

OverEasy

User Reference Guide

High Speed File Reorganization Utility for Wang/VS Computers

Tenth Edition, March 2000

Copyright

© Copyright, Utility Systems, Inc., 1991-2000. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Utility Systems, Inc.

Trademarks

OverEasy is a registered trademark of Utility Systems, Inc.

Software License Agreement

Use of OverEasy and EggShell is restricted to those specifically licensed by Utility Systems, Inc. That license contains restrictions relating to the copying and use of the OverEasy and EggShell software and related documentation.

Address comments and questions to:

usi@overeasy.com

Table of Contents

1

Introduction

OverEasy 4 with EggShell	1-1
Features and Functions	1-2
Version 4 Enhancements	1-3
Version 3 Enhancements	1-4
Technical Overview	1-4
System Requirements	1-5

2

Installation

Preparation	2-1
Run INSTALL	2-1
Running OverEasy and EggShell	2-1

3

General Information

File Organization	3-1
Block Placement	3-1
Volume Analysis	3-2
Extent Consolidation	3-2
About XDMS	3-3
Indexed+ Organization	3-3
DMS/TX	3-4
Shared and Special Input Mode	3-4
Block Packing Density	3-5
Alternate Index Packing Density	3-6
U File Format	3-7
Growth Factor	3-8
Secondary Extent Specification	3-9
Alternate Index Specification	3-9
Changing Record Specifications	3-10
Reorganization Report	3-11

4

Getparm Reference

INPUT Getparm	4-1
OPTIONS Getparm	4-2
ALTINDEX Getparm	4-5
OUTPUT Getparm	4-6
PRINT Getparm	4-8

5

About EggShell

Overview	5-1
Working with Lists	5-2
Creating a List	5-2
PF Key Functions	5-4
Overriding Parameters	5-6
Saving a List	5-9
Loading a List	5-11
Reorganizing Files	5-12
Multiple File Reorganization	5-12
Delaying the Start of Reorganization	5-14
Crash Recovery	5-15
Reorganizing Files from a Procedure	5-15
Sample Command Procedure	5-16
EggShell–OverEasy Reorganization Report	5-17
Stopping the Reorganization	5-18
Checkpoint Recovery	5-19
Volume and Free Extent Management	5-20
Free Extent Analysis	5-21
Free Extent Consolidation	5-21

6

Return Codes

OverEasy Return Codes	6-1
EggReorg Return Codes	6-2

7

[Getparm Quick Reference](#)

[OverEasy Getparm Parameters](#) 7-1

A

[Appendix A](#)

[WAITER Program](#)..... A-1
[Operation](#)..... A-1
[WAITER Return Codes](#) A-2
[Sample Procedures](#) A-2

B

[Appendix B](#)

[EGGCIP Program](#)..... B-1
[Operation](#)..... B-1
[EGGCIP Return Codes](#)..... B-2
[Sample Procedure](#)..... B-2

1 INTRODUCTION

OverEasy 4 with Eggshell

Since its introduction in 1989, OverEasy has become the standard for indexed file management on Wang VS systems. It broke the speed barrier by reorganizing indexed files 10 to 15 times faster than Wang's COPY program. Since that time we have enhanced OverEasy to be faster, added support for Wang's new XDMS indexed file format, and added EggShell, a powerful indexed file management program.

EggShell enables you to create lists of files, libraries, and volumes for reorganization by OverEasy. A unique 'point and shoot' interface actually makes it fun to create or edit lists which you can reorganize right away, or set up to run overnight. EggShell also has a special "Crash Recovery" feature which very quickly identifies and reorganizes files left open after a system crash.

Thousands of USI customers continue to benefit from these improvements through free upgrades provided as part of their software license or maintenance agreement. Our annual maintenance program supports the ongoing research and development that has made these improvements possible.

Release 4, the latest release of OverEasy/EggShell, adds free extent consolidation which many of you have asked for. It has been integrated into the EggShell disk management program. As with file reorganization, you can perform free extent consolidation interactively using EggShell or directly through a proceduralizable program that we have provided.

We call the new program EGGCIP. You use EGGCIP primarily to make more of the free space on your disk volumes available for new file creation. At the same time it will try to consolidate files that are in multiple extents into a single extent. EGGCIP can even be used on the system volume, at any time of the day. It is designed to keep a low profile and not interfere with your system, even while reading the VTOC!

Section 5 describes EggShell in more detail. Take a few minutes to read the tutorial and then try EggShell. You will be amazed at how easy indexed file maintenance can be.

Features and Functions

- **10–15 times faster than COPY** – Copy/Reorg time is typically reduced by 90% for indexed and alternate indexed files.
- **Less Workfile Space** – OverEasy reduces the alternate index workfile space by 75% or more.
- **Saves Disk Space** – OverEasy allocates space much more precisely and eliminates hidden unused blocks.
- NEW* • **Disk Volume Summary** – View statistics about all your disk volumes at a glance. Shows number of files and libraries, percent utilization, free space and more.
- NEW* • **Extent Consolidation** – Make free space more usable on your volumes by reducing the number of free extents.
- **Faster Crash Recovery** – Identifies and reorganizes any files left open after a crash.
- **Reorg Files, Libraries and Volumes** – Reorganize anything from a single file to your entire system in a single, easy step.
- **Unattended File Reorganization** – can be set up to run at night or other quiet system times.
- **Supports XDMS (Indexed+) files.**
- **Converts files from DMS to XDMS format** – and vice-versa at speeds up to 150 times faster than COPY. Entire libraries may be converted in one step.
- **Use instead of CREATE** – for converting files from consecutive to indexed or Indexed+ format.
- **100% Compatible** – All files reorganized with OverEasy are 100% compatible with Wang's DMS, DMS-TX and XDMS standards.
- **Set Minimum Filesize** – For transaction type files which expand and contract frequently, you can set a minimum file size for reorganization.

Version 4 Enhancements

OverEasy version 4 introduces the following enhancements:

Extent Consolidation - The free extent consolidator will analyze the VTOC of a volume and then consolidate your unused blocks to create more usable free space. It will also consolidate files that are in multiple extents. Learn more about this feature and how to use it in section 5.

Enhanced Catalog Utilization - EggShell accesses the file catalog to set file attributes such as Growth Space, Secondary Extent Size, and Minimum Block Allocation whenever it reorganizes a file. Now, OverEasy will access the catalog to set defaults when executed by a program other than EggShell. This means you can control your indexed file attributes even if you run OverEasy from your own procedures, from the new Wang COPY enhancement, or from the numerous other disk management programs that call OverEasy for file reorganization.

New Install Utility - Keeping the installation simple while adding feature after feature to OverEasy has not been easy. In order to fit all of these features onto a single low density diskette we have used the data compression software we developed for BackBurner. To install OverEasy you still just run the INSTALL program from the diskette, we do the rest, decompressing and copying the files to the target library. Since the OverEasy package has now increased by several more program files, we have provided the ability to designate a library other than @SYSTEM@.

Links with BackBurner® - This version of OverEasy can be mapped into BackBurner as an SSL (Shared Subroutine Library). BackBurner can then use OverEasy to create a reorganized file during restoration. A soon to be released version of BackBurner (4th quarter, 1994) will allow indexed files to be backed up directly to USI record format and restored as reorganized files. This will provide the capability to reorganize huge files using tape and also provides the foundation for backing up Indexed+ files in open/shared mode.

Version 3 Enhancements

In addition to EggShell, OverEasy 3 introduced these new features:

- **New Compressed file format** A new compressed file format, FILEORG=U, has been added. This file format compresses the essential data of an indexed file into significantly less disk space than the original file. This is very useful for reorganizing large files when there is insufficient disk space available to make a regular copy of the file.
- **Secondary extent size** – You may now specify the size of secondary extents when running on VS operating 7.21.05 or greater.

Technical Overview

This utility copies the file and builds new indexes in the process. Input files with damaged primary and/or alternate indexes are handled just as easily as non-damaged files. Like Wang's COPY, OverEasy will recover any file that has an intact primary data chain.

OverEasy will verify the integrity of the primary data chain as it runs and will inform you of any problem. The total time to perform the verified copy/reorg is much less than would

be required just to run Wang's VERIFY program against the file. You will find that it is quicker and easier to simply run the EggShell "Crash Recovery" option against all your indexed files after a system crash.

In addition to detailing the file block usage, time and disk I/Os used for the run, OverEasy will give valuable record and block size statistics to help you better manage and fine tune your files.

The primary index structure is improved and positioned within the data blocks to optimize random retrieval when buffer pools are used.

System Requirements

OverEasy and EggShell will run on any VS system using VS/OS release 7.10 or greater.

Release 7.20 or greater is required to create Indexed+ (XDMS) files. Although not required, installation of Wang's XDMS product is recommended prior to creating Indexed+ files.

The OverEasy program files take approximately 220 blocks on your system volume.

OverEasy with EggShell requires a minimum of 256K modifiable data area (MDA). Increasing the MDA will significantly improve reorganization speed, especially for alternate indexed files.

Free extent consolidation from EggShell requires a subtask quota of at least one. When running free extent consolidation via EGGCIP, the task must have a minimum MDA size of 512K bytes.

2**INSTALLATION**

Preparation

Loading OverEasy is a one-step process managed by the `INSTALL` utility. You must have System Administrator rights while installing OverEasy.

Run INSTALL

Run the installation program, `INSTALL` in library `OVEREASY` on diskette volume `OVEREZ`.

`INSTALL` automatically copies all of OverEasy's files into `@SYSTEM@` or the library of your choice. If you already have OverEasy installed in the specified library, `INSTALL` replaces the existing files with the newer versions.

The OverEasy program files are stored on the installation diskette in compressed format. The `INSTALL` program must be used to decompress and copy the files to your system volume.

Installation takes about 1 minute.

Running OverEasy and EggShell

Many of our customers use OverEasy implicitly through programs or procedures supplied by their application software vendor, another software utility company, or written in house for their specific needs. Those programs and procedures will work with the new version.

For new users and those who previously used OverEasy interactively on a per-file basis, we encourage you to try our EggShell interactive file reorganization program. It is a powerful and easy to use indexed file management tool. A tutorial is provided in the EggShell section of this manual.

To use OverEasy directly on a particular file, simply run the program `OVEREASY`. The subsequent screens allow you to specify the input file, reorganization options, and the output file. You may simply accept the defaults on the `OPTIONS` screen, and OverEasy will create a well organized file. If you wish to adjust these options to tune your file, see sections 3 and 4.

3**GENERAL INFORMATION**

File Organization

The organization of a file refers to how the data, index, and alternate index blocks of a file are formatted and how those blocks are related to one another. Both DMS and XDMS expect that files will be organized in a standard format. Files created by OverEasy conform to that standard format and can be validated with the Wang supplied VERIFY or VERPLS utility as appropriate.

Block Placement

Blocks may be placed anywhere in the file space as long as they adhere to the standard DMS or XDMS structure for that file organization. A file should be reorganized when the blocks are not placed for optimal retrieval speeds or when the blocks contain excessive unused space. When a file is reorganized with OverEasy, the placement of data and index blocks is determined using unique algorithms designed by Utility Systems, Inc. to optimize both sequential and random retrieval speeds.

DMS places all of the primary index blocks in the middle of the prime data area and all of the alternate indexes at the end of the prime data area. Its file size algorithm works correctly only if the estimated record count, record length, and assumed compression factor are accurate. When they are not, COPY can produce files that contain many unused blocks and perform poorly in random retrieval mode.

Indexed+ files created using the new XDMS product have no organization by design. Data, index, and alternate index blocks are simply assigned sequentially as needed during file load. The resulting organization is somewhat better than DMS when no alternate indexes are present. When alternate indexes are present they are scattered all over the file space causing not only reduced alternate path access times, but decreased sequential and random retrieval speeds along the primary path as well.

OverEasy takes a completely different approach to file organization than either DMS or XDMS. The file size is derived by examining the input file to make an accurate space allocation. The prime index blocks are distributed within the prime data area to optimize random retrieval and are never separated from the data blocks by unused blocks.

OverEasy places alternate index structures immediately after the prime data/index area. The higher level index blocks of each alternate path are distributed within the area for optimized retrieval in the same way that the primary index is distributed in the prime data area. Free space for file growth is always allocated at the end of the file.

The reorganization report provides detailed information about the number of blocks used for the various portions of the file. Both the original estimate and the actual used block count are reported. Because OverEasy assumes that the input file was created by

OverEasy, the estimations will be more accurate the second and subsequent times that OverEasy is used on a particular file.

Volume Analysis

The Volume Analysis screen provided by EggShell provides a comprehensive look at the status of all of your disk drives at a glance. It lists all of the mounted volumes with utilization statistics, an analysis of the free space on the volume, and the size of the largest file that can be created on each volume.

Select a volume from this screen and view additional detail, including all of the free extents. Press a key to enter free extent consolidation and quickly consolidate the free space on the volume.

Extent Consolidation

Extent consolidation reduces fragmentation of your disk volumes by consolidating the unused space on the volume into fewer larger free extents. Consolidating the free space on a volume improves system performance and capacity in several ways. When free space is all in one place, your data files are packed closer together for faster inter-file access. New files are created in fewer extents, which reduces intra-file access times. This is particularly beneficial for large work files such as those used by sort. In general, a volume which is consolidated regularly will have greater usable capacity. Consider a 326mb volume that is typically 75% full. Increasing utilization by 10% means making another 33mb available.

About XDMS

XDMS is the acronym that Wang uses for its new Extended Data Management System. That software product offers a number of improvements over the original Data Management System (DMS). The most notable improvements are advanced sharing and roll forward recovery.

Advanced sharing eliminates most of the delay typically associated with accessing a file in shared mode. There is no intertask messaging to a sharer task, in fact, there is no sharer task. The net result is that applications that were bottlenecked on shared file requests run much faster.

Roll forward recovery is a DMS/TX feature available with XDMS that permits damaged file recovery by restoring a backup copy and “rolling forward” the changes from the after image journal.

Aside from being a relatively new product, the XDMS problems noted by early users have been inflated file size and intolerable conversion and reorganization times. OverEasy minimizes the increased file size and eliminates the conversion and reorganization problem.

Indexed+ Organization

Indexed+ is the name used to describe the organization of files created by XDMS. Within the Indexed+ organization there are two file sequence types available and two index types.

Key sequenced (KSDS) files have records stored in the order of ascending primary key. New records are inserted between the records with lower primary keys and those with higher primary keys. When read sequentially, the records are returned in primary key sequence. Key sequencing is supported by OverEasy.

Entry sequenced (ESDS) files have records stored in the order that they are entered into the file. New records are added to the end of the file. When accessed sequentially, the records are returned in the order that they were written to the file. Entry sequencing is not supported by OverEasy.

B-tree indexes are very much like the indexes in DMS indexed files. Particular entries are located by searching the tree from the top down to the lowest level member. All alternate indexes are maintained using B-trees. OverEasy supports B-tree type indexes.

Hashed indexes provide a new way to maintain the primary index. When properly used for selected applications, record retrieval times may be significantly reduced. In general hashed indexes require more disk space than B-trees. Hashed indexes are not supported by OverEasy.

When files are converted from indexed to Indexed+, they are converted to key-sequenced B-tree format. That combination is functionally equivalent to the old DMS indexed file format. It is the only format that is fully compatible with DMS. OverEasy supports the key-sequenced B-tree format.

DMS/TX

Indexed and Indexed+ files that are part of DMS/TX databases are fully supported. OverEasy handles the recovery blocks exactly like COPY and can be used instead of COPY during reorganization, as part of soft crash recovery, when the file is damaged. Note that OverEasy should not be used when BACKUP is specified in the DMS/TX “Operator Instructions for Soft crash Recovery”.

When files are converted between indexed and Indexed+, the DMS/TX database information is also converted, and the new file will be a part of the same DMS/TX database. Files requiring soft crash recovery should not be converted between DMS and

XDMS format with either OverEasy or COPY as the recovery information is lost and the database will be inconsistent when used with the newly converted file.

Indexed+ files do not support the notion of pre-allocated recovery blocks. When attached to a DMS/TX database, a recovery block is allocated dynamically. The RECBLK keyword of the OPTIONS getparm is used to specify that DMS/TX database attachment is to be preserved (RECBLK=U) or removed (RECBLK=N).

Shared and Special Input Mode

Sometimes a need arises to provide a copy of an indexed file that is currently or perpetually open in shared mode. OverEasy can use shared record input to process those files. The resultant reorganization will be about 25% slower than with the standard INPUT mode, but a lot faster than using COPY in shared mode.

Use the MODE keyword of the INPUT getparm to specify MODE=SHARED. A LOCK getparm will be issued that is substantially the same as the lock getparm issued by COPY, SORT and other Wang utilities.

The MODE keyword can also be set to SPECIAL to use special input mode. While this mode is slightly faster than shared, it does not allow you to coordinate your file access with updates made by other users. As a result, special input mode is not 100% reliable and we recommend that you use SHARED if the need arises.

Block Packing Density

Like DMS and XDMS, OverEasy allows for user specification of data and index block packing density using the OPTION getparm keywords DPACK and IPACK respectively. The packing density is specified as a percent, with 100% being the maximum.

DPACK specifies the percent of available space in prime data blocks that is to be filled. The excess is available to accommodate increases in the size of existing records (caused by updates) as well as added records.

In much the same way, IPACK specifies the percent of available space in prime index blocks that is to be used for index entries. The unused space is available for new index entries.

OverEasy determines the number of bytes that are to be filled by the simple formula:

$$\text{Bytes} = (\text{packing-factor} \times 2048) / 100$$

Blocks are considered full when the physical size of the next record would cause the limit to be exceeded. Like DMS, OverEasy insures that regardless of the DPACK value, at least one record is placed in each block.

Both DMS and XDMS add control information to each block, and XDMS adds control information to each record. The size of that control information is counted as data when determining if a block is filled. DMS adds 5 bytes to each block as well as 2 bytes to each record when the file is alternate indexed. XDMS adds 12 bytes to each block and from 8 to 12 bytes per record. In addition, XDMS will replicate the primary key of each record that is stored in compressed form.

The reorganization report shows the minimum, average, and maximum size of the prime data and primary index blocks, which will allow you to fine tune your DPACK and IPACK specifications for optimum disk usage. The last data block is not considered for the minimum data block statistic unless it is the only data block.

Alternate Index Packing Density

The packing density of primary data blocks and primary index blocks can be controlled using DPACK and IPACK. By adjusting these values, you can control the amount of block splitting that occurs to primary data and index blocks when records are updated or added. Minimizing block splits reduces the average time required to add new records and access existing records.

OverEasy also allows you to control block splits in the alternate index area. Wang's DMS always creates alternate index records with a packing density of 100%. When you add a record to a file with alternate indexes, blocks in each index must split to accommodate the new alternate path entry. In fact, for large files, you may have to split blocks at several levels of the alternate index tree. This not only consumes time, but causes your alternate index efficiency for random retrieval to degenerate rapidly.

The APACK factor in OverEasy's OPTIONS screen allows you to specify a packing density for all of the alternate indexes in the file. You may also provide specific values for each alternate path in the file by specifying ALTINDEX=YES in the OPTIONS getparm. The ALTINDEX getparm is then displayed which allows you to specify a unique APACK value for each alternate index.

Specifying an APACK value of less than 100% will usually cause the number of blocks used for the index to increase. An APACK value of 95% (a good value) would cause the initial size of the alternate index to increase by about 5%. We recommend using a value in the 90% to 95% range to reduce block splits and improve performance.

U File Format

A common backup practice is to reduce the size of large files by using OverEasy to remove alternate indexes prior to backup. If the file is subsequently needed, it is restored from tape and the alternate indexes quickly rebuilt using OverEasy.

The new file organization, FILEORG=U or “USI Compressed”, is an improvement on that process. When a file is converted to USI compressed, the prime index and all alternate indexes are removed. The data records are then compressed and packed as tightly as possible, even spanning blocks, so that there is no wasted space. All of the original file specifications are recorded, including the per-record alternate index masks.

The USI compressed file appears as a consecutive fixed file with record size 2048. When specified as input, OverEasy recognizes the USI compressed file format and sets the options just as if the original file were specified as input.

This new format is used extensively by EggShell when large files must be reorganized and there is insufficient disk space to build the reorganized files on the input disk. The USI file format not only saves disk space, but also saves time. The end to end reorg times of off-volume reorganizations are reduced considerably.

Consider converting large indexed files to USI compressed form and backing up the USI compressed copy. The file will be much smaller, but more importantly, will be verified to be free of prime data damage by virtue of having been processed by OverEasy.

Growth Factor

The secondary extent allocation algorithm that DMS uses allocates as much as 50% of the original file size when extra space is needed. It is important to plan for small changes in file size so that secondary allocation does not occur.

When compressed records are updated, the physical record size on disk may increase. Indexed files with compressed records should be created with a small amount of unused space in each block to allow for this normal fluctuation in record size. Block splits can be avoided in this case by allowing a little extra space in each block with a DPACK of less than 100%

When new records are added randomly, block splits may occur. If you typically add many random records to your file between reorganizations, you may want to allow extra space in each block for one or more records by further reducing the DPACK specification.

When many records are added and/or updated, block splits cannot be avoided. You can reserve extra blocks for splits by specifying a file growth factor in the OPTIONS getparm of OverEasy. The number of growth blocks may be specified as a percentage of the overall file size or as an absolute number of blocks to be reserved.

The keyword GROWTHB is used to specify an absolute number of blocks to be reserved. OverEasy will include the growth blocks in its initial block estimate and file allocation. If the estimate is low however, OverEasy will not force secondary extent allocation just to provide the growth blocks.

The keyword GROWTHP is used to specify the number of growth blocks as a percentage of file size. GROWTHP may be specified as a real number with any number of decimal positions. A GROWTHP value of 0.1 would mean that one extra block should be allocated for every thousand blocks in the initial file. GROWTHP factors should rarely exceed 1%. You can determine if your growth factor is adequate by watching for an increase in the number of extents between reorganizations.

Both GROWTHP and GROWTHB may be specified. OverEasy will compute GROWTHP and attempt to satisfy the larger value.

Growth blocks are always allocated at the end of the file. Be careful not to run any file compression utilities against the reorganized file. They will remove the growth blocks and cause DMS to expand the file size by up to 50% the next time a block split occurs.

Secondary Extent Specification

As described above, when an indexed file runs out of space, a secondary extent of as much as 50% of the original file size is allocated. Going into secondary extent for a large file could consume as many as 65,000 blocks, a situation that could exhaust all available

disk space and force immediate reorganization. It is unlikely that anyone who reorganizes regularly would ever use the gross over allocation of 50%.

When using OverEasy on VS/OS release 7.21.05 and above, secondary extent size can be set as a percentage of original file size. The SECEXT keyword is included in the OUTPUT getparm when running OverEasy on systems which support it.

Alternate Index Specification

Alternate index specifications default to the specifications of the input file. If you want to add, delete or change the specification of any alternate index, you should reply YES to the ALTINDEX keyword in the OPTIONS getparm. That will cause the ALTINDEX getparm to be issued which allows you to specify the key position, length, and duplication attributes for all of the alternate indexes. The defaults of the ALTINDEX getparm screen will be set to the input file alternate index specification. See the section titled "Getparm Reference" for more detailed information.

Changing Record Specifications

Three attributes describe records and how they are stored in a file:

RECTYPE Describes whether the records are Fixed or Variable in length.

RECSIZE Describes the fixed record size or the maximum variable record size.

COMPRESS Specifies whether or not records are to be compressed when stored on the disk.

These record specifications will default to the same specifications as the input file. If they are changed, OverEasy will reformat the records to conform to the new specifications using the following rules:

- **When the record size must be increased**, the record is extended by adding spaces on the right.
- **When the record size must be decreased**, the record is truncated on the right.
- **When changing from RECTYPE=V to RECTYPE=F** the output record length will be the specified RECSIZE and records will be adjusted according the above rules.
- **When changing from RECTYPE=F to RECTYPE=V** the output record length will be the smaller of the input record length or the specified RECSIZE.
- **When COMPRESS=YES is specified**, the logical record length is unaffected unless the RECSIZE parameter is reduced.
- **When RECTYPE=F and COMPRESS=YES is specified**, all of the logical records will be converted to the specified fixed length according to the rules above, then they will be compressed. The resulting file will, however, have the VARIABLE attribute as required by DMS.

Reorganization Report

OverEasy creates a report after the reorganization is complete. The report provides the following information:

- File, library and volume of the input and output files
- Options that were selected for the run
- Estimated records and blocks required for each portion of the file
- Actual records and blocks in each portion of the file
- Wall time, CPU time, and disk operations consumed while loading each portion of the file.
- Minimum, average, and maximum record and block sizes in the prime data area

The report is one page and requires only one block to save on disk. We suggest that you designate a library to contain all of the reports and direct them there with the `PRINT` getparm. Specifying `QUEUE=NO` will suppress printing and keep the file on disk for reference.

Periodic review of the reports can be used to insure that your reorganization parameters are up to date. For instance, consider a compressed file in which you want to avoid block splits by allowing for two inserted records in each block. If the average physical record length is 109 bytes, you would set `DPACK` to 89%. That would fill the blocks to 1818 bytes, leaving 225 bytes to accommodate two more records. Over time, the addition of new fields to the records may cause the average physical record size to increase to 131 bytes. The `DPACK` should be adjusted to 87% to allow for the same number of insertions.

Also of interest might be the size, in blocks, of the alternate index structures and the number of levels in the index structure. Very large alternate keys will cause the number of index levels to increase, which will increase the random retrieval time. Shortening the alternate keys will reduce disk space requirements and potentially reduce the number of levels in the index.

The report also shows you how much time and how many disk blocks are attributable to each segment of the file. This information can be used to estimate the reorganization time and disk space savings that can be realized by removing an alternate index path.

Whether or not you produce the report, OverEasy accumulates all of this valuable information during the reorganization. With a little analysis, you can use it to fine tune your files for peak performance.

4

GETPARM REFERENCE

This section describes the OverEasy Getparm requests and associated keywords. A summary table is provided in section 7 for quick reference.

```

Wang VS GETPARM v 7                               Parameter Reference Name: INPUT
                                                    Message ID: 100
                                                    Component: OVEREZ
Information Required by OVEREASY
Active Subprogram is OVEREASY
-----
Please identify the file to be reorganized.

OverEasy High Speed Indexed File Reorganization Utility      Version 3.00
-----
FILE      = *****      LIBRARY = INLIB***      VOLUME  = INVOL*

Specify the input file processing mode.
May be INPUT (recommended), SHARED, or SPECIAL.  MODE      = INPUT**

Press PF16 to terminate

(c)Copyright 1989-92, Utility Systems, Inc.
All rights reserved. Unauthorized use, duplication, or distribution
is strictly prohibited.

SERIAL#: 000000, WANGNET-ID: USI

```

INPUT Getparm

m

This Getparm identifies the file which is to be copied and reorganized. It is displayed first because many of the subsequent Getparms contain initial values that are derived from the attributes of the input file.

Overeasy requires that the input file be organized as indexed.

- FILE** Specify the name of the file. This parameter is required.
- LIBRARY** Specify the name of the library. This parameter is required. The default value is your INLIB usage constant.
- VOLUME** Specify the name of the volume. This parameter is required. The default value is your INVOL usage constant.
- MODE** Specify input processing mode. Defaults to INPUT. Optional modes are SHARED and SPECIAL input. When SHARED is used, a standard LOCK Getparm is issued to allow locking of the file during input. For more information about shared mode, refer to section 3.

Wang VS GETPARAM v 7	Parameter Reference Name: OPTIONS
	Message ID: 200
	Component: OVEREZ
Information Required by OVEREASY Active Subprogram is OVEREASY	
Please review the following reorganization parameters.	
Should a reorganization report be generated?	REPORT = YES (YES or NO)
What is the output file organization?	FILEORG = I (C, I, or N)
Are the records Fixed or Variable length?	RECTYPE = F (F or V)
What is the F or maximum V record size?	RECSIZE = 128* (Min 26)
Should the records be compressed?	COMPRESS = YES (YES or NO)
What is the primary key start position?	KEYPOS = 19**
What is the primary key length?	KEYLEN = 8**
What is the data block packing density?	DPACK = 100 Percent
What is the index block packing density?	IPACK = 95* Percent
What is the alternate index block packing density?	APACK = 95* Percent
Specify file growth space: In blocks...	GROWTHB = ***** Blocks
or as a percentage of file size...	GROWTHP = 1.0** Percent
DMS/TX recovery blocks: None or Allocated?	RECBLK = N
There are 2 alternate indexes. Want to modify?	ALTINDEX = NO* (YES or NO)
(c)Copyright 1989-94, Utility Systems Inc.	

OPTIONS
Getparm

m

This Getparm is used to accept or change most of the attributes of the file that is to be created. The defaults are set to match the attributes of the input file as closely as possible.

REPORT Specify whether to print the reorganization report. Valid responses are YES and NO. The default is YES. When YES is specified, the PRINT Getparm will be issued at the completion of the run to solicit the report file name.

FILEORG Specify the DMS or XDMS file organization for the output file. Supported organizations are C=Consecutive, I=Indexed, N=Indexed+. Indexed+ is supported on operating system release 7.20 and above. The default will be the file organization of the input file.

RECTYPE Specify whether the output records are F (fixed length) or V (Variable length). When fixed is specified with COMPRESS=YES, the resulting records are considered to be variable by DMS but will all be the length specified in RECSIZE. The default value is taken from the input file's specifications.

- RECSIZE** Specify the integer fixed record size or maximum variable record size. When RECTYPE=F this value is the logical record size that is to be assigned to all records. When RECTYPE=V this value is the maximum logical record size. If necessary, input records will be truncated or extended to conform to the specified value. The default value is taken from the input file.
- COMPRESS** Specifies whether records are to be compressed when stored on disk. Valid responses are YES and NO. Compressing records can save considerable space which can also result in faster retrieval. Consult the Wang DMS manual for more information about compressed record format. The default value is taken from the input file's specifications.
- KEYPOS** Specify the position of the primary key. The valid range is 1 through RECSIZE. For indexed and indexed+ input files the value defaults to the position of the primary key of the input file. With consecutive input files this keyword defaults to spaces. KEYPOS must be specified when FILEORG is I or N.
- KEYLEN** Specify the length of the primary key. The sum of KEYPOS and KEYLEN cannot exceed RECSIZE+1. For indexed and indexed+ input files the default is the length of the primary key of the input file. For consecutive input files the default is spaces. KEYLEN must be specified when FILEORG is I or N.
- DPACK** Specify the packing density for data blocks as an integer percentage value from 1 to 100. The maximum data block size will be set to the rounded product of 2048 times the DPACK percent. The default will be the value associated with the input file or 100% if the input file is consecutive.
- IPACK** Specify the packing density for index blocks as an integer percentage value from 1 to 100. The maximum index block size will be set to the rounded product of 2048 times the IPACK percent. The default will be the value associated with the input file or 95% if the input file is consecutive.
- APACK** Specify the packing density for alternate index blocks as an integer percentage value from 1 to 100. The maximum index block size will be set to the rounded product of 2043 times the APACK percent. The default will be the IPACK value associated with the input file or 95% if the input file is consecutive.
- GROWTHB** An absolute number of blocks to be reserved for file growth. The number of blocks that will be reserved may be less if the block requirements for the file are underestimated. When GROWTHP results in a greater number of blocks, GROWTHP overrides this specification. The default value is 0.
- GROWTHP** The growth factor specified as a percentage of extra unused blocks that are to be allocated to accommodate future growth in the file. The value may contain up to three decimal positions. The following examples show how the growth factor is applied.

- GROWTHP=0.01% — 1 Block/10000 Blocks in file
- GROWTHP=0.5% — 1 Block/200 Blocks in file
- GROWTHP=1% — 1 Block/100 Blocks in file
- GROWTHP=100% — 1 Block/Block in file

When GROWTHP results in fewer blocks than are specified in GROWTHB, then GROWTHB will be used. The default value is 1.0

RECBLK Specifies whether DMS/TX recovery blocks are to be allocated or used. The valid responses are N=None (no recovery blocks), A=Allocate blocks (indexed only), U=Use recovery blocks (valid if the input file has recovery blocks in use). OverEasy will add, remove, or copy recovery blocks in the same way that the Wang COPY program would. The default value is determined from the input file's recovery block status. Before changing this keyword, consult the DMS/TX Reference Manual for a full discussion of this topic.

ALTINDEX Specifies whether you want to change the alternate index specification. The default is always NO. Specifying YES will cause the ALTINDEX Getparm screen to be presented for review and modification of the alternate index specifications.

```

Wang VS GETPARM v 7                               Parameter Reference Name: ALTINDEX
                                                    Message ID: 300
                                                    Component: OVEREZ
Information Required by OVEREASY
Active Subprogram is OVEREASY

Specify alternate index parameters.

KEYPOS01 = ****    KEYLEN01 = ***    DUPSOK01 = ***    APACK01 = 95*
KEYPOS02 = ****    KEYLEN02 = ***    DUPSOK02 = ***    APACK02 = 95*
KEYPOS03 = ****    KEYLEN03 = ***    DUPSOK03 = ***    APACK03 = 95*
KEYPOS04 = ****    KEYLEN04 = ***    DUPSOK04 = ***    APACK04 = 95*
KEYPOS05 = ****    KEYLEN05 = ***    DUPSOK05 = ***    APACK05 = 95*
KEYPOS06 = ****    KEYLEN06 = ***    DUPSOK06 = ***    APACK06 = 95*
KEYPOS07 = ****    KEYLEN07 = ***    DUPSOK07 = ***    APACK07 = 95*
KEYPOS08 = ****    KEYLEN08 = ***    DUPSOK08 = ***    APACK08 = 95*
KEYPOS09 = ****    KEYLEN09 = ***    DUPSOK09 = ***    APACK09 = 95*
KEYPOS10 = ****    KEYLEN10 = ***    DUPSOK10 = ***    APACK10 = 95*
KEYPOS11 = ****    KEYLEN11 = ***    DUPSOK11 = ***    APACK11 = 95*
KEYPOS12 = ****    KEYLEN12 = ***    DUPSOK12 = ***    APACK12 = 95*
KEYPOS13 = ****    KEYLEN13 = ***    DUPSOK13 = ***    APACK13 = 95*
KEYPOS14 = ****    KEYLEN14 = ***    DUPSOK14 = ***    APACK14 = 95*
KEYPOS15 = ****    KEYLEN15 = ***    DUPSOK15 = ***    APACK15 = 95*
KEYPOS16 = ****    KEYLEN16 = ***    DUPSOK16 = ***    APACK16 = 95*
    
```

- ALTINDEX Getparm

arm

This Getparm is issued when ALTINDEX=YES is specified in the OPTIONS Getparm. It allows for respecification of the alternate indexes. The default values will be from the alternate index specification of the input file. You may add, delete or modify any of the 16 alternate index paths.

There are sixteen sets of three keywords. Each individual keyword is qualified by the alternate index path number 01 thru 16 where nn is specified below. To remove an alternate key path, simply blank out its associated key specification fields.

KEYPOSnn Specify the starting position of the alternate key where the first character of the record is position 1.

KEYLENnn Specify the length of the alternate key field. This keyword is required in all lines where **KEYPOSnn** is not blank.

DUPSOKnn Specify YES or NO to allow duplicate alternate key fields or not. When duplicates are allowed, the alternate index structure that is created is different. You should specify YES unless you are certain that there is no possibility of duplicates occurring.

APACKnn Specify the packing density for this alternate index as an integer percentage value from 1 to 100. The maximum index block size will be set to the rounded product of 2043 times the APACK percent. The default will be the APACK value specified in the **OPTIONS** Getparm.

```

Wang VS GETPARM v 7                               Parameter Reference Name: OUTPUT
                                                    Message ID: 400
                                                    Component: OVEREZ
Information Required by OVEREASY
Active Subprogram is OVEREASY

-----

Please specify the following output file information:

FILE      = *****      LIBRARY = OUTLIB**      VOLUME = OUTVOL

Optional file protection period:      RETAIN = *** Days
Optional file protection class:      FILECLAS = *
Total estimated file blocks:          BLOCKS = 12597***
Modify Access Control List?          ACL = NO*
Secondary extent as % of original file size: SECEXT = 50 %

c)Copyright 1989-94, Utility Systems, Inc.
    
```

O
U
T
P
U
T

G
e
t
p
a
r
m

m

This Getparm is the standard output file open Getparm which is issued by DMS. It specifies the name and location of the reorganized indexed file that is to be created by OverEasy.

FILE Specify the name of the file. This parameter is required.

LIBRARY Specify the name of the library. This parameter is required. The default value will be your **OUTLIB** usage constant.

VOLUME Specify the name of the output volume. This parameter is required. The default value will be your **OUTVOL** usage constant.

- RETAIN** Specify the number of days to retain the file. Retention period is the period during which the file cannot be scratched without explicitly altering the file's expiration date. The VS operating system does not automatically scratch files when the expiration date has been exceeded, however, you may have other disk management software that does. If this field is left blank the expiration date will be set to the current date.
- FILECLAS** Specify the file protection class that is to be associated with the file. This parameter defaults to your FILECLAS usage constant.
- BLOCKS** Specify the number of blocks to be allocated for the output file. This value of this keyword will be set to the best estimate that OverEasy can make. You should avoid modifying it. Increasing the value will usually have no effect other than to require more space while actually reorganizing the file. Decreasing the value will at best cause growth space to be eliminated and possibly fragment the file on the disk. At worst it will cause OverEasy to fail if it reaches the secondary extent limit.
- ACL** This keyword is present when running on OS 7.20 and above. The input file Access Control List is normally copied to the output file. Specify YES to modify that ACL with the standard USERS and GROUPS Getparms.
- SECEXT** This keyword is presented when running on OS 7.32 and above. Specify the secondary extent size that is to be associated with the file as a percentage of original file size. Valid values are in the range of 1 to 99. The default value is 50%.

```

Wang VS GETPARAM v 7
Parameter Reference Name: PRINT
Message ID: 500
Component: OVEREZ
Information Required by OVEREASY
Active Subprogram is OVEREASY
-----
Please specify the reorganization report file.

FILE      = OUTFILE1   LIBRARY = SPOOLLIB   VOLUME = PRTVOL

Should the file be queued for printing?    QUEUE = YES (YES or NO)

If Queue=YES:
  What is the print spooling class?        PRTCLASS = X (A - Z)
  What printer form number should be used? FORM#  = 000 (0-255)

          You may press PF1 to skip the report.

          (c)Copyright 1989-94, Utility Systems, Inc.

```

P
R
I
N
T

G
e
t
p
a
r
m

This Getparm describes the reorganization report file. The file name defaults to the filename specified in the OUTPUT Getparm. Library and volume default to your SPOOLIB and SPOOLVOL usage constants respectively.

- FILE** Specify the name of the report file. Default is the name of the new indexed file specified in the OUTPUT Getparm.
- LIBRARY** Specify the name of the report library. Default is the SPOOLIB usage constant.
- VOLUME** Specify the name of the report volume. Default is the SPOOLVOL usage constant.
- RECORDS** Specifies the estimated number of print lines. Defaults to 80 and should be left at the default value.
- QUEUE** Specify YES if you want the file to be automatically queued to print. Specify NO if the file is not to be queued to the printer.
- PRTCLASS** Specify the printer class which is to be used when the file is added to the system print queue. Defaults to your PRTCLASS usage constant
- FORM#** Specify the form type that is to be associated with the print queue entry. Default value is your FORM# usage constant.

5

ABOUT EGGSHELL**Overview**

You know how powerful OverEasy is for reorganizing your indexed and alternate indexed files quickly and efficiently. EggShell is literally the shell that encases OverEasy, adding another layer of functionality.

The core of EggShell is the power it gives you to build a list of what you need to reorganize — files, libraries, volumes or some combination -- rather than reorganizing one file at a time.

Using EggShell is a straightforward process. You follow these steps:

1. Create a list.
2. Reorganize your files right then.

-- or--
3. Save the list.
4. At a later time, reload the list and run the reorganization.

And at any time you can also use the crash recovery option to reorganize only the indexed files that EggShell finds in a crash state.

All of these options are available from the first EggShell menu, shown here.

```

EggShell: Indexed File Manager 1.02                (c)1992 Utility Systems, Inc.
-----
* Multiple File Reorganization
* Crash Recovery
* Save a list of marked files
* Load a previously stored list of marked files
* Volume and free extent management
* XDMS support: Convert to/from indexed+

Position cursor by option and press ENTER.
-----
ENT Perform selection
16) Exit EggShell

```

You will first learn how to work with a list (creating, saving, and loading) and then how to do the reorganization procedures (multiply file reorganization and crash recovery).

Volume and free extent

management is a new feature with EggShell version 4 that will present usage and free space statistics for all of your disk drives. Consolidation of unused disk space is quick

and easy using the free extent consolidation function which is available from this menu option.

The last option “XDMS Support” appears only if you have XDMS installed on your system. This function allows you to convert files in a list from indexed to Indexed+ and vice-versa.

Working with Lists

You can do basically three things with a list:

- Create a list
- Save a list
- Load a list

Creating a List

Creating a list with EggShell is simply a matter of picking and choosing ... or “marking” and “unmarking” files, in EggShell terms. On your disks, files fall into a hierarchy of volume ... library ... file. In the same way, EggShell takes you through marking the items to include in the list.

To begin, choose the “Multiple File Reorganization” option on the main menu. A screen appears listing all of the volumes on your system. You can then move further to a screen listing the libraries on the volume you select, then on to a screen listing the indexed files in the library you select.

As you move through marking files, libraries and volumes, EggShell highlights them. So you can see at a glance which files will be reorganized with this list. If you mark or unmark an entire volume, it affects all of the libraries and files within. One step down, if you mark or unmark an entire library, it affects all of the files within. You can work just in the volume world, only down at the file level, or anyplace in-between. And you'll probably find yourself combining these levels as you create your list.

Let's see how this works. First, choose the “Multiple File Reorganization” option from the main menu.

```

EggShell: OverEasy Multiple File Reorganization                               Volumes 1 - 11 of 11
-----
Volume      Num  Num  <-----Parameters----->
Name        List type Libs Files Dpck Ipck Apck Grwthp Grwthb 2Xtns MinBks
-----
* 1MEG1          0   0   ***  ***  ***  ***** ***** ***  *****
* 1MEG2          0   0   ***  ***  ***  ***** ***** ***  *****
* SCRACH        18  470  ***  ***  ***  ***** ***** ***  *****
* SYS000        32 1183  ***  ***  ***  ***** ***** ***  *****
* SYS004       348 5505  ***  ***  ***  ***** ***** ***  *****
* SYS005        12  216  ***  ***  ***  ***** ***** ***  *****
* SYS006       151 2736  ***  ***  ***  ***** ***** ***  *****
* SYS041         1   1   ***  ***  ***  ***** ***** ***  *****
* TEMP          8  278  ***  ***  ***  ***** ***** ***  *****
* USITST        58 1903  ***  ***  ***  ***** ***** ***  *****
* WORK          2   2   ***  ***  ***  ***** ***** ***  *****

      System Defaults:          ***  ***  ***  ***** ***** ***  *****

Mark/unmark volumes or press ENTER to view libraries
-----
ENT Select Libs
1) Mark/unmark 1
17) Mark all Volumes
9) Search Volume
12) Reorg Marked Vol
16) Return
    
```

This VOLUMES screen appears, listing all of your volumes (or at least as many as fit on one screen) along with a quick reference of the number of libraries and indexed files in each. There is also a “List type” column,

which is blank now since you haven't yet created a list, and a parameters section. We'll look at the parameters section further on.

You can approach the task of creating a list in basically three ways:

- **Mark a volume for reorganization** (it will be highlighted). EggShell automatically marks all of the libraries and indexed files within it.
- **Select a volume.** You can then mark specific libraries and/or on the succeeding screens (again, EggShell highlights them). This builds an “Include” list (“INCLUDE LIST” appears in the top right corner to remind you).
- **Mark a volume and then unmark for reorganization** specific libraries and files within it (you see the highlighting disappear, meaning those files will be skipped during the reorganization). This builds an “Exclude” list (“EXCLUDE LIST” appears in the top right corner).

There are endless permutations of these possibilities: mark an entire volume, specific libraries on a volume, all of the files in a library excluding one or two, or crate a list that combines all of these specifications. You control the level of detail.

Using the PF keys listed at the bottom of the screen to perform these functions, as explained in the following section. Note that only the PF keys with functions applicable to you will appear on the screen. The exception is PF32, which is always available, even when not listed on the screen.

PF Key Functions

ENTER (Select Libs) Displays a Libraries screen, listing the libraries in that volume when you position the cursor and press ENTER.

PF1 (Mark/unmark1)	Lets you mark or unmark one volume at at time. Simply move the cursor to the volume name and press PF1. (Note that marking a volume automatically marks all of the libraries and indexed files associated with it.)
PF17 (Mark all Volumes)	Marks all of the volumes listed (and so automatically marks all of the associated libraries and indexed files). If one or all of the volumes are already marked, pressing PF17 unmarks them all.
PF2 (First Page)	Displays the first 12 entries in the volume list.
PF3 (Last Page)	Displays the last 12 entries in the volume list.
PF4 (Prev Page)	Displays the previous page.
PF5 (Next Page)	Displays the next page.
PF6 (Prev Line)	Scrolls text down a line.
PF7 (Next Line)	Scrolls text up a line.

PF9 (Search Volume) Searches for a specified volume. This feature is particularly helpful with lengthy volume listings. Pressing PF9 causes the following message to appear in the middle of the screen:

Search for Volume:*****

Type the volume name or a part of it and press ENTER. EGG SHELL displays the next volume name that matches your entry.

PF12 (Reorg Marked Vol) Starts the file reorganization, working on all marked volumes on that screen and below in the hierarchy (see page 3-1 for a detailed explanation).

PF16 (Return) Returns to the previous screen.

PF32 Quickly returns to the main menu, skipping any intervening screens.

Overriding Parameters

You will notice that the right side of the Volumes screen lists several file parameters.

OverEasy allows you to specify data, index and alternate index block packing density parameters, file growth parameters, a secondary extent size parameter (available only with VS/OS release 7.21.05 and later) and a minimum output file size parameter, MINBKS. For files that fluctuate significantly in size, such as transaction files, MINBKS is particularly useful for specifying a minimum file size.

EggShell goes one step further, letting you more finely tune your file by overriding the OverEasy defaults. And you can do so at the file, library, volume or system level (just input the parameter settings in the appropriate screen — file, library or volume).

These levels interact in a hierarchy, so that

- any parameters set in the file screen override any values set on the library screen,
- which in turn override any volume parameters,
- which in turn override any system parameters,
- which then override the OverEasy default parameter settings.

On this Volumes screen you can specify the values for any volume and also the system defaults. After typing a parameter value, you must press a PF key or ENTER to catalog it. If you simply press the PF16 to return to the previous screen, it will not record the setting.

Along those same lines, you should be aware that when you enter these parameter settings on the screen, you are not yet affecting the file. The file does not use those values until you reorganize with EggShell.

The libraries screen appears when you position the cursor by a specific volume and press ENTER.

Library		<-----Parameters----->						
Name	List type	Dpck	Ipck	Apck	Grwthp	Grwthb	2Xtns	MinBks
		INCLUDE LIST						
* #RFNPRT		***	***	***	*****	*****	***	*****
* @SYSDUMP		***	***	***	*****	*****	***	*****
* ATFADDEL		***	***	***	*****	*****	***	*****
* ATFALT		***	***	***	*****	*****	***	*****
* ATFAMANY		***	***	***	*****	*****	***	*****
* ATFBROKE		***	***	***	*****	*****	***	*****
* ATFBUGS		***	***	***	*****	*****	***	*****
* ATFDAMA#		***	***	***	*****	*****	***	*****
* ATFDAMAG		***	***	***	*****	*****	***	*****
* ATFDMSTX		***	***	***	*****	*****	***	*****
* ATFFIX		***	***	***	*****	*****	***	*****
* ATFOLAP		***	***	***	*****	*****	***	*****
Volume Defaults:								
Mark Libraries on Volume USITST or press ENTER to view files.								
ENT Select Files		3) Last Page			7) Next Line			
1) Mark/unmark 1					9) Search Library			
17) Mark all Libraries		5) Next Page			12) Reorg Marked LIBs			
					16) Return			

As you can see, it is almost identical to the Volumes screen, except that it presents a list of libraries. If on the proceeding Volumes screen you marked this volume for reorganization, you'll notice that all of the libraries on that volume are automatically

marked (highlighted).

You have the same sort of options:

- Mark some or all of the libraries (building an Include list)
- Unmark some or all of the libraries (building an Exclude list)
- Position the cursor by a library and press ENTER to go on to mark or unmark specific files
- Do any combination of these options

In the section on the right of the screen, you can set parameters for each library. The parameters in effect, as described in the preceding section, "Overriding Parameters," appear as defaults.

The PF keys work almost identically to those on the Volumes screen, except that they now refer to the library level. Refer to the section entitled "PF Key Functions" for detailed PF key information.

This Files screen appears when you select a library and press ENTER. It lists the indexed files in that library, along with the status of each file (this information is useful if you are doing crash recovery), number of extents, blocks allocated and blocks used.

```

EggShell: OverEasy Multiple File Reorganization                               Files 1 - 12 of 16
-----
File          Num Blks   Blks  <-----Parameters----->
Name         Status Xts Alloc  Used  Dpck Ipck Apck Grwthp Grwthb 2Xtns MinBks
-----
* FUBIG01A  CLOSE 1    10     9    100 100  ***  *****  *****  ***  *****
* FUMED02A  CLOSE 1    15    14    100 100  ***  *****  *****  ***  *****
* FUMED03A  CLOSE 1    22    21    100 100  ***  *****  *****  ***  *****
* FUSML03A  CLOSE 1    10     9    100 100  ***  *****  *****  ***  *****
* FUSML04A  CLOSE 1    18    17    100 100  ***  *****  *****  ***  *****
* FUSML08A CLOSE 1    13    12    100 100  ***  *****  *****  ***  *****
* VCALL01A  CLOSE 1     6     5    100 100  ***  *****  *****  ***  *****
* VCBIG04A  CLOSE 1    19    18    100 100  ***  *****  *****  ***  *****
* VCMED04A  CLOSE 1    17    16    100 100  ***  *****  *****  ***  *****
* VCSML01A  CLOSE 1     7     6    100 100  ***  *****  *****  ***  *****
* VUALL01A  CLOSE 1    24    23    100 100  ***  *****  *****  ***  *****
* VUBIG04A  CLOSE 1    20    19    100 100  ***  *****  *****  ***  *****

Library Defaults:

Mark files in Lib ATFALT on Vol USITST or press ENTER to show file stats.
-----
ENT Show File Stats          3) Last Page          7) Next Line
 1) Mark/Unmark 1           5) Next Page         9) Search File
17) Mark all files          12) Reorg Marked File
                             16) Return
    
```

You have the same options:

-
-
-
-

In the section on the right of the screen, you can set parameters for each file. The parameters

in effect, as described in the preceding section, “Overriding Parameters,” appear as defaults.

The PF keys work almost identically to those on the Volumes screen, except that they now refer to the file level. Refer to the section entitled “PF Key Functions” for detailed PF key information.

The File Statistics screen, shown here, appears when you select a file and press ENTER. EggShell gathers these file statistics from VS/OS system services, providing them for easy reference.

```

EggShell: OverEasy Multiple File Reorganization
-----
Statistics for Volume: USITST      Library: ATFALT      File: FUSML08A
-----
File Organization: Indexed      Record Size:      12
Blocks Allocated:  13          Record Type:      Fixed/Uncompressed
Blocks Used:       12          Primary Key Pos:  1
Extents:           1           Primary Key Len:  4
Records:           250         DMS/TX Recov Blks: None
Owner:             PAL          FP Class:         (Blank)
Secondary Alloc:   6

4 Alternate Indices are Present

Index#  Pos  Len  Dups?  # Entries      Index#  Pos  Len  Dups?  # Entries
-----  -  -  -  -  -  -----  -  -  -  -  -  -----
  1     1   1   YES     250           1     1   1   YES     250
  2     2   2   YES     250           2     2   2   YES     250
  3     3   3   YES     250           3     3   3   YES     250
  4     4   4   YES     250           4     4   4   YES     250

-----
16) Return

```

This screen simply presents information — you cannot input anything.

**S
a
v
i
n
g
a**

List

When you are satisfied with what you have marked for your list, you are ready to save it. Doing so means you never have to make those decisions again. Select the “Save a list of marked files” option from the main menu. (To quickly return to the main menu, press PF32.)

This window appears in the middle of the screen after you choose the save option:

```

                Save a marked list
Please enter a name for the list.
-----
Name:*****

```

Type a 1 to 8 character filename for the list, then press ENTER. (If a default appears, it is the name of the last list that you loaded during the session.) Your list is now stored in EggShell, ready for later use.

To replace an existing list, enter the same filename. EggShell displays this window:

```
                Save a marked list
-----
List ***** already exists.
Press ENTER to replace, PF16 to return.
```

If you press ENTER, EggShell scratches your old list and the new list takes that name. If you don't wish to replace that list, press PF16 to return to the window to specify a different name.

NOTE: To delete a list entirely, you must use the VS Manage Files and Libraries utility. The lists are stored as files in the library @EGGLST@ on the system volume.

But what happens when you've saved a list, and then new files, libraries or volumes are added to your system? EggShell handles that situation by looking at what you marked or unmarked for the next higher level — and marking it the same way.

- For example, suppose you marked volume SYS001 for reorganization. If you add library TEST to that volume, EggShell will automatically mark that library (and all of the files within it) for reorganization.
- Now suppose you had marked volume SYS001 for reorganization, but excluded library USER. If you add a file to USER after saving the list, EggShell will automatically exclude that file, using the characteristics of the library.
- And of course you can check this by going to the appropriate EggShell screen and seeing whether the file, library or volume is highlighted.

Loading a List

You've created a list and saved it. Loading that list recalls the markings so that you can make changes or run the file reorganization.

Select the "Load a previously stored list of marked files" option on the main menu. This window appears in the middle of the screen:

```

Load a marked list.
Enter the name of list to be loaded.
-----
Name: *******
```

Type the list name, then press ENTER. If you cannot remember the name of your list, you can find it in library @EGGLST@ on your system volume.

EggShell loads the markings (or unmarkings) of that list. Then, when you select either "Multiple File Recovery" or "Crash Recovery," you will see the appropriate highlighted markings on the Volumes, Libraries and Fields screens. You may then update the list or run the reorganization.

If you have loaded the file to make changes, remember to save that changed list before leaving EggShell.

Note that if you were interactively marking files before choosing the Load option, the following will appear:

```

Load a marked list
New list will replace currently marked entries.
-----
* Save currently marked entries.
* Discard currently marked entries.
```

This gives you a chance to save those entries, if you wish, before loading a different list. Position the cursor at the desired option and press ENTER.

Reorganizing Files

When you reorganize files through EggShell, you are actually using the OverEasy utility. But instead of specifying each file individually, you can load a list that you have built, and EggShell takes it from there.

EggShell has two reorganization options:

- Multiple file reorganization
- Crash recovery

They operate identically, except that crash recovery is a more selective reorganization.

This section also discusses delaying the reorganization until a later time, the EggShell-OverEasy Reorganization Report, cancelling a reorganization and reorganizing files from a command procedure.

Multiple File Reorganization

Now you've gotten to why you're working with lists in the first place. You've marked the files, libraries and volumes that you want to reorganize (either interactively just now or by loading a saved list) and you're ready to start the reorganization.

You can start the reorganization on either the Volumes, Libraries or Files screen. But note that you will get different results depending on which screen you choose. EggShell again uses the hierarchy of file structure to give you more control over which files are reorganized. It reorganizes all of the files marked on the screen you are running the file reorganization from, plus any marked files on levels lower in the hierarchy.

In practice, this concept translates into the following:

- Pressing PF12 (Reorg Marked Vol) on the *Volumes* screen reorganizes all of the marked volumes, and by default the associated marked libraries and marked files. So running it from the Volumes screen takes into account *your entire list*.
- Pressing PF12 (Reorg Marked Lib) on the *Libraries* screen reorganizes all of the marked libraries and by default the associated marked files on *just the volume you selected on the previous screen*.
- Pressing PF12 (Reorg Marked File) on the *Files* screen reorganizes all of the marked files on *just the library you selected on the previous screen*.

This window appears when you press PF12:

```

Please select when to start file reorganizations
-----
* Start file reorganizations immediately
* Specify a time to start file reorganizations
    
```

Select the first choice to start the reorganization of the marked files immediately. As the reorganization runs, information on the number of reorganizations attempted, the number of successes and the number of errors scrolls on your screen, as seen in the following sample screen.

Selection of the second choice will cause the WAITER program (described in Appendix A) to run. WAITER allows you to specify the start time for the reorganization to run.

```

EggShell: Indexed File Manager 1.02                (c)1992 Utility Systems, Inc
-----
Currently: Reorganizing file on volume
Volume Library File                Dpck Ipck Apck Grwthp Grwthb 2Xtns MinBks
-----
SYS004 ALTDB  CISDBTE                85  95      75

Completion Report
Volume Library File                Blocks Before Reorg   Blocks After Reorg   Rtn
Allocated      Used      Allocated      Used      Code
-----
SYS004 ALTDB  CISDBAX                224   221     387     221     0
SYS004 ALTDB  CISDBCU                307   303     531     303     0
SYS004 ALTDB  CISDBDE                488   483     846     483     0
SYS004 ALTDB  CISDBLE               1184   758    1311     749     0
SYS004 ALTDB  CISDBSE                608   602    1054     602     0

Press PF1 to interrupt.
Searched: Files(5)  Reorgs: Attempted(5) Success(5) Errors(0)
    
```

The reorganization screen displays the volume, library and file being reorganized, along with any parameter values that you set at the file, library, volume or system level.

It also shows you the number of disk blocks

that were allocated and used both before and after the reorganization. This information lets you see the effect of the parameter settings on the file. You can then fine tune the file parameters to optimize the file size for the next reorganization.

Any error messages also appear, along with a return code. The return code indicates the success of an OverEasy/EggShell run or the reason for failure. Section 6 provides a complete list of return codes and their meanings.

The following window appears when your reorganization is done:

```
Multiple File Reorg completed. RC=0
Report file is EGG0004 in PRT on SYS004.
-----
* Return to EggShell
* Scratch the report file
* Display report file
* Print report file
```

This window presents several options for working with the EggShell-OverEasy Reorganization Report or you can choose to return to EggShell. Refer to the EggShell-OverEasy Reorganization Report section, page 3-5, for details on the report itself.

Position the cursor by the option you want and press ENTER.

Delaying the Start of the Reorganization

As explained in the previous section, when you press PF12 a window appears with choices for starting the file reorganization. Usually you will select “Start file reorganizations immediately.”

However, you could also specify another time for the file reorganization to begin. To do so, select “Specify a time to start file reorganizations.” A screen appears on which you specify a start date and time. Once you have done so, the program waits until that time and then runs the reorganization. Refer to the WAITER program documentation in Appendix A for more information about this feature.

Crash Recovery

When your system crashes, you know that files may be lurking in the system in a crashed state. But until you open an application that uses a file affected by the crash, you don't know which need to be reorganized.

EggShell takes that guesswork out of crash recovery. The crash recovery option operates identically to multiple file reorganization -- but with one vital difference. It searches for and reorganizes only the indexed files in your list that are in a crashed state.

This lets you reorganize immediately only the files that need it. And you can tell how many files are crashed, and their sizes, by looking at the Files screen to see which show a status of "CRASH."

Follow these steps to run the crash recovery:

1. Interactively mark the files, libraries and volumes to reorganize, or load a previously saved list.
2. Select the "Crash Recovery" option from the main menu.
3. Press PF12 on the screen that you want the reorganization to start from (either the volume, library or file level).

Refer to the preceding Multiple File Reorganization section for details, as it runs identically.

Reorganizing Files from a Procedure

You can also reorganize files with EggShell from within a command procedure, using the EGGREORG program. This allows you to reorganize files in background mode.

EGGREORG issues the following Getparm requesting the name of a previously saved EggShell list and whether you want to reorganize only crashed files. It operates just as though you had chosen PF12 from the EggShell Volumes screen.

```

Wang VS GETPARAM v 7                               Parameter Reference Name: LISTNAME
                                                    Message Id: 7000
                                                    Component: EGGORG

Information Required by EGGREORG
Active Subprogram is EGGREORG

-----

What is the name of the EggShell list to be reorganized?  LISTNAME = *****
Do you want to reorganize only crashed files?           CRASH   = NO*
XDMS users, to convert files to/from Indexed+ you may  FILEORG = *
enter an overriding file organization.
Enter N to force Indexed+, I to force Indexed.

PF16 to terminate

EggReorg: Indexed file reorg manager Vers: 1.02 (c)1992 Utility Systems, Inc.

```

The LISTNAME screen parameters and valid responses are described below:

LISTNAME

CRASH

FILEORG

-
S
a

Example Command Procedure

```

Procedure
  Run EGGREORG
  Enter LISTNAME Listname=WEEKLIST, Crash=NO
  Return

```

If you wish to start the reorganization at a specific time, call the WAITER utility supplied with EggShell as the first step of your command procedure. (See the WAITER documentation in Appendix A for details.)

EggShell-OverEasy Reorganization Report

When the reorganization completes, you can look at the EggShell/OverEasy Reorganization Report on-line or print a copy. This window appears if you choose to print the report file:

```

Press ENTER to print the report
or PF16 to return
-----
          Print Class:  X
          Form Number: 000
    Number of Copies:  1****
    Scratch after print? YES
  
```

The defaults that appear are from your usage constants. You can change any of the field values. When you press ENTER, EggShell queues the report to print and returns you to the EggShell main menu.

The report contains all of the information that you saw on-line during the reorganization or crash recovery (including detailed error messages), along with the following statistics:

File Data	The filename, library, volume and organization (I for Indexed, N for Indexed+)
Record	The type (fixed or variable), compression (yes or no) and size of the record.
Packing	The Dpack, Ipack, and Apack parameters used.
Growth	The Growthb and Growthp parameters used (either the defaults of OverEasy or the override values specified in EggShell).
Blocks	The number of blocks allocated and used, shown both before and after the reorganizations.
Key	Primary data key and alternate indexed structure information, including the position of the start of the key in the record, the key length and whether duplicates are allowed (yes or no).
Data	The number of records and blocks for the prime data key and alternate indexes.

EggShell stores the report in the current spool library on the current spool volume. If EggShell runs interactively, the report is named EGGS##### (EggShell generates the number, #####). If EggShell runs as a background job, the report is named EGGR#####. These reports should be purged periodically.

Stopping the Reorganization

Normally, the EggShell file reorganization simply runs and completes. However, if you do find partway through an EggShell reorganization that you need to stop it, the procedure is simple.

To interrupt the reorganization, press PF1. (Do not cancel using the HELP/PF16 keys.) The following window then appears in the middle of your screen. You must specify what to do about the interrupted file reorganization.

```
The current file reorg has been interrupted
```

- ```

* Resume the current file reorganization
* Abort the current file reorganization
```

You can continue reorganizing the current file or abort the reorganization immediately. Position the cursor by your choice and press ENTER.

Once you've specified what to do about the reorganization of the current file, EggShell prompts you with the following window for what to do with the rest of the marked files that have not yet been reorganized. (Note that you will see only this window and not the preceding one if EggShell was in between files when interrupted.)

```
The current file reorg will be resumed [aborted]
Then...
```

- ```
-----  
* Continue with the next FILE  
* Continue with the next LIBRARY  
* Continue with the next VOLUME  
* Cancel remaining reorganizations
```

You may have interrupted the reorganization for a variety of reasons. EggShell gives you a variety of ways to proceed. You can cancel the reorganization completely or move on to the next file, library, or volume. Position the cursor by your choice and press ENTER.

Checkpoint Recovery

EggShell was designed to keep your indexed files intact should the system crash while a reorganization is in progress.

When you re-IPL the VS after a system crash during which EggShell was reorganizing files, run EggShell again immediately. It will prompt you to initiate recovery if needed. If so, you must run the EggShell recovery before EggShell will allow any other reorganizations.

During the recovery, EggShell checks each file that was under reorganization when the system failed, prompting you for input when it finds one that needs to be recovered. Upon completing the necessary cleanup, it displays a window listing any errors it encountered. EggShell proceeds normally when you press ENTER.

During the recovery EggShell will complete all reorganizations which were in progress when the system crashed. No one else can use EggShell while the recovery is in progress.

Volume and Free Extent Management

The volume and free extent management selection provides a view of all the disks mounted on the system with usage and free space statistics.

EggShell Volume Analysis - Copyright 1993-94, Utility Systems, Inc.									
Volume	Size	Total Libs	Total Files	% Used	Free Blocks	Free Xtnts	Max Blks New File	Largest Free	Second Free
* OFFICE	102 Mb	13	1,480	67	16,436	9	15,876	15,206	304
PACE01	319 Mb	504	8,483	84	24,909	32	20,067	14,078	5,208
PACE02	1.3 Gb	237	3,307	58	266,601	1	266,601	266,601	
SCRACH	319 Mb	57	765	47	98,129	11	93,208	85,985	5,428
SYSTEM	319 Mb	143	3,647	87	20,249	48	19,389	18,894	365
TEST	142 Mb	92	2,188	75	17,333	15	17,287	17,113	114
WORK	319 Mb	0	0	1	154,204	2	154,204	84,857	69,347

ENT Show/Consolidate ext
 1) Refresh Screen
 16) Return

% Used:

Max Blks New File:

Largest Free:

Second Free:

Select any volume by positioning the cursor and press enter to show all of the unused extents on the volume and to access the extent consolidator for that volume.

Free Extent Analysis

Select a volume from the Volume Analysis screen and press enter to view the Free Extent Analysis for that volume.

EggShell Free Extent Analysis - (c)Copyright 1993-94, Utility Systems, Inc.								
Volume SYSTEM contains 21349 free blocks in 48 extents.								
Size	First	Last	Size	First	Last	Size	First	Last
9,630	145,541	155,170	20	89,106	89,125	5	89,039	89,043
4,058	103,722	107,779	18	101,913	101,930	5	45,827	45,831
3,405	122,751	126,155	18	89,081	89,098	3	57,922	57,924
1,815	109,294	111,108	17	90,557	90,573	3	41,167	41,169
944	126,165	127,108	15	89,335	89,349	3	36,785	36,787
499	102,714	103,212	14	95,611	95,624	3	67,416	67,418
326	98,210	98,535	12	42,692	42,703	3	69,682	69,684
273	101,531	101,803	11	42,597	42,607	2	45,809	45,810
68	95,651	95,718	11	42,626	42,636	2	58,369	58,370
34	96,327	96,360	9	42,667	42,675	2	33,289	33,290
33	89,257	89,289	8	74,414	74,421	2	33,051	33,052
25	89,050	89,074	7	95,630	95,636	1	93,299	93,299
21	89,023	89,043	7	42,667	42,675	1	87,091	87,091
21	90,633	90,653	6	74,414	74,421	1	3,756	3,756

ENT Refresh	4) Prev Page	12) Consolidate Extents
2) First Page	5) Next Page	16) Return
3) Last Page	6) Prev Line	
	7) Next Line	

The free extents on the volume are listed in descending order of size. Use the standard PF2–PF7 keys when all of the free extents do not fit on the screen.

Free Extent Consolidation

To initiate free extent consolidation, press PF12 from the Free Extent Analysis screen.

The extent consolidator moves and consolidates the free space by moving selected files on the volume. If a print file is moved that is queued to print, it will be removed from the print queue. A pop-up menu asks you if print files should be moved. We suggest that you specify YES for maximum efficiency.

The consolidator will run until it reaches a point of diminishing returns, typically when the number of free extents has been reduced to less than 10. Progress will be constantly updated on the screen and you may stop the consolidator via PF1 at any time. You may also limit the time that the consolidator may run by specifying the hours and minutes to run on the pop-up menu that is presented during startup. To run the extent consolidator from a procedure, see the description of EGGCIP in Appendix B.

6**RETURN CODES****OverEasy Return Codes**

Return codes are used to inform a linking program of the success of OverEasy or the reason for failure. Procedures or linking programs should always check the return code before proceeding to use the newly created file. Non-zero return codes indicate that some abnormal condition occurred. Lower values indicate that the new file was created but that there were minor problems. In general, higher values mean more severe problems.

- | | |
|----|--|
| 0 | Successful reorganization, no exceptions encountered. |
| 4 | Successful reorganization. Record count of input file did not agree with FDR1 record count. The estimated and actual record counts are available in the reorganization report. Minor discrepancies in record count are common on VS systems. Larger discrepancies should be justified by the history and use of the particular file involved. |
| 8 | Unsuccessful. Duplicate keys were detected for an alternate path in which duplicates are not allowed. When this error occurs during foreground processing, the offending alternate key and associated primary key values are displayed. |
| 12 | Unsuccessful. Output file size was underestimated and file could not be extended. The most common cause of this error is that there is erroneous information in the FRD1 or AXD1. Try overriding the BLOCKS keyword in the OUTPUT getparm screen with the number of blocks currently allocated to the input file. |
| 16 | Unsuccessful. Program integrity halt. Print the halt file in library OVERDIAG on the system volume and contact USI for assistance. |
| 20 | Unsuccessful. Insufficient modifiable data area to perform reorganization. Have the systems administrator increase the size of your MDA, logoff and logon, then retry OverEasy. |
| 24 | Unsuccessful. User cancelled via PF-16. |
| 28 | Unsuccessful. Disk I/O error on file other than input file. |
| 32 | Unsuccessful. Primary key sequence error. When converting a consecutive file to indexed, this condition probably means that the input file must be sorted and/or duplicate keys removed. When this condition occurs with indexed input, the primary data chain of the input file is probably corrupted. Run verify against the input file if there is any doubt and proceed with your alternate recovery strategy. |
| 36 | Unsuccessful. Disk I/O error on input file. |

-
- | | |
|----|---|
| 40 | Unsuccessful. OverEasy program expiration date has been exceeded or this copy of OverEasy is not configured for your system. Run OverEasy in foreground to obtain more precise information. |
|----|---|

EggReorg Return Codes

A return code indicates the success or reason for failure of a particular file reorganization. Here are the possible values and their associated conditions:

- | | |
|-----|---|
| 0 | All file reorganizations completed successfully. |
| 2 | All file reorganizations completed successfully. Crashed DMS/TX files were encountered. Suggest DMS/TX Database Recovery be run. |
| 4 | One or more file reorganizations completed with exceptions. Check the EggShell–OverEasy Reorganization Report for a completed message. |
| 6 | One or more file reorganizations completed with exceptions. Check the EggShell–OverEasy Reorganization Report for a complete message. Additionally, crashed DMS/TX files were encountered. Suggest DMS/TX Database Recovery be run. |
| 16 | User aborted the file reorganization through the “Cancel remaining reorganizations” option. |
| 32 | User pressed HELP/PF16 to abort the file reorganization. |
| 110 | EggShell recovery needed. Run EGG SHELL to recover an interrupted file reorganization. |

7 GETPARM QUICK REFERENCE

This section is a quick reference guide for use when writing procedures. Detailed information about any of the Getparms or keywords is provided in section 4.

OverEasy Getparms — Quick Reference

PRNAME	KEYWORD	LEN	TYPE	DEFAULT	PERMITTED VALUES
INPUT	FILE	8	AN		
	LIBRARY	8	AN	INLIB	
	VOLUME	6	AN	INVOL	
	MODE	7	AN	INPUT	INPUT, SHARED or SPECIAL
LOCK	LOCK	3	AN	NO	YES or NO
	TIMEOUT	3	AN	10	0-255 or NO
	BYPASS	3	AN	NO	YES or NO
OPTIONS	REPORT	3	AN	YES	YES or NO
	FILEORG	1	AN	from input file	C, I, N or U
	RECTYPE	1	AN	from input file	F or V
	RECSIZE	4	INT	from input file	DMS maximum
	COMPRESS	3	AN	from input file	YES or NO
	KEYPOS	4	INT	from input file	<= RECSIZE
	KEYLEN	3	INT	from input file	< 255 and within record
	DPACK	3	INT	from input file	10-100
	IPACK	3	INT	from input file	10-100
	APACK	3	INT	same as IPACK	50-100
	GROWTHB	5	INT		0-99999
	GROWTHP	5	NUM	1.0	any positive real number
	RECBK	1	AN	from input file	N, A, or U
	ALTINDEX	3	AN	NO	YES or NO

OverEasy Getparms — Quick Reference (contd.)

PRNAME	KEYWORD	LEN	TYPE	DEFAULT	PERMITTED VALUES
ALTINDEX	KEYPOS01- KEYPOS16	4	INT	from input file	
	KEYLEN01- KEYLEN16	3	INT	from input file	
	DUPSOK01- DUPSOK16	3	AN	from input file	YES or NO
	APACK01- APACK16	3	INT	APACK option	50-100
OUTPUT	FILE	8	AN		
	LIBRARY	8	AN	OUTLIB	
	VOLUME	6	AN	OUTVOL	
	RETAIN	3	INT		0-999 or blank
	FILECLAS	1	AN	from input file	A-Z, #, \$ or blank
	BLOCKS	8	INT	from OverEasy	
	ACL	3	AN	NO	YES or NO
	SECEXT	3	INT	50	1-100
PRINT	FILE	8	AN		
	LIBRARY	8	AN	SPOOLIB	
	VOLUME	6	AN	SPOOLVOL	
	QUEUE	3	AN	YES	YES or NO
	PRTCLASS	1	AN	PRTCLASS	A-Z
	FORM#	3	INT	FORM#	

A

APPENDIX A

WAITER Program

The WAITER program allows you to delay the start of a program or procedure until a specified time and date in the future.

```

Wang VS GETPARM v 7                               Parameter Reference Name: INPUT
                                                    Message Id: 100
                                                    Component: WAITER

Information Required by EGGSHELL
Active Subprogram is WAITER

-----
WAITER - Version 1.01

Please specify the amount of time for this program to wait before terminating

The current date and time is: 04/21/92 15:21.

Specify the end of the wait period as WAITDATE (MMDDYY) and WAITTIME (HHMMSS)

WAITDATE = 042192      WAITTIME = ****

Press ENTER to wait, or
Press PF13 to exit without waiting, or
Press PF16 to terminate.

(c)Copyright 1991-92, Utility Systems, Inc.

```

-
O
p
e
r
a
t
i
o
n

When invoked,
the WAITER

program requires you to specify a date and time in the future until which you want to delay the start of a reorganization. These two parameters are specified as the keywords WAITDATE and WAITTIME. EggShell will suspend operation until the VS time-of-the-day clock crosses the date and time boundary specified by the wait.

After the WAITDATE and WAITTIME fields have been filled in, pressing the ENTER key causes the wait to commence. The wait may be prematurely terminated by pressing PF13 at any time during the wait. Pressing PF16 also causes a premature termination of the wait, but also sends an abort indication (different return code) to the calling program or procedure.

The format for WAITDATE and WAITTIME are as follows:

WAITDATE yymmdd

WAITTIME hhmm (in 24-hour format)

WAITER Return Codes

0 Normal completion at end of wait period. The system time-of-day clock has passed the date and time specified.

- 4 Forced completion before end of wait period. The user pressed PF13 to terminate the wait period prior to the date and time specified.
- 16 Abort. The user pressed PF16 to terminate.

Return codes 4 and 16 both indicate a premature termination of the wait. However, the different return codes may be used by the calling program or procedure to distinguish between the user's intent to continue with the next step in the job stream, or to abort the entire job.

Sample Procedures

```
Procedure
  Run WAITER
  Enter INPUT Waitdate="041292",Waitempty="1700"
  .
  .
```

The procedure above waits until 5:00 p.m. on April 12, 1992 before continuing with the remaining procedure steps.

```
Procedure
WTR: Run WAITER
  Enter INPUT Waitdate="062392",Waitempty="0900"
  If (WTR=16) Return Code = 99 [User Aborted]
* WAITER terminated with return code 0 or 4
EGG: Run EGGREORG
  Enter LISTNAME Listname=DATABASE
  Return Code = EGG
```

This procedure invokes WAITER to delay the start of a reorganization run until 9:00 a.m. on June 23, 1992. If the user aborts WAITER by pressing PF16, this procedure terminates immediately, without running EGGREORG, and returns a code of 99 to the caller of this procedure. If the wait terminates normally or if it is forced to completion by pressing PF13, EGGREORG is run and the return code from EGGREORG is passed to the caller of this procedure.

B

APPENDIX B

EGGCIP Program

The EGGCIP program allows you to run the extent consolidator as part of a procedure, in either background or foreground.

```

Wang VS GETPARAM v 7                                Parameter Reference Name: INPUT
                                                    Message Id: 100
                                                    Component: EGGCIP

Information Required by EGGCIP
Active Subprogram is EGGCIP

-----
Please specify the volume and run options.

OverEasy/EggShell Free Extent Consolidation

Enter the volume name of the volume to compress.    VOLUME = *****
You may set a time limit for the consolidation.     MAXTIME = **** (HHMM)
Do you want to move print files?                   MOVEPRT = YES
YES allows the consolidator to do a better job
but may cause print files that are queued to
print to be removed from the queue.

(c)Copyright 1994, Utility Systems, Inc.

```

-
O
p
e
r
a
t
i
o
n

The EGGCIP program consolidates free

extents in exactly the same way as EggShell Free Extent Consolidation. The main difference is that this method of using the consolidator may be invoked from a procedure and does not have an interactive screen display while running.

The keyword fields for the INPUT Getparm screen are as follows:

- VOLUME** The volume name of the volume to be consolidated. This parameter is required.
- MAXTIME** The maximum time that the consolidator is to run against this volume. The consolidator will terminate automatically when the free extents have been reduced to a reasonable number, typically less than 10. Specify a time limit in hours and minutes to force completion within the specified time. Leave blank (or zeroes) for automatic termination. Minimum time limit is 5 minutes. The default maximum runtime is 8 hours.
- MOVEPRT** The extent consolidator reorganizes the free extents by moving files on the volume. If it moves a print file which is queued to print, that file will be removed from the print queue. The consolidator works most efficiently when print files can be moved but you may specify NO to this keyword to suppress moving of print files. The default value is YES.

EGGCIP Return Codes

0	Normal completion. The consolidator was invoked and run to completion.
16	Abort. PF16 was entered in response to the INPUT getparm.

Sample Procedure

Following is a simple procedure to run EGGCIP on volume "SYSTEM" for one hour and twenty minutes, allowing printfiles to be moved.

```
Procedure
CIP: Run EGGCIP
  Enter INPUT
  Volume=SYSTEM,MAXTIME=0120,MOVEPRT=YES
  Return Code=CIP
```

