Oracle® GoldenGate for Base24

Administration Guide

Version 3.0

October 2009

Oracle® GoldenGate for Base24 Administration Guide, version 3.0

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.This documentation is in prerelease status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Contents

CHAPTER 1

Introduction to GoldenGate for BASE24

. . . .

This chapter introduces GoldenGate for BASE24 features and capabilities in the following topics:

Contents

GoldenGate for BASE24 overview

GoldenGate for BASE24 offers comprehensive data movement and management solutions for the ATM and POS electronic funds transfer (EFT) application from ACI Worldwide. GoldenGate for BASE24 enables users to continuously capture data from BASE24 / HP NonStop systems to one or more secondary systems, which supports the following business initiatives:

- Achieve **Business Availability** objectives and implement comprehensive disaster recovery plans.
- Ensure smooth **Application Integration** by linking together legacy and web-enabled systems, and integrate customer applications with core transactional systems.
- Improve **Business Visibility** by closing the batch window on end-of-day account and inter-bank funds settlements.
- Achieve **Compliance** and audit regulations cost effectively.

Add-on modules

You can extend your GoldenGate for Base24 functionality through custom add-on modules including:

- **D24 - Dual site processing for high availability**

  Allows transactional data to be synchronized bi-directionally in real time throughout the day. In the event of an outage on one system, the full transaction load will be processed on the remaining machine, ensuring continuous availability.

- **M24 – Zero downtime system migrations**

  Enables system administrators to load new databases while the application remains active on the production system.

- **E24 – Super extract real-time feed to host**

  Automatically captures, transforms, and delivers BASE24 transactions in real-time to settlement systems, including platforms based on Oracle Sybase, MySQL and other databases. This reduces the overall batch processing time previously required and data no longer sits on tape for hours at a time.

- **N24 – Notification of refreshed files**

  Coordinates the notification message to all BASE24 satellite processes once the file refresh process has completed. Coordination is available for bi-directional configurations as well as uni-directional without the need to modify BASE24.

- **T24 - Moving tokenized and segmented data into a structured format**

  Resolves problems with moving tokenized or segmented data to other platforms and databases. Reorganizes and reformats the transaction log (TLF and PTLF) tokens for all transaction types into a configurable order. Also re-organizes and re-formats PBF and CAF segments into a defined flat structure that can be propagated into any supported database.

Document scope

This guide is a supplement to the GoldenGate for HP NonStop Operations Guide. It provides an overview to GoldenGate for BASE24 processing, and focuses on how GoldenGate manages BASE24 transactional data. BASE24 implementation and operating instructions are detailed in the ACI BASE24 documentation.

Except where stated, all references to BASE24 are for BASE24 6.0 version 4 and XPNET 3.0. In some instances, examples for BASE24 5.3 are included.

GoldenGate for BASE24 processing

Three methods of capturing files are used in BASE24: one for audited Enscribe TMF files, one for non-audited Enscribe files, and one for directly extracting Enscribe entry-sequenced data.

GoldenGate for BASE24 manages transactional data in different ways depending on the record type being sourced. For TMF audited files, the applications make changes to the database. For non-audited files, a GoldenGate intercept library (SKELBN for BASE24) is bound into the BASE24 application so records can be transmitted to a Logger process.

Regardless of how messages are transmitted, all messaging between GoldenGate processes or libraries is done in a 'nowait' manner. This means that GoldenGate does not delay the user application. Using log/audit trails (i.e. files) to hold record and file operations and data means that no data is lost in the event communication is lost between the source and target site, or a GoldenGate component process abends. The component can always be re-positioned to recover data that might have been lost.

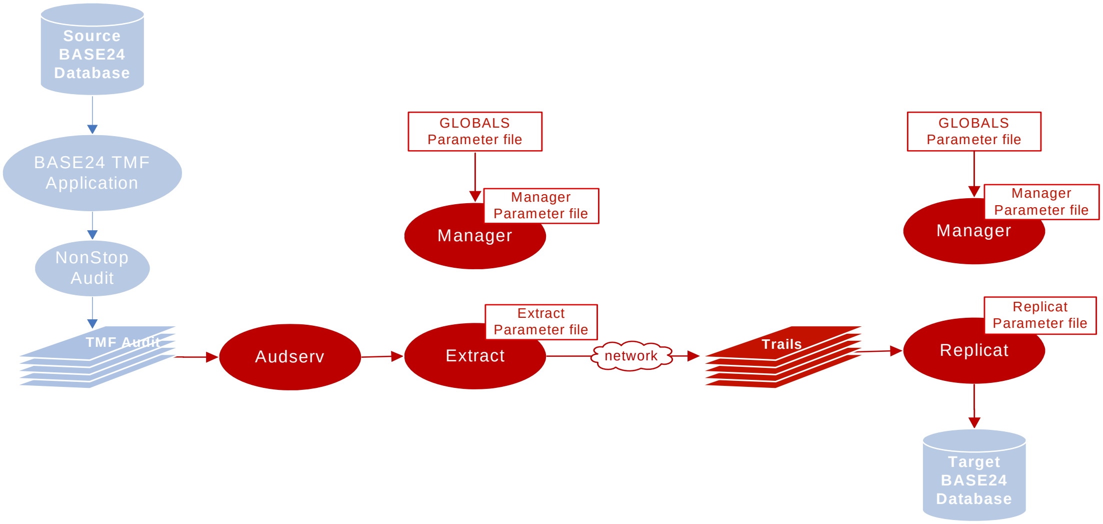
Capturing data changes from TMF applications

Capturing TMF audit from BASE24 TMF audit trails provide the central resource for retrieving database changes in TMF-enabled applications. Changes to these files and SQL tables are recorded in TMF audit trails for transaction integrity and recoverability. Figure 1 shows the processing flow for TMF-audited applications.

Extract and Audserv work together to retrieve and process BASE24 database changes. When started, Extract starts an Audserv process to read audit trails from their original location on disk, from a disk or tape dump, or from a user-specified alternate location. Audserv also determines the location of all required audit.

Note Audserv can only return data accessible to the user who has PROGID'd and licensed the application.

Figure 1 GoldenGate processing – TMF audit files



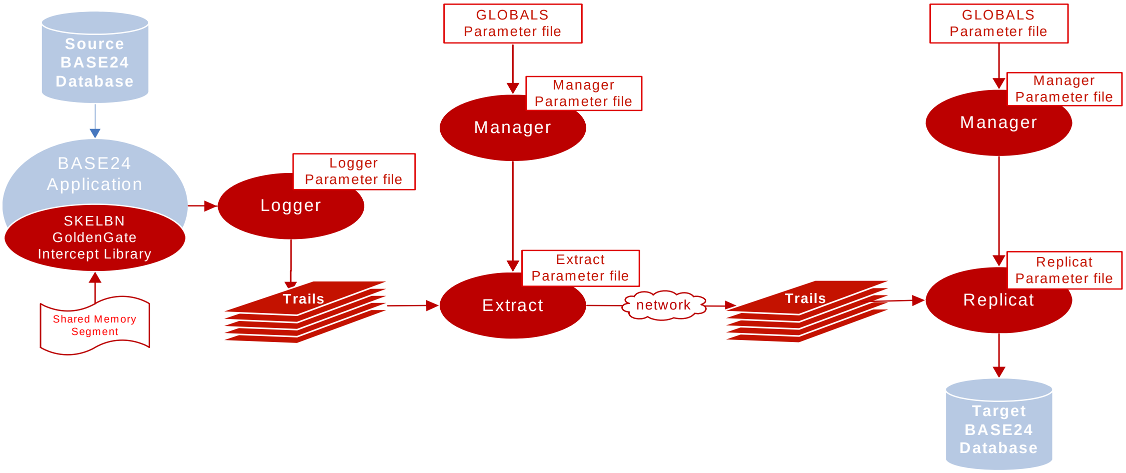
Database changes include insert, update or delete operations, along with transaction metadata. Insert and update records are after images, or the format of the database record after the operation completes; delete records are returned as before images. Audserv will also return before images for updates.

Extract saves each image in memory until an associated transaction commit record is received. If the transaction aborts, the associated records are discarded. Committed records can be output to one or more user-designated files called trails.

Capturing data changes from non-TMF applications

In some instances, customers do not generate TMF-audited files in BASE24. To capture non-audit records, GoldenGate binds itself to the BASE24 application, and captures data via the Logger component. Logger creates a trail that is read by Extract, which then continues to process the data as usual. Figure 2 displays the processing flow.

Figure 2 GoldenGate processing - non-TMF



To make non-TMF data captures possible, GoldenGate binds in a special library designed to work with BASE24 called SKELBN. This library is a group of functions with the same names as the Guardian operating system and BASE24 application procedures.

For example, when a BASE24 function is called, SKELBN performs the task. The application is unaware of the substitution and executes, from an application programming standpoint, exactly as it did before.

If the function succeeds, it sends its data to Logger, which writes it to a log trail. Extract reads the log trail and writes the update to a remote trail on the target site. Replicat reads the updates from the trail file and applies the trail records to the target database.

Direct extract processing

This capture method is limited to Enscribe entry-sequenced files that are only inserted and not updated. In BASE24 this applies to the transaction log files (i.e. TLF, PTLF and TTLF files). This method does not require Logger or any intercept libraries. Once Extract process reads the record, processing is the same for all methods.

CHAPTER 2

Installing GoldenGate for Base24

This chapter outlines the steps required to successfully implement GoldenGate for BASE24. These include:

Contents

Determining requirements

Before installing GoldenGate for BASE24, you must make a number of decisions. These include determining the files you wish to replicate, your data management environment, and how replication impacts your source system. Your business needs influence these decisions; ensure you have clearly defined objectives and uses for your replicated data. For example, disaster recovery needs differ from hot backup needs, which differ again from data warehousing and reporting needs. Once you have clarified the purpose of your transactional management system, you can answer the following questions:

- **Which BASE24 configurations cannot be replicated?**

  Changes to the N1ANEF and other NEF data files cannot be captured because these files are updated by the XPNET process. ACI does not allow any libraries to be bound with XPNET. Most other configuration files such as LNMAP, DESTMAP, SPANCNTL and SPANMAP are created when BASE24 is installed and are not replicated.

- **Which files should be included for replication?**

  The LCONF files should be replicated.

- **Which files should be excluded?**

  Typically files that can be re-created on the target system, such as report files, should not be replicated, additionally some of the BASE24 configuration files.

- **How large should GoldenGate trails be?**

  Considerations include:

  - Peak transaction volumes
  - Replicated file size
  - Compression usage
  - Number of days to store data if it cannot be moved to the target
  - Refresh type (full, incremental)
  - Frequency, size, and location of replication resulting from a full refresh, if applicable.

- Location of GoldenGate trails.

  The location of the trail files is very important. The trail files should be on disk volumes that are not used or minimally used for any other application. The less contention while accessing the disk the better. Trails should not share the disk volumes with other trails.

- BASE24 processes that impact data replication

  GoldenGate allows files to be excluded from capture by process name or program object file name. Changes to certain files (i.e. IDF, PRDF, TDF, PTDF) by the Settlement process on the source site should not be replicated if the BASE24 Settlement process is running on the target system.

- Other applications running that impact system availability.

Once you have established your transactional data management system requirements, you are ready to install and configure GoldenGate for BASE24.

Prerequisites

Before you can install GoldenGate for BASE24, you must verify required dictionary files and install GoldenGate for HP NonStop.

Verify required dictionary files

The BASE24 dictionary subvolumes BA60DDL, PS60DDL and AT60DDL are required by GoldenGate for its Replicat mapping. If the system where GoldenGate is installed does not contain these subvolumes, FUP DUP them to the BASE24 volume in your GoldenGate environment. The dictionary files all begin with the constant DICT.

```
FUP DUP \PROD\BASE24.BA60DDL.DICT*,
                           \MASTER.BASE24.BA60DDL.*,SOURCEDATE

FUP DUP \PROD\.BASE24.AT60DDL.DICT*,
                            \MASTER.BASE24.AT60DDL.*,SOURCEDATE
```

```
FUP DUP \PROD.BASE24.PS60DDL.DICT*,
                            \MASTER.PS60DDL.*,SOURCEDATE
```

Verify the following files are pointing to their correct alternate key file locations:

- DICTKDF
- DICTOBL
- DICTODF
- DICTOUF
- DICTRDF

Install GoldenGate for HP NonStop

Before you install GoldenGate for BASE24, you must install GoldenGate for HP NonStop in the same volume as your BASE24 application. Find installation instructions in the supplemental guide for the module you are installing.

Installing GoldenGate for BASE24

This section describes how to install GoldenGate for BASE24 and bind the GoldenGate intercept library to BASE24's SKELB.

Installing GoldenGate for BASE24

To install GoldenGate for BASE24, you must install the application in the same volume as your BASE24 application and point your application files to GoldenGate.

- Save the GoldenGate for BASE24 zip file to a temporary folder on your workstation and unzip its contents.
- Upload the files in BINARY mode to the BASE24 volume and GoldenGate subvolume where you wish to install and run GoldenGate.
- After the files are loaded onto the HP, alter the file code of UNPAKIT to code 101.

  ```
  TACL 1> FUP ALTER UNPAKIT, CODE 101
  ```

- At the TACL prompt type: `RUN UNPAKIT`

  The UNPAKIT process will prompt you with the location that it is going to install into and ask you to confirm. Either confirm your location, or enter N. You will then be prompted to enter your desired location.
- Change the TDF and PTDF to contain the logical network name where GoldenGate is installed.
- Change the LCONF to contain the site name, disk volume names, and logical network name where GoldenGate is installed.

Your GoldenGate for BASE24 installation is complete. However, if your environment runs non-audited files, you must bind GoldenGate's intercept library to SKELB.

Binding the GoldenGate intercept library to SKELB

To bind the GoldenGate intercept library BASELIB into SKELB, you must modify the BINDSKEL macro with the location of your current SKELB library on <BASE24 vol>.XPNET.SKELB.

The following is an example of this modification. Either use EDIT or TEDIT to modify the BINDSKEL file:

```
?tacl macro
#frame
#push bindin fl modts modts2

sink [#definedelete =skelb]
sink [#definedelete =skelbn]
sink [#definedelete =baselib]

add define =skelb,  class map, file <BASE24 volume>.xpnet.skelb
add define =skelbn, class map, file <BASE24 volume>.xpnet.skelbn
add define =baselib, class map, file baselib
```

Once the BINDSKEL macro has been modified, run BINDSKEL to generate the new SKELBN library.

The XPNET release determines which subvolume the SKELB is located.

For example:

XPNET 2.1:

```
add define =skelb,  class map, file <base24 volume>.spannet.skelb
add define =skelbn, class map, file <base24 volume>.spannet.skelbn
```

XPNET 3.0:

```
add define =skelb,  class map, file <base24 volume>.xpnet.skelb
add define =skelbn, class map, file <base24 volume>.xpnet.skelbn
```

There are no expected GoldenGate warnings or errors for the BIND and AXCEL programs. Should you encounter exceptions to the warnings, see the ACI XPNET 2.1 or 3.0 Implementation Guide.

CHAPTER 3
Configuring GoldenGate for BASE24

This chapter guides you through the steps required to configure GoldenGate for BASE24. These steps are outlined in the following topics:

Contents

Overview

GoldenGate is configured by the number of components you run, and the parameters applied to those components. This section outlines sample configurations for a simple BASE24 implementation of two Extracts and two Replicats. While each GoldenGate for BASE24 component has required parameters, there are many additional optional parameters that control replication. You will see these described in the comments of the sample files below.

Figure 3 Direct Extract parameter file for a TLF and PTLF file

```
--******************************************************************
-- EXTTLFA  Direct Extract Parameter File
--          Directly reads the BASE24 TLF file:
--          <Base24 volume>.PRO1ATLF.TLYYMMDD
--          and moves everything to the Extract Trail
--          \BACKUP.<trail volume>.TLFATRLS.AA
--
--          This Extract is 1 of 2 Extracts reading the BASE24 TLF
--          files in the PRO1ATLF subvolume.  The Extracts will
--         alternate reading the PTLF files.  This allows extracting
--          next day’s PTLF while still extracting the current day.
--*****************************************************************
EXTRACT EXTTLFA

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTTLFA, PURGE
-- Do not pad records that are not maximum size
NOFILLSHORTRECS
-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
--TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
--RMTHOST <tcp/ip address>, MGRPORT <manager port number>
```

```
-- Specify the number of Direct Extracts and which number for this
-- Extract
ALTINPUTRANGE(1OF2),TEMPLATE<Base24volume>.PRO1TLF.TL*

-- Set the location of the Remote Trail (if tcp/ip)
--RMTTRAIL \BACKUP.<volume>.TLFATRLS.AA
-- Set the location of the Remote Trail (if expand)
EXTTRAIL \BACKUP.<volume>.TLFATRLS.AA

-- Move all files in the Logtrail to \BACKUP
FILE $*.*.*;
```

Figure 4 Direct Extract file for a TLF file

```
--*****************************************************************
-- EXTTLFB  Direct Extract Parameter File
--          Directly reads the BASE24 TLF file:
--          <Base24 volume>.PRO1ATLF.TLYYMMDD
--          and moves everything to the Extract Trail
--          \BACKUP.<trail volume>.TLFBTRLS.AA
--
--          This Extract is 1 of 2 Extracts reading the BASE24 TLF
--          files in the PRO1ATLF subvolume.  The Extracts will
--         alternate reading the PTLF files.  This allows extracting
--          next day's PTLF while still extracting the current day.
--*****************************************************************
EXTRACT EXTTLFB

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTTLFB, PURGE
-- Do not pad records that are not maximum size
NOFILLSHORTRECS
-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (if tcp/ip)
--TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (if tcp/ip)
--RMTHOST <tcp/ip address>, MGRPORT <manager port number>
```

```
--Specify the number of Direct Extracts & the number for this Extract
ALTINPUT RANGE (2 OF 2), TEMPLATE <Base24 volume>.PRO1TLF.TL*

-- Set the location of the Remote Trail (if tcp/ip)
--RMTTRAIL\BACKUP.<volume>.TLFBTRLS.AA

-- Set the location of the Remote Trail (if expand)
EXTTRAIL \BACKUP.<volume>.TLFBTRLS.AA

-- Move all files in the Logtrail to \BACKUP
FILE $*.*.*;
```

Figure 5 Replicat parameter file for Replicat reading the TLFATRLS.AA extract

```
**********************************************************************
-- REPTLFA - Reads the Exttrails <Base24 volume>.TLFATRLS.AA
--          and creates and maintains the TLF files
--
**********************************************************************
REPLICAT REPTLFA

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPTLFA, PURGE

-- Do block reads
FASTREADS

-- Map the TLF files
MAP \PRODUCTION.<Base24 volume>.PRO1ATLF.T*,
TARGET \BACKUP.<Base24 volume>.*.*,
CREATETEMPLATE <volume>.GGTMPL.TLYYMMDD,
ALTFILECHAR 2;
```

Figure 6 Replicat parameters reading the TLFBTRLS.AA extract

```
--********************************************************************
-- REPTLFB - Reads the Exttrails <Base24 volume>.TLFBTRLS.AA
--          and creates and maintains the TLF files
--********************************************************************
REPLICAT REPTLFB
```

```
-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPTLFB, PURGE

-- Do block reads
FASTREADS

-- Map the TLF files
MAP \PRODUCTION.<Base24 volume>.PRO1ATLF.T*,
                                TARGET \BACKUP.<Base24 volume>.*.*,
    CREATETEMPLATE <volume>.GGTMPL.TLYYMMDD,
    ALTFILECHAR 2;
```

Configuring Extract and Replicat

To configure Extract and Replicat, you must create their individual parameter files, then add the Extract and Replicat groups.

Creating Extract and Replicat parameter files

To create Extract and/or Replicat parameter files, use EDIT or TEDIT to open a blank NonStop file, then complete the following steps:

1. Enter the name of your component (Extract or Replicat) and the name of the component group:

   ```
   EXTRACT EXTTLFA
   ```

2. Enter the discard file location:

   ```
   DISCARDFILE <$ggvolume.subvolume.filename>
   ```

3. Enter desired parameters based on your environment and performance considerations.
4. Save your file. For more assistance in creating parameter files, see the examples in the “Overview” on page 14.

Adding Extracts and Replicats

Once you have created your parameter files, you can add the Extract and Replicat groups using GGSCI.

Add Extracts

```
GGSCI> ADD EXTRACT EXTTLFA,
       FILETYPE ACITLF <Base24 volume>.PRO1ATLF.TL,
       EXTSEQNO <yymmdd of first TLF file to be extracted>,
       EXTRBA 0
```

This command adds the EXTTLFA Extract process, which uses the EXTTLFA parameter file created in the previous step. This Extract reads the TLF file that matches the yymmdd in the EXTSEQNO parameter.

```
GGSCI> ADD EXTRACT EXTTLFB,
       FILETYPE A
       EXTSEQNO <yymmdd of first TLF file to be extracted>,
       EXTRBA 0
```

This command adds the EXTTLFB Extract process, which uses the EXTTLFB parameter file you created in the previous step. This Extract reads the TLF file that matches the yymmdd in the EXTSEQNO parameter. Since there are two Extracts that will read from the same yymmdd file, the hashing algorithm will decide which extract reads which file sets.

Add Replicats

```
GGSCI> ADD REPLICAT REPTLFA, LOGTRAILSOURCE <trail volume>.TLFLOGS.AA
```

This command adds the REPTLFA Replicat process which uses the REPTLFA parameter file already created. This Replicat reads the <trail volume>.TLFATRLS.AA logtrail.

```
GGSCI> ADD REPLICAT REPTLFB, LOGTRAILSOURCE <trail volume>.TLFLOGS.AA
```

This command adds the REPTLFB Replicat process, which uses the REPTLFB parameter file already created. This Extract reads the <trail volume>.TLFBTRLS.AA logtrail.

Configuring templates

Each Extract above reads files from your source BASE24 application. For the files to be useful on the target system, they must follow a consistent format. GoldenGate prepackages templates to guide you in setting up your own file layouts. Examples of the template files for BASE24 are below.

File to read: GGTMPL.TLYYMMDD

Template:

```
<BASE24 volume>.GGTMPL.TLYYMMDD
   ENSCRIBE
   TYPE E
   FORMAT 1
   EXT ( <# of primary extents> PAGES, <# of secondary extents> PAGES)
   REC 4072
   BLOCK  4096
   ALTKEY ( "CR", FILE 0, KEYOFF 38, KEYLEN 30 )
   ALTKEY ( "TR", FILE 0, KEYOFF 14, KEYLEN 24 )
   ALTFILE ( 0, <Base24 volume>.PRO1ATLF.T0YYMMDD )
   MAXEXTENTS 100
   BUFFERED
   OWNER 200,255
   SECURITY (RWEP): OOOO
   DATA MODIF:   1 Oct 2003,  5:51
   CREATION DATE:   1 Oct 2003,  5:51
   LAST OPEN:  28 Jul 2004, 17:00
   FILE LABEL: 192 (4.7% USED)
   EOF: 0 (0.0% USED)
   EXTENTS ALLOCATED: 0
```

File to read: PRO1ATLF.T0YYMMDD

Template:

```
<BASE24 volume>.PRO1ATLF.T0YYMMDD
   ENSCRIBE
   TYPE K
   FORMAT 1
   EXT ( <# of primary extents> PAGES, <# of secondary extents> PAGES)
   REC 36
```

```
      BLOCK  4096
      KEYLEN 36
      KEYOFF 0
      MAXEXTENTS 100
      BUFFERED
      OWNER 200,255
      SECURITY (RWEP): OOOO
      DATA MODIF:   1 Oct 2003,  5:51
      CREATION DATE:   1 Oct 2003,  5:51
      LAST OPEN:   2 Aug 2004, 21:47
      FILE LABEL: 150 (3.6% USED)
      EOF: 0 (0.0% USED)
      EXTENTS ALLOCATED: 0
```

File to read: GGTMPL.POYYMMDD

Template:

```
   <Base24 volume>.GGTMPL.POYYMMDD
      ENSCRIBE
      TYPE E
      FORMAT 1
      EXT ( 50 PAGES, 200 PAGES )
      REC 4072
      BLOCK  4096
      ALTKEY ( "CR", FILE 0, KEYOFF 10, KEYLEN 30 )
      ALTKEY ( "RK", FILE 4, KEYOFF 145, KEYLEN 26 )
      ALTKEY ( "RT", FILE 1, KEYOFF 40, KEYLEN 57 )
      ALTKEY ( "TK", FILE 3, KEYOFF 129, KEYLEN 42 )
      ALTKEY ( "TT", FILE 2, KEYOFF 97, KEYLEN 32 )
      ALTFILE ( 0, <Base24 volume>.PRO1PTLF.P0YYMMDD )
      ALTFILE ( 1, <Base24 volume>.PRO1PTLF.P1YYMMDD )
      ALTFILE ( 2, <Base24 volume>.PRO1PTLF.P2YYMMDD )
      ALTFILE ( 3, <Base24 volume>.PRO1PTLF.P3YYMMDD )
      ALTFILE ( 4, <Base24 volume>.PRO1PTLF.P4YYMMDD )
      MAXEXTENTS 500
      BUFFERED
      OWNER 200,255
      SECURITY (RWEP): OOOO
      DATA MODIF:  10 May 2004,  7:38
      CREATION DATE:  10 May 2004,  7:38
      LAST OPEN:  28 Jul 2004, 17:00
      FILE LABEL: 312 (7.6% USED)
      EOF: 0 (0.0% USED)
      EXTENTS ALLOCATED: 0
```

File to read: PRO1PTLF.P0YYMMDD

Template:

```
   <Base24 volume>.PRO1PTLF.P0YYMMDD
      ENSCRIBE
      TYPE K
      FORMAT 1
      EXT ( 50 PAGES, 200 PAGES )
      REC 36
      BLOCK  4096
      KEYLEN 36
      KEYOFF 0
      MAXEXTENTS 500
      BUFFERED
      OWNER 200,255
      SECURITY (RWEP): OOOO
      DATA MODIF:  10 May 2004,  7:38
      CREATION DATE:  10 May 2004,  7:38
      LAST OPEN:   2 Aug 2004, 21:47
      FILE LABEL: 150 (3.6% USED)
      EOF: 0 (0.0% USED)
      EXTENTS ALLOCATED: 0
```

File to read: PRO1PTLF.P1YYMMDD

Template:

```
   <Base24 volume>.PRO1PTLF.P1YYMMDD
      ENSCRIBE
      TYPE K
      FORMAT 1
      EXT ( 50 PAGES, 200 PAGES )
      REC 63
      BLOCK  4096
      KEYLEN 63
      KEYOFF 0
      MAXEXTENTS 500
      BUFFERED
      OWNER 200,255
      SECURITY (RWEP): OOOO
      DATA MODIF:  10 May 2004,  7:38
      CREATION DATE:  10 May 2004,  7:38
      LAST OPEN:   2 Aug 2004, 21:47
      FILE LABEL: 150 (3.6% USED)
      EOF: 0 (0.0% USED)
      EXTENTS ALLOCATED: 0
```

File to read: PRO1PTLF.P2YYMMDD

Template:

```
<Base24 volume>.PRO1PTLF.P2YYMMDD
   ENSCRIBE
   TYPE K
   FORMAT 1
   EXT ( 50 PAGES, 200 PAGES )
   REC 38
   BLOCK  4096
   KEYLEN 38
   KEYOFF 0
   MAXEXTENTS 500
   BUFFERED
   OWNER 200,255
   SECURITY (RWEP): OOOO
   DATA MODIF:  10 May 2004,  7:38
   CREATION DATE:  10 May 2004,  7:38
   LAST OPEN:   2 Aug 2004, 21:47
   FILE LABEL: 150 (3.6% USED)
   EOF: 0 (0.0% USED)
   EXTENTS ALLOCATED: 0
```

File to read: PRO1PTLF.P3YYMMDD

Template:

```
<Base24 volume>.PRO1PTLF.P3YYMMDD
   ENSCRIBE
   TYPE K
```

```
    FORMAT 1
    EXT ( 50 PAGES, 200 PAGES )
    REC 48
    BLOCK  4096
    KEYLEN 48
    KEYOFF 0
    MAXEXTENTS 500
    BUFFERED
    OWNER 200,255
    SECURITY (RWEP): OOOO
    DATA MODIF:  10 May 2004,  7:38
    CREATION DATE:  10 May 2004,  7:38
    LAST OPEN:   2 Aug 2004, 21:47
    FILE LABEL: 150 (3.6% USED)
    EOF: 0 (0.0% USED)
    EXTENTS ALLOCATED: 0
```

File to read: PRO1PTLF.P4YYMMDD

Template:

```
<Base24 volume>.PRO1PTLF.P4YYMMDD
    ENSCRIBE
    TYPE K
    FORMAT 1
    EXT ( 50 PAGES, 200 PAGES )
    REC 32
    BLOCK  4096
    KEYLEN 32
    KEYOFF 0
    MAXEXTENTS 500
    BUFFERED
    OWNER 200,255
    SECURITY (RWEP): OOOO
    DATA MODIF:  10 May 2004,  7:38
    CREATION DATE:  10 May 2004,  7:38
    LAST OPEN:   2 Aug 2004, 21:47
    FILE LABEL: 150 (3.6% USED)
    EOF: 0 (0.0% USED)
    EXTENTS ALLOCATED: 0
```

Configuring the PATHCONF file

The GoldenGate library GGSLIB needs to be set for the Pathway servers that maintain data files. This must be done for all servers except:

- SERVER-NCS
- SERVER-NCP
- SERVER-NCPI-XX
- SERVER-MENUHELP
- SERVER-NCSS
- SERVER-NCSP

To configure the file, enter the following line in the Pathway Configuration file

```
BASE24 vol>.PRODCNTL.PATHCONF:
SET SERVER GUARDIAN-LIB <GoldenGate volume.subvolume>.GGSLIB
```

The following is a PATHCONF CAF Server example:

```
[ CARD ACCOUNT FILE SERVER                ]
RESET SERVER
SET SERVER CPUS                   0:1
SET SERVER PROGRAM                <BASE24 vol>.BA60OBJ.SVCAF
SET SERVER DELETEDELAY            10 MINS
SET SERVER TIMEOUT                60 SECS
SET SERVER PRI                    135
SET SERVER HOMETERM               $VHS
SET SERVER GUARDIAN-LIB   <GoldenGate vol>.<GoldenGate subvol>.GGSLIB
ADD SERVER SERVER-CAF
```

For the SERVER-DPCT set the GUARDIAN-LIB to <BASE24 volume>.XPNET.SKELBN.

```
[ DEVICE CONTROL TERMINAL SERVER             ]
RESET SERVER
SET SERVER cpus                   0:1
SET SERVER PROGRAM                <BASE24 vol>.BA60OBJ.SVDPCT
SET SERVER GUARDIAN-LIB           <BASE24 vol>.XPNET.SKELBN
SET SERVER HIGHPIN                OFF
SET SERVER DELETEDELAY            12 HRS
SET SERVER PRI                    135
SET SERVER PROCESS                $PPCT
SET SERVER NUMSTATIC              0
SET SERVER HOMETERM               $VHS
ADD SERVER SERVER-DPCT
```

Note If GoldenGate has been installed for more than one BASE24 network on the same HP NonStop Server, the second GoldenGate environment needs its own prefix (i.e. $XX) and AUDCFG file (i.e. $SYSTEM.<GoldenGate subvol>>.AUDCFG). In addition to setting the GUARDIAN-LIB parameter in the PATHCONF file, two additional define settings are required.

Add the following settings to each server listed above.

```
SET SERVER DEFINE =GGS_PREFIX, CLASS MAP, FILE $<two character prefix>
SET SERVER DEFINE =GGS_AUDCFG, CLASS MAP, FILE $SYSTEM.<GoldenGate
subvol>.AUDCFG
```

These DEFINES must be entered for all the servers listed for the GUARDIAN-LIB setting plus the servers SERVER-NCPI-XX (i.e. SERVER-NCPI-1A).

The following example illustrates both the GUARDIAN-LIB and DEFINE settings.

```
[ CARD ACCOUNT FILE SERVER                    ]
RESET SERVER
SET SERVER cpus                   0:1
SET SERVER PROGRAM                <BASE24 vol>.BA6TOBJ.SVCAF
SET SERVER DELETEDELAY            10 MINS
SET SERVER TIMEOUT                60 SECS
SET SERVER PRI                    135
SET SERVER HOMETERM               $vhs
SET SERVER GUARDIAN-LIB           <BASE24 volume>.XPNET.SKELBN
SET SERVER DEFINE =GGS_PREFIX,CLASS MAP,FILE $<two character prefix>
SET SERVER DEFINE =GGS_AUDCFG,CLASS MAP,FILE $SYSTEM.<GoldenGate
subvol>.AUDCFG
ADD SERVER SERVER-CAF
```

Configuring the NEF file

The name of the BASE24 run time library SKELB will change when the GoldenGate BASELIB is bound into it, creating the new file named SKELBN. SKELBN can be created on the <BASE24 volume>.XPNET.SKELBN.

If the new SKELBN file replaces the SKELB references in the N1ACONF file, the library configuration setting should be changed to the following:

```
SET PROCESS LIBRARY <BASE24 vol>.XPNET.SKELBN
```

The following example illustrates this new library location using a GoldenGate volume/subvolume in the N1ACONF file:

```
RESET PROCESS
SET PROCESS BCPU 1
SET PROCESS LIBRARY <BASE24 vol>.XPNET.SKELBN
SET PROCESS PROGRAM <BASE24 vol>.PS60obj.RTAU
SET PROCESS PPD $p1R1
SET PROCESS PRIORITY 175
SET PROCESS CPU 0
SET PROCESS STARTUP AUTOMATIC
SET PROCESS QAT 64
ADD PROCESS P1A^RTAU1, UNDER SYSNAME \SITEA, UNDER NODE P1A^NODE
```

Note The library setting must be changed for all BASE24 processes except the generic simulator process such as P1A^GSIM or P1A^GENSIM

Changing the BASE24 online environment

Either the NCS Pathway screen or the NCPCOM conversational interface maintains the satellite processes that make up BASE24's online environment. If you are using the new SKELBN library, you will have to incorporate this change into the satellite processes.

Updating the satellite processes

The new SKELBN must be incorporated into BASE24. If this is an initial installation of GoldenGate that uses the default settings, BASE24 does not need to be brought down. By stopping and starting groups of processes it is possible to phase in the new SKELBN library. The time to stop and start the groups of processes can be further reduced by making obey files to run the actual commands. If this is not an initial installation, contact GoldenGate Technical Services to discuss the best method of upgrading your site.

If you are using the same SKELB name for the new library, perform the following steps:

1. From the TACL prompt, go to <BASE24 volume>.XPNET. If the new SKELBN file is located elsewhere, duplicate it to this location using FUP.
2. Rename SKELB to OSKELB.
3. Rename SKELBN to SKELB.
4. In NCPCOM or the NCS screen, stop all satellite processes that share the same program object file at the same time.
5. Restart each group of satellite processes that share the same program object file until all satellite processes have switched to the new library. If the STARTUP option is AUTOMATIC or DEMAND, the satellite processes will automatically startup by themselves.

If you are using a new SKELBN name, perform the following steps:

1. From the TACL prompt, go to <BASE24 volume>.XPNET. If the new SKELBN file is located elsewhere, duplicate it to this location using FUP.
2. In the NCPCOM or the NCS screen, stop the satellite processes that share the same program object file at the same time. If the STARTUP option is AUTOMATIC or DEMAND, it may be necessary to reissue the STOP command until the processes have permanently stopped. This is because BASE24 will restart a process a configurable number of times before it stops trying. Processes must be stopped and restarted as a group, otherwise an Error 18 will occur since the same object cannot be bound with two different libraries at the same time.

3. Alter the satellite processes to point to the new library name, such as:

   ```
   ALTER PROCESS *RTAU*, LIBRARY <Base24 volume>.XPNET.SKELBN
   ```

4. Restart the processes.

Repeat these steps until all satellite processes have had their library name altered.

Updating the Pathway Servers

To point your Pathway servers to the new GGSLIB, you must update the interface program on all but the following servers:

- SERVER-NCS
- SERVER-NCP
- SERVER-NCPI-xx
- SERVER-MENUHELP
- SERVER-NCSS
- SERVER-NCSP
- SERVER-DPCT

To update the Pathway servers:

1. From the TACL prompt, move to the main BASE24 subvolume PRODCNTL.
2. Run the Pathway interface program Pathcom

   ```
   TACL> PATHCOM $PPMN
   ```

3. Add the GGSLIB intercept library to the Pathway environment by entering the following for all file maintenance servers:

   ```
   = FREEZE SERVER-xxxxx
   = STOP   SERVER-xxxxx
   = STOP   SERVER-xxxxx (sometimes the stop must be issued twice)
   = ALTER  SERVER-xxxxx, GUARDIAN-LIB <GoldenGate
   volume.subvolume>.GGSLIB
   ```

These next two lines are only required if GoldenGate's default configuration is not used.

```
= ALTER SERVER-xxxxx, DEFINE =GGS_AUDCFG, CLASS MAP, FILE
<vol.subvol>.AUDCFG
= ALTER SERVER-xxxxx, DEFINE =GGS_PREFIX, CLASS MAP, FILE $<two
characters>
= THAW SERVER-xxxxx
```

4. For SERVER-DPCT set the GUARDIAN-LIB command to SKELBN

```
= FREEZE SERVER-DPCT
= STOP SERVER-DPCT
= STOP SERVER-DPCT
= ALTER SERVER-DPCT, GUARDIAN-LIB <BASE24 volume>.XPNET.SKELBN
= THAW SERVER-DPCT
```

Binding the intercept library to other programs

The GGSCI program uses the command BIND PROGRAMS to bind either the BASELIB or GGSLIB intercept library to any program. The BASE24 Refresh program requires the File Utility Program (FUP) to actually load the file that is to be fully refreshed. For GoldenGate to capture the file load, the Refresh process must use a copy of FUP with the GoldenGate library GGSLIB bound into it.

The following will bind the GoldenGate intercept library to a copy of FUP:

1. Go to the GoldenGate volume and subvolume
2. FUP DUP the program FUP to the GoldenGate subvolume.

```
TACL> FUP DUP $SYSTEM.SYSxx.FUP, GGSFUP
```

3. Run GGSCI:

```
TACL> RUN GGSCI
```

4. Run BIND PROGRAMS:

   ```
   GGSCI 1> BIND PROGRAMS
   Program: GGSFUP
   Program: GO
   Begin with BIND process (Y/N)? Y
   ```

5. Change the HIGHREQUESTERS option to ON for the GGSFUP program. The following command will set HIGHREQUESTERS ON for the GGSFUP program:

   ```
   TACL> BIND CHANGE HIGHREQUESTERS ON IN GGSFUP
   ```

6. License the GGSFUP program.

   ```
   TACL> FUP LICENSE GGSFUP
   ```

Unbinding GGSLIB from a program

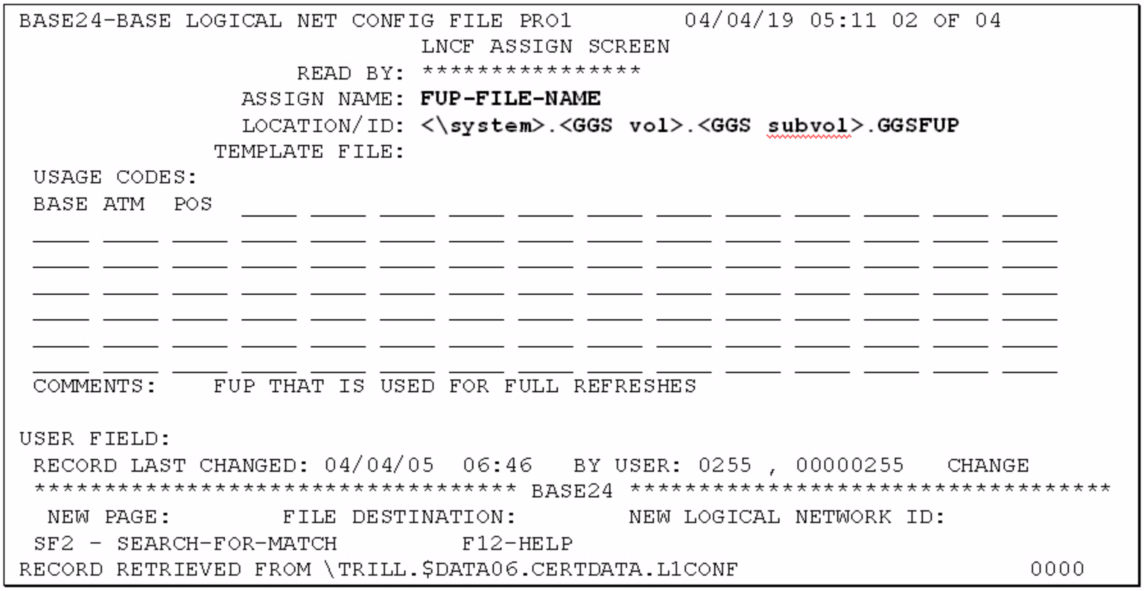
A reverse of this method will unbind a library to a program. Run the program but do not have a value for the LIB option. For example if you want to take the intercept library out of a program:

```
TACL> RUN ENTLCONF /LIB   /
```

Configuring the Refresh FUP LCONF screen

The following is an example of what your LCONF screen should look like once it is ready to run with GoldenGate for BASE24.

The LCONF Assign screen FUP-FILE-NAME defines the location of the FUP used by the Refresh process.



```
BASE24-BASE LOGICAL NET CONFIG FILE PRO1       04/04/19 05:11 02 OF 04
                             LNCF ASSIGN SCREEN
                    READ BY: ****************
                ASSIGN NAME: FUP-FILE-NAME
                LOCATION/ID: <\system>.<GGS vol>.<GGS subvol>.GGSFUP
              TEMPLATE FILE:
 USAGE CODES:
 BASE ATM  POS   ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
 COMMENTS:    FUP THAT IS USED FOR FULL REFRESHES

USER FIELD:
 RECORD LAST CHANGED: 04/04/05  06:46   BY USER: 0255 , 00000255   CHANGE
 *********************************** BASE24 ***********************************
  NEW PAGE:        FILE DESTINATION:        NEW LOGICAL NETWORK ID:
 SF2 - SEARCH-FOR-MATCH         F12-HELP
RECORD RETRIEVED FROM \TRILL.$DATA06.CERTDATA.L1CONF                     0000
```

Configuring Transaction Security Services

BASE24 offers services that interface with different devices to supply additional layers of transactional security (Transaction Security Services, or TSS). GoldenGate for BASE24 can capture these files as part of a Hot Site replication solution in the following scenario:

- TSS is running on an HP NonStop operating system
- TSS is running as an audited application
- You implement one TSS configuration per logical network.

When you set up TSS replication, the following files are captured by default:

CHNPKD	CRDVD	CSECD	EMVSD	ENVMTD	EPNUMD	EVDOCD
HSPKPD	HPSPKSD	IDESD	IDNTD	NCRD	SECDCD	VPVVD

Implementing TSS replication

In a standard, single-direction implementation of GoldenGate for BASE24, TSS files are captured with no special configuration. If you have multiple BASE24 environments, you must configure an Extract and Delivery process for each one. Follow the procedures in Configuring Extract and Replicat.

For a bi-directional implementation, you must:

- Configure an Extract and Replicat process on each logical network where BASE24 operates.
- Configure the locator IDs in each KEYF file to ensure they are different for each enviornment. This allows you to identify your data's source.
- Build the TSS database with the logical network option enabled. This allows the TSS database to manage data in a bi-directional manner based on the Extended Memory Tables and process requirements from ACI.

Non-audited files in TSS

Certain files are non-audited, even if you run TSS as an audited application. These files include:

ASAFCD ASAFD ASAFOD CRDVOD EVENTD EXLOGD IDESOD

ISECOD KEYF VPVVOD IDNTD NCRD SECDCD VPVVD

You can still capture non-audited TSS files; however, you must work with ACI to ensure you have the correct version of their libraries. Once you have obtained the correct libraries, bind the appropriate GGSLIB as described in "Binding the intercept library to other programs" on page 29. Once the library is bound, configure a Logger to capture the non-audited files.

CHAPTER 4

Initial Database Synchronization

This chapter guides you through loading your target database for the first time. You can use this same technique to synchronize databases at any time (e.g. if your target database was unavailable for a time while your source continued to process records). Database synchronization is discussed in the following topics:

Contents

Database synchronization prerequisites

Before you begin your synchronization work, you should know which files have network-dependent file names, either on the BASE24 or HP NonStop node. These files include:

- LCONF
- IDF
- PTDF, PTDD1
- TDF, ATDD1

The fields that contain these values must be changed to match the target environment.   A FUP LOAD or DUP is not adequate to change the field values. GoldenGate Software uses Extract and Replicat to load the new database and change the dependent values at the same time. The Extract process can be configured to read directly from a BASE24 file and write the data to a file which is picked up by Replicat and loaded into the target schema. Either the Extract or Replicat can be configured to perform the required value substitution during this process.

To prepare for synchronization, perform the following steps:

1. For non-audited files, make sure the GoldenGate Logger is running and all processes that access the BASE24 database either have the GoldenGate library BASELIB or GGSLIB bound into them. Skip this step for TMF audited files.
2. If GoldenGate DEFINES are required, make sure all processes have the correct GoldenGate files and processes open.
3. The online Extracts may be running, however the online Replicats on the backup site must NOT be running.
4. Prepare the Initial Load Extract parameter file and the inital load Replicat parameter files for LCONF. The following examples illustrate initial load parameter files for the Extract and Replicat processes for the LCONF file.

Figure 7 Initial load Extract parameter file

```
--******************************************************************
-- EILCONF  Initial Load Extract Parameter File
--         Reads the LCONF file and creates the EXTFILE on the BACKUP
--          site for initial loading
-- *****************************************************************
-- Directly read the IDF source file
SOURCEISFILE

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.EILCONF, PURGE

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Use the length of the record read.  Do not pad with spaces
NOFILLSHORTRECS

-- Output Extract file to be read by the Replicat RILCONF
EXTFILE \BACKUP.<volume>.B24INIT.LCONF, MEGABYTES <megabytes>, PURGE

-- LCONF file to be loaded
FILE <Base24 volume>.PRO1CNTL.L1CONF;
```

Figure 8 Initial load Replicat parameter file

```
--******************************************************************
-- RILCONF  Initial Load Replicat Parameter File
--          This Replicat reads the EXTFILE and loads the LCONF file
--          on the BACKUP site.
-- *****************************************************************

-- Identifies this as an Initial Load Replicat
SPECIALRUN

-- Tells the Replicat to stop when end of file is reached on the
EXTFILE
END RUNTIME
```

```
-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.RILCONF, PURGE

-- Set the dictionary location
DICTIONARY <base24 base volume>.BA60DDL

-- Use the target dictionary
ASSUMETARGETDEFS

-- Do block reads
FASTREADS

-- Map the LCONF file and substitute NonStop node and volume.
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute -- the correct SITE, VOLUME or SUBVOLUME name.
MAP \PRODUCTION.<volume>.PRO*CNTL.L*CONF,
   TARGET \BACKUP.<volume>.*.*,
   TARGETDEF LCONF,
   COLMAP
   (PRIKEY        = PRIKEY,
   PROD-IND       = PROD-IND,
   LCONF.COMMENTS = @STRSUB(LCONF.COMMENTS,"\PRODUCTION","\BACKUP",
                  “PRODVOL”, “BKUPVOL”, “PRODSUBV”,   “BKUPSUBV”),
   LAST-CHNG-TIME = LAST-CHNG-TIME,
   FILE-NAME      = @STRSUB (FILE-NAME, "\PRODUCTION", "\BACKUP",
                  “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
   TEMPLATE       = @STRSUB (TEMPLATE,  "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
   USER-FIELD     = USER-FIELD,
   USER-FLD2      = USER-FLD2,
   PROD-IND-ADNL  = PROD-IND-ADNL,
   USER-FLD4      = USER-FLD4,
   LAST-AFM       = LAST-AFM),

   WHERE (ITEM-TYP = "A");
```

```
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \PRODUCTION.<volume>.PRO*CNTL.L*CONF,
   TARGET \BACKUP.<volume>.*.*,
   TARGETDEF LCONF,
   COLMAP
   (PRIKEY         = PRIKEY,
   PROD-IND       = PROD-IND,
   LCONF.COMMENTS = @STRSUB
   (LCONF.COMMENTS,"\PRODUCTION",”\BACKUP”),
   LAST-CHNG-TIME = LAST-CHNG-TIME,
   PLGTH          = PLGTH,
   PTXT           = @STRSUB (PTXT,    "\PRODUCTION", "\BACKUP"),
   USER-FLD3      = USER-FLD3,
   PROD-IND-ADNL  = PROD-IND-ADNL,
   USER-FLD4      = USER-FLD4,
   LAST-AFM       = LAST-AFM),

   WHERE (ITEM-TYP = "P");
```

Running the initial load programs

Once the parameter files have been prepared, the initial load Extract and Replicat must be run. This can now be done in one of two ways:

Running the initial load from TACL:

This method works for all versions of GoldenGate.

Source site

```
TACL> VOLUME <GoldenGate main volume.subvolume>
TACL> RUN EXTRACT /in ggsparm.eilconf, name/
```

Target site

The target data file must be created but contain no records. If the data file does exist, perform a FUP PURGEDATA. For example:

```
TACL> FUP PURGEDATA <Base24 volume>.prodcntl.l1conf
```

If the file does not exist, create the file by setting the target file like the data file on the source side. For example:

```
TACL> VOLUME <BASE24 volume>.prodcntl
TACL> FUP CREATE l1conf, LIKE \source.<Base24 volume>.prodcntl.l1conf
TACL> VOLUME <GoldenGate main volume.subvolume>
TACL> RUN REPLICAT /in ggsparm.rilconf, name/
```

Running the initial load from GGSCI

This method works for GoldenGate version 7.0.2.0 or later. If you do not explicitly start the Replicat, the Manager will.

Source site

```
TACL> VOLUME <GoldenGate main volume.subvolume>
TACL> RUN GGSCI
GGSCI> ADD EXTRACT eilconf, SOURCEISTABLE
GGSCI> START eilconf
```

Target site

```
TACL> VOLUME <GoldenGate main volume.subvolume>
TACL> RUN GGSCI
GGSCI> ADD REPLICAT rilconf, SPECIALRUN
GGSCI> START rilconf
```

CHAPTER 5

GoldenGate for Base24 Solutions: Hot Site Backup

This chapter outlines how to implement GoldenGate for BASE24 as a hot-site backup/disaster recovery solution. Each step required for such an implementation is discussed in the following topics:

Contents

Overview

Keeping a copy of data off-site enables businesses to recover from temporary outages and disasters with minimal loss of information. GoldenGate for BASE24 provides replicated copies of transactions off-site within seconds of their entry on primary systems, reducing the impact of catastrophic events.

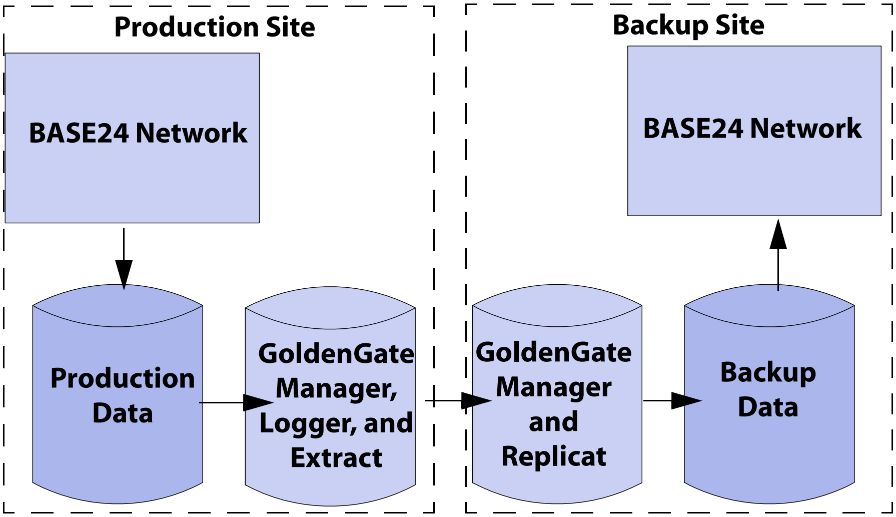
GoldenGate for BASE24 can replicate between two HP NonStop systems using TCP/IP or Expand. A benefit of using TCP/IP is that a disaster recovery or backup system can have the same name and node number as the primary system, allowing for more transparent recovery.

There are two modes of operating a hot site backup site for BASE24. If the BASE24 system on the backup site has been configured but none of the satellite applications are running, the site is considered to be in a passive mode. In this case all database changes are replicated from the production site to the backup site. If the BASE24 system is running on the backup site, the site is considered to be in an active mode. In this case replicating database changes (i.e. creating the next day's TLF file and updating the transaction/business dates in the database) from specific processes such as the settlement processes is excluded & not replicated.

An active backup affects BASE24's full refresh replication. The authorization processes on the backup site must be notified when to close and reopen the file that has just been fully refreshed. The backup site's mode of operation affects how GoldenGate is configured.

The following solution is for an active backup, which replicates the entire BASE24 database. The BASE24 environment is divided into two networks. The PRO1 network is only for ATMs and the PRO2 network is for a POS configuration. This includes both TMF Audited files and non-audited files.

Figure 9 Sample active hot site backup configuration



Production Site	Backup Site
BASE24 Network The network may have multiple nodes and process both ATM and POS transactions.	**BASE24 Network** The backup BASE24 network is configured like the Production BASE24 network.
Transaction Processing All transactions processing is done on the production site.	**Transaction Processing** No transactions processed.
Refresh Production site files are refreshed from the host files. GoldenGate replicates the refresh to the backup site.	**Refresh** The refreshed files are replicated from the Production site.

Production Site	Backup Site
Extract Super Extract is only run from the Production site.	**Extract** No Super Extract processing.
Settlement Settlement is run on the production site, but is not replicated.	**Settlement** Settlement is run on the backup site.

Required hot site backup components

The GoldenGate setup below is for a typical active hot site backup solution that contains both ATM and POS records. This configuration has the following GoldenGate entities:

- Manager Processes: $GGMGR and $GGMGX (on \PRODUCTION and \BACKUP sites)
- Three Loggers (\PRODUCTION site) A simple rule to determine which file goes to which logger is to split them up by volume of I/Os (updates and inserts). Each Logger process can handle approximately 200 to 300 transactions a second. However this will vary depending on processor speeds. One example of Logger distribution:
    - **TLF:** This Logger is dedicated to only logging file creations and record inserts to the ATM TLF files. These log records will be generated for every ATM transaction whether it is approved or not. These records are always the full record image because only complete records (inserts) are logged.
    - **PTLF:** This Logger is dedicated to only logging file creations & record inserts to the POS PTLF files. These log records will be generated for every POS transaction whether it is approved or not. These records are always the full record image because only complete records (inserts) are logged.
    - **BASE24:** This logger is dedicated to logging everything else. Even though some of the files included in this Logger are high volume, (i.e. TDF, PTDF, PRDF) the updates will be compressed. This minimizes the amount of data

that is moved between source and target nodes. However, some files such as the LCONF and IDF may need to do string substitutions for the HP NonStop Server node name or volume and subvolume names. The changes to these files must be in the NOCOMPRESSUPDATE (i.e. full record image) format to allow field mapping.

- Four Extracts (\PRODUCTION site):
    - **EXTTLF:** This Extract acts as a data pump. It moves all the data in the TLFLOGS.AA log trails to the TLFTRLS.AA extract trails on the backup site.
    - **EXTPTLF:** This Extract acts as a data pump. It moves all the data in the PTLFLOGS.AA log trails on the production site to the PTLFTRLS.AA extract trails on the backup site.
    - **EXTB24:** This Extract acts as a data pump. It moves all the data in the B24LOGS.AA log trails on the production site to the B24TRLS.AA extract trails on the backup site.
    - **EXTTMF:** This Extract moves the selected data from the TMF Audit Logs to the TMFTRLS.AA extract trails on the backup site.
- Four Replicats (\BACKUP site):
    - **REPTLF:** This Replicat applies all the TLF records from the TLFTRLS.AA extract trails to the BASE24 Backup database. If the Backup system is passive, the TLF file operations are also replicated.
    - **REPPTLF:** This Replicat applies all the PTLF records from the PTLFTRLS.AA extract trails to the BASE24 Backup database. If the Backup system is passive, the PTLF file operations are also replicated.
    - **REPB24:** This Replicat applies all the BASE24 data records from the B24TRLS.AA extract trails to the BASE24 backup database. This includes all file operations.
    - **REPTMF:** This Replicat applies all the BASE24 TMF data records from the TMFTRLS.AA extract trails to the BASE24 backup database.

Sample parameter files

The parameter files for each of the GoldenGate components can be created and edited using the following command:

```
GGSCI> EDIT PARAM <group or parameter file name>
```

The <group> is the Extract or Replicat group name and the <parameter file> is the name GoldenGate uses for the parameter file, such as Mgrparm or Logparm.

Refer to the *GoldenGate for HP NonStop Administrator Guide* for more information on creating and editing parameter files.

This section provides sample parameter files for each of the components required for a GoldenGate for BASE24 hot site backup solution.

Global and Manager parameter files

Global and Manager parameter files control how other GoldenGate components such as Extract and/or Replicat behave.

Figure 10 Sample Global parameter file

```
-- ******************************************************
-- GLOBALS CONFIGURATION FILE                           *
-- ******************************************************
--
-- Sets defaults if required by GoldenGate
APPLICATION HOTSITE

-- Sets up a global process home terminal for GoldenGate processes.
HOMETERM $VHS
```

Figure 11 Sample Manager parameter file

```
-- ******************************************************
-- MGRPARM     Manager Parameters                       *
-- ******************************************************
PORT 8000
-- TCPIP parameters are not required if using EXPAND
TCPIPPROCESSNAME $ZTC0
```

```
-- Keep the TLF Extract Trails for at least 5 days and processed
PURGEOLDEXTRACTS <Base24 volume>.TLFTRLS.AA, USECHECKPOINTS,
MINKEEPDAYS 5

-- Keep the PTLF Extract Trails for at least 5 days and processed
PURGEOLDEXTRACTS <Base24 volume>.PTLFTRLS.AA, USECHECKPOINTS,
MINKEEPDAYS 5

-- Keep the BASE24 Extract Trails for at least 5 days and processed
PURGEOLDEXTRACTS <Base24 volume>.B24TRLS.AA,  USECHECKPOINTS,
MINKEEPDAYS 5
```

Logger parameter files

Each Logger parameter file can control how each Logger reads and writes the data it is configured to capture. In this example, four Loggers handle different source records.

Figure 12 Sample Logger parameter file #0

```
LOGPARM Parameter File
-- ******************************************************
-- LOGPARM     Logger Parameters   Logger 0             *
-- ******************************************************
-- LOGGER 0

-- Location, number and size of the Logger trails
LOG <trail volume>.TLFLOGS.AA, MEGABYTES <value>, NUMFILES <number>,
SECURE "NCNC"

-- Primary and backup CPU for Logger 0
CPU 0,1

-- Get unstructured files
GETUNSTRUCTURED
-- Get bulk loads
GETBULKIO

-- Use the full record image, do not compress the updates
NOCOMPRESSUPDATES
-- Make priority higher than BASE24 applications
PRI 180
```

```
-- List the files to be captured
FILE $*.PRO1ATLF.T*
```

Figure 13 Sample Logger parameter file #1

```
-- ******************************************************
-- LOGPARM     Logger Parameters   Logger 1             *
-- ******************************************************
-- LOGGER 1

-- Location, number and size of the Logger trails
LOG <trail volume>.PTLFLOGS.AA, MEGABYTES <value>, NUMFILES <number>,
SECURE "NCNC"

-- Primary and backup CPU for Logger 1
CPU 2,3

-- Get unstructured files
GETUNSTRUCTURED
-- Get bulk loads
GETBULKIO

-- Use the full record image, do not compress the updates
NOCOMPRESSUPDATES
-- Make priority higher than BASE24 applications
PRI 180

-- List the files to be captured
FILE $*.PRO2PTLF.P*
```

Figure 14 Sample Logger parameter file #2

```
-- ******************************************************
-- LOGPARM     Logger Parameters   Logger 2             *
-- ******************************************************
-- LOGGER 2

-- Location, number and size of the Logger trails
LOG <trail volume>.B24LOGS.AA, MEGABYTES <value>, NUMFILES <number>,
SECURE "NCNC"

-- Primary and backup CPU for Logger 2
CPU 4,5
```

```
-- Get unstructured files
GETUNSTRUCTURED
-- Get bulk loads
GETBULKIO
-- Compress the updates
COMPRESSUPDATES

-- Make priority higher than BASE24 applications
PRI 180

-- List the files to be captured
FILE $*.PRO1DATA.IDF,     NOCOMPRESSUPDATES
FILE $*.PRO2DATA.IDF,     NOCOMPRESSUPDATES
FILE $*.PRO1EXEC.L1CONF,  NOCOMPRESSUPDATES
FILE $*.PRO2EXEC.L2CONF,  NOCOMPRESSUPDATES
FILE $*.PRO1DATA.*
FILE $*.PRO1EXEC.*
FILE $*.PRO1CNTL.*
FILE $*.PRO2DATA.*
FILE $*.PRO2EXEC.*
FILE $*.PRO2CNTL.*
-- List the files to be excluded only if BASE24 on the Backup site is
-- ACTIVE
EXCLUDEFILE <BASE24 vol>.PRO1DATA.IDF*,  PROGRAM <BASE24
vol>.BA60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO1DATA.TDF*,  PROGRAM <BASE24
vol>.AT60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO1DATA.ATD*,  PROGRAM <BASE24
vol>.AT60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO1DATA.FF*,   PROGRAM <BASE24
vol>.AT60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO1DATA.F0*,   PROGRAM <BASE24
vol>.AT60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO2DATA.PTDF*, PROGRAM <BASE24
vol>.PS60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO2DATA.PTD*,  PROGRAM <BASE24
vol>.PS60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO2DATA.PRDF*, PROGRAM <BASE24
vol>.PS60OBJ.*SETL
```

```
EXCLUDEFILE <BASE24 vol>.PRO2DATA.FF*,   PROGRAM <BASE24
vol>.PS60OBJ.*SETL
EXCLUDEFILE <BASE24 vol>.PRO2DATA.F0*,   PROGRAM <BASE24
vol>.PS60OBJ.*SETL
```

About Logger parameter files

You can split or merge Loggers based on the volume of I/Os. These statistics can be obtained using MEASURE statistics, a LOGDUMP COUNT command, or Replicat LAG STATS parameter. You can then change the Logger configuration without impacting the system by executing the GGSCI> ALTER LOGGER command.

There aren't any special BASE24 considerations for the Logger parameters. Your decision can be based solely on volume and size. However, see "Frequently Asked Questions" on page 104 with regards to UAF and Settlement.

All updates that are not column (field) mapped can be logged COMPRESSUPDATES. COMPRESSUPDATES means that only the data that has changed in a record is logged to the log trail. It is not by field, but by byte that the data is saved. For example, if only five consecutive bytes of a 300 character record has changed, the following is saved in the GoldenGate log trail:

- Character offset from the beginning of the record where the change has occurred (2 bytes)
- Number of consecutive characters changed (2 bytes)
- The changed characters (5 bytes)

A total of 9 bytes is written to the log trail rather than 300. This saves disk space and transmission time. At least an additional 48 bytes is inserted for each record as header information.

Extract

It is highly recommended that you extract the log trails on the source node and move them across to the target as fast as you can. The fastest way to do this is for Extract processes to copy log trails to the backup nodes as extract trails. The FASTREADS and FASTIO parameters allow up to 28K block-mode reads and writes.

The parameters in *italics* are for configuring the Extracts if the communication link is TCP/IP rather than Expand.

Figure 15 Sample Extract parameter for Extract group EXTTLF

```
********************************************************************
-- EXTTLF Extract Parameter File
--        Reads the Logtrails \PRODUCTION.<trail volume>.TLFLOGS.AA
--        and moves everything to the Extract Trail
--        \BACKUP.<trail volume>.TLFTRLS.AA
********************************************************************
EXTRACT EXTTLF

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTTLF, PURGE
-- Get all file operations
GETFILEOPS

-- Do not pad records that are not maximum size
NOFILLSHORTRECS

-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL <trail volume>.TLFTRLS.AA

-- Set the location of the Extract Trail (If EXPAND)
EXTTRAIL \BACKUP.<trail volume>.TLFTRLS.AA

-- Move all TLF files in the Logtrail to \BACKUP
FILE $*.*.*;
```

Figure 16 Sample Extract parameter file for Extract group EXTPTLF

```
********************************************************************
-- EXTPTLF  Extract Parameter File
--         Reads the Logtrails \PRODUCTION.<trail volume>.PTLFLOGS.AA
--         and moves everything to the Extract Trail
--         \BACKUP.<trail volume>.PTLFTRLS.AA
********************************************************************
EXTRACT EXTPTLF

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTPTLF, PURGE
-- Get all file operations
GETFILEOPS

-- Do not pad records that are not maximum size
NOFILLSHORTRECS

-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL <trail volume>.PTLFTRLS.AA

-- Set the location of the Extract Trail (If EXPAND)
EXTTRAIL \BACKUP.<trail volume>.PTLFTRLS.AA

-- Move all PTLF files in the Logtrail to \BACKUP
FILE $*.*.*;
```

Figure 17 Sample Extract parameter for Extract group EXTB24A

```
********************************************************************
-- EXTB24A  Extract Parameter File
--          Reads the Logtrails \PRODUCTION.<trail volume>.B24LOGS.AA
--          and moves everything to the Extract Trail
--          \BACKUP.<trail volume>.B24TRLS.AA
********************************************************************
EXTRACT EXTB24A

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTB24A, PURGE

-- Get all file operations
GETFILEOPS

-- Do not pad records that are not maximum size
NOFILLSHORTRECS
-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \BACKUP.<volume>.B24TRLS.AA

-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \BACKUP.<volume>.B24TRLS.AA

-- Move all files in the Logtrail to \BACKUP
FILE $*.*.*;
```

Figure 18 Sample Extract parameter file for Extract group EXTTMF

```
********************************************************************
-- EXTTMF   Extract Parameter File
--          Reads from the AUDSERV process and moves everything to the
--          Extract Trail \BACKUP.<trail volume>.TMFTRLS.AA
********************************************************************
EXTRACT EXTTMF

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTTMF, PURGE

-- Get all file operations
GETFILEOPS
-- Do not pad records that are not maximum size
NOFILLSHORTRECS

-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \BACKUP.<volume>.TMFTRLS.AA

-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \BACKUP.<volume>.TMFTRLS.AA

-- Move all files in the Logtrail to \BACKUP
FILE $*.*.*;
```

Replicat

The Replicat processes running on the target node are going to be the slowest of all GoldenGate components, as it has to replay all the I/Os in the source node.

Generally you will have a Replicat for each set of extract trails, however if required, more Replicats can be added to share the processing of a set of extract trails.

Figure 19 Sample Replicat parameter file for Replicat group REPTLF (passive mode)

```
********************************************************************
-- REPTLF   Replicat Parameter File
--         Reads the Remote/Extract Trails <trail volume>.TLFTRLS.AA
--         and applies all records to the target files.
********************************************************************
REPLICAT REPTLF

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPTLF, PURGE

-- Do block reads
FASTREADS

-- Replicat the file operations to the files
GETFILEOPS

-- Map financial and exception TLF records
MAP \PRODUCTION.<Base24 volume>.PRO1ATLF.T*,
TARGET \BACKUP.<Base24 volume>.*.*;
```

Figure 20 Sample Replicat parameter file for Replicat group REPPTLF (passive mode)

```
********************************************************************
-- REPPTLF  Replicat Parameter File
--         Reads the Remote/Extract Trails <trail volume>.PTLFTRLS.AA
--         and applies all records to the target files.
********************************************************************
REPLICAT REPPTLF

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPPTLF, PURGE

-- Replicat the file operations to the files
GETFILEOPS
```

```
-- Do block reads
FASTREADS

-- Map financial and exception PTLF records
MAP \PRODUCTION.<Base24 volume>.PRO1PTLF.P*,
TARGET \BACKUP.<Base24 volume>.*.*;
```

Figure 21 Sample Replicat parameter file for Replicat group REPTLF (active mode)

```
********************************************************************
-- REPTLF   Replicat Parameter File
--          It assumes the ‘create’ will always come before the first
--          insert for the new TLF file.
--
--          The ‘create’ will drop through the MAP statement and wait
--          for the file to be created.  Once the ‘create’ is
--          satisfied, the records will be inserted.
********************************************************************
REPLICAT REPTLF

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPTLF, PURGE

-- Set the dictionary location
DICTIONARY <base24 atm volume>.AT60DDL
-- Assume the target definitions
ASSUMETARGETDEFS

-- Do block reads
FASTREADS
-- Ignore the file operations to the files
IGNOREFILEOPS

-- Map financial and exception TLF records
MAP \PRODUCTION.<volume>.PRO1ATLF.T*, TARGET
\BACKUP.<volume>.PRO1ATLF.*,
    TARGETDEF TLF,
    NOCOLMAP,
    WHERE (TLF.HEAD.REC-TYP <> "00");

-- Get the file operations to the files
GETFILEOPS
```

```
-- Wait for the file to be created on the \BACKUP site
WAITFILEEVENT \BACKUP.<Base24 volume>.PRO1ATLF.T*, CREATE
```

Figure 22 Sample Replicat parameter file for REPPTLF (active mode)

```
********************************************************************
-- REPPTLF  Replicat Parameter File
--         It assumes the ‘create’ will always come before the first
--         insert for the new PTLF file.
--
--         The ‘create’ will drop through the MAP statement and then
--         wait for the file to be created.  Once the ‘create’ is
--         satisfified, the records will be inserted.
********************************************************************
REPLICAT REPPTLF

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPPTLF, PURGE

-- Set the dictionary location
DICTIONARY <base24 pos volume>.PS60DDL
-- Assume the target definitions
ASSUMETARGETDEFS

-- Do block reads
FASTREADS
-- Ignore the file operations to the files
IGNOREFILEOPS

-- Map financial and exception PTLF records
MAP \PRODUCTION.<volume>.PRO1PTLF.P*, TARGET
\BACKUP.<volume>.PRO1PTLF.*,
   TARGETDEF PTLF,
   NOCOLMAP,
    WHERE (PTLF.HEAD.REC-TYP <> "00");

-- Get the file operations to the files
GETFILEOPS

-- Wait for the file to be created on the \BACKUP site
WAITFILEEVENT \BACKUP.<Base24 volume>.PRO1PTLF.P*, CREATE
```

Figure 23 Sample Replicat parameter file for Replicat group REPB24A

```
*****************************************************************
-- REPB24A  Replicat Parameter File
--          This Replicat reads the Remote/Extract Trail:
-- <trail volume>.B24TRLS.AA and maps the BASE24 records to
--          the Target database.
*****************************************************************
REPLICAT REPB24A

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.REPB24A, PURGE

-- Set the dictionary location
DICTIONARY <base24 base volume>.BA60DDL
-- Only used during Initial Loads
-- HANDLECOLLISIONS
-- Use the target dictionary
ASSUMETARGETDEFS

-- Replicat the file operations to the files
GETFILEOPS
-- Do block reads
FASTREADS

-- Map the LCONF file and substitute NonStop node and volume.
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
--  substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \PRODUCTION.<volume>.PRO*CNTL.L*CONF, TARGET
\BACKUP.<volume>.*.*,
    TARGETDEF LCONF,
    COLMAP
        (PRIKEY  = PRIKEY,
        PROD-IND = PROD-IND,
        LCONF.COMMENTS = @STRSUB (LCONF.COMMENTS, "\PRODUCTION",
            "\BACKUP", “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
        LAST-CHNG-TIME = LAST-CHNG-TIME,
        FILE-NAME  = @STRSUB (FILE-NAME, "\PRODUCTION", "\BACKUP",
                          “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
        TEMPLATE   = @STRSUB (TEMPLATE,  "\PRODUCTION", "\BACKUP",
                           “PRODVOL”, “BKUPVOL”,“PRODSUBV”, “BKUPSUBV”),
        USER-FIELD = USER-FIELD,
        USER-FLD2  = USER-FLD2,
        PROD-IND-ADNL = PROD-IND-ADNL,
        USER-FLD4     = USER-FLD4,
        LAST-AFM      = LAST-AFM),
    WHERE (ITEM-TYP = "A");

-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \PRODUCTION.<volume>.PRO*CNTL.L*CONF, TARGET
\BACKUP.<volume>.*.*,
    TARGETDEF LCONF,
    COLMAP
        (PRIKEY  = PRIKEY,
        PROD-IND = PROD-IND,
        LCONF.COMMENTS = @STRSUB (LCONF.COMMENTS, "\PRODUCTION",
                                                  "\BACKUP"),
        LAST-CHNG-TIME = LAST-CHNG-TIME,
        PLGTH     = PLGTH,
        PTXT      = @STRSUB (PTXT, "\PRODUCTION", "\BACKUP"),
        USER-FLD3 = USER-FLD3,
        PROD-IND-ADNL = PROD-IND-ADNL,
        USER-FLD4     = USER-FLD4,
        LAST-AFM      = LAST-AFM),
    WHERE (ITEM-TYP = "P");

-- MAP the CAF file with EXCEPTIONSONLY mapping
MAP \PRODUCTION.<vol>.PRO1DATA.CAF,    TARGET
\BACKUP.<vol>.PRO1DATA.CAF;
MAP \PRODUCTION.<vol>.PRO1DATA.CAF,    TARGET
\BACKUP.<vol>.PRO1DATA.OLDCAF,
    EXCEPTIONSONLY;
MAP \PRODUCTION.<vol>.PRO1DATA.OLDCAF, TARGET
\BACKUP.<vol>.PRO1DATA.OLDCAF;
MAP \PRODUCTION.<vol>.PRO1DATA.OLDCAF, TARGET
\BACKUP.<vol>.PRO1DATA.CAF,
    EXCEPTIONSONLY;
```

```
-- Map the IDF file
MAP \SITEA.<volume>.PRO1DATA.IDF, TARGET
\SITEB.<volume>.PRO1DATA.IDF,
   TARGETDEF IDF,
   COLMAP (USEDEFAULTS,
         NEG-NAME  = @STRSUB (NEG-NAME,  "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
         UAF-NAME  = @STRSUB (UAF-NAME,  "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
         CAF-NAME  = @STRSUB (CAF-NAME,  "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
         PBF1-NAME = @STRSUB (PBF1-NAME, "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
         PBF2-NAME = @STRSUB (PBF2-NAME, "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”),
         PBF3-NAME = @STRSUB (PBF3-NAME, "\PRODUTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”,  “BKUPSUBV”),
         PBF4-NAME = @STRSUB (PBF4-NAME, "\PRODUCTION", "\BACKUP",
                    “PRODVOL”, “BKUPVOL”, “PRODSUBV”, “BKUPSUBV”));

-- MAP the PBF file with EXCEPTIONSONLY mapping
MAP \PRODUCTION.<vol>.PRO1DATA.PBF,    TARGET
\BACKUP.<vol>.PRO1DATA.PBF;
MAP \PRODUCTION.<vol>.PRO1DATA.PBF,    TARGET
\BACKUP.<vol>.PRO1DATA.OPBFDA,
   EXCEPTIONSONLY;
MAP \PRODUCTION.<vol>.PRO1DATA.OPBFDA, TARGET
\BACKUP.<vol>.PRO1DATA.OPBFDA;
MAP \PRODUCTION.<vol>.PRO1DATA.OPBFDA, TARGET
\BACKUP.<vol>.PRO1DATA.PBF,
   EXCEPTIONSONLY;

MAP \PRODUCTION.<vol>.PRO1DATA.NEWC*,  TARGET
\BACKUP.<vol>.PRO1DATA.*;
MAP \PRODUCTION.<vol>.PRO1DATA.NCAF*,  TARGET
\BACKUP.<vol>.PRO1DATA.*;
MAP \PRODUCTION.<vol>.PRO1DATA.NPBF*,  TARGET
\BACKUP.<vol>.PRO1DATA.*;

-- Map all the BASE24 data files not replicated in other Replicats
MAP \PRODUCTION.<vol>.PRO1DATA.*,      TARGET
\BACKUP.<vol>.PRO1DATA.*;
```

```
MAP \PRODUCTION.<vol>.PRO2DATA.*,      TARGET
\BACKUP.<vol>.PRO2DATA.*;
MAP \PRODUCTION.<vol>.PRO1EXEC.*,      TARGET
\BACKUP.<vol>.PRO1EXEC.*;
MAP \PRODUCTION.<vol>.PRO2EXEC.*,      TARGET
\BACKUP.<vol>.PRO2EXEC.*;
MAP \PRODUCTION.<vol>.PRO1CNTL.*,      TARGET
\BACKUP.<vol>.PRO1CNTL.*;
MAP \PRODUCTION.<vol>.PRO2CNTL.*,      TARGET
\BACKUP.<vol>.PRO2CNTL.*;
```

Figure 24 Sample Replicat parameter file for Replicat group REPPTMF

```
*******************************************************************
*
-- REPTMF   Replicat Parameter File
--         Reads the Remote/Extract Trails <trail volume>.TMFTRLS.AA
--          and applies all records to the target files.
*******************************************************************
REPLICAT REPTMF

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.REPTMF, PURGE

-- Replicat the file operations to the files
GETFILEOPS

-- Do block reads
FASTREADS

-- Map financial and exception PTLF records
MAP \PRODUCTION.<volume>.*.*, TARGET \BACKUP.<volume>.*.*;
```

Adding and starting GoldenGate components

To begin hot site backups, you must start all the GoldenGate components you added in the last steps. Components to add and start include:

- Manager on the production site
- Manager on the backup site

- Logger on the production site
- Extract on the production site
- Replicat on the backup site

Start Manager on the production site

Issue the following command:

```
GGSCI> START MANAGER
```

The Manager process is responsible for keeping Extracts and Replicats running and maintaining extract trails. The Manager process is part of the default GoldenGate environment. It is automatically added and only has to be started.

Start Manager on the backup site

```
GGSCI> START MANAGER
```

The Manager process is responsible for keeping Extracts and Replicats running and maintaining extract trails. The Manager process is part of the default GoldenGate environment. It is automatically added and only has to be started.

Add GoldenGate components for the production site

Issue the following commands:

```
GGSCI> ADD LOGGER
```

This command adds the Logger processes to the GoldenGate environment.

```
GGSCI> ADD EXTRACT EXTTLF, LOGTRAILSOURCE <trail volume>.TLFLOGS.AA
```

This command adds the EXTTLF Extract process to read the <trail volume>.TLFLOGS.AA logtrail.

```
GGSCI> ADD EXTRACT EXTPTLF, LOGTRAILSOURCE <trail volume>.PTLFLOGS.AA
```

This command adds the EXTPTLF Extract process to read the <trail volume>.PTLFLOGS.AA log trail.

```
GGSCI> ADD EXTRACT EXTB24A, LOGTRAILSOURCE <trail volume>.B24LOGS.AA
```

This command adds the EXTB24A Extract process to read the <trail volume>.B24LOGS.AA log trail.

```
GGSCI> ADD EXTTRAIL \BACKUP. <trail volume>.TLFTRLS.AA, EXTRACT
EXTTLF
```

This command adds the Extract Trail to the GoldenGate environment. The actual file is not created until the first log record is written to the target site.

```
GGSCI> ADD EXTTRAIL \BACKUP. <trail volume>.PTLFTRLS.AA, EXTRACT
EXTPTLF
```

This command adds the Extract Trail to the GoldenGate environment. The actual file is not created until the first log record is written to the target site.

```
GGSCI> ADD EXTTRAIL \BACKUP. <trail volume>.B24TRLS.AA, EXTRACT
EXTB24A
```

This command adds the Extract Trail to the GoldenGate environment. The actual file is not created until the first log record is written to the target site.

Add GoldenGate components on the backup site

Add Replicat on your backup site by issuing the following commands:

```
GGSCI> ADD REPLICAT REPTLF, EXTTRAILSOURCE <trail volume>.TLFTRLS.AA
```

This command adds the Replicat process REPTLF. This Replicat is used to deliver the transaction log file changes from \PRODUCTION to \BACKUP.

```
GGSCI> ADD REPLICAT REPPTLF, EXTTRAILSOURCE <trail
volume>.PTLFTRLS.AA
```

This command adds the Replicat process REPPTLF. This Replicat is used to deliver the transaction log file changes from \PRODUCTION to \BACKUP.

```
GGSCI> ADD REPLICAT REPB24A, EXTTRAILSOURCE <trail volume>.B24TRLS.AA
```

This command adds the Replicat process REPB24A. This Replicat is used to deliver the BASE24 file changes from \PRODUCTION to \BACKUP.

Start GoldenGate components on the production site:

Start GoldenGate on the production site by issuing the following commands:

Start Logger

```
GGSCI> START LOGGER
```

This command starts the Logger process(es). The information in the Logparm file that is used by the Logger and intercept libraries is loaded into each CPU's memory.

Start Extract

```
GGSCI> START EXTTLF
```

This command starts the EXTTLF Extract process. At this point this Extract will start reading the log trails on \PRODUCTION.<trail volume>.TLFLOGS.AA and writing the log records to \BACKUP.<trail volume>.TLFTRLS.AA extract trails.

```
GGSCI> START EXTPTLF
```

This command starts the EXTPTLF Extract process. At this point this Extract will start reading the log trails on \PRODUCTION.<trail volume>.PTLFLOGS.AA and writing the log records to \BACKUP.<trail volume>.PTLFTRLS.AA extract trails.

```
GGSCI> START EXTB24A
```

At this point the EXTB24A Extract will start reading the log trails on \PRODUCTION.<trail volume>.B24LOGS.AA and writing the log records to \BACKUP.<trail volume>.B24TRLS.AA extract trails.

Start GoldenGate on the backup site

Start GoldenGate on the backup site by issuing the following commands:

Start Replicat

```
GGSCI> START REPTLF
```

This command starts the Replicat process REPTLF. This process will read the extract records from its extract trail and update the BASE24 transaction log files

```
GGSCI> START REPPTLF
```

This command starts the Replicat process REPPTLF. This process will read the extract records from its extract trail and update the BASE24 transaction log files

```
GGSCI> START REPB24A
```

This command starts the Replicat process REPB24A. This process will read the extract records from its extract trail and update the other BASE24 data files.

CHAPTER 6

GoldenGate for Base24 Solutions: Bi-Directional Processing

This chapter outlines how to implement GoldenGate for BASE24 as a bi-directional processing solution. Each step required for such an implementation is discussed in the following topics:

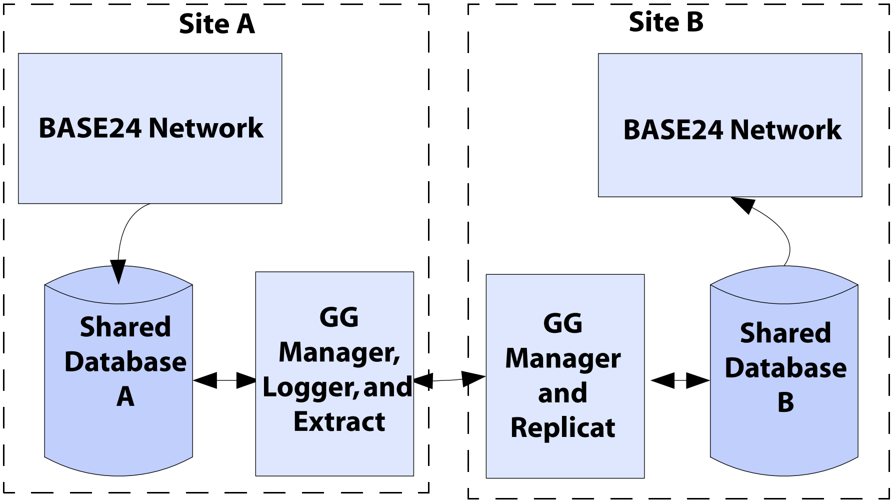
Contents

Overview

Bi-directional processing provides the most complete business continuity service available. Processing is performed on both sites, with database changes replicated to both sites. The databases are kept in-sync with each other so either site can take over processing if the other site fails. This means that in the event a site is disabled, only a portion of the network is momentarily down. The other site is configured to process the total environment.

For planned down situations such as installing a new operating system, BASE24 release, or GoldenGate version, the total system would continue to run without interruption until the 'downed' site is ready to come back on-line. Then processing would go back to being shared between the two systems.

Figure 25 Bi-directional solution



Site A	Site B
BASE24: The terminal files are the same on both sides. However only half the terminals are active on site A; these same terminals are inactive on site B.	**BASE24**: The terminal files are the same on both sides. However only half the terminals are active on site B; these same terminals are inactive on site A.
Transaction Processing: All transactiona acquired on site A are authorized on site A. There is no BASE24 routing between sites. All database changes are replicated to site B.	**Transaction Processing**: All transactions acquired on site B are authorized on site B. There is no BASE24 routing between sites. All database changes are replicated to site A.
Refresh: Refreshes run on one site and replicate to the other.	**Refresh**: Refreshes run on one site and replicate to the other.
Extract: Super Extract can be run from either site.	**Extract**: Super Extract can be run from either site.
Settlement: Settlement is run at each site and is not replicated.	**Settlement**: Settlement is run at each site and is not replicated.

GoldenGate components

The GoldenGate setup described in this solution is for a bi-directional environment with the following GoldenGate components:

Site A components

- One Manager
- Two Loggers:
  - **TLF**: This Logger is dedicated to logging file creations and record inserts to the ATM TLF files and POS PTLF files on site A. These log records will be generated for every ATM and POS transaction whether it is approved or not. These records are always the full record image because only record inserts are logged.
  - **BASE24:** This Logger is dedicated to logging everything else. Even though some of the files included in this Logger are high volume, (i.e. TDF, PTDF, PRDF) the updates will be compressed. This will minimize the amount of data that is moved between the source and target nodes. However, some files such as the LCONF and IDF may need to do string substitutions for the HP NonStop Server node name or the volume and subvolume names. The changes to these files must be in the uncompressed update format (full record image) to allow such field mapping.
- Three Extracts:
  - **EXTTLFA:** This Extract acts as a data pump. It moves all data in the TLFLOGS.AA log trails on site A to the TLFTRLS.AA extract trails on site B.
  - **EXTB24A:** This Extract acts as a data pump. It moves all the data in the B24LOGS.AA log trails to the B24TRLS.AA extract trails on site B.
  - **EXTTMFA:** This Extract moves the selected data from the TMF Audit Logs to the TMFTRLS.AA extract trails on site B.
- Three Replicats:

- **REPTLFB:** This Replicat applies all the TLF records from the TLFTRLS.BB extract trails to the BASE24 database on site B.
- **REPB24B:** This Replicat applies all the BASE24 data records from the B24TRLS.BB extract trails to the BASE24 downside on site B. This includes all file operations.
- **REPTMFB:** This Replicat applies all the BASE24 TMF data records from the TMFTRLS.BB extract trails to the BASE24 database on site B.

Site B components

- One Manager
- Two Loggers:
  - **TLF:** This Logger is dedicated to logging creation and inserts to the ATM TLF files and POS PTLF files on site B. These log records will be generated for every ATM and POS transaction whether it is approved or not. These records are always the full record image because only record inserts are logged.
  - **BASE24:** This Logger is dedicated to logging everything else. Even though some of the files included in this Logger are high volume, (i.e. TDF, PTDF, PRDF) the updates will be compressed. This will minimize the amount of data that is moved between source and target nodes. However, some files such as the LCONF and IDF may need to do string substitutions for the HP NonStop Server node name or volume and subvolume names. The changes to these files must be in the uncompressedupdate format to allow such field mapping.
- Three Extracts:
  - **EXTTLFB:** This Extract acts as a data pump. It moves all the data in the TLFLOGS.BB log trails to the TLFTRLS.BB extract trails on site A.
  - **EXTB24B:** This Extract acts as a data pump. It moves all the data in the B24LOGS.BB log trails to the B24TRLS.BB extract trails on site A.
  - **EXTTMFB:** This Extract moves the selected data from the TMF Audit Logs to the TMFTRLS.BB extract trails on site A.

- Three Replicats:
    - **REPTLFA:** This Replicat applies all the TLF records from the TLFTRLS.AA extract trails to the BASE24 database on site A.
    - **REPB24A:** This Replicat applies all the BASE24 data records from the B24TRLS.AA extract trails to the BASE24 database on site A. This includes all file operations.
    - **REPTMFA**: This Replicat applies all the BASE24 TMF data records from the TMFTRLS.AA extract trails to the BASE24 database on site A.

Sample parameter files

This section details sample parameter files for the following GoldenGate components:

Site A components

Figure 26 Global parameter file

```
HOST \SITEB, GGSSUBVOL <volume>.GGS(version) , NODENUM <node number>
DELETE DEFINE =GGS_AUDCFG
DELETE DEFINE =GGS_PREFIX
ADD DEFINE =GGS_AUDCFG, CLASS MAP, FILE $SYSTEM.GGS.AUDCFG
ADD DEFINE =GGS_PREFIX, CLASS MAP, FILE $GG
```

Figure 27 Manager parameter file

```
--*****************************************************************
-- MANAGER
--*****************************************************************
-- Manager port for the Extracts on the \SITEB site to use as
-- RMTHOST mgrport
PORT <manager port number> (IF TCP/IP)

-- Keep the (P)TLF Remote/Extract Trails for at least 2 days and
-- processed
PURGEOLDEXTRACTS <volume>.TLFTRLS.BB, USECHECKPOINTS, MINKEEPDAYS 2
```

```
-- Keep the rest of the BASE24 data files Remote/Extract Trails for
-- at least 2 days and processed
PURGEOLDEXTRACTS <volume>.B24TRLS.BB, USECHECKPOINTS, MINKEEPDAYS 2
```

Figure 28 Logger 0 parameter file

```
--****************************************************************
-- LOGGER 0  -  Captures TLF  from PRO1ATLF
--              Captures PTLF from PRO1PTLF
--****************************************************************

-- Location, number and size of the Logger trails
LOG <volume>.TLFLOGS.AA , MEGABYTES <megabytes>, NUMFILES <num>,
SECURE "NCNC"

-- Primary and backup CPU for Logger 0
CPU 0,1

-- Get unstructured files
GETUNSTRUCTURED

-- Get bulk loads
GETBULKIO

-- Use the full record image, do not compress the updates
NOCOMPRESSUPDATES

-- Make priority higher than BASE24 nucleus (NETWORK)
PRI 180

-- List the files to be captured
FILE <volume>.PRO1ATLF.T*
FILE <volume>.PRO1PTLF.P*
```

Figure 29 Logger 1 parameter file

```
--****************************************************************
-- LOGGER 1  -  Captures all BASE24 data files
--****************************************************************

LOG <volume>.B24LOGS.AA , MEGABYTES <megabytes>, NUMFILES <num>,
SECURE "NCNC"
```

```
-- Primary and backup CPU for Logger 1
CPU 2,3

-- Get unstructured files
GETUNSTRUCTURED

-- Get bulk loads
GETBULKIO

-- Make priority higher than BASE24 nucleus (NETWORK)
PRI 180

-- List the files to be captured
FILE <volume>.PRO1DATA.*       COMPRESSUPDATES
FILE <volume>.PRO1EXEC.*,      COMPRESSUPDATES
FILE <volume>.PRO1CNTL.L*CONF  NOCOMPRESSUPDATES
FILE <volume>.PRO1CNTL.*,      COMPRESSUPDATES

-- List the files to be excluded by the Settlement program
EXCLUDEFILE <volume>.PRO1DATA.TDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.PTDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.ATD*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.PTD*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.IDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.PRDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.FF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.F0*, PROGRAM <volume>.PRO1OBJ.*SETL
```

Extract Parameter File for Extract Group EXTB24A

```
--*****************************************************************
-- EXTB24A - This Extract reads the Log trails
--           \SITEA.<volume>.B24LOGS.AA and moves everything to
--           the Remote Trail \SITEB.<volume>.B24TRLS.AA
-- This includes all BASE24 data files not in the other Extracts
--*****************************************************************

EXTRACT EXTB24A

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.EXTB24A, PURGE
```

```
-- Get all file operations
GETFILEOPS

-- Do not pad records that are not maximum size
NOFILLSHORTRECS
-- Do not check if the source files exist, just pass all files through
PASSTHRU
-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>

-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \SITEB.<volume>.B24TRLS.AA
-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \SITEB.<volume>.B24TRLS.AA

-- Move all files in the Logtrail to \SITEB
FILE $*.*.*;
```

Figure 30 Extract parameter file for Extract group EXTTLFA

```
--****************************************************************
-- EXTTLFA - This Extract reads the Log trails on
--           \SITEA.<volume>.TLFLOGS.AA and moves everything to
--           the Remote Trail \SITEB.<volume>.TLFTRLS.AA
--           This includes TLF and PTLF files.
--****************************************************************

EXTRACT EXTTLFA

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.EXTTLFA, PURGE

-- Ignore all file operations
IGNOREFILEOPS
```

```
-- Do not pad records that are not maximum size
NOFILLSHORTRECS

-- Do block writes
FASTIO
-- Do block reads
FASTREADS
-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>

-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \SITEB.<volume>.TLFTRLS.AA
-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \SITEB.<volume>.TLFTRLS.AA

-- Move all TLF files in the Logtrail to \SITEB except header record
FILE $*.*.*;
```

Figure 31 Extract parameter file for Extract group EXTTMFA

```
_***************************************************************************
-- EXTTMFA  Extract Parameter File
--          Reads from the AUDSERV process and moves everything
--          to the Extract Trail \SITEB.<trail volume>.TMFTRLS.AA
--**************************************************************************

EXTRACT EXTTMFA

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTTMFA, PURGE

-- Get all file operations
GETFILEOPS
-- Do not pad records that are not maximum size
NOFILLSHORTRECS
```

```
-- Do not check if the source files exist, just pass all files through
PASSTHRU
-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>

-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \SITEB.<volume>.TMFTRLS.AA

-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \SITEB.<volume>.TMFTRLS.AA

-- Move all files in the Logtrail to \SITEB
FILE <Base24 volume>.PRO1DATA.*;
```

Figure 32 Replicat parameter file for Replicat group REPB24B

```
--***************************************************************
-- REPB24B - This Replicat reads the Remote/Extract Trail
--           <volume>.B24TRLS.BB and Replicates all
--           BASE24 data files from Site B
--***************************************************************

REPLICAT REPB24B

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPB24B, PURGE
-- Exception Processing for Error 11
REPERROR 11, EXCEPTION

-- Set the dictionary location
DICTIONARY <base24 base volume>.BA60DDL
-- Only used during Initial Loads
-- HANDLECOLLISIONS
-- Use the target dictionary
ASSUMETARGETDEFS
```

```
-- Replicat the file operations to the files
GETFILEOPS
-- Do block reads
FASTREADS

-- MAP the CAF file with EXCEPTIONSONLY mapping
MAP \SITEB.<volume>.PRO1DATA.CAF,
   TARGET \SITEA.<volume>.PRO1DATA.CAF;
MAP \SITEB.<volume>.PRO1DATA.CAF,
   TARGET \SITEA.<volume>.PRO1DATA.OLDCAF,
   EXCEPTIONSONLY;
MAP \SITEB.<volume>.PRO1DATA.OLDCAF,
   TARGET \SITEA.<volume>.PRO1DATA.OLDCAF;
MAP \SITEB.<volume>.PRO1DATA.OLDCAF,
   TARGET \SITEA.<volume>.PRO1DATA.CAF,
   EXCEPTIONSONLY;

-- MAP the PBF file with EXCEPTIONSONLY mapping
MAP \SITEB.<volume>.PRO1DATA.PBF,    TARGET
\SITEA.<volume>.PRO1DATA.PBF;
MAP \SITEB.<volume>.PRO1DATA.PBF,    TARGET
\SITEA.<volume>.PRO1DATA.OPBFDA,
   EXCEPTIONSONLY;
MAP \SITEB.<volume>.PRO1DATA.OPBFDA, TARGET
\SITEA.<volume>.PRO1DATA.OPBFDA;
MAP \SITEB.<volume>.PRO1DATA.OPBFDA, TARGET
\SITEA.<volume>.PRO1DATA.PBF,
   EXCEPTIONSONLY;

MAP \SITEB.<volume>.PRO1DATA.NEWC*,
   TARGET \SITEA.<volume>.PRO1DATA.*;
MAP \SITEB.<volume>.PRO1DATA.NCAF*,
   TARGET \SITEA.<volume>.PRO1DATA.*;
MAP \SITEB.<volume>.PRO1DATA.NPBF*,
   TARGET \SITEA.<volume>.PRO1DATA.*;

-- Map the IDF file
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
```

```
MAP \SITEB.<volume>.PRO1DATA.IDF,
   TARGET \SITEA.<volume>.PRO1DATA.IDF,
   TARGETDEF IDF,
   COLMAP (USEDEFAULTS,
      NEG-NAME     = @STRSUB (NEG-NAME,  "\SITEB",  "\SITEA”,
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
      UAF-NAME     = @STRSUB (UAF-NAME,  "\SITEB",  "\SITEA",
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
      CAF-NAME     = @STRSUB (CAF-NAME,  "\SITEB",  "\SITEA",
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
      PBF1-NAME      = @STRSUB (PBF1-NAME, "\SITEB", "\SITEA",
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
      PBF2-NAME    = @STRSUB (PBF2-NAME, "\SITEB", "\SITEA",
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
      PBF3-NAME    = @STRSUB (PBF3-NAME, "\SITEB",  "\SITEA",
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
      PBF4-NAME    = @STRSUB (PBF4-NAME, "\SITEB",  "\SITEA",
                      “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”));

-- Map the LCONF file and substitute NonStop node and volume.
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \SITEB.<volume>.PRO1CNTL.L*CONF,
   TARGET \SITEA.<volume>.PRO1CNTL.*,
   TARGETDEF LCONF,
   COLMAP (PRIKEY       = PRIKEY,
          PROD-IND      = PROD-IND,
          LCONF.COMMENTS = @STRSUB (LCONF.COMMENTS, "\SITEB", "\SITEA",
                          “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
          LAST-CHNG-TIME = LAST-CHNG-TIME,
          FILE-NAME     = @STRSUB (FILE-NAME, "\SITEB", "\SITEA",
                           “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
          TEMPLATE      = @STRSUB (TEMPLATE, "\SITEB",  "\SITEA",
                            “VOLB”, “VOLA”, “SUBVOLB”, “SUBVOLA”),
          USER-FIELD    = USER-FIELD,
          USER-FLD2     = USER-FLD2,
          PROD-IND-ADNL  = PROD-IND-ADNL,
          USER-FLD4     = USER-FLD4,
          LAST-AFM      = LAST-AFM),
   WHERE (ITEM-TYP = "A");
```

```
-- Map the PRO1CNTL file
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \SITEB.<volume>.PRO1CNTL.L*CONF,
   TARGET \SITEA.<volume>.PRO1CNTL.*,
   TARGETDEF LCONF,
   COLMAP (PRIKEY       = PRIKEY,
          PROD-IND      = PROD-IND,
          LCONF.COMMENTS = @STRSUB (LCONF.COMMENTS, "\SITEB", "\SITEA"),
          LAST-CHNG-TIME = LAST-CHNG-TIME,
          PLGTH         = PLGTH,
          PTXT          = @STRSUB (PTXT, "\SITEB", "\SITEA"),
          USER-FLD3     = USER-FLD3,
          PROD-IND-ADNL  = PROD-IND-ADNL,
          USER-FLD4     = USER-FLD4,
          LAST-AFM      = LAST-AFM),
   WHERE (ITEM-TYP = "P");

-- Map the remaining BASE24 data files
MAP \SITEB.<volume>.PRO1DATA.*, TARGET \SITEA.<volume>.PRO1DATA.*;
MAP \SITEB.<volume>.PRO1EXEC.*, TARGET \SITEA.<volume>.PRO1EXEC.*;
MAP \SITEB.<volume>.PRO1CNTL.*, TARGET \SITEA.<volume>.PRO1CNTL.*;
```

Figure 33 Replicat parameter file for Replicat group REPTLFB

```
--******************************************************************
-- REPTLFB - This Replicat reads the Remote/Extract Trails
--           <volume>.TLFTRLS.BB from Site B and
--           MAPS the ATM and POS Transaction Log Files
--******************************************************************
REPLICAT REPTLFB

-- Set the dictionary location
DICTIONARY <base24 atm volume>.AT60DDL
-- Use the target site definitions
ASSUMETARGETDEFS
-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPTLFB, PURGE

-- Ignore the file operations to the combined A+B (P)TLF files
IGNOREFILEOPS
```

```
-- Do block reads
FASTREADS

-- Map financial and exception TLF records
MAP \SITEB.<volume>.PRO1ATLF.T*, TARGET \SITEA.<volume>.PRO1ATLF.*,
   TARGETDEF TLF,
   NOCOLMAP,
   WHERE (TLF.HEAD.REC-TYP <> "00");

-- Set the dictionary location
DICTIONARY <base24 pos volume>.PS60DDL

-- Map financial and exception PTLF records
MAP \SITEB.<volume>.PRO1PTLF.P*, TARGET \SITEA.<volume>.PRO1PTLF.*,
   TARGETDEF PTLF,
   NOCOLMAP,
   WHERE (PTLF.HEAD.REC-TYP <> "00");
```

Figure 34 Replicat parameter file for Replicat group REPTMFB

```
--******************************************************************
-- REPTMFB  Replicat Parameter File
--         Reads the Remote/Extract Trails <trail volume>.TMFTRLS.BB
--          and applies all records to the target files.
--******************************************************************

REPLICAT REPTMFB

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.REPTMFB, PURGE
-- Replicat the file operations to the files
GETFILEOPS
-- Do block reads
FASTREADS

-- Map financial and exception PTLF records
MAP \SITEB.<Base24 volume>.*.*, TARGET \SITEA.<Base24 volume>.*.*;
```

Site B components

Figure 35 Global parameter file

```
HOST \SITEA, GGSSUBVOL <volume>.GGS(version) , NODENUM <node number>
DELETE DEFINE =GGS_AUDCFG
DELETE DEFINE =GGS_PREFIX
ADD DEFINE =GGS_AUDCFG, CLASS MAP, FILE $SYSTEM.GGS.AUDCFG
ADD DEFINE =GGS_PREFIX, CLASS MAP, FILE $GG
```

Figure 36 Manager parameter file

```
--*****************************************************************
-- MANAGER
--*****************************************************************
-- Manager port for the Extracts on the \SITEA site
-- to use as RMTHOST mgrport
PORT <manager port number> (If TCP/IP)

-- Keep the (P)TLF Remote/Extract Trails for at least 2 days
-- and processed
PURGEOLDEXTRACTS <volume>.TLFTRLS.AA, USECHECKPOINTS, MINKEEPDAYS 2

-- Keep the rest of the BASE24 data files Remote/Extract Trails for
-- at least 2 days and processed
PURGEOLDEXTRACTS <volume>.B24TRLS.AA, USECHECKPOINTS, MINKEEPDAYS 2
```

Figure 37 Logger 0 parameter file

```
--***************************************************************
-- LOGGER 0  -  Captures TLF  from PRO1ATLF
--              Captures PTLF from PRO1PTLF
--****************************************************************

-- Location, number and size of the Logger trails
LOG <volume>.TLFLOGS.BB , MEGABYTES <megabytes>, NUMFILES <num>,
SECURE "NCNC"

-- Primary and backup CPU for Logger 0
CPU 2,3
```

```
-- Get unstructured files
GETUNSTRUCTURED

-- Get bulk loads
GETBULKIO

-- Use the full record image, do not compress the updates
NOCOMPRESSUPDATES

-- Make priority higher than BASE24 nucleus (NETWORK)
PRI 180

-- List the files to be captured
FILE <volume>.PRO1ATLF.T*
FILE <volume>.PRO1PTLF.P*
```

Figure 38 Logger 1 parameter file

```
--*****************************************************************
-- LOGGER 1  -  Captures all BASE24 data files
--*****************************************************************

LOG <volume>.B24LOGS.BB , MEGABYTES <megabytes>, NUMFILES <num>,
SECURE "NCNC"
-- Primary and backup CPU for Logger 1
CPU 2,3

-- Get unstructured files
GETUNSTRUCTURED

-- Get bulk loads
GETBULKIO

-- Make priority higher than BASE24 nucleus (NETWORK)
PRI 180

-- List the files to be captured
FILE <volume>.PRO1DATA.*,       COMPRESSUPDATES
FILE <volume>.PRO1EXEC.*,       COMPRESSUPDATES
FILE <volume>.PRO1CNTL.L*CONF,  NOCOMPRESSUPDATES
FILE <volume>.PRO1CNTL.*,       COMPRESSUPDATES
```

```
-- List the files to be excluded by the Settlement program
EXCLUDEFILE <volume>.PRO1DATA.TDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.PTDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.ATD*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.PTD*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.IDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.PRDF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.FF*, PROGRAM <volume>.PRO1OBJ.*SETL
EXCLUDEFILE <volume>.PRO1DATA.F0*, PROGRAM <volume>.PRO1OBJ.*SETL
```

Figure 39 Extract parameter file for Extract group EXTB24B

```
--*****************************************************************
-- EXTB24B - This Extract reads the Log trails
--           \SITEB.<volume>.B24LOGS.BB and
--           moves everything to the Remote Trail
--           \SITEA.<volume>.B24TRLS.BB
-- This includes all BASE24 data files not in the other Extracts
--*****************************************************************

EXTRACT EXTB24B

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.EXTB24B, PURGE
-- Get all file operations
GETFILEOPS
-- Do not pad records that are not maximum size
NOFILLSHORTRECS
-- Do not check if the source files exist, just pass all files through
PASSTHRU
-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \SITEA.<volume>.B24TRLS.BB
-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \SITEA.<volume>.B24TRLS.BB

-- Move all files in the Logtrail to \SITEA
FILE $*.*.*;
```

Figure 40 Extract parameter file for Extract group EXTTLFB

```
--*****************************************************************
-- EXTTLFB - This Extract reads the Log trails
--           \SITEB.<volume>.TLFLOGS.BB and
--           moves everything to the Remote Trail
--           \SITEA.<volume>.TLFTRLS.BB
-- This includes TLF and PTLF files.
--*****************************************************************
EXTRACT EXTTLFB

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.EXTTLFB, PURGE

-- Ignore all file operations
IGNOREFILEOPS

-- Do not pad records that are not maximum size
NOFILLSHORTRECS

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \SITEA.<volume>.TLFTRLS.BB
-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \SITEA.<volume>.TLFTRLS.BB

-- Move all TLF files in the Logtrail to \SITEA except header record
FILE $*.*.*;
```

Figure 41 Extract parameter file for Extract group EXTTMFB

```
--*******************************************************************
-- EXTTMFB  Extract Parameter File
--         Reads from the AUDSERV process and moves everything to
--         the Extract Trail \SITEA.<trail volume>.TMFTRLS.BB
-********************************************************************

EXTRACT EXTTMFB

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.EXTTMFB, PURGE

-- Get all file operations
GETFILEOPS

-- Do not pad records that are not maximum size
NOFILLSHORTRECS

-- Do not check if the source files exist, just pass all files through
PASSTHRU

-- Do block writes
FASTIO
-- Do block reads
FASTREADS

-- Set the TCP/IP process name (If TCP/IP)
-- TCPIPPROCESSNAME <tcp/ip process name>
-- set the TCP/IP address and the manager port number (If TCP/IP)
-- RMTHOST <tcp/ip address> , MGRPORT <manager port number>
-- Set the location of the Remote Trail (If TCP/IP)
-- RMTTRAIL \SITEA.<volume>.TMFTRLS.BB

-- Set the location of the Remote Trail (If EXPAND)
EXTTRAIL \SITEA.<volume>.TMFTRLS.BB
```

```
-- Move all files in the Logtrail to \SITEA
FILE <Base24 volume>.PRO1DATA.*;
```

Figure 42 Replicat parameter group; moves files from site B to site A

```
-- **************************************************************
-- REPB24A - This Replicat reads the Remote/Extract Trail
--           <volume>.B24TRLS.BB
--           and replicates all BASE24 data files from Site A
--***************************************************************

REPLICAT REPB24A

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPB24A, PURGE
-- Exception Processing for Error 11
REPERROR 11, EXCEPTION

-- Set the dictionary location
DICTIONARY <base24 base volume>.BA60DDL
-- Only used during Initial Loads
-- HANDLECOLLISIONS
-- Use the target dictionary
ASSUMETARGETDEFS

-- Replicat the file operations to the files
GETFILEOPS
-- Do block reads
FASTREADS

-- MAP the CAF file with EXCEPTIONSONLY mapping
MAP \SITEA.<volume>.PRO1DATA.CAF,
   TARGET \SITEB.<volume>.PRO1DATA.CAF;
MAP \SITEA.<volume>.PRO1DATA.CAF,
   TARGET \SITEB.<volume>.PRO1DATA.OLDCAF,
   EXCEPTIONSONLY;
MAP \SITEA.<volume>.PRO1DATA.OLDCAF,
   TARGET \SITEB.<volume>.PRO1DATA.OLDCAF;
MAP \SITEA.<volume>.PRO1DATA.OLDCAF,
   TARGET \SITEB.<volume>.PRO1DATA.CAF,
   EXCEPTIONSONLY;
```

```
-- MAP the PBF file with EXCEPTIONSONLY mapping
MAP \SITEA.<volume>.PRO1DATA.PBF,
   TARGET \SITEB.<volume>.PRO1DATA.PBF;
MAP \SITEA.<volume>.PRO1DATA.PBF,
   TARGET \SITEB.<volume>.PRO1DATA.OPBFDA,
   EXCEPTIONSONLY;
MAP \SITEA.<volume>.PRO1DATA.OPBFDA,
   TARGET \SITEB.<volume>.PRO1DATA.OPBFDA;
MAP \SITEA.<volume>.PRO1DATA.OPBFDA,
   TARGET \SITEB.<volume>.PRO1DATA.PBF,
   EXCEPTIONSONLY;

MAP \SITEA.<volume>.PRO1DATA.NEWC*,
   TARGET \SITEB.<volume>.PRO1DATA.*;
MAP \SITEA.<volume>.PRO1DATA.NCAF*,
   TARGET \SITEB.<volume>.PRO1DATA.*;
MAP \SITEA.<volume>.PRO1DATA.NPBF*,
   TARGET \SITEB.<volume>.PRO1DATA.*;

-- Map the IDF file
MAP \SITEA.<volume>.PRO1DATA.IDF,
   TARGET \SITEB.<volume>.PRO1DATA.IDF,
   TARGETDEF IDF,
   COLMAP (USEDEFAULTS,
      NEG-NAME  = @STRSUB (NEG-NAME,  "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”),
      UAF-NAME  = @STRSUB (UAF-NAME,  "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”),
      CAF-NAME  = @STRSUB (CAF-NAME, "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”),
      PBF-NAME = @STRSUB (PBF1-NAME, "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”),
      PBF2-NAME = @STRSUB (PBF2-NAME, "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”),
      PBF3-NAME = @STRSUB (PBF3-NAME, "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”),
      PBF4-NAME = @STRSUB (PBF4-NAME, "\SITEA", "\SITEB",
                 “VOLA”, “VOLB”, “SUBVOLA”, “SUBVOLB”));
```

```
-- Map the LCONF file and substitute NonStop node and volume.
-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \SITEA.<volume>.PRO1CNTL.L*CONF,
   TARGET \SITEB.<volume>.PRO1CNTL.*,
   TARGETDEF LCONF,
   COLMAP
      (PRIKEY        = PRIKEY,
      PROD-IND       = PROD-IND,
      LCONF.COMMENTS = @STRSUB (LCONF.COMMENTS, "\SITEA", "\SITEB",
                       “VOLA”, “VOLB”,“SUBVOLA”,“SUBVOLB”),
      LAST-CHNG-TIME = LAST-CHNG-TIME,
      FILE-NAME      = @STRSUB (FILE-NAME, "\SITEA", "\SITEB",
                       “VOLA”, “VOLB”, “SUBVOLA”,“SUBVOLB”),
      TEMPLATE       = @STRSUB (TEMPLATE, "\SITEA", "\SITEB",
                       “VOLA”, “VOLB”,“SUBVOLA”,“SUBVOLB”),
      USER-FIELD     = USER-FIELD,
      USER-FLD2      = USER-FLD2,
      PROD-IND-ADNL  = PROD-IND-ADNL,
      USER-FLD4      = USER-FLD4,
      LAST-AFM       = LAST-AFM),
   WHERE (ITEM-TYP = "A");

-- If the SITE, VOLUME or SUBVOLUME are different between sites,
-- substitute the correct SITE, VOLUME or SUBVOLUME name.
MAP \SITEA.<volume>.PRO1CNTL.L*CONF,
   TARGET \SITEB.<volume>.PRO1CNTL.*,
   TARGETDEF LCONF,
   COLMAP
      (PRIKEY        = PRIKEY,
      PROD-IND       = PROD-IND,
      LCONF.COMMENTS=@STRSUB(LCONF.COMMENTS, "\SITEA", "\SITEB"),
      LAST-CHNG-TIME = LAST-CHNG-TIME,
      PLGTH        = PLGTH,
      PTXT       = @STRSUB (PTXT,   "\SITEA", "\SITEB"),
      USER-FLD3      = USER-FLD3,
      PROD-IND-ADNL  = PROD-IND-ADNL,
      USER-FLD4      = USER-FLD4,
      LAST-AFM       = LAST-AFM),
   WHERE (ITEM-TYP = "P");
```

```
-- Map the remaining BASE24 data files
MAP \SITEA.<volume>.PRO1DATA.*, TARGET \SITEB.<volume>.PRO1DATA.*;
MAP \SITEA.<volume>.PRO1EXEC.*, TARGET \SITEB.<volume>.PRO1EXEC.*;
MAP \SITEA.<volume>.PRO1CNTL.*, TARGET \SITEB.<volume>.PRO1CNTL.*;
```

Figure 43 Replicat parameter file for Replicat group REPTLFA

```
--*******************************************************************
-- REPTLFA - This Replicat reads the Remote/Extract Trails
--          <volume>.TLFTRLS.AA from Site A and MAPS the
--          ATM and POS Transaction Log Files
--*******************************************************************

REPLICAT REPTLFA

-- Set the dictionary location
DICTIONARY <base24 atm volume>.AT60DDL
-- Use the target site definitions
ASSUMETARGETDEFS

-- Set the discard file
DISCARDFILE <volume>.GGSDISC.REPTLFA, PURGE

-- Ignore the file operations to the combined A+B (P)TLF files
IGNOREFILEOPS

-- Do block reads
FASTREADS

-- Map financial and exception TLF records
MAP \SITEA.<volume>.PRO1ATLF.T*, TARGET \SITEB.<volume>.PRO1ATLF.*,
   TARGETDEF TLF,
   NOCOLMAP,
   WHERE (TLF.HEAD.REC-TYP <> "00");

-- Set the dictionary location
DICTIONARY <base24 pos volume>.PS60DDL

-- Map financial and exception PTLF records
```

```
MAP \SITEA.<volume>.PRO1PTLF.P*, TARGET \SITEB.<volume>.PRO1PTLF.*,
    TARGETDEF PTLF,
    NOCOLMAP,
    WHERE (PTLF.HEAD.REC-TYP <> "00");
```

Figure 44 Replicat parameter file for Replicat group REPTMFA

```
--*****************************************************************
-- REPTMFA  Replicat Parameter File
--         Reads the Remote/Extract Trails <trail volume>.TMFTRLS.AA
--          and applies all records to the target files.
--*****************************************************************

REPLICAT REPTMFA

-- Set the discard file
DISCARDFILE <GG volume>.GGSDISC.REPTMFA, PURGE

-- Replicat the file operations to the files
GETFILEOPS

-- Do block reads
FASTREADS

-- Map financial and exception PTLF records
MAP \SITEA.<Base24 volume>.*.*, TARGET \SITEB.<Base24 volume>.*.*;
```

Adding GoldenGate components

Once your parameter files are in place, you must add the GoldenGate components required for bi-directional processing. Begin with the components on site A, and finish with site B.

Adding site A components

- Add your Logger by issuing the following command:

  ```
  GGSCI> ADD LOGGER
  ```

- Add your Extracts by issuing the following commands:

  ```
  GGSCI> ADD EXTRACT EXTTLFA, LOGTRAILSOURCE <trail volume>.TLFLOGS.AA
  ```

  This command adds the EXTTLFA Extract process, which reads the <log trail volume>.TLFLOGS.AA logtrail.

  ```
  GGSCI> ADD EXTRACT EXTB24A, LOGTRAILSOURCE <trail volume>.B24LOGS.AA
  ```

  This command adds the EXTB24A Extract process, which reads the <log trail volume>.B24LOGS.AA logtrail.

  ```
  GGSCI> ADD EXTRACT EXTTMFA, BEGIN NOW
  ```

  This command adds the EXTTMFA Extract process, which reads the TMF Audit Log.

- Add your trails with the following commands. The actual files are not created until the first log record is written.

  ```
  GGSCI> ADD EXTTRAIL \SITEB.<trail volume>.TLFTRLS.AA, EXTRACT EXTTLFA
  GGSCI> ADD EXTTRAIL \SITEB.<trail volume>.TMFTRLS.AA, EXTRACT EXTTMFA
  GGSCI> ADD EXTTRAIL \SITEB.<trail volume>.B24TRLS.AA, EXTRACT EXTB24A
  ```

- Add your Replicats with the following commands.

  ```
  GGSCI> ADD REPLICAT REPTLFB, EXTTRAILSOURCE <trail volume>.TLFTRLS.BB
  ```

  This command adds the Replicat process REPTLFB that delivers the transaction log file changes from \SITEB to \SITEA.

  ```
  GGSCI> ADD REPLICAT REPTMFB, EXTTRAILSOURCE <trail volume>.TMFTRLS.BB
  ```

  This command adds the Replicat process REPTMFB that delivers the TMF Audit log records from \SITEB to \SITEA.

  ```
  GGSCI> ADD REPLICAT REPB24B, EXTTRAILSOURCE <trail volume>.B24TRLS.BB
  ```

  This command adds the Replicat process REPB24B that delivers the BASE24 file changes from \SITEB to \SITEA.

Adding site B components

- Add your Loggers with the following commands.

  ```
  GGSCI> ADD LOGGER
  ```

- Add your Extracts with the following commands:

  ```
  GGSCI> ADD EXTRACT EXTTLFB, LOGTRAILSOURCE <trail volume>.TLFLOGS.BB
  ```

  This command adds the EXTTLFB Extract process that reads the <log trail volume>.TLFLOGS.BB logtrail.

  ```
  GGSCI> ADD EXTRACT EXTB24B, LOGTRAILSOURCE <trail volume>.B24LOGS.BB
  ```

  This command adds the EXTB24B Extract process that reads the <log trail volume>.B24LOGS.BB logtrail.

  ```
  GGSCI> ADD EXTRACT EXTTMFB, BEGIN NOW
  ```

  This command adds the EXTTMFB Extract process that reads the TMF Audit Log.

- Add your trails with the following commands. The actual files are not created until the first log record is written to the target site.

  ```
  GGSCI> ADD EXTTRAIL \SITEA.<trail volume>.TLFTRLS.BB, EXTRACT EXTTLFB
  GGSCI> ADD EXTTRAIL \SITEA.<trail volume>.TMFTRLS.BB, EXTRACT EXTTMFB
  GGSCI> ADD EXTTRAIL \SITEA.<trail volume>.B24TRLS.BB, EXTRACT EXTB24B
  ```

- Add your Replicats with the following commands:

  ```
  GGSCI> ADD REPLICAT REPTLFA, EXTTRAILSOURCE <trail volume>.TLFTRLS.AA
  ```

  This command adds the Replicat process REPTLFA that delivers the transaction log file changes from \SITEA to \SITEB.

  ```
  GGSCI> ADD REPLICAT REPTMFA, EXTTRAILSOURCE <trail volume>.TMFTRLS.AA
  ```

  This command adds the Replicat process REPTMFA that delivers the transaction log file changes from SITEA to \SITEB.

  ```
  GGSCI> ADD REPLICAT REPB24A, EXTTRAILSOURCE <trail volume>.B24TRLS.AA
  ```

This command adds the Replicat process REPB24A that delivers the BASE24 file changes from SITEA to \SITEB.

Starting GoldenGate components

Once you have added the other GoldenGate components on both site A and site B, you are ready to start bi-directional processing. Begin with site A, and finish with site B.

Starting site A components

- Start Manager on site A by issuing the following command:

```
GGSCI> START MANAGER
```

- Start Logger with the following comands. Data from the Logparm file and intercept libraries Logger requires is loaded into the CPU'S memory.

```
GGSCI> START LOGGER
```

- Start your Extracts with the following commands.

```
GGSCI> START EXTTLFA
```

This command starts the EXTTLFA process, which reads the log trails on \SITEA.<trail volume>.TLFLOGS.AA and writes the log records to \SITEB.<trail volume>.TLFTRLS.AA extract trails.

```
GGSCI> START EXTTMFA
```

This command starts the EXTTMFA process, which reads the TMF Audit Logs on \SITEA and writes the records to \SITEB.<trail volume>.TMFTRLS.AA extract trails.

```
GGSCI> START EXTB24A
```

This command starts the EXTB24A process, which reads the log trails on \SITEA.<trail volume>.B24LOGS.AA and writes the log records to \SITEB.<trail volume>.B24TRLS.AA extract trails.

- Start your Replicats with the following commands.

  ```
  GGSCI> START REPTLFB
  ```

  This command starts the Replicat process REPTLFB, which reads records from its Extract Trail and updates the BASE24 transaction log files

  ```
  GGSCI> START REPTMFB
  ```

  This command starts the Replicat process REPTMFB, which read the extract records from its Extract Trail and updates the BASE24 TMF Audit log files.

  ```
  GGSCI> START REPB24B
  ```

  This command starts the Replicat process REPB24B, which reads records from its Extract Trail and update the BASE24 data files.

Starting site B components

- Start Manager on site B by issuing the following command:

  ```
  GGSCI> START MANAGER
  ```

- Start Logger with the following command. Data from the Logparm file and intercept libraries Logger requires is loaded into the CPU'S memory.

  ```
  GGSCI> START LOGGER
  ```

- Start your Extracts with the following commands.

  ```
  GGSCI> START EXTTLFB
  ```

  This command starts the EXTTLFB process, which reads the log trails on \SITEB.<trail volume>.TLFLOGS.BB and writes the log records to \SITEA.<trail volume>.TLFTRLS.BB extract trails.

  ```
  GGSCI> START EXTTMFB
  ```

  This command starts the EXTTMFB process, which reads the TMF Audit Logs on \SITEB and writes the records to \SITEA.<trail volume>.TMFTRLS.BB extract trails.

  ```
  GGSCI> START EXTB24B
  ```

This command starts the EXTTB24B process, which reads the log trails on \SITEB.<trail volume>.B24LOGS.BB and writes the log records to \SITEA.<trail volume>.B24TRLS.BB extract trails.

- Start your Replicats with the following commands.

  ```
  GGSCI> START REPTLFA
  ```

  At this point, this process will read records from its Extract Trail and update the BASE24 transaction log files

  ```
  GGSCI> START REPTMFA
  ```

  At this point, the process will read records from its Extract Trail and update the BASE24 TMF Audit log files

  ```
  GGSCI> START REPB24A
  ```

  At this point, this process will read records from its Extract Trail and update the BASE24 data files.

Bi-directional processing with c-tree

If you are processing c-tree records, GoldenGate offers a log-based product to replicate that data bi-directionally. Designed specifically to work with GoldenGate for Windows and UNIX, GoldenGate for c-tree captures data from transaction logs and delivers relevant data to the database of your choice.

C-tree processing overview

C-tree records are written to transaction logs that are flat files; these are the source for GoldenGate. Extract captures the record's key and data and passes it straight through to Replicat on your target system. Metadata such as column length is not replicated due to c-tree constraints. As a workaround, ACI has developed user exits; check your ACI documentation for further details.

About c-tree transaction logs

C-tree transaction logs are stored in the TRANLOG.DIR. GoldenGate captures just the key and data area of the transaction. Size limitations are as follows:

- The maximum key length is 1024 bytes.
- The maximum row length is 10,000 bytes.

Transaction logs do not automatically truncate; so you must monitor log size and clean out old logs. Periodically review Extract checkpoints to determine which logs can be removed and which can be kept, or set your KEEPLOGS parameter to KEEPLOGS = -1.

CHAPTER 7

GoldenGate for BASE24 Solutions: Data Warehousing

. . . .

This chapter outlines how to implement GoldenGate for BASE24 as a data warehousing solution. Each step required for such an implementation is discussed in the following topics:

Contents

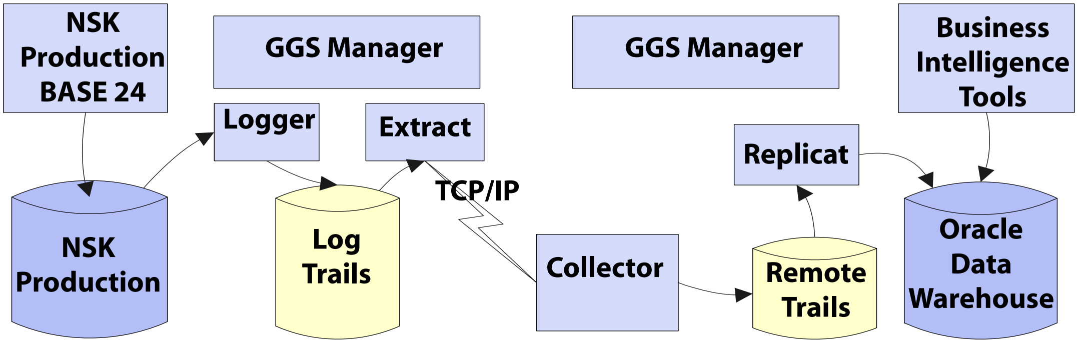
Overview

GoldenGate delivers up-to-the-second data to support real-time reporting and analytical applications on HP and other hardware platforms. This means the real-time data analysis can be done on non-production systems, which is more cost effective. Maintaining additional keys or indexes for data reporting does not affect production performance or compete for production system resources.

Typically the BASE24 transaction log files are used as the main input file for data warehousing. All the information about the transaction is contained in the TLF. This information can be used for everything from fraud detection to retailer analysis.

The following environment illustrates capturing ATM TLF records on the HP NonStop server and replicating the records to an Oracle database on a Windows system.

Figure 45 Sample data warehouse solution



GoldenGate components

The GoldenGate example shown in Figure 45 is for a typical Data Warehouse environment. This configuration has the following GoldenGate entities:

- **Two Managers**: The HP NonStop and Windows systems each have a Manager process running. These two Manager processes establish a link between the systems by starting the Collector when the NonStop Extract needs to send trail records to the Windows system.
- **One Collector:** The Windows Manager will start a Collector process and give the Collector's port number to the Manager on the HP NonStop Server. The HP Manager will then give the Collector's port number to the Extract process. From then on the Extract will communicate directly with the Collector process.
- **One Logger:** This Logger is dedicated to capturing inserts to the ATM TLF files on the production network. These log records will be generated for every ATM transaction whether it is approved or not. Since only record inserts are logged, these records are always the full record image.
- **One Log Trail:** This trail contains all the TLF records captured by the TLF Logger process.
- **One Extract**: This Extract reads the log trails and sends the data to the Collector on the Windows system.
- **One Remote Trail:** The remote trail files are numbered in the same manner as on the HP; there is a two character constant followed by six numbers beginning with 000000. The number and size of the remote trails can be modified with the EXTRACT ALTER command on the HP NonStop system.
- **One Replicat:** This Replicat applies all the TLF records from the remote trails to the database on the NT. The database can be Oracle, MSSQL, SYBASE or any ODBC.

Sample HP NonStop parameter files

This section provides sample parameter files for each of the components required for an HP NonStop source in a BASE24 data warehousing solution.

Sample Manager file

```
-- *****************************************************
-- MGRPARM     Manager Parameters
-- *****************************************************
-- TCPIP parameters
PORT 7809
TCPIPPROCESSNAME $ZTC0

-- Keep the TLF Extract Trails for at least 5 days and processed
PURGEOLDEXTRACTS <Base24vol>.TLFTRLS.AA, USECHECKPOINTS, MINKEEPDAYS 5
```

Sample Logger file

The Logparm parameter file defines the scope and configuration for each Logger process. The information contained in the Logparm is used to set the values in the shared memory segment files.

```
--*************************************************************
-- LOGGER 0  -  Captures TLF from PRO1ATLF
--*************************************************************
-- Location, number and size of the Logger trails
LOG <trail volume>.NTLOGS.AA , MEGABYTES <megabytes>, NUMFILES <num>,
SECURE "NCNC"

-- Primary and backup CPU for Logger 0
CPU 0,1

-- Get unstructured files
GETUNSTRUCTURED
-- Get bulk loads
GETBULKIO

-- Use the full record image, do not compress the updates
NOCOMPRESSUPDATES
-- Make priority higher than BASE24 nucleus (NETWORK)
PRI 180

-- List the files to be captured
FILE <volume>.PRO1ATLF.T*
```

Sample Extract file

The Extract parmeter file is read at startup of the Extract process. The parameters identify the log trails to be read and determine which files and target system(s) receive the trail records. In this example, the EXTNT Replicat reads the NTLOG trail and sends the data to the Collector on the Windows system. We are using the Logger to capture TLF inserts, however the Direct Extract can also be used for data warehousing.

TLF files are created daily. Each day's file has its own name. To be able to use a single file name on the Windows system, the Extract uses the TARGETNAME parameter. This parameter uses its file name value to replace the original TLF file name. This means that whatever the source file name was, it will now be changed to a consistent name for the target system.

The token area of the TLF does not lend itself to mapping because of its variability. GoldenGate's T24 product can be used to map the token area.

```
--**************************************************************--
--                      Extract EXTNT
--
-- Replicates the BASE24 transaction log files from BASE24 to
-- the Windows Source Log Trail:
-- \PRODUCTION.<log trail volume>.NTLOGS.AA --
**************************************************************--
EXTRACT EXTNT

DISCARDFILE <GG volume>.GGSDISC.EXTNT, PURGE
NOFILLSHORTRECS

--Windows REMOTE HOST
RMTHOST  135.3.249.148 , MGRPORT 7809

-- ATM
RMTTRAIL c:\ggs7011\dirdat\tl
FILE <Base24 volume>.PRO1ATLF.TL*,
    ALTNAME <Base24 volume>.PRO1TPLT.TLYYMMDD,
      TARGETNAME <Base24 volume>.PRO1ATLF.TLF;
```

Sample Windows parameter files

This section provides sample parameter files for each of the components required for a Windows target in a BASE24 data warehousing solution. In this example the remote trails are c:\GGS7011\DIRDATA\AA.

Sample Manager file

```
-- *******************************************************
-- MGRPARM    Manager Parameters                   *
-- *******************************************************
-- TCPIP parameters
PORT 7801

-- Keep the TLF Remote Trails for at least 5 days and processed
PURGEOLDEXTRACTS C:\GGS7011\DIRDAT\AA, USECHECKPOINTS, MINKEEPDAYS 5
```

Sample Replicat file

This file is read by the example REPTLF Replicat at start up. For disaster recovery sites the updates that occurred on the source site are replicated to the target site without any changes.

```
--***************************************************************--
--                      Replicat REPTLF
-- Replicates the BASE24 transaction log files from BASE24
-- to the Windows Source Log Trail:
--               \PRODUCTION.<log trail volume>.NTLOGS.AA
-- *************************************************************** --
REPLICAT REPTLF

DISCARDFILE c:\ggs7011\dirdisc\reptlf, purge
TARGETDB GGS   ??User and Password??
COLMATCH PREFIX TLFO_

-- DEFs file created by DEFGEN on the HP and transferred to theWindows
SOURCEDEFS c:\ggs7011\dirdef\ntdefs.def
```

```
MAP <Base24 volume>.PRO1ATLF.TLF, TARGET ATMDATA,
   COLMAP (USEDEFAULTS,
         TLFO_TERM_TERM_ID = TERM-ID,
         TLFO_CRD_NUM = NUM,
         TLFO_CRD_MBR_NUM = MBR-NUM,
         TLFO_TRAN_T_CDE = T-CDE,
         TLFO_TRAN_T_FROM = T-FROM,
         TLFO_TRAN_T_TO = T-TO,
         TLFO_FROM_ACCT_NUM = FROM-ACCT-ACCT-NUM,
         TLFO_TO_ACCT_NUM = TO-ACCT-ACCT-NUM,
         TLFO_SETL_CDE = SETL-CRNCY-CDE,
         TLFO_ACQ_INST_ID = ACQ-INST-ID-NUM,
         TLFO_NEW_PIN_OFST = @IF( @COLTEST(PIN-OFST , PRESENT),
                         @NUMSTR(PIN-OFST),@COLSTAT(NULL)),
         TLFO_CHG_RESET_IND = "",
         TLFO_PIN_CHG_MODE = "",
         TLFO_CRD_TYPE = TYP-CDE,
         TLFO_TRAN_TIME = TRAN-TIM,
         TLFO_TRAN_DAT = @DATE ("YYYY-MM-DD", "YY", TRAN-DAT-YY,
                           "MM", TRAN-DAT-MM, "DD", TRAN-DAT-DD),
         TLFO_POST_DATE = @DATE ("YYYY-MM-DD","YY", POST-DAT-YY,
                           "MM", POST-DAT-MM, "DD", POST-DAT-DD),
         TLFO_ISS_ICHG_DATE =
            @DATE("YYYY-MM-DD","YY", ISS-ICHG-SETL-DAT-YY,
                  "MM", ISS-ICHG-SETL-DAT-MM,
                  "DD", ISS-ICHG-SETL-DAT-DD),
         TLFO_TRAN_DATE =
            @DATE ("YYYY-MM-DD:HH:MI:SS","YY", TRAN-DAT-YY,
                   "MM", TRAN-DAT-mm, "DD", TRAN-DAT-DD,
                   "HHMISS", TRAN-TIM)),

         WHERE (REC-TYP = "01" OR REC-TYP = "20" OR REC-TYP = "21");
```

Creating DEFS and table files

GoldenGate requires that the Replicat have a source record definition file to enable mapping fields and columns from the source to the target. If the target files/tables do not exist on the target system, GoldenGate creates a file with the commands to create the files/tables.

The GoldenGate utility programs to create these files are DEFGEN and DDLGEN. Once the source definition file and target table command file have been created, they must be transferred to the Windows or target system.

Adding GoldenGate components

Components can be added to your HP NonStop site with the following commands. The examples are taken from the sample parameter files on page 97.

- Add your Logger with the following command.

  ```
  GGSCI> ADD LOGGER
  ```

- Add your Extract with the following command.

  ```
  GGSCI> ADD EXTRACT EXTNT, LOGTRAILSOURCE <trail volume>.NTLOGS.AA
  ```

  This command adds the EXTNT Extract process that reads the <GoldenGate volume>.NTLOGS.AA logtrail.

- Add your remote trail with the following command.

  ```
  GGSCI> ADD RMTTRAIL C:\GGS7011\DIRDAT.AA, EXTRACT EXTNT
  ```

  This command adds the extract trail to the GoldenGate environment. The actual file is not created until the first log record is written to the target site.

The components can be added to your Windows site using the following commands:

- Add your Replicat with the following command.

  ```
  GGSCI> ADD REPLICAT REPTLF, RMTTRAILSOURCE C:\GGS7011\DIRDAT\AA
  ```

  This command adds the Replicat process REPTLF. This Replicat is used to deliver the transaction log file changes from \PRODUCTION to the NT system.

Starting GoldenGate components

Start the following components on your HP NonStop site:

- Start Manager with the following command.

  ```
  GGSCI> START MANAGER
  ```

  The Manager process is responsible for keeping the Extracts and Replicats running and maintaining the extract trails. The Manager process is part of the default GoldenGate environment; it is automatically added during installation.

- Start Logger with the following command.

  ```
  GGSCI> START LOGGER
  ```

  This command starts the Logger process(es). The information in the Logparm file that is used by the Logger and intercept libraries is loaded into each CPU's memory.

- Start Extract with the following command.

  ```
  GGSCI> START EXTNT
  ```

  At this point this Extract will start reading the logtrails on \PRODUCTION.<trail volume>.TLFLOGS.AA and writing the log records to \BACKUP.<trail volume>.TLFTRLS.AA extract trails.

Start the following components on your Windows Site:

- Start Manager with the following command.

  ```
  GGSCI> START MANAGER
  ```

  The Manager is also responsible for starting the Collector process when the Extract process on the source HP system first starts to send data to the target system.

- Start Replicat with the following command.

  ```
  GGSCI> START REPTLF
  ```

  This command starts the Replicat process REPTLF that reads records from its extract trail and updates the BASE24 transaction log files.

APPENDIX 1A:
Frequently Asked Questions

. . . .

This chapter answers questions frequently asked about GoldenGate for BASE24.

Questions and answers

What files should be compressed (should have the Logger parameter COMPRESSUPDATES set)? Which files should not?

COMPRESSUPDATES can only be used for configurations that are “like to like” and need no transformation. So it should not be used when platforms other than HP NonStop are involved or with files that require column mapping. It should not be used for entry-sequenced files.

Identify the entry-sequenced files and the unique alternate key files.

Entry-sequenced files that have updates must have the Replicat parameter

```
ENTRYSEQUPDATES, EXACTKEYS
```

ILFs replication needs this parameter, whereas PTLF and TLF do not. Not specifying this improves the performance of the Replicator.

What are the cutover issues, especially with bi-directional replication?

A bi-directional environment will probably want to settle each network separately. As such, settlement will not replicate I/Os for the PRDF and IDF. Also, the terminals that aren’t connected in that node should be deactivated in the PTDF and TDF. In this way, each node will settle its own terminals. During a disaster recovery these deactivated terminals must be activated but their business day will be correct.

What site-related changes are needed in the data and configuration files?

The TDF and PTDF contain the logical network name. The LCONF contains the site name, disk volume names, logical network name.

If a link is down, at what point is it better to recreate a database than keep replicating?

- Initial database creation.
- Logger stop (abend) problems can cause the target to be inconsistent. In most cases, selected files need to be recreated and not the whole database.

- When the link between the target and source is down so long that the log trails wrap around.

When should a signal be sent in bi-directional replication that a new refresh file is ready ?

When the new refresh file is successfully replicated across from the refresh node to the non-refresh node.

What are some common issues for bi-directional replication?

If you choose to run SETL at each node, logging of UAF, PRDF, RDF and IDF must be excluded for SETL. Example: EXCLUDEFILE $*.PRO1DATA.UAF, PROGRAM <Base24 vol>.B24OBJ.PSETL.

Glossary

The following explains terminology used in this manual.

Term	Definition
BASE24 Full Refresh	The BASE24 method of reloading an entire file. Once the new file is loaded to the BASE24 database, the old file is renamed and the newly refreshed file is renamed to its production name. The Refresh process then notifies all processes that have the file open to close the old file and open the new file.
BASE24 Tokens	BASE24's method to save additional data associated with a transaction. The token area is located at the end of BASE24's internal messages and can be moved into BASE24's external messages, Super Extract records, and transaction log records.
BASELIB	GoldenGate's user library that intercepts file and record operations to create audit records to be written by the Logger process.

Term	Definition
Bi-Directional Replication	Mutual replication between two BASE24 databases that share the same information. Each BASE24 database is updated locally by BASE24 applications and replicated to the other database.
BINDEXIT	The TACL macro that binds the user exit with the GoldenGate program (i.e. Extract or Replicat).
BINDSKEL	The TACL macro that binds the BASELIB and SKELB into a new SKELB file.
COMPRESSUPDATES	Identifies GoldenGate audit records that contain only the changes to a record operation and not the entire record. It is a technique to reduce the amount of data that is transferred and replicated.
Data Warehousing	The process of replicating BASE24 database file/records to another platform for data reporting and analysis.
Direct Extract	Extracting directly from Enscribe entry-sequenced files rather than using GoldenGate's intercept libraries and extracting from logtrails.
End Of Day (EOD)	The time when all processing is completed for the business day for a BASE24 product. All activity to the BASE24 product's transaction log file is complete.
Extract	Receives database changes and processes them by formatting and sending them to a target database.
GGS Intercept Library	See GGSLIB

Term	Definition
GGS Marker	Records inserted into the audit trails to identify application-specific events in the context of Extract and Replicat processing.
GGSLIB	The GoldenGate intercept library that combines GoldenGate's user library (i.e. C or TAL) with the COBOL run time libraries.
Hot Site Backup	One or more BASE24 networks on a separate HP NonStop system that are continuously updated from the primary site. In case the primary site can no longer process transactions, processing will continue from the backup site with no loss of data or delay.
Impacting	The BASE24 term for applying changes to BASE24 files that occurr from the time of the last BASE24 Extract to the time of the full refresh to the BASE24 file.
Initial Load	Extract can retrieve data directly from a file/table to be loaded by the Replicat to the target file/table. It is a way to initially synchronize databases.
Logger	Captures database changes to files unprotected by NonStop's Transaction Monitoring Facility (TMF). The Logger also captures file operations that can be duplicated by the Replicat.
Logdump	A GoldenGate utility that reads a Logtrail or Extract Trail.
Logtrails / Extract Trails	Files created by Logger or Extract that contain a GoldenGate header and data portion of the file/record operation.

Term	Definition
Replicat	Performs the database replication step. Replicat reads from either Logtrails, extract trails or extract files, created either by Logger or Extract.
SKELB	BASE24 user library that contains the Application Programmatic Interface (API) for BASE24 applications to XPNET
User Exit	Allows the customer to extend and customize the functionality of the Extract and Replicat programs. These are programs written either in C or COBOL that are independently compiled and bound to the GoldenGate program. Processing control is passed for each data record from the GoldenGate program to the user exit.