATE SELECTION LIST
                  {REPLACE ## WITH YOUR DEVICE NUMBER
                  {nn = your process network number)

RM V1>REGCTL.DB V1>REGCTL.EL
                  {RM=RECONSTITUTE MULTIPLE USING SEL LIST

PI V1>REGCTL.DB V1>REGCTL.EL  V1>REGCTL.EB
                  {PI=PRINT ENTITIES FROM IDFS TO EBS
```

## UCN EC FILE

```
{SOURCE FILE:     filename
{AUTHOR:          your name
{CREATED:         date
{MODIFICATIONS:

{PURPOSE:         THIS COMMAND FILE CREATES A SELECTION LIST
{                 OF REGULATORY CONTROL POINTS, THEN USES IT
{                 TO PERFORM A RECONSTITUTE MULTIPLE. IT
{                 THEN CREATES EB FILES USING THE PRINT IDF
{                 ENTITIES COMMAND.

V1=NET>S###       {SET VARIABLE 1 = DEVICE/VOLUME
                  {REPLACE ### WITH YOUR PARTITION NUMBER

RM V1>REGCTL.DB V1>REGCTL.EL
                  {RECONSTITUTE MULTIPLE USING SEL LIST
PI V1>REGCTL.DB V1>REGCTL.EL V1>REGCTL.EB
                  {PRINT ENTITIES FROM IDFS TO EBS
LS V1>REGCTL.EL ## nn NIM REGCTL
                  {LS = CREATE SELECTION LIST
                  {REPLACE ## WITH YOUR DEVICE NUMBER
                  {nn = YOUR PROCESS NETWORK NUMBER
```

**END OF LAB 3**

# APPENDIX A

## Data Storage Forms

## Comparison of data storage Forms

Tables A.1-A.3 summarize the purpose, advantages, and disadvantages of using checkpoints, intermediate data files (IDFs), and exception build (EB) files as forms of storing data points.

**Table 8 - CHECKPOINTS AS A DATA STORAGE FORM**

| Database Storage Form | Primary Purpose | Advantages | Disadvantages |
|---|---|---|---|
| Checkpoint | Checkpoints are database "saves" that are taken periodically for the purpose of quickly restoring the device's database in case of a device failure.<br><br>Devices such as controllers and LCN modules should have their databases checkpointed on an interval no less than twice the amount of time it takes to complete the checkpoint (TB(1)=2xTB(3)) or on demand by the user. | 1. Checkpoints are most likely to have the most current and up to date image of the device's database.<br>2. Checkpoints can be configured to occur automatically.<br>3. Checkpoints can be requested by the user and require no special system configuration.<br>4. Internal identifiers are retained.<br>5. Faster to restore data. | 1. Additional backup checkpoints are usually made in case the media (that is, floppy, cartridge, or HM disk) becomes corrupted.<br>2. The user needs to remember to make backup copies and maintain control of them to require all copies to have the same image of the database.<br>3. The user might overwrite a good checkpoint with a bad one.<br>4. The information in the checkpoint files cannot be viewed. |

**Table 9 - IDFs as a Data Storage Form**

| Database Storage Form | Primary Purpose | Advantages | Disadvantages |
|---|---|---|---|
| Intermediate Data Files (IDFs) | Intermediate Data Files are used for the initial loading of the database or for reloading the database when the checkpoint is no longer valid, as in the case of some software upgrades. | 1. IDFs can be created directly from the DEB by using a RECONSTITUTE MULTIPLE command.<br>2. IDFs are faster to load than Exception Build files.<br>3. IDFs can be read from and written to while in the DEB.<br>4. IDFs have some database validity checks performed.<br>5. The order of the points in the file is not critical because it makes two passes. | 1. IDFs are more susceptible to becoming obsolete when moving to a new software release.<br>2. IDFs cannot be viewed or modified without using the DEB.<br>3. IDFs are larger files than EBs.<br>4. Internal identifiers are not retained. |

**Table 10 - Exception Build Files as a Data Storage Form**

| Database Storage Form | Primary Purpose | Advantages | Disadvantages |
|---|---|---|---|
| Exception Build Files (EBs) | Exception Build Files (EBs) are text files built by using the Text Editor and can be used for initial database loading or for the modification of the existing database. | 1. Prebuilding - EBs allow the user to prebuild database before system installation.<br>2. Flexibility - EBs allow the user to directly view point data. The Text Editor can be used to edit the data point(s).<br>3. Transportability - EB files can be easily transported between systems. This includes PCs, other storage media, or other LCNs.<br>4. EB files are smaller than IDFs.<br>5. EB files are less susceptible to obsolescence because of a software upgrade.<br>6. EB files can be built using WORKBOOK, Point Builder, or TPS Builder, which incorporates some, database validity checking. | 1. Unless WORKBOOK is used, no database validity checks are performed.<br>2. EB loading is slower than loading from IDFs.<br>3. Internal identifiers are not retained. |

# APPENDIX B

## Checkpoint Guidelines

## Checkpoint guidelines

While the checkpoint strategies you decide to implement may vary, the following section lists some suggested guidelines.

## Formatting Removable Media

Use initialized cartridges or zip disks. Use the command processor Create Volume command (CR) to format the media. When formatting removable media for checkpoint files, follow the guidelines listed below.

**HG Checkpoints**

Format with at least 240 files to ensure that a file access error does not occur.

Previous HG master and logic block checkpoint files are not discarded until the checkpoint is successfully completed. This means that two copies of these files are maintained while a checkpoint is in progress. On a fully loaded Data Hiway with 63 devices, a checkpoint could occupy almost 1,000 kb.

**ATTENTION**

Formatting removable media by using "Initialize Removable Media" from the System Menu will default the file number to 156 files. This may or may not be sufficient to contain all your checkpoint files.

## Demand Save vs. Copy File

It is best to *demand* checkpoints to removable media as a backup for HM checkpoint data, rather than copying the files from the HM to the disk. "Demand checkpoints" ensure that the data is split correctly between media and is therefore usable for loading or restoring a node. When using removable media, demand a checkpoint twice for each node.

## Fast Load Cartridge

Maintain checkpoints on your Fast Load cartridge.

## How Many Backup Copies Should Be Made

The number of backup copies you maintain is a matter of user preference, but usually two checkpoint backup copies on removable media are sufficient. More than two copies add responsibilities in keeping them updated and secure.

## When and How Often to Checkpoint?

Periodically request checkpoints during initial point building. This ensures that your data is backed up. After your system is online with your process, determine how many backup copies you need. You may want to checkpoint to removable media about once a week after the system is online. Determine what type of application is shared between the LCN-level advanced control devices and the process network devices. This will determine how often you checkpoint, if at all.

## When to Disable Autocheckpoint

Depending on the type of application you are using, you may decide to disable autocheckpointing. For example, if your Application Module and process-connected devices are used in a batch application, the auto-checkpoint you may be saving may be a "middle of the batch snapshot." This may not necessarily be the checkpoint you wish to restore to your Application Module. If your process is one where parameters do not get changed as often, then autocheckpointing would be desirable.

> **ATTENTION**
>
> If you wish to maintain a system "startup" checkpoint, you should perform a checkpoint at the time your process is in that phase.

## When to Restore a Checkpoint

When deciding whether or not to restore a node with a previous checkpointed database, consider where the data resides. Knowing where data resides and how it is backed up helps you decide whether a database restore is necessary for a process-connected device that was previously reset. Consider the following scenario: An HG and NIM are reset and restarted after a power outage.

Q. Do the associated process-connected devices need to be restored as well?

A. No. In most cases all that is required is to restart the device. Many of the controllers have backup memory, maintaining the database for an extended period of time.

#### Memory Backup of Process-connected Devices

Table 6 lists the **TotalPlant** Solution (TPS) system process connected devices and how they back up their database. Some process-connected devices provide their memory circuits with an operating voltage through a backup battery. This maintains the memory contents and may make a database restore unnecessary. Additionally, some devices may be supported by a battery backup system or other UPS options.

**Table 11 - Memory Backup**

| Type of Device | Memory Backup ? | Backup Time[2] | Effect of Database Restore from Checkpoint |
|---|---|---|---|
| Basic Controller | Yes | Up to 30 minutes | All control points return to manual mode. |
| Multifunction Controller[1, 5] | Yes | Up to 50 hours | All control and PCFA points return to a program manual mode.[3] |
| Advanced Multifunction Controller[1, 5] | Yes | Up to 72 hours | All control and PCFA points return to program manual mode.[3] |
| Extended Controller | Yes | Up to 30 minutes | All control points return to manual mode, external mode switching disabled. |
| Process Interface Units (HLPIU, LLPIU, LEPIU) | No | N/A | Device returns to processing. |
| Data Hiway Port | Yes | Up to 120 hours | Device returns to processing if hot start configured. |
| Emulated Data Hiway Port in PLCG, EPLCG | No | N/A | Emulated DHPs return to processing after restore. |
| Process Manager and Advanced Process Manager[1, 4, 5, 7] | Yes | Up to 72 hours[8] | Alive restores to Idle.<br>Idle restores to Idle. |
| High Performance Process Manager [1, 4, 5, 7] | Yes | 12 hours<br>or<br>45 hours<br>or<br>55 hours | Alive restores to Idle.<br>Idle restores to Idle.<br>—PM/APM power system versions with 3 AA-size NiCad = 12 hours<br>—HPM power system version with 3 C-size NiCad = 45 hours<br>—AC only power system with 3 alkaline = 55 hours |
| Smart Transmitter | Yes | Non-volatile EEPROM | Restore to box does not restore smart transmitter. For PM/APM/HPM, request a download from STIM Detail Displays to restore transmitter database, if needed. |
| Logic Manager Module[5] | Yes | Up to 6 months | Alive restores to Idle.[6]<br>Idle restores to Idle. |
| 620 Logic Controller System | Yes | Up to 1.5 years | |

**Table 11 - Memory Backup, *continued***

Notes:

1. If reloading CL/MC or CL/PM programs, the programs must be loaded from process module displays. A checkpoint file cannot be used to load or save CL/MC or CL/PM programs.
2. Length of backup depends on battery condition. Change batteries regularly.
3. Requires controller to be in "Idle" state, which can affect logic block, pulsed output processing driven by CL/MC. All CL/MC programs transition to off state upon return to processing state.
4. CL/PM programs restart option configurable.
5. Device must be in Idle.
6. Digital outputs may show “undefined” on some releases.
7. PM/APM/HPM has warm/cold startup options. Cold startup means REGCTL points with outputs are changed to Manual. PM has this option in R401 and later.
8. AC-only power supplies provide CMOS backup only.

# APPENDIX C

## EC File Backup Example

***(Excerpt from Command Processor Operation Manual)***

## Miscellaneous Commands

### Execute Command

The following example depicts how an EC can be used to back up files. In this example, you may do a backup from any source path to any destination path. The EC asks several questions that you must answer to determine where the files are located and which suffixes they have. The EC also requires the file time and date in order to allow selective file backup; for example, performing backups for a week or an entire month. This command file uses several of the commands described in Section 12— Decision-Making Commands, page 101. In the following example, the command file

- gathers the source and destination pathnames,
- uses an LS command to list the desired files,
- writes them to a temporary file named FILES.LT,
- opens file FILES.LT with the &FILE directive,
- reads each line in the file (the third and fourth fields are the file names and suffixes, respectively),
- uses ^R3 and^R4 to copy the file from the source pathname to the destination pathname, and
- repeats this process for each file in FILES.LT.

```
&P**Begin Backup of Files**
&P
&IF EXISTS FILES.LT & THEN DL FILES.LT

& Q2 "Are you using a Cartridge in the backup? (y or n)"
&IF NOT ^Q2 & THEN &G AGAIN
&Q3 "Do you want to format your Cartridge? (y or n)"
&IF NOT ^Q3 &THEN &G AGAIN
&V9 "Enter the device where the Cartridge is located:"
&V10 "Enter the volume name to initialize the Cartridge with:"

&P
&P The Cartridge in device ^V9 will be formatted and initialized
&P and everything currently on the Cartridge will be lost.
&P
&Q4 "Do you wish to continue to format and initialize the Cartridge? (y
or n)"
&IF NOT ^Q4 &THEN &G AGAIN
CR ^V9>^V10 -FMT -MF 2000

&L AGAIN
&V1 "Enter the extension of the files you wish to backup (e.g. CL, *,
EB)."
&V2 "Enter the device from where the files should be read:"
&V3 "Enter the volume from where the files should be read."
&Q5 "Do you wish to specify a date range to choose the files? (y or n)"
&IF NOT ^Q5 & THEN &G NORANG
&V7 "All ^V1's will be backed up, including and after this date
(mm/dd/yy):"
&V8 "All ^V1's will be backed up, prior to this date (mm/dd/yy):"
```

```
&L NORANG
&P
DO FILES.LT
&IF NOT ^Q5 & THEN &G LSTWO
LS ^V2>^V3>*.^V1 -FROM ^V7 -PRIOR ^V8 -NOHDG
&G LSONE
&L LSTWO
LS ^V2>^V3>*.^V1 -NOHDG
&L LSONE
DO
&Q1 "Do you wish to backup any more files? (y or n)"
&IF ^Q1 &THEN &G AGAIN
&V4 "Enter the device where the files should be copied to:"
&V5 "Enter the volume where the files should be copied to."
&IF EXISTS ^V4>^V5> &THEN &G OKPATH
&P
&P The designated destination does not exist, hit <ENTER> to continue
&P and it will be created for you.
PAUSE
&V6 "Enter the main volume name under the destination device:"
CD ^V4>^V6> ^V5
&L OKPATH
&FILE FILES.LT
&L NEXT
&IF NOT READ &THEN &G DONE
CP ^V2>^V3>^R3.^R4 ^V4>^V5>=
&G NEXT
&L DONE
&P
&P ** Completed Backup of Files**
The following shows what appears on the screen as the backup command
file, described above, executes:
EC EXAMPLE.EC
 **BEGIN BACKUP OF FILES**
ARE YOU USING A CARTRIDGE IN THE BACKUP? (Y OR N)
Y
ENTER THE DEVICE WHERE THE CARTRIDGE IS LOCATED:
$F6
ENTER THE VOLUME NAME TO INITIALIZE THE CARTRIDGE WITH:
EC2
THE CARTRIDGE IN DEVICE $F6 WILL BE FORMATTED AND INITIALIZED AND
EVERYTHING CURRENTLY ON THE CARTRIDGE WILL BE LOST. DO YOU WISH TO
CONTINUE TO FORMAT AND INITIALIZE THE CARTRIDGE? (Y OR N)
Y
CR $F6>EC2 -FMT -MF 2000
Create Volume Complete
ENTER THE EXTENSION OF THE FILES YOU WISH TO BACKUP (E.G. CL, *, EB):
ENTER THE DEVICE FROM WHERE THE FILES SHOULD BE READ:
$F7
ENTER THE VOLUME FROM WHERE THE FILES SHOULD BE READ:
```

```
EC
DO YOU WISH TO SPECIFY A DATE RANGE TO CHOOSE THE FILES? (Y OR N)
N

DO FILES.LT
Data Out Complete
LS &F7>EC>*.XX -NOHDG
The list called for by the LS command appears here.

Catalog Complete
DO
Data Out Complete
DO YOU WISH TO BACKUP ANY MORE FILES? (Y OR N)
N
ENTER THE DEVICE WHERE THE FILES SHOULD BE COPIED TO:
$F6
ENTER THE VOLUME WHERE THE FILES SHOULD BE COPIED TO:
EC2
CP $F7>EC>ATTACH.EC $F6>EC2>=
Copy Complete
CP $F7>EC>BLANKS.XX $F6>EC2>=
Copy Complete
CP $F7>EC>BLANKS.XX $F6>EC2>=
Copy Complete
CP $F7>EC>EMPTY.XX $F6>EC2>=
Copy Complete
CP $F7>EC>F1.XX $F6>EC2>=
Copy Complete
CP $F7>EC>F2.XX $F6>EC2>=
Copy Complete
CP $F7>EC>F3.XX $F6>EC2>=
Copy Complete
CP $F7>EC>FILE.XX $F6>EC2>=
Copy Complete
CP $F7>EC>FILE1.XX $F6>EC2>=
Copy Complete
CP $F7>EC>FILES.XX $F6>EC2>=
Copy Complete
CP $F7>EC>FILES1.XX $F6>EC2=
Copy Complete
CP $F7>EC>LONGLINE.XX $F6>EC2>=
CP $F7>EC>SUB_STR.XX $F6>EC2>=
Copy Complete

 **COMPLETED BACKUP OF FILES**
EC Complete
```

LAST PAGE