

MainView[®]
Alternate Access
Implementation and User's Guide

TD-733-3

Version 3.1

February 28, 1999



This document is published by the BMC Software, Inc., Technical Publications Department.

Copyright © 1995–1999 Boole & Babbage, Inc. All rights reserved.

Copyright © 1999 BMC Software, Inc. All rights reserved.

BMC Software, the BMC Software logo, and all other product or service names are trademarks or registered trademarks of BMC Software, Inc., in the USA and in other select countries. The ™ and ® indicate USA trademark or USA registration.

IBM and DB2 are registered trademarks of International Business Machines Corp. All other third-party logos and product/trade names are registered trademarks or trademarks of their respective companies.

RESTRICTED RIGHTS LEGEND

Use, distribution, or disclosure by the Government is subject to restrictions set forth in subparagraph (c) (1) (ii) of Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

BMC Software
2101 CityWest Blvd.
Houston, TX 77042-2827
USA

Contacting BMC Software

World Wide Web Address

For a complete list of all BMC Software international offices and subsidiaries, see the BMC Software home page at:

<http://www.bmc.com>

USA and Canada

In the USA and Canada, you can contact BMC Software by telephone, fax, or email:

Telephone	713 918 8800 (Corporate Headquarters) 800 538 1872 (Customer Support/USA) 408 526 3040 (Customer Support/Canada)
Fax	713 918 8000 (Corporate Headquarters) 408 526 3897 (Customer Support)
Email	support@boole.com (Customer Support)

International

Outside the USA and Canada, you can contact a BMC Software international support center.

Fax: (61) 3 9810 2030
Email: asiapacific_support@bmc.com

Asia/Pacific/Australia

BMC Software (Australia) Pty. Ltd.
3rd Floor
290 Burwood Road
Hawthorn Victoria 3122
Australia
Telephone: (61) 3 9810 2010
Customer Support: (61) 3 9810 2020

Europe, the Middle East, Africa, and Latin America

BMC Software, Inc.
2101 CityWest Blvd.
Houston, TX 77042-2827
USA
Telephone: 713 918 8800
Fax: 713 918 1302
Email: support@boole.com

You can also call the BMC Software office in your country, if applicable.

Contents

Chapter 1. Understanding MainView Alternate Access	1
MainView Alternate Access Functions	1
Security Considerations	2
An Overview of the Architecture	2
Logon Address Space (LAS) Functions	4
Terminal Address Space (TAS) Functions	5
Limitations of the Architecture	6
Technical Requirements	7
Hardware Considerations	7
Software Considerations	8
ISPF Maintenance	8
Chapter 2. Customizing MainView Alternate Access	9
Using AutoCustomization	9
Accessing AutoCustomization for CMF MONITOR Customers	9
Executing the BBV Steps	10
After AutoCustomization	12
Using Manual Customization	13
Step 1: Defining the ISPF Environment	13
Defining ISPF Profile Data Sets for Concurrent TSO and BBV Sessions	14
Defining ISPF LOG and LIST Data Sets for Concurrent TSO and BBV Sessions ..	14
Step 2: (Optional) Defining the VTAM Major and Minor Nodes	17
Step 3: (Optional) Defining a USS Table Entry	19
Step 4: (Optional) Defining Logon Interpret Table Entries	20
Step 5: Defining a Default TAS Initialization Parameter List	21
Step 6: Defining a TAS Startup Procedure	23
Step 7: (Optional) Defining a Default LAS Initialization Parameter List	25
Step 8: (Optional) Defining a LAS Startup Procedure	27
Step 9: (Optional) Enabling VTAM Terminal Support	28
Step 10: (Required) Verifying Security for BBV Sessions	28
Step 11: (Optional) Starting an EXCP Session	28
Step 12: (Optional) Starting a VTAM Session	28
Verifying Your BBV Load Modules	29
Defining Other LAS and TAS Initialization Parameter Lists	31
Understanding TAS Initialization Parameters	31
Understanding LAS Initialization Parameters	35
Verifying Your VTAM Configuration	37
Migrating from a Test to a Production Environment	37
Chapter 3. Defining Security for MainView Alternate Access	39
Understanding AutoLogon Session Security Issues	39
Defining Additional Security for Terminals	40
Defining Additional Security for Applications	41
Configuring CA-ACF2	42
Customizing Class TERMINAL Access Authorization for CA-ACF2	45
Customizing Class APPL Access Authorization for CA-ACF2	45
Configuring RACF	47
Customizing Class TERMINAL Access Authorization for RACF	48
Customizing Class APPL Access Authorization for RACF	48
Configuring CA-TOP SECRET	49
Customizing Class TERMINAL Access Authorization for CA-TOP SECRET	51

Customizing Class APPL Access Authorization for CA-TOP SECRET	51
Chapter 4. Logging On and Off from a VTAM Terminal	53
Activating the VTAM Application Major Node	54
Starting and Stopping the Logon Address Space (LAS)	54
Starting the LAS	55
Starting Multiple LASs	55
Controlling the LAS Initialization Parameters	56
Stopping the LAS	56
Using the LOGON Command	57
Examples of Using the VTAM LOGON Command	59
Using the Abbreviated Logon Command	60
Examples of Using the Abbreviated Logon Command	61
Using the Logon Panels	61
Using the User ID Panel	62
Using the Password Panel	63
Logging Off	64
Chapter 5. Configuring, Starting, and Terminating VTAM AutoLogon Sessions	65
Understanding the Three Methods of VTAM AutoLogon Session Initialization	65
Controlling TAS Initialization Parameters	66
Defining the START Command for VTAM AutoLogon Sessions	67
Examples of the START Command for VTAM AutoLogon Sessions	72
Defining BBVTASxx Members for VTAM AutoLogon Sessions	73
Examples of Defining the LOGON Command in a BBVTASxx Member	74
Terminating a VTAM AutoLogon Session	75
Chapter 6. Configuring, Starting, and Terminating EXCP AutoLogon Sessions	77
Understanding an EXCP Terminal Session	77
Inactivating a VTAM Terminal for EXCP Communications	78
Understanding the Three Methods of EXCP AutoLogon Session Initialization	78
Controlling the TAS Initialization Parameters	79
Defining the START Command for EXCP AutoLogon Sessions	79
Examples of the START Command for EXCP AutoLogon Sessions	84
Defining BBVTASxx Members for EXCP AutoLogon Sessions	85
Examples of Defining the LOGON Command in a BBVTASxx Member	86
Terminating an EXCP AutoLogon Session	87
Appendix A. Libraries and Load Modules	89
Load Modules	89
Libraries	89
BBPARM Data Set Members	90
BBSAMP Data Set Members	90
Appendix B. Cross-Reference of Parameters	93
Appendix C. Customer Support	97
Contacting Customer Support	97
After-Hours	97
Mailing Address	97
For Product Information	97
Glossary	99

Index	103
--------------------	-----

Tables

1.	CSA Requirements	8
2.	AutoCustomization Steps	11
3.	TAS Initialization Parameters	32
4.	LAS Initialization Parameters	35
5.	START Command Parameters (VTAM)	67
6.	START Command Parameters (EXCP)	80
7.	BBPARM Library Members	90
8.	BBSAMP Library Members	91
9.	TAS and LAS Parameter Cross-Reference	95

Figures

1.	How the BBV Architecture Establishes VTAM or EXCP Sessions	3
2.	LAS Operation.	4
3.	Example of the BBV Steps on a Product Customization Steps Panel.	10
4.	Example of the BBSAMP Member BBVINIT, Part. 1	15
5.	Example of the BBSAMP Member BBVINIT, Part. 2	16
6.	Example of the BBSAMP Member BBVTASA	18
7.	Example of the BBSAMP Member BBVUSST	19
8.	Example of the BBSAMP Member BBVLIT	20
9.	Example of the BBPARM Member BBVTAS00.	22
10.	Example of the BBSAMP Member BBVTAS.	24
11.	Example of the BBPARM Member BBVLAS00	26
12.	Example of the BBSAMP Member BBVLAS	27
13.	Example of the BBSAMP Member BBVLVER	29
14.	Example of the Output Created by BBVLVER.	30
15.	BBV Startup and Logon Procedure from VTAM Terminals.	53
16.	User ID Panel.	62
17.	Password Panel	63

About This Book

This book explains how to customize, maintain, and use MainView Alternate Access to access Boole & Babbage online products.

Who Should Read This Book

You should read this book if you need to

- Customize MainView Alternate Access
- Start or stop MainView Alternate Access sessions
- Define an AutoLogon session using MainView Alternate Access
- Access a Boole & Babbage product through VTAM or EXCP using MainView Alternate Access
- Maintain MainView Alternate Access

How This Book Is Organized

The following table describes how this book is organized.

Location	Description
Chapter 1, “Understanding MainView Alternate Access” on page 1	Introduces you to MainView Alternate Access functions and architecture, discusses the security interface, outlines technical requirements, and describes ISPF maintenance considerations.
Chapter 2, “Customizing MainView Alternate Access” on page 9	Describes how to customize MainView Alternate Access using AutoCustomization or manual customization; verify the load modules and your VTAM configuration; define parameter list members; and migrate MainView Alternate Access from your test environment to your production environment.
Chapter 3, “Defining Security for MainView Alternate Access” on page 39	Describes how to customize security in CA-ACF2, RACF, or CA-TOP SECRET for AutoLogon sessions so that user IDs that do not require clear text passwords can be used.
Chapter 4, “Logging On and Off from a VTAM Terminal” on page 53	Explains how to activate the VTAM application major node, start the LAS, log on a VTAM session from the terminal, use the logon panels, and log off a VTAM session.
Chapter 5, “Configuring, Starting, and Terminating VTAM AutoLogon Sessions” on page 65	Provides instructions for configuring, starting, and terminating a VTAM AutoLogon session.
Chapter 6, “Configuring, Starting, and Terminating EXCP AutoLogon Sessions” on page 77	Provides instructions for configuring, starting, and terminating an EXCP AutoLogon session.

Location	Description
Appendix C, “Customer Support” on page 97	Explains what to do if you need to contact Boole & Babbage for product assistance or other customer services.
Appendix A, “Libraries and Load Modules” on page 89	Describes each library and load module associated with MainView Alternate Access.
Appendix B, “Cross-Reference of Parameters” on page 93	Provides a cross-reference of parameters in the startup procedures, the parameter list members, and the START command.
“Glossary” on page 99	Describes a list of terms that are used throughout this book.
“Index” on page 103	Provides a cross-reference of topics and the pages on which they are discussed.

Required Reading

While using this book, you are referred to information in the Boole & Babbage *Product Installation and Maintenance Guide*, TD-2024.

Related Reading

While using this book, you may want to refer to the documentation listed in the following table.

For information about	Refer to
MainView products	<i>Using MainView</i> , TD-906 <i>MainView Implementation Guide</i> , TD-2071 <i>MainView Administration Guide</i> , TD-M52 <i>Implementing Security for MainView Products</i> , TD-742
CA-ACF2	<i>CA-ACF2 MVS Administrator Guide</i> <i>CA-ACF2 MVS General Information Guide</i> <i>CA-ACF2 MVS Systems Programmer Guide</i> <i>CA-ACF2 MVS Implementation Planning Guide</i>
CA-TOP SECRET	<i>CA-TOP SECRET MVS Planning Guide</i> <i>CA-TOP SECRET MVS Control Options Guide</i> <i>CA-TOP SECRET MVS TSS Command Functions Guide</i> <i>CA-TOP SECRET MVS Implementation: General Guide</i> <i>CA-TOP SECRET MVS Implementation: Batch and STC Guide</i> <i>CA-TOP SECRET MVS Implementation: TSO Guide</i> <i>CA-TOP SECRET MVS General Concepts Guide</i>
RACF	<i>RACF Command Language Reference</i> <i>RACF Security Administrator's Guide</i> <i>RACF General User's Guide</i> <i>RACF General Information</i>

For information about	Refer to
TSO/E REXX	<i>TSO Extensions Version 2 REXX/MVS User's Guide</i> <i>TSO Extensions Version 2 REXX/MVS Reference</i>
VTAM	<i>VTAM Installation and Resource Definition</i> <i>VTAM Messages and Codes</i> <i>VTAM Operation</i>

Changes to This Book

The following changes have been made to this book:

- A new LAS initialization parameter, GNAME, provides support for VTAM generic names. See [“Understanding LAS Initialization Parameters”](#) on page 35 for details.
- The previous edition of this book contained “Chapter 2: Replacing VTM 1.1 with MainView Alternate Access.” That chapter contained obsolete material and has been removed from this edition.

Chapter 1. Understanding MainView Alternate Access

MainView Alternate Access is a product component that provides EXCP and VTAM interfaces for access to MainView products. Boole & Babbage developed MainView Alternate Access, Version 3, Release 1, as a

- Replacement product for VTAM/EXCP Terminal Monitor, Version 1, Release 1
- Common VTAM and EXCP session access component for all Boole & Babbage products

Throughout this book, MainView Alternate Access is referred to by its formal name, MainView Alternate Access, and by its component code, BBV. The component code is associated with the MainView Alternate Access FMID (functional module identifier) BBVVT31.

This chapter

- Explains the functions MainView Alternate Access performs
- Discusses MainView Alternate Access and security (see page 2)
- Describes the MainView Alternate Access architecture and its limitations (see page 2)
- Outlines software and hardware requirements (see page 7)
- Explains a maintenance issue related to ISPF (see page 8)

MainView Alternate Access Functions

BBV performs the following functions:

- Provides separate VTAM and EXCP interfaces to ISPF for accessing your Boole & Babbage online products, but without requiring the TSO subsystem to be active
- Supports AutoLogon for selected VTAM or EXCP terminals by an automation product, such as AutoOPERATOR
Note: AutoLogon is a function that allows both EXCP and VTAM terminal sessions to be configured so they are logged directly on to an application at session initialization.
- Initializes and can automatically log on individual terminal sessions at IPL time
- Initiates/terminates terminal sessions for user logons/logoffs dynamically
- Runs up to 32 EXCP terminal sessions and any number of defined VTAM sessions concurrently per MVS image
- Supports full ISPF functionality for each session
- Offers new password support
- Provides full data set allocation through a REXX EXEC procedure at session initialization
- Allows passing of an optional parameter string (OPT subparameter) at session initialization

Security Considerations

BBV uses the MVS SAF interface to verify security and supports the following external security manager (ESM) products:

- CA-ACF2
- RACF
- CA-TOP SECRET

Security is provided only for the VTAM or EXCP sessions that BBV initiates and supports. Once you have accessed your Boole & Babbage product, BBV supports but does not manage product-specific security—this is managed by your product.

Any RACF USERID, CA-ACF2 LOGONID, or CA-TOP SECRET ACID (all are referred to as *user ID* throughout this book) to be used with a BBV session must be defined to your site's ESM. If an ID is not defined to your ESM, it is denied logon access by BBV; if an ID is defined, security checks associated with the user ID are supported and verified by BBV through the standard MVS SAF interface.

Your ESM may require further customization for BBV. For example, if RACF is not customized, any user ID is allowed to log on through BBV without specifying a password. Each ESM requires specific customization by your site security administrator for session support; see [“Defining Security for MainView Alternate Access” on page 39](#) for more information.

Optionally, BBV provides facilities to secure user IDs for use with AutoLogon sessions. Depending on your site requirements, AutoLogon sessions may be configured with user IDs that do not require manual entry of clear text passwords. In addition, sessions can be further secured so that they can be used only at specifically authorized VTAM terminals or only with BBV.

An Overview of the Architecture

The BBV architecture is comprised of the Logon Address Space (LAS) and Terminal Address Space (TAS).

The LAS is required only to initialize a TAS for each user logging on from a VTAM terminal.

A TAS is required to support each session, whether it is an EXCP or a VTAM session. Each TAS interacts with the external security manager directly using the SAF interface and allocates the ISPF environment during session initialization.

There are three ways that a TAS can be initialized:

- VTAM terminal users log on from the terminal.
- A START command is issued for a VTAM terminal session.
- A START command is issued for an EXCP terminal session.

[Figure 1](#) outlines how these address spaces interact. The following sections discuss the functions of each address space and the limitations of the architecture:

- [“Logon Address Space \(LAS\) Functions” on page 4](#)

- “Terminal Address Space (TAS) Functions” on page 5
- “Limitations of the Architecture” on page 6

START command issued by operator, automation product, or IPL procedure...

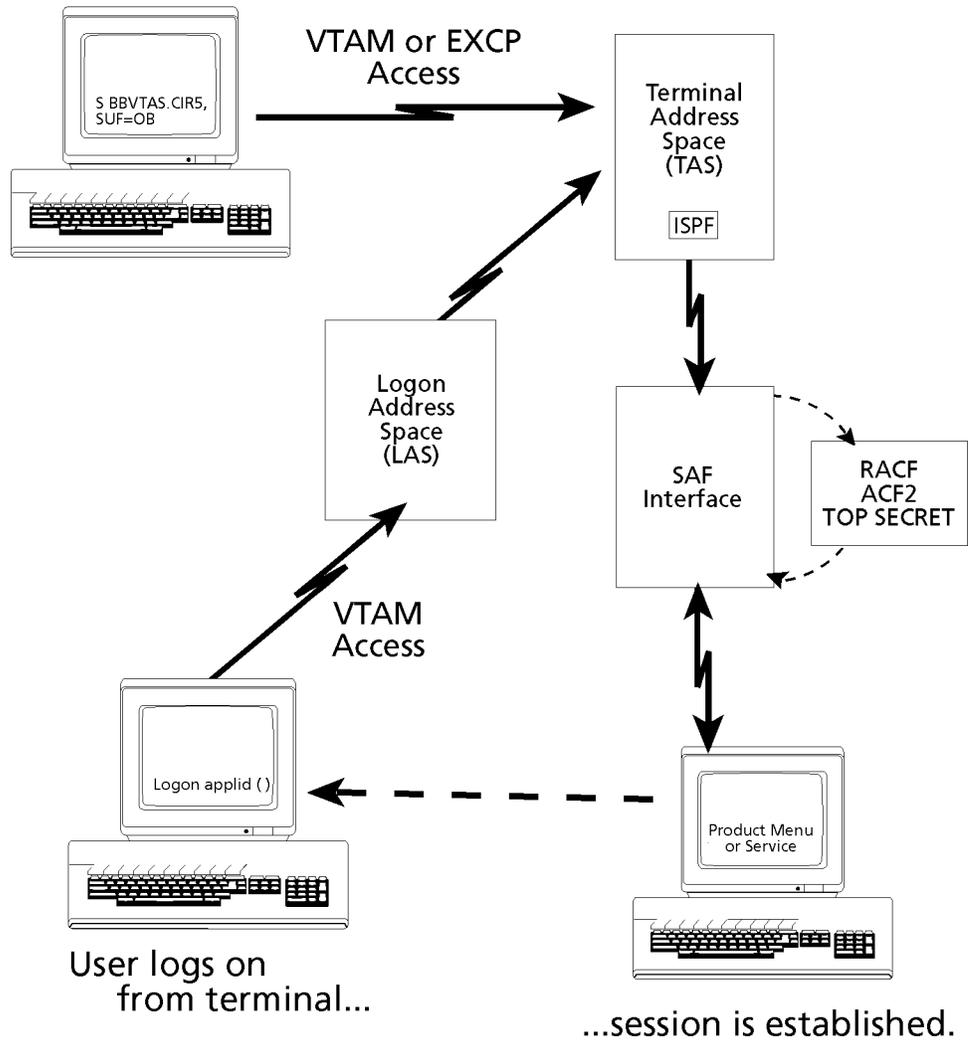


Figure 1. How the BBV Architecture Establishes VTAM or EXCP Sessions

Logon Address Space (LAS) Functions

The Logon Address Space (LAS) is required only for VTAM terminal users that log on from the terminal; the LAS is not required for EXCP or VTAM terminal sessions that are initialized by issuing an MVS START command.

Operational Considerations:

- VTAM must be available and the BBV VTAM application major node must be active before the LAS can be started (see [“Activating the VTAM Application Major Node” on page 54](#) for more information).
- The address space for the LAS must run as a started task and can be started by an operator or during IPL (see [“Starting the LAS” on page 55](#) for more information).
- The startup procedure (proc) for the LAS must be placed in a JES-defined data set, such as SYS1.PROCLIB, that is referenced by started tasks.
- Only one LAS needs to run per system; however, other LASs can run but must be named uniquely (see [“Starting Multiple LASs” on page 55](#) for more information).
- The LAS should stay active at all times; otherwise, VTAM terminal users cannot log on.
- The LAS acknowledges the LOGON command (see [“Using the LOGON Command” on page 57](#) for more information) or an abbreviated logon command that has been customized to your site’s USS table or logon interpret table (see [“Using the Abbreviated Logon Command” on page 60](#) for more information).

When active, the LAS requests the user ID for a terminal session and issues a START command containing the user ID to initiate a TAS. Once a TAS is initialized, it becomes independent of the LAS, so the LAS is not a single point of control (or failure) for a user session. Specifically, the sequence of events that occurs in association with the LAS is shown in [Figure 2](#).

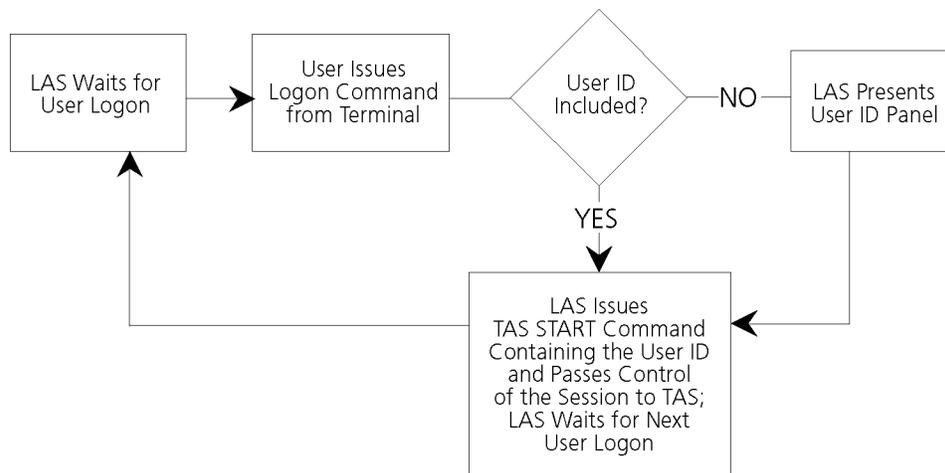


Figure 2. LAS Operation

Terminal Address Space (TAS) Functions

A Terminal Address Space (TAS) is required for each user session, whether VTAM or EXCP. A TAS cannot support both modes in a single session, however.

Up to 32 EXCP TAS sessions can run concurrently on a system, along with any number of defined VTAM sessions. The EXCP 32-session maximum is an artificial limit set internally by BBV. The maximum number of VTAM sessions that can run is determined by the number of minor nodes defined (see [Figure 6 on page 18](#)); the default number is 10.

Note: If your site requires more than 32 concurrent EXCP sessions, contact Boole & Babbage Customer Support (see [“Customer Support” on page 97](#) for information).

Operational Considerations:

- For VTAM sessions, VTAM must be available and the BBV VTAM application major node must be active before a TAS can be started (see [“Activating the VTAM Application Major Node” on page 54](#) for more information).
- For EXCP communications, the terminals for EXCP sessions must be inactive on the VTAM network before a TAS can be initialized (see [“Inactivating a VTAM Terminal for EXCP Communications” on page 78](#)).
- Each TAS must run as a started task.
- The priority of each TAS should be set above batch jobs running in your system and set similar to the Boole & Babbage online components.
- The started task for each TAS is created at session initialization and terminates when the user logs off or the started task is stopped.
- A TAS initializes when a START command is issued:
 - By the LAS because of a user logon
 - By an operator
 - By an automation or scheduling product
 - During the IPL procedure

Initialization Parameters: When a TAS initializes, it reads parameters that define the session environment from a parameter library (parmlib) member, from the proc, and from the START command, if defined. If appropriate logon parameters are defined in one or more of these locations, a session on a specific terminal can be established without any user intervention.

For example, if the OPT() subparameter is specified with other appropriate logon parameters, a customized character string defined to OPT() gets passed to ISPF at session initialization. ISPF processes this character string to display a product menu, service, view, or screen. When a TAS's session parameters contain OPT() and the session initializes directly into an application like this, the session is known as an *AutoLogon session*.

EXCP sessions require that all logon parameters be defined to parmlib, the proc, and/or the START command because logon services are not provided for EXCP communications. However, you do not have to define all logon parameters for VTAM sessions. In fact, LAS-issued START commands typically contain only the terminal LU name and the user ID values.

Session Security: During session initialization, the TAS checks with the external security manager (ESM) to see if a password is required for the specified user ID. If a password is required, the TAS passes the password to the SAF interface for verification by the ESM.

In the case of an EXCP or an AutoLogon session start, the session fails if the user ID and/or password values are invalid. In the case of a VTAM session start, the TAS displays the password panel if the password value is invalid or if a password has not been provided but is required.

BBVINIT REXX EXEC: Each TAS uses the BBVINIT REXX EXEC (see [Figure 4 on page 15](#) and [Figure 5 on page 16](#)), which

- Provides flexible allocation of ISPF data sets and creates the ISPF environment
- Executes the MAINVIEW CLIST for the MainView Selection Menu
- Passes the OPT() character string to ISPF, if defined

BBVINIT dynamically allocates the ISPPROF profile data set called *syspref.BBVISPF.PROFILE* for each TAS, if it detects that one does not exist at logon. This avoids a potential conflict with the ISPPROF data set at session initialization if the user is concurrently logged on to TSO with the same user ID. In addition, BBV can be customized to use a different name prefix for its ISPF profile data set.

Initialization Sequence of Events: The following sequence of events occurs during TAS initialization:

- The TAS is started by the LAS (for a VTAM terminal user logon) or by a START command issued by an operator, by an automation or scheduling product, or during the IPL procedure.
- The TAS verifies the user ID with the ESM through the SAF interface.
- The TAS displays the password panel, if needed, and verifies the password with the ESM through the SAF interface.
- Upon receiving verification from the ESM, the TAS invokes BBVINIT to
 - Allocate session data sets
 - Start ISPF for the session
 - Execute the MAINVIEW CLIST
 - Pass the OPT() character string

Limitations of the Architecture

BBV supports only the features and functions required by Boole & Babbage products. ISPF and TSO/E must be installed on the same system for BBV to run because the Boole & Babbage products that use BBV require ISPF, and ISPF requires TSO/E.

BBV does not

- Support access to non-Boole & Babbage products
- Provide access to the TSO READY prompt
- Allow terminal reconnects
- Receive broadcast messages

- Support TSO TEST
- Support concurrent BBV and TSO sessions running under the same user ID and using the same ISPF profile data set
- Support the ATTN key; BBV uses the PA1 key to invoke the ATTENTION function
- Support sessions through 3x74 Model A and Model C controllers
- Support sessions on 327x Model 1 terminals

Technical Requirements

BBV requires no changes to the operating system, hardware, or any Boole & Babbage online components; however, BBV must execute from an APF-authorized library.

The priority of TASs should be set above batch jobs and set similar to the Boole & Babbage online components you are running.

The following sections explain specific hardware and software considerations.

Hardware Considerations

BBV can run on any system hardware configuration that supports MVS/XA or MVS/ESA.

Terminal Types: BBV supports the following terminal types:

- All model 327x, except Model 1
- 3742
- Plasma display
- Dialup
- Terminals with screen sizes up to 62 x 162
- Terminals that can be queried

Controller Types: BBV sessions are not supported through 3x74 Model A and Model C controllers.

DASD: The approximate number of 3390-cylinders required for the BBV target libraries is 32; for the BBV distribution libraries, the approximate number is 38.

Virtual Storage: Boole & Babbage recommends that you allow 4096K of private storage to run BBV. The default LAS and TAS procedures specify REGION=4096 on the EXEC statement. Boole & Babbage recommends that you do not change this value.

The CSA requirements for BBV are shown in the following table:

Table 1. CSA Requirements

Address Space Type	CSA
LAS	128 Bytes
Each VTAM TAS	284 Bytes
All EXCP TASs	6K
Note: The first EXCP TAS to initialize after an IPL allocates 6K of CSA; this allocation supports any number of EXCP sessions and remains allocated until the next IPL.	

Software Considerations

BBV requires the following software configuration:

- MVS/XA (all versions) or MVS/ESA (all versions)
- ISPF 2.1 or higher
- TSO/E 2.1 or higher
- VTAM 2.2 or higher (VTAM session access only)

BBV operates with the following external security manager system software products:

- RACF 1.8.0 or higher
- CA-TOP SECRET 4.2 or higher
- CA-ACF2 5.2 or higher

ISPF Maintenance

Aside from any normal product maintenance, BBV may require library updates associated with ISPF maintenance. BBV uses the ISPF libraries to establish an ISPF environment for each session at TAS initialization. The ISPF library allocations are contained in *hi level*. UBBSAMP, in a REXX EXEC called BBVINIT. If maintenance is applied to ISPF that changes the library names, you must change the library names in BBVINIT as well.

You can update BBVINIT by

- Invoking AutoCustomization and performing the step associated with creating the ISPF environment (see [“Using AutoCustomization” on page 9](#) for more information).
- Manually updating BBVINIT (see [“Step 1: Defining the ISPF Environment” on page 13](#) for more information).

Chapter 2. Customizing MainView Alternate Access

There are two methods of customizing MainView Alternate Access:

- AutoCustomization
- Manual customization (see page 13)

This chapter describes both methods and explains the following BBV customization topics:

- Using the BBV(YES) parameter, which is required for CMF MONITOR customers.
- Verifying the BBV Load modules and PTF maintenance levels (see page 29).
- Defining other initialization parameter list members (see page 31).
- Understanding TAS initialization parameters (see page 31).
- Understanding LAS initialization parameters (see page 35).
- Verifying your VTAM configuration (see page 37).
- Moving BBV components from a test environment to a production environment (see page 37).

Using AutoCustomization

You can customize BBV using AutoCustomization by completing the BBV steps presented for any MainView product. If you own more than one MainView product, BBV is customized for all products when you complete the BBV steps in AutoCustomization for any one product.

If you choose to use AutoCustomization to customize BBV, see the Boole & Babbage Product Installation and Maintenance Guide for AutoCustomization startup and usage instructions. The steps shown in [Figure 3 on page 10](#) are displayed for BBV.

Accessing AutoCustomization for CMF MONITOR Customers

If you are a CMF MONITOR customer and you want to use AutoCustomization to customize BBV, you must specify a required parameter when you invoke AutoCustomization to make the BBV steps appear on the CMF MONITOR Product Customization Steps panel. Without this parameter, only the SHR and CMF steps appear.

The required parameter is BBV(YES). It should be entered with the AutoCustomization startup command as follows:

```
TSO EX ' hi level. BBCLIB(BBCUST) ' ' BBV(YES) '
```

where

hi level Is the high level qualifier for your Boole & Babbage product libraries.

' **BBV(YES)** ' Is the required parameter you must enter with the AutoCustomization startup command.

Executing the BBV Steps

The BBV steps appear when you invoke AutoCustomization and access the Product Customization Steps panel for any Boole & Babbage product that uses BBV. An example of the BBV steps is shown in [Figure 3](#).

```

BOOLE & BABBAGE ----- PRODUCT CUSTOMIZATION STEPS ----- ROW 1 TO 21 OF 21
COMMAND ==> SCROLL ==> PAGE

Valid line commands: Step Status(S) Step Flag(F)
S - Select a step. (Must be selected in sequence) -----
B - Browse a step. (No actions will be taken and + completed o optional
    may be browsed out of sequence) - bypassed

Step S F Description Product
-----
 1 + Specify jobcards and other operational defaults SHR
 2 + o Implement GDDM/PGF support SHR
 3 + o Implement Katakana terminal support SHR
 4 + Create site data sets for use with MainView products SHR
 5 + o Create historical data sets for use with MainView products SHR
 6 + Add our load library to your system APF list SHR
 7 - o Add our load library to your system linklist SHR
 8 + Create Clist for invoking MainView products SHR
 9 - o Reload all BBX services SHR
10 - Create procedure to start the CAS (Coordinating Address Space) SHR
11 - o Create procedure to start the COMMON STORAGE MONITOR (CSMON) SHR
12 + Allocate PARMFILE data set for MVS products SHR
13 - Specify Extractor operating environment SHR
14 + Create the PAS procedure for MainView MVS products SHR
15 + Customize Alternate Access to Boole & Babbage products BBV
16 Customize Alternate Access environment for ISPF BBV
17 o Customize VTAM for Alternate Access BBV
18 Create TAS initialization parms and start up procedure BBV
19 o Create LAS initialization parms and start up procedure BBV
20 Copy sample parameter members to UBBPARM for MV MVS products MMR
21 o Copy sample screen definitions to SBBSEDEF for MV MVS products MMR
***** BOTTOM OF DATA *****

```

Figure 3. Example of the BBV Steps on a Product Customization Steps Panel

Each of the BBV steps is described briefly in [Table 2 on page 11](#). The panels for each step prompt you for specific customization information; HELP is available by pressing PF1. Should you need additional information, you can refer to the corresponding manual customization instructions provided in [“Using Manual Customization” on page 13](#).

Table 2. AutoCustomization Steps

Step	Description
Customize Alternate Access to Boole & Babbage products	<p>(required) Allows you to specify whether you want to customize BBV.</p> <p>By responding NO, the remaining BBV steps are marked as bypassed for you automatically. You can continue with any incomplete product customization steps or exit the Product Customization Steps panel and your product status is once again OPERATIONAL.</p> <p>By responding YES, the remaining BBV steps are changed to indicate whether they are truly required or optional to BBV customization. You can continue with the next BBV step.</p>
Customize Alternate Access environment for ISPF	<p>(required) Defines the ISPF environment and creates the BBVINIT REXX EXEC that activates the ISPF environment for each TAS. See Figure 4 on page 15 and Figure 5 on page 16 for a complete example of the REXX EXEC.</p>
Customize VTAM for Alternate Access	<p>(optional) Is necessary if VTAM is to be used to access your Boole & Babbage products. During this step, the VTAM application major node definition is created, and the VTAM application prefix name and number of minor nodes are defined. See Figure 6 on page 18 for an example of the VTAM application major definition.</p> <p>Optionally, you can create a sample USS table (USSTAB) entry and/or a sample logon interpret table (LOGTAB entry). See Figure 7 on page 19 for an example of the sample USSTAB table entry; see Figure 8 on page 20 for an example of the BBVLASxx member</p> <p>Note: AutoCustomization creates VTAM samples and an application member, but does not modify your VTAM configuration. See “Verifying Your VTAM Configuration” on page 37 for a list of VTAM configuration issues.</p>
Create TAS initialization parms and start up procedure	<p>(required) Creates both the TAS proc and initialization parameter list member.</p> <p>See Figure 9 on page 22 for an example of the default TAS initialization parameter list; see Figure 10 on page 24 for an example of the default TAS proc.</p>
Create LAS initialization parms and start up procedure	<p>(optional) Creates both the LAS proc and initialization parameter list member. The LAS proc and parameter list are required only for VTAM access by users who log on from the terminal.</p> <p>See Figure 11 on page 26 for an example of the default LAS initialization parameter list; see Figure 12 on page 27 for an example of the default LAS proc.</p>

After AutoCustomization

After BBV is customized, you can

1. Verify the installation of your BBV load modules (optional)
See [“Verifying Your BBV Load Modules” on page 29](#) for information about running a job that verifies the presence of all CSECTs in the BBV load modules.
2. Enable VTAM terminal session support (required for VTAM access)
See [“Verifying Your VTAM Configuration” on page 37](#) to verify support for VTAM terminal sessions.
3. Configure security for BBV sessions (required for some ESMs)
See [Chapter 3, “Defining Security for MainView Alternate Access” on page 39](#) for information about securing sessions.
4. Log on from a terminal connected to your VTAM network (optional)
See [Chapter 4, “Logging On and Off from a VTAM Terminal” on page 53](#) for instructions on how to start the LAS and log on from a VTAM terminal.
5. Configure and start a VTAM AutoLogon session (optional)
See [Chapter 5, “Configuring, Starting, and Terminating VTAM AutoLogon Sessions” on page 65](#) for information about VTAM AutoLogon terminal sessions.
6. Configure and start an EXCP AutoLogon session (optional)
See [Chapter 6, “Configuring, Starting, and Terminating EXCP AutoLogon Sessions” on page 77](#) for information about EXCP AutoLogon terminal sessions.

Using Manual Customization

This section provides step-by-step instructions on how to manually customize BBV. Some steps are optional.

Important

Because of library allocations, BBV manual customization cannot be performed until *after* you have completed manual customization for all your MainView products. See the appropriate product *Installation Checklist* letter for product customization information.

Step 1: Defining the ISPF Environment

An ISPF environment is identified for each VTAM or EXCP session during TAS initialization by a REXX EXEC called `BBVINIT`. `BBVINIT` should include a basic set of ISPF libraries, but should not contain any user or application libraries. (See your ISPF documentation for more information about ISPF libraries.)

An active TSO session and an active TAS session cannot share the same ISPF profile data set concurrently; therefore, Boole & Babbage recommends a separate ISPF profile data set name format for the ISPF environment. See [“Defining ISPF Profile Data Sets for Concurrent TSO and BBV Sessions” on page 14](#) for more information about running TSO and BBV sessions under the same user ID on the same system. As well, you may want to configure the `SYSPREF=` TAS parameter for your session data set allocations (see [page 34](#) for more information) or modify `BBVINIT` to allocate separate ISPF LOG and LIST data sets (see [“Defining ISPF LOG and LIST Data Sets for Concurrent TSO and BBV Sessions” on page 14](#) for more information).

If you want to specify a distinct location for dynamic allocation of your session data sets, see information about the TAS parameter, `DEVGROUP=`, on [page 33](#).

To define the ISPF environment

1. Create *hi level*. `UBBSAMP` member `BBVINIT`.

Notes:

- You must name your new member `BBVINIT`. Do not use a different name because BBV will not initialize sessions.
 - The `UBBPARM` data set cannot be a PDSE type data set and must have the following attributes:
 - `DSORG=PO`
 - `LRECL=80`
 - `RECFM=F` or `RECFM=FB` (FBS is not supported)
2. Copy `BBSAMP` member `BBVINIT` to the new member. `BBVINIT` is shown in [Figure 4 on page 15](#) and [Figure 5 on page 16](#).
 3. Follow the instructions at the top of the `UBBSAMP(BBVINIT)` member to customize the REXX EXEC.
 4. Add unit and volume information to the allocation statement of the ISPF profile data set if your site does not support SMS. Refer to the appropriate IBM book for REXX to find out

how unit and volume parameters should be coded on the ALLOCATE statement for the ISPPROF data set.

Important

Boole & Babbage recommends that you make only the modifications to the REXX EXEC that are listed in these instructions and in the instructions at the top of the BBVINIT member. Boole & Babbage does not support other changes to the REXX EXEC.

Defining ISPF Profile Data Sets for Concurrent TSO and BBV Sessions

The BBVINIT sample provides an instruction for defining the name of the ISPF profile data set for BBV sessions through the ALLOCATE statement for ISPPROF. There is also an optional SYSPREF= parameter that can be defined in the TAS parameter member (see page 34) when sites require a high level qualifier that is different from the user ID name for the ISPF profile data set name. The resulting default ISPF profile data set name formats for concurrent TSO and BBV sessions using the same user ID on the same system are

- *userID*. ISPF. PROFILE for a TSO session
- *userID*. BBV ISPF. PROFILE or
- *syspref*. BBV ISPF. PROFILE for a BBV session

Defining ISPF LOG and LIST Data Sets for Concurrent TSO and BBV Sessions

A conflict over the ISPF LOG and LIST data sets can result if the BBV session needs to access these data sets when the TSO session already has them allocated. If your site requires specific support for LOG and/or LIST data sets for concurrent TSO and BBV sessions under the same user ID:

1. Complete all BBVINIT customization instructions presented in “[Step 1: Defining the ISPF Environment](#)” on page 13.
2. Add the following ALLOCATE statements in your customized BBVINIT directly before the END statement for the ISPPROF ALLOCATE statement.

```
"ALLOCATE F(ISPLOG) DSN(' "SYSPREF". SPFLOGx. LIST' ) SHR REUSE"  
"ALLOCATE F(ISPLIST) DSN(' "SYSPREF". SPFx. LIST' ) SHR REUSE"
```

where **x** is any unique value that does not conflict with a value used by ISPF such as a letter character, since ISPF uses only numbers in these data sets.

Note: If a SYSPREF= parameter is defined in the TAS parameter member, this value is used when allocating the LOG and LIST data sets.

3. Preallocate the LIST and LOG data sets. Refer to ISPF documentation for information about preallocating data sets.

```

/* REXX - IDENTIFY THE ENVIRONMENT - DO NOT DELETE THIS LINE          */
/*- DISTRIBUTED IN BBVVT31.BBVINIT -----*/
/*                                                                    */
/*          REXX EXEC FOR INITIALIZING ISPF ENVIRONMENT IN TAS        */
/*-----*/
/* INSTRUCTIONS:                                                    */
/*                                                                    */
/* - CHANGE ALL ?HILVL TO THE HIGH LEVEL QUALIFIER OF YOUR PRODUCT  */
/* DATA SETS.                                                       */
/*                                                                    */
/* - CHANGE ALL ?ISPPROF TO THE FORMAT OF THE ISPF PROFILE DATA SET */
/* FOR YOUR SITE.  VARIABLES "SYSPREF" AND "SYSUID" ARE PRESET TO    */
/* PROVIDE SYSTEM PREFIX AND USER ID VALUES, RESPECTIVELY.  FOR    */
/* EXAMPLE:      "SYSPREF". "SYSUID". BBVISPF.PROFILE                */
/*                                                                    */
/* - CHANGE ALL ?ISPLLIB TO THE FIRST ISPF DATA SET FOR THE ISPLLIB */
/* CONCATENATION.  INSERT THE REMAINING ISPF DATA SETS IN THIS    */
/* CONCATENATION.  PAY CLOSE ATTENTION TO THE QUOTES, DOUBLE        */
/* QUOTES, AND SEMI COLONS.  SEE THE ISPLLIB ALLOCATE AS AN EXAMPLE.*/
/*                                                                    */
/* - CHANGE ALL ?ISPLIB TO THE FIRST ISPF DATA SET FOR THE ISPLIB  */
/* CONCATENATION.  INSERT THE REMAINING ISPF DATA SETS IN THIS    */
/* CONCATENATION.  PAY CLOSE ATTENTION TO THE QUOTES, DOUBLE        */
/* QUOTES, AND SEMI COLONS IN THIS EXAMPLE.                          */
/*                                                                    */
/* - CHANGE ALL ?ISPLMLIB TO THE FIRST ISPF DATA SET FOR THE ISPLMLIB */
/* CONCATENATION.  INSERT THE REMAINING ISPF DATA SETS IN THIS    */
/* CONCATENATION.  PAY CLOSE ATTENTION TO THE QUOTES, DOUBLE        */
/* QUOTES, AND SEMI COLONS.  SEE THE ISPLLIB ALLOCATE AS AN EXAMPLE.*/
/*                                                                    */
/* - CHANGE ALL ?ISPLSLIB TO THE FIRST ISPF DATA SET FOR THE ISPLSLIB */
/* CONCATENATION.  INSERT THE REMAINING ISPF DATA SETS IN THIS    */
/* CONCATENATION.  PAY CLOSE ATTENTION TO THE QUOTES, DOUBLE        */
/* QUOTES, AND SEMI COLONS.  SEE THE ISPLLIB ALLOCATE AS AN EXAMPLE.*/
/*                                                                    */
/* - CHANGE ALL ?ISPLTLIB TO THE FIRST ISPF DATA SET FOR THE ISPLTLIB */
/* CONCATENATION.  INSERT THE REMAINING ISPF DATA SETS IN THIS    */
/* CONCATENATION.  PAY CLOSE ATTENTION TO THE QUOTES, DOUBLE        */
/* QUOTES, AND SEMI COLONS.  SEE THE ISPLLIB ALLOCATE AS AN EXAMPLE.*/
/*                                                                    */
/* - CHANGE ALL ?SYSPROC TO THE FIRST ISPF DATA SET FOR THE SYSPROC */
/* CONCATENATION.  INSERT THE REMAINING ISPF DATA SETS IN THIS    */
/* CONCATENATION.  PAY CLOSE ATTENTION TO THE QUOTES, DOUBLE        */
/* QUOTES, AND SEMI COLONS.  SEE THE ISPLLIB ALLOCATE AS AN EXAMPLE.*/
/*                                                                    */
/* - IF YOU USE SYSEXEC, INSERT AN ALLOCATE STATEMENT FOR YOUR      */
/* SYSEXEC DATA SETS.  USE THE SYSPROC ALLOCATE AS AN EXAMPLE.    */
/*                                                                    */
/* - IF MMR OR CMF IS INSTALLED, CHANGE ALL ?MVLIB TO THE DATA SET */
/* CONTAINING THE MAINVIEW CLIST.  CHANGE ALL ?MVNME TO THE NAME    */
/* ASSIGNED TO THE MAINVIEW CLIST.                                  */
/*                                                                    */
/* - IF ONLY BBI2 PRODUCTS ARE INSTALLED, CHANGE ?MVLIB TO THE     */
/* DATA SET CONTAINING THE BBI-TS CLIST.  CHANGE ?MVNME TO THE    */
/* NAME ASSIGNED TO THE BBI-TS CLIST.                               */
/*                                                                    */
/* - MODIFY THE CHANGE LOG TO REFLECT WHEN THIS MEMBER WAS ALTERED.*/
/*                                                                    */
/* END OF INSTRUCTIONS.                                             */
/*-----*/
/* CHANGE LOG:   CREATED BY ?USER ON ?DATE AT ?TIME                */
/*-----*/
/*                                                                    */
/* WHEN NECESSARY, UNCOMMENT THE FOLLOWING LINE TO ALLOW TRACING OF */
/* THE BBVINIT EXEC:                                              */
/*   TRACE I

```

```

ADDRESS TSO
SYSPREF = SYSVAR(SYSPREF)
SYSUID = SYSVAR(SYSUID)
PARSE ARG MENU OPT
UPPER MENU OPT

IF OPT = '' THEN AUTOLOG = ''
ELSE          AUTOLOG = 'AUTOLOG'

```

Figure 4. Example of the BBSAMP Member BBVINIT, Part. 1

```

/*****
/*
/* SET SMFID VARIABLE
/*
/*****

@CVT = STORAGE(10, 4);          /* GET CVT ADDRESS FROM   */
/* HEX LOCATION 10           */
@SMCA = STORAGE(D2X(C2D(@CVT)+197), 3); /* GET 24-BIT SMCA ADDRESS */

SMFID = STORAGE(D2X(C2D(@SMCA)+16), 4); /* SET SMFID VARIABLE   */

/*****
/*
/* ALLOCATE ISPF USER PROFILE DATA SET FOR BBV
/*
/*****

CALL MSG(OFF)
"ALLOCATE F(ISPPROF) ",
"DSN(' ?ISPPROF' ) ",
"SHR REUSE"
IF RC /= 0 THEN ;
DO
CALL MSG(ON)
"ALLOCATE F(ISPPROF) ",
"DSN(' ?ISPPROF' ) ",
"SPACE(2 1) TRACKS DIR(10) ",
"NEW CATALOG",
"RECFM(F B) LRECL(80) BLKSIZE(8880) "
END
/*****
/*
/* ALLOCATE OTHER NECESSARY ISPF DATA SETS FOR BBV
/*
/*****

CALL MSG(OFF)
"ALLOCATE F(ISPLLIB) DSN(' ?ISPLLIB' ",
") SHR";

"ALLOCATE F(ISPLLIB) DSN(' ?ISPLLIB' ",
"' ?HI LVL. BBPLIB' ",
") SHR";

"ALLOCATE F(ISPMLIB) DSN(' ?ISPMLIB' ",
") SHR";

"ALLOCATE F(ISPSLIB) DSN(' ?ISPSLIB' ",
") SHR";

"ALLOCATE F(ISPTLIB) DSN(' ?ISPTLIB' ",
") SHR";

"ALLOCATE F(SYSPROC) DSN(' ?SYSPROC' ",
") SHR REUSE";

CALL MSG(ON)
/*****
/*
/* INVOKE APPROPRIATE BOOLE & BABBAGE PRODUCT MENU
/*
/*****
IF MENU = "MV" | MENU = "PMG" | MENU = "MAINVIEW" THEN DO

IF MENU = "MV" | MENU = "MAINVIEW" THEN DO
"ISPSTART CMD(EX ' ?MVLIB(?MVNME)' ' ?AUTOLOG' )";
EXIT
END

IF MENU = "PMG" THEN DO
"ISPSTART CMD(EX ' ?HI LVL. BBCLIB(BBVCPMG)' ",
"' ?HI LVL " ?AUTOLOG' )";
EXIT
END
ELSE DO
SAY "INVALID COMMAND: COMMAND MUST BE 'MV' OR 'PMG' "
END
RETURN 0;

```

Figure 5. Example of the BBSAMP Member BBVINIT, Part. 2

Step 2: (Optional) Defining the VTAM Major and Minor Nodes

If you require VTAM session access, the instructions in this step are required; if your site needs only EXCP session access, skip this step.

For VTAM access, a VTAM application major node definition must be created and placed in a data set that resides in the VTAMLST DD statement concatenation. Also, the VTAM minor node name prefix and number of minor nodes must be defined. The number of minor nodes that are defined determines the maximum number of VTAM sessions that can run concurrently.

To define the VTAM application major node definition, minor node name, and number of minor nodes:

1. Create member **BBVTASA** in a data set that belongs to the VTAMLST DD statement concatenation, or make sure you move this new member to a VTAMLST DD data set once you have completed your modifications.
2. Copy *hi level*. **BBSAMP** member **BBVTASA** to the new member.
3. This member, shown in [Figure 6 on page 18](#), can be used without modification to define the default VTAM application major node name (**BBVTASA**) and minor node root name (**BBVTAS**), if these names are acceptable to your site. If not, change the names.

If you change the names

- And use **NETVIEW** to manage nodes and applications, the major node name must be the same as the minor node root name, plus one character. For example, if the minor node root name is **BBTAS**, the major node name must be **BBTASx**, such as **BBTASA**.
 - The prefix of the minor node **APPL** statements must match the minor node root name.
4. One minor node **APPL** statement must be defined for each VTAM session. A default of 10 statements (**BBVTAS01 APPL** through **BBVTAS10 APPL**) is provided in the **BBVTASA** member. Delete or add statements to suit your site requirements.

Note: The maximum number of minor node **APPL** statements that can be defined is determined by the number of characters in the minor node root name plus a unique numeric identifier. The combination of these two elements cannot exceed a total of 8 characters.

For example, if you use the default minor node root name of **BBVTAS**, you can define up to 99 **APPL** statements (**BBVTAS01 APPL** through **BBVTAS99 APPL**); if you use a default minor root node name of **BBTAS**, you can define up to 999 **APPL** statements (**BBTAS001 APPL** through **BBTAS999 APPL**); and so on.

5. After you have verified the root names and **APPL** statements, delete any comment lines that occur after the **VBUILD** statement. Some versions of VTAM do not support comments in the definition area so Boole & Babbage recommends removing the comments to avoid encountering a problem.

```

*--DISTRIBUTED IN BBVVT31.BBVTASA-----
*
* SAMPLE BBV VTAM APPLICATION MAJOR NODE DEFINITION
*
* INSTRUCTIONS:
*
* - REMOVE ALL COMMENTS AFTER THE VBUILD STATEMENT.
*
* - NO ADDITIONAL CUSTOMIZATION IS REQUIRED UNLESS THE NAMES, BBVTASA
*   AND BBVTAS ARE UNACCEPTABLE.  CHANGE EACH OCCURRENCE OF THESE NAMES
*   AS REQUIRED.
*
* END OF INSTRUCTIONS.
*-----*
* CHANGE LOG:
*   CREATED BY ?USER ON ?DATE AT ?TIME
*-----*
BBVTASA VBUILD TYPE=APPL
*
*-----*
*           MINOR NODE ROOT NAME
* THIS NAME MUST BE USED WHEN CUSTOMIZING THE DEFAULT PARAMETER LIST
* FOR THE LAS ADDRESS SPACE.
*-----*
BBVTAS  APPL  AUTH=(PASS, ACQ)
*
*-----*
*           MINOR NODES (ONE PER BBVTAS)
*-----*
BBVTAS01 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS02 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS03 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS04 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS05 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS06 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS07 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS08 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS09 APPL  AUTH=(PASS, ACQ), EAS=1
BBVTAS10 APPL  AUTH=(PASS, ACQ), EAS=1
*
*-----*
* THIS EXAMPLE ALLOWS 10 CONCURRENT VTAM SESSIONS TO ALTERNATE
* ACCESS.  INCREASE OR DECREASE THESE ENTRIES BASED ON YOUR
* REQUIREMENTS.
*-----*

```

Figure 6. Example of the BBSAMP Member BBVTASA

Step 3: (Optional) Defining a USS Table Entry

If you require VTAM session access from a VTAM terminal, the instructions in this step may be required; if your site needs only EXCP session access or will configure only VTAM AutoLogon sessions, you can skip this step.

Optionally, you may need to update your USS table (USSTAB) if you want to define a shorter logon command for users that log on from a VTAM terminal; the default command is BBV. To update the USS table, a USSTAB table entry must be added to your production USS table and the table must then be recompiled.

Note: Your site may require that you check with your VTAM site administrator about adding the sample USSTAB entry and recompiling your production USS table.

A sample USSTAB table entry is provided as shown in [Figure 7](#). You are not required to use this sample table entry; you can create your own.

To define the sample USSTAB table entry:

1. Create UBBSAMP member BBVUSST.
2. Copy *hi level*. BBSAMP member BBVUSST to the new member.
3. Follow the instructions at the top of the UBBSAMP(BBVUSST) member to make modifications, if necessary.
4. Add the table entry to your production USS table.
5. Recompile the USS table.
6. Ensure that you define the USS=YES parameter defined in your LAS parameter list member (see [“Step 7: \(Optional\) Defining a Default LAS Initialization Parameter List”](#) on [page 25](#) for more information).

```
*-- DISTRIBUTED IN BBVVT31. BBVUSST-----*
*
*                               SAMPLE BBV USS TABLE ENTRY
*-----*
* INSTRUCTIONS:
*
* - CHANGE ALL ?VAPPL TO THE MINOR NODE ROOT NAME (DEFAULT IS BBVTAS)
*   SPECIFIED IN THE VTAM APPLICATION MAJOR NODE DEFINITION.
*
* END OF INSTRUCTIONS.
*-----*
*
*USSTAB  USSTAB  FORMAT=V3R2
*
* INSERT THESE FOUR STATEMENTS INTO YOUR USS TABLE:
*
BBV      USSCMD  CMD=BBV, REP=LOGON, FORMAT=BAL
         USSPARM PARM=P1, REP=DATA
         USSPARM PARM=APPLID, DEFAULT=?VAPPL
         USSPARM PARM=LOGMODE
*
*
*          USSEND
*          END
```

Figure 7. Example of the BBSAMP Member BBVUSST

Step 4: (Optional) Defining Logon Interpret Table Entries

If you require VTAM session access from a VTAM terminal, the instructions in this step may be required; if your site needs only EXCP session access or will configure only VTAM AutoLogon sessions, you can skip this step.

Optionally, you may need to update your logon interpret table (LOGTAB) if you want to define a shorter logon command for users that log on from a VTAM terminal; the default command is BBV. To update the logon interpret table, LOGTAB table entries must be added to your production logon interpret table and the table must then be recompiled.

Note: Your site may require that you check with your VTAM site administrator about adding sample LOGTAB entries and recompiling your production logon interpret table.

Sample LOGTAB table entries are provided as shown in [Figure 8](#). You are not required to use these sample table entries; you can create your own.

To define sample LOGTAB table entries:

1. Create UBBSAMP member BBVLIT.
2. Copy *hi level*. BBSAMP member BBVLIT to the new member.
3. Follow the instructions at the top of the UBBSAMP(BBVLIT) member to make modifications, if necessary.
4. Add the table entry to your production logon interpret table.
5. Recompile the logon interpret table.
6. Ensure that you define the USS=NO parameter defined in your LAS parameter list member (see [“Step 7: \(Optional\) Defining a Default LAS Initialization Parameter List”](#) on [page 25](#) for more information).

```
*-- DISTRIBUTED IN BBVVT31. BBVLIT-----*
*
*           SAMPLE BBV LOGON INTERPRET TABLE ENTRIES
*-----*
* INSTRUCTIONS:
*
* - CHANGE ALL ?VAPPL TO THE MINOR NODE ROOT NAME (DEFAULT IS BBVTAS)
*   SPECIFIED IN THE VTAM APPLICATION MAJOR NODE DEFINITION.
*
* END OF INSTRUCTIONS.
*-----*
*
*LOGTAB  INTAB
*
* INSERT THESE TWO STATEMENTS INTO YOUR LOGON INTERPRET TABLE:
*
*           LOGCHAR APPLID=(APPLICID, ?VAPPL), SEQNCE=' BBV'
*           LOGCHAR APPLID=(APPLICID, ?VAPPL), SEQNCE=' bbv'
*
*           ENDINTAB
*           END
```

Figure 8. Example of the BBSAMP Member BBVLIT

Step 5: Defining a Default TAS Initialization Parameter List

An initialization parameter list is used when the Terminal Address Space (TAS) initializes. This parameter list must reside in your *hi level . sysid*. UBBPARM dataset and it must be named BBVTAS00. If you do not define a parameter list member, the TAS initializes using default parameters (see [“Understanding TAS Initialization Parameters” on page 31](#) for information about the default values).

You can create other parameter lists containing different initialization parameters if a single default list does not suit your site requirements or you want to configure AutoLogon sessions; see [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#) for more information.

To define a default TAS initialization parameter list:

1. Create UBBPARM member BBVTAS00.

Notes:

- Create UBBPARM member BBVTAS00.

You must create this member in *hi level . sysid*. UBBPARM

- You must name your default member BBVTAS00. If defining additional parameter list members, you must name them BBVTASxx, where xx is a unique value. Do not use a different naming convention for your parameter list members.
- The UBBPARM data set cannot be a PDSE type data set and must have the following attributes:
 - DSORG=PO
 - LRECL=80
 - RECFM=F or RECFM=FB (FBS is not supported)

2. Copy *hi level . BBPARM* member BBVTAS00 to the new member.

3. Follow the instructions at the top of the UBBPARM(BBVTAS00) member to customize the BBVTAS00 initialization parameter list. The instructions and sample member are shown in [Figure 9 on page 22](#).

For more information about the TAS initialization parameters, see [“Understanding TAS Initialization Parameters” on page 31](#).

```

*--DISTRIBUTED IN BBVVT31.BBVTAS00-----*
*
*
*          TERMINAL ADDRESS SPACE (TAS) PARAMETERS
*-----*
* INSTRUCTIONS:
*
* - IF EXECUTING ON KATAKANA TERMINALS, CHANGE ALL ?CAPS TO YES.
*   IF NOT EXECUTING ON KATAKANA TERMINALS, DELETE THE LINE
*   CONTAINING ?CAPS.
*
* - CHANGE ALL ?DEVGROU P TO AN ESOTERIC UNIT NAME FOR DYNAMICALLY
*   ALLOCATING SESSION DATA SETS.
*
* - MODIFY CHANGE LOG TO REFLECT WHEN THIS MEMBER WAS ALTERED.
*
* END OF INSTRUCTIONS.
*-----*
* CHANGE LOG:   CREATED BY ?USER ON ?DATE AT ?TIME
*-----*
CAPS=?CAPS      UPPERCASE FOR KATAKANA TERMINAL SUPPORT
DEVGROU=?DEVGROU  DEFAULT UNIT FOR SESSION DATA SET ALLOCATIONS
*-----*
* TAS PARAMETERS, DEFAULTS, AND DESCRIPTIONS:
*
*CAPS=NO          NO|YES      CAPITALIZE ALL TEXT IN MESSAGES
*                  AND LOGON PANELS
*
*DEVGROU=SYSALLDA      1- 8 CHARACTER GENERIC DEVICE NAME
*                  TO BE USED FOR DYNAMIC ALLOCATION
*                  OF SESSION DATA SETS
*
*SYSPREF=prefix      1- 8 CHARACTER ISPF PROFILE PREFIX
*
*ACCOUNT=            DEFAULT ACCOUNT NUMBER TO BE USED
*                  WHEN ACCOUNT # IS NOT AVAILABLE IN
*                  THE USER SECURITY PROFILE
*-----*
* AUTOLOGON PARAMETERS AND DESCRIPTIONS:
*
*   SEE THE MAINVIEW ALTERNATE ACCESS IMPLEMENTATION AND USER GUIDE
*   FOR MORE INFORMATION ABOUT USING AUTOLOGON.
*
*   NOTE: LOGON STATEMENTS CANNOT CONTAIN COMMENTS.
*
*LOGON APPLID(applid)          VTAM MINOR NODE ROOT NAME
*
*LOGON TERMI D(termid)         VTAM LUNAME
*
*LOGON UCB(ucb)                NON-SNA TERMINAL UCB
*
*LOGON DATA(userid/password)  USER ID & PASSWORD
*
*LOGON OPT(X; X; X; XXXX)      OPT STRING PASSED TO APPLICATION
*                  TO INVOKE PRODUCT PANEL, SERVICE,
*                  OR DISPLAY

```

Figure 9. Example of the BBPARM Member BBVTAS00

Step 6: Defining a TAS Startup Procedure

Each TAS must run as a started task, so a startup procedure is required to initialize each address space. The started task must be executed out of a procedure library.

To create a TAS startup procedure:

1. Create UBBSAMP member BBVTAS.
2. Copy *hi level*. BBSAMP member BBVTAS to the new member.
3. Follow the instructions at the top of the UBBSAMP(BBVTAS) member to customize the BBVTAS proc. The instructions and sample member are shown in [Figure 10 on page 24](#).

For more information about defining the BBVTAS PROCP= and SUF= parameters, see [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#).

Notes:

- If you customize the parameters on the BBVTAS EXEC statement, the SUFFIX= subparameter must be the first parameter defined to the PARM= parameter. If you remove the SUFFIX= subparameter, the default is SUFFIX=00.
 - If you do not have your ISPF modules in LPA, add the libraries containing the ISPF modules to the BBVTAS STEPLIB DD concatenation. You must APF-authorize the ISPF libraries to have them in the STEPLIB DD concatenation.
4. Verify that the TAS proc resides in a procedure library.

```

/**-- DISTRIBUTED IN BBVT31. BBVTAS-----*
/**
/**          TERMINAL ADDRESS SPACE (TAS) STARTUP PROCEDURE
/**-----*
/** INSTRUCTIONS:
/**
/** - CHANGE ALL ?HILVL TO THE HIGH LEVEL QUALIFIER OF THE PRODUCT
/**   LIBRARIES.
/**
/** - CHANGE ALL ?SUF TO THE 2-BYTE SUFFIX ASSIGNED TO THE DEFAULT
/**   BBVTASXX PARAMETER LIST MEMBER.
/**
/** - DELETE THE STEPLIB IF BBLINK IS IN THE LINK LIST CONCATENATION.
/**   IF NOT, CHANGE ALL ?BBLINK TO THE NAME YOU ASSIGNED TO BBLINK.
/**
/** - CHANGE ALL ?UBBPARM TO THE NAME YOU ASSIGNED TO THE USER
/**   PARAMETER LIBRARY.
/**
/** - CHANGE ALL ?UBBSAMP TO THE NAME YOU ASSIGNED TO THE USER
/**   SAMPLE LIBRARY.
/**
/** - MODIFY CHANGE LOG TO REFLECT WHEN THIS MEMBER WAS ALTERED.
/**
/** END OF INSTRUCTIONS.
/**-----*
/** CHANGE LOG:   CREATED BY ?USER ON ?DATE AT ?TIME
/**-----*
/**
/**BBVTAS  PROC P=,          * TAS PARAMETER OVERRIDES
/**          SUF=?SUF      * BBVTASXX MEMBER SUFFIX
/**
/**-----*
/** TAS PROC PARAMETERS AND DESCRIPTIONS:
/**
/** P   - USED TO PASS A LOGON COMMAND STRING OR OTHER TAS
/**       PARAMETERS WHEN USING MVS START TO CREATE A TAS SESSION.
/**       SEE THE MAINVIEW ALTERNATE ACCESS IMPLEMENTATION AND USER
/**       GUIDE FOR FURTHER INFORMATION.
/**
/** SUF  - SPECIFIES THE SUFFIX OF THE BBVTASXX CONTROL STATEMENT
/**        MEMBER YOU WANT THE TAS TO USE.  TAS CONTROL STATEMENT
/**        MEMBERS ARE STORED IN THE PARAMETER LIBRARY ALLOCATED BY
/**        THE DDNAME BBVPARM  SEE THE MAINVIEW ALTERNATE ACCESS
/**        IMPLEMENTATION AND USER GUIDE FOR FURTHER INFORMATION.
/**-----*
/**
/**BBVTAS  EXEC PGM=BBVTLOAD, REGION=4096K, TIME=1440, DYNAMNBR=400,
/**        PARM=' SUFFIX=&SUF: &P '
/**
/**STEPLIB DD  DISP=SHR, DSN=?BBLINK
/**
/**BBVPARM DD  DISP=SHR, DSN=?UBBPARM
/**        DD  DISP=SHR, DSN=?HILVL. BBPARM
/**
/**SYSPROC DD  DISP=SHR, DSN=?UBBSAMP
/**

```

Figure 10. Example of the BBSAMP Member BBVTAS

Step 7: (Optional) Defining a Default LAS Initialization Parameter List

If you require VTAM session access from VTAM terminals, the instructions in this step are required; if your site needs only EXCP session access and/or access for VTAM AutoLogon sessions, you can skip this step.

An initialization parameter list is required when the Logon Address Space (LAS) initializes. This parameter list must reside in your *hi level . sysid . UBBPARM* data set and it must be named BBVLAS00.

You can create other parameter lists containing different initialization parameters if a single default list does not suit your site requirements; see [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#) for more information.

To define the default LAS initialization parameter list:

1. Create UBBPARM member BBVLAS00.

Notes:

- You must create this member in *hi level . sysid . UBBPARM*
 - You must name your default member BBVLAS00. If defining additional parameter list members, you must name them BBVLASxx, where xx is a unique value. Do not use a different naming convention for your parameter list members
2. Copy *hi level . BBPARM* member BBVLAS00 to the new member.
 3. Follow the instructions at the top of the UBBPARM(BBVLAS00) member to customize the BBVLAS00 initialization parameter list. The instructions and sample member are shown in [Figure 11 on page 26](#).

For more information about the LAS initialization parameters, see [“Understanding LAS Initialization Parameters” on page 35](#).

Step 8: (Optional) Defining a LAS Startup Procedure

If you require VTAM session access, the instructions in this step are required; if your site needs only EXCP session access and/or access for VTAM AutoLogon sessions, you can skip this step.

The LAS must run as started task, so a startup procedure is required to initialize the address space. The started task must be executed out of a procedure library.

To create a LAS startup procedure:

1. Create UBBSAMP member BBVLAS.
2. Copy *hi level*. BBSAMP member BBVLAS to the new member.
3. Follow the instructions at the top of the UBBSAMP(BBVLAS) member to customize the BBVLAS proc. The instructions and sample member are shown in [Figure 12](#).

For more information about defining the BBVLAS PROC parameter, SUF=, see [“Controlling the LAS Initialization Parameters” on page 56](#).

4. Verify that the LAS proc resides in a procedure library that is used by started tasks.

```
/*-DISTRIBUTED IN BBVVT31. BBVLAS-----*
/*
/*          LOGON ADDRESS SPACE (LAS) STARTUP PROCEDURE
/*-----*
/* INSTRUCTIONS:
/*
/* - CHANGE ALL ?HILVL TO THE HIGH LEVEL QUALIFIER OF THE PRODUCT
/* LIBRARIES.
/*
/* - CHANGE ALL ?SUF TO THE 2-BYTE SUFFIX ASSIGNED TO THE DEFAULT
/* BBVLASXX PARAMETER LIST MEMBER.
/*
/* - DELETE THE STEPLIB IF BBLINK IS IN THE LINK LIST CONCATENATION.
/* IF NOT, CHANGE ALL ?BBLINK TO THE NAME YOU ASSIGNED TO BBLINK.
/*
/* - CHANGE ALL ?UBBPARM TO THE NAME YOU ASSIGNED TO THE USER
/* PARAMETER LIBRARY.
/*
/* - MODIFY CHANGE LOG TO REFLECT WHEN THIS MEMBER WAS ALTERED.
/*
/* END OF INSTRUCTIONS.
/*-----*
/* CHANGE LOG:   CREATED BY ?USER ON ?DATE AT ?TIME
/*-----*
/*
/*BBVLAS  PROC SUF=?SUF
/*
/*-----*
/* LAS PROC PARAMETER AND DESCRIPTION:
/*
/* SUF - SPECIFIES THE SUFFIX OF THE BBVLASXX CONTROL STATEMENT
/* MEMBER YOU WANT THE LAS TO USE. LAS CONTROL STATEMENT
/* MEMBERS ARE STORED IN THE PARAMETER LIBRARY ALLOCATED BY
/* THE DDNAME BBVPPARM. SEE THE MAINVIEW ALTERNATE ACCESS
/* IMPLEMENTATION AND USER GUIDE FOR FURTHER INFORMATION.
/*-----*
/*
/*BBVLAS  EXEC PGM=BBVLLOAD, TIME=1440, DYNAMNBR=100, REGION=4096K,
/*          PARM=' SUFFIX=&SUF'
/*
/*STEPLIB DD DISP=SHR, DSN=?BBLINK
/*
/*BBVPPARM DD DISP=SHR, DSN=?UBBPARM
/*          DD DISP=SHR, DSN=?HILVL. BBPPARM
```

Figure 12. Example of the BBSAMP Member BBVLAS

Step 9: (Optional) Enabling VTAM Terminal Support

If you require VTAM session access, the instructions in this step are required; if your site needs only EXCP session access, you can skip this step.

There are tasks associated with your VTAM configuration that must be completed and the VTAM application major node must be activated. Furthermore, the LAS must be initialized to enable user logons from VTAM terminals.

To ensure the proper VTAM configuration, activate the VTAM application major node, and initialize the LAS:

1. Verify that your VTAM configuration is complete by reviewing the list in [“Verifying Your VTAM Configuration” on page 37](#) and performing any tasks that are required.
2. Follow the instructions in [“Starting the LAS” on page 55](#) to initialize the LAS for user logons from VTAM terminals.

Step 10: (Required) Verifying Security for BBV Sessions

The instructions in this step are required to provide for proper security of BBV sessions.

All user IDs for BBV sessions must be defined to your external security manager (ESM). Other customization may be required depending on the ESM installed in your system and the security requirements at your site. Additional support can also be configured for AutoLogon sessions to support user IDs that do not require clear text passwords or for VTAM sessions where you want to associate environmental information, such as a terminal ID.

Read the information in [Chapter 3, “Defining Security for MainView Alternate Access” on page 39](#) to verify security and follow any instructions to configure security, if necessary

Step 11: (Optional) Starting an EXCP Session

If you require EXCP session access, the instructions in this step are required; if your site needs only VTAM session access, you can skip this step.

To issue an operator START command to start an EXCP session or to customize a BBVTASxx member to create an EXCP AutoLogon terminal session, follow the instructions in [Chapter 6, “Configuring, Starting, and Terminating EXCP AutoLogon Sessions” on page 77](#).

Step 12: (Optional) Starting a VTAM Session

If you require VTAM session access, the instructions in this step are required; if your site needs only EXCP session access, you can skip this step.

[“Step 9: \(Optional\) Enabling VTAM Terminal Support” on page 28](#) must be completed before you can start a VTAM terminal session. There are two ways to start a VTAM terminal session:

- Follow the instructions in [“Using the LOGON Command” on page 57](#) to log on through a VTAM terminal.
- Follow the instructions in [Chapter 5, “Configuring, Starting, and Terminating VTAM AutoLogon Sessions” on page 65](#) to issue an operator START command to start a VTAM

session or to customize a BBVTASxx member to create a VTAM AutoLogon terminal session.

Verifying Your BBV Load Modules

Boole & Babbage has provided member BBVLVER in *hi level*. BBSAMP, which is a job that verifies the presence of all CSECTs in the BBV load modules. This job creates output that lists each CSECT, the actual CSECT name found, and the PTF indicating the current maintenance level of each CSECT, as well as other information.

To use BBVLVER, shown in [Figure 13](#), follow the instructions at the top of the member and submit the job.

```
//JOBNAME JOB other-job-card-information
/**-----*
/**
/**                      BBV LOAD MODULE VERIFY ROUTINE
/**-----*
/** This JCL can be used to:
/**
/** o Verify that the BBV load modules are linked together correctly
/** o Determine the PTF level for CSECTS in the BBV load modules
/**
/** This routine has no parameters, but the BBV load modules must be
/** loadable by this verification routine. Boole & Babbage suggests
/** that the easiest way to do this is to add a STEPLIB pointing to
/** the BBV load library.
/**
/** INSTRUCTIONS:
/**
/** - CHANGE THE ?BBVHLV HIGH LEVEL QUALIFIER FOR THE STEPLIB
/**   TO THE HIGH LEVEL QUALIFIER OF THE BBV LOAD MODULE LIBRARY.
/**
/** END OF INSTRUCTIONS.
/**-----*
/** CHANGE LOG:   CREATED BY ?USER ON ?DATE AT ?TIME
/**-----*
/**
/**BBVLVER EXEC PGM=BBVLOAD
/**
/**STEPLIB DD DISP=SHR,DSN=?bbvhlv.BBLINK
/**
/**SYSPRINT DD SYSOUT=*
/**
```

Figure 13. Example of the BBSAMP Member BBVLVER

An example of the output that is created is shown in [Figure 14](#).

```

BBVC223I  BBVTBI ND  BBVTBI ND  3. 1. 0  BPV0083  01/07/94  16. 52
BBVC223I  BBVTBTVD  BBVTBTVD  3. 1. 0  BPV0084  01/10/94  12. 16
BBVC223I  BBVTCPAR  BBVTCPAR  3. 1. 0  BPV0046  10/22/93  05. 59
BBVC223I  BBVTDATA
BBVC223I  BBVTDQRY  BBVTDQRY  3. 1. 0  BPV0073  12/23/93  17. 54
BBVC223I  BBVTEATI  BBVTEATI  3. 1. 0  BPV0071  12/10/93  15. 34
BBVC223I  BBVTEATP  BBVTEATP  3. 1. 0  BPV0077  12/23/93  17. 58
BBVC223I  BBVTEATS  BBVTEATS  3. 1. 0  BPV0083  01/07/94  16. 52
BBVC223I  BBVTEATX  BBVTEATX  3. 1. 0  BPV0061  11/08/93  12. 58
BBVC223I  BBVTECTL  BBVTECTL  3. 1. 0  BPV0014  09/21/93  12. 12
BBVC223I  BBVTEDEV  BBVTEDEV  3. 1. 0  BPV0083  01/07/94  16. 53
BBVC223I  BBVTEDYN  BBVTEDYN  3. 1. 0  BPV0083  01/07/94  16. 53
BBVC223I  BBVTETRM  BBVTETRM  3. 1. 0  BPV0077  12/23/93  17. 59
BBVC223I  BBVTEUCB  BBVTEUCB  3. 1. 0  BPV0014  09/17/93  17. 45
BBVC223I  BBVTLOGH  BBVTLOGH  3. 1. 0  BPV0046  10/22/93  10. 25
BBVC224E
BBVC222E  CSECT  BBVTLOGX is missing, check link
BBVC224E
BBVC223I  BBVTLTRM  BBVTLTRM  3. 1. 0  BPV0041  10/13/93  22. 28
BBVC223I  BBVTMAIN  BBVTMAIN  3. 1. 0  BPV0059  11/05/93  18. 28
BBVC223I  BBVTPARM  BBVTPARM  3. 1. 0  BPV0076  12/13/93  12. 51
BBVC223I  BBVTRACE  BBVTRACE  3. 1. 0  BPV0083  01/07/94  16. 53

```

Figure 14. Example of the Output Created by BBVLVER

Each field in this output, from left to right, is explained in the following list:

Message Number

Contains one of three message number values:

- BBVC222E Issued when a CSECT cannot be found; names the missing CSECT.
- BBVC223I Provides information about a CSECT that is present and verifies correctly. If only the CSECT name appears after this message number, the CSECT is actually a sub-CSECT of the preceding primary CSECT.
- BBVC224E Issued before and after the BBVC222E message without any message text. Serves to offset the BBVC222E message so that missing CSECTs are easier to locate in the output.

Expected CSECT Name

Is the name of the CSECT that is expected to be found. There is one entry for each CSECT and sub-CSECT in the BBV load modules.

Actual CSECT Name

Is the name of the CSECT actually found during the verification procedure.

If no value appears in this field, the named CSECT is a sub-CSECT.

If this name is different from the Expected CSECT Name value, contact Boole & Babbage Customer Support (see [Appendix C, "Customer Support" on page 97](#)).

Release Is the release level of the BBV component.

PTF Number

Is the number of the PTF indicating the level of maintenance for the CSECT. If a CSECT has no PTFs associated with it, this field is skipped.

Date Is the date the PTF was created.

Time Is the time the PTF was created.

Defining Other LAS and TAS Initialization Parameter Lists

A single default parameter list may not suit your site requirements for all sessions. You can create other parameter list members containing different initialization parameters and point to them at startup using the `SUF=` parameter (see page 56 for the LAS and page 79 for the TAS).

To define other LAS and TAS initialization parameter list members, follow the instructions for defining the default lists in either “[Step 5: Defining a Default TAS Initialization Parameter List](#)” on page 21 or “[Step 7: \(Optional\) Defining a Default LAS Initialization Parameter List](#)” on page 25, but be sure to name the additional list members `BBVTASxx` or `BBVLASxx`, respectively, where `xx` is a unique identifier.

Note: Boole & Babbage recommends that you always retain default `BBVLAS00` (if needed) and `BBVTAS00` members in `UBBPARM`.

Understanding TAS Initialization Parameters

The default TAS initialization parameters are contained in a list member called `BBVTAS00` that resides in the `hi level . sysid . UBBPARM` data set (see [Figure 9 on page 22](#) for an example of the default member).

You can create additional TAS parameter list members containing other initialization parameters that also reside in `UBBPARM` (see “[Defining Other LAS and TAS Initialization Parameter Lists](#)”). These other TAS parameter list members require a naming convention of `BBVTASxx`, where `xx` is a unique identifier.

A `BBVTASxx` member can be configured to initialize a single AutoLogon session or one or more TASs that require parameters other than those contained in the default `BBVTAS00` member; see “[Defining the START Command for EXCP AutoLogon Sessions](#)” on page 79 and “[Defining the START Command for VTAM AutoLogon Sessions](#)” on page 67, respectively, for more information.

The TAS initialization parameters that can be defined to any list member are shown in the following table.

Table 3. TAS Initialization Parameters

Parameter	Description
ACCOUNT=	<p>(optional) Defines a default account value of up to 39 alphanumeric characters. This account value is used for job accounting in any job card that is generated by ISPF when an account number is not available in the security profile for the specified user ID.</p> <p>BBV cannot access the account information for a user ID in UADS, so when a user submits a job, BBV attempts to obtain the account information from the security profile for the user ID in the external security manager, such as RACF, CA-ACF2, or CA-TOP SECRET. If an account number is not available in the security profile or a security profile for the user ID does not exist, a TAS obtains an accounting value from this parameter.</p> <p>If you do not need to define a default account number, delete this parameter or leave it commented out as *ACCOUNT=.</p>
APPLID()	<p>(optional subparameter) Is used for VTAM terminal sessions only and is discussed in “Using the LOGON Command” on page 57 and “Defining the START Command for VTAM AutoLogon Sessions” on page 67.</p>
CAPS={ YES NO}	<p>(optional) Determines if the TAS should function with or without Katakana terminal support. The default is NO.</p> <p>If you need Katakana terminal support, define YES. All BBV panels and messages are converted to uppercase.</p> <p>If you do not need Katakana terminal support, delete the CAPS=?CAPS parameter or define CAPS=NO.</p>
DATA()	<p>(optional subparameter) Is used for both EXCP and VTAM sessions and is discussed in “Using the LOGON Command” on page 57 and “Defining the START Command for VTAM AutoLogon Sessions” on page 67 for VTAM terminals, and “Defining the START Command for EXCP AutoLogon Sessions” on page 79 for EXCP terminals.</p>

Table 3. TAS Initialization Parameters (Continued)

Parameter	Description
<p>DEVGROU={ device SYSALLDA}</p>	<p>(optional) Defines a one- to eight-character esoteric device name used for dynamic allocations of any session data sets, when a device name is not available in the security profile for the specified user ID.</p> <p>BBV cannot access the device name information for a user ID in UADS, so when a TAS initializes, BBV attempts to obtain the esoteric device name information from the security profile for the user ID in the external security manager, such as RACF, CA-ACF2, or CA-TOP SECRET. If a device name is not available in the security profile or a security profile for the user ID does not exist, the TAS obtains the device name from the DEVGROU= parameter.</p> <p>If a DEVGROU= parameter is not specifically defined and the device name is not available in the security profile, the default value for the esoteric device name is SYSALLDA, which means that data sets can be allocated dynamically to any available device at TAS initialization.</p> <p>Your site may have devices specifically grouped for dynamic allocations, such as ISPF allocations, and require a particular device value be defined.</p> <p>Note: Boole & Babbage recommends that you specify this parameter only if you experience problems with the default allocation unit at TAS initialization. For RACF or CA-TOP SECRET sites, the device name is obtained from the security product database; for CA-ACF2 sites, the device name is obtained from the LOGONID profile.</p>
<p>LOGON</p>	<p>(optional subparameter) Is used for both EXCP and VTAM sessions and is discussed in “Using the LOGON Command” on page 57 and “Defining the START Command for VTAM AutoLogon Sessions” on page 67 for VTAM terminals, and “Defining the START Command for EXCP AutoLogon Sessions” on page 79 for EXCP terminals.</p>
<p>OPT()</p>	<p>(optional subparameter) Is used for both EXCP and VTAM AutoLogon sessions and is discussed in “Defining the START Command for VTAM AutoLogon Sessions” on page 67 for VTAM terminals and “Defining the START Command for EXCP AutoLogon Sessions” on page 79 for EXCP terminals.</p>

Table 3. TAS Initialization Parameters (Continued)

Parameter	Description
<p><code>SYSPREF={ <i>userID</i> <i>prefix</i> }</code></p>	<p>(optional) Defines a one- to eight-character TSO profile prefix value for the ISPF profile data sets that are allocated to a user ID at logon. The <code>SYSPREF=<i>prefix</i></code> value must be defined when</p> <ul style="list-style-type: none"> • A high level qualifier or index name that is different from the user ID is required • The prefix value is not available in the security profile If a <code>SYSPREF=</code> parameter is not defined and the prefix value is not available in the security profile, the default for the prefix value for the ISPF profile data sets allocated at logon is the user ID. <p>If your site uses the user ID as the high level for ISPF profile data sets, delete this parameter or leave it commented out as <code>*SYSPREF=<i>prefix</i></code>; if your site requires a prefix value, manually add the <code>SYSPREF=</code> parameter to the parameter list member with a prefix value.</p> <p>BBV cannot access the prefix value for a user ID in UADS, so when a TAS initializes, BBV obtains the prefix value from the security profile for the user ID in the external security manager.</p> <p>Note: For RACF sites, the prefix value is obtainable when TSO segment data is defined.</p> <p>If a prefix value is not available in the security profile or a security profile does not exist, the TAS obtains the prefix value from the <code>SYSPREF=</code> parameter.</p>
<p><code>TERMI D()</code></p>	<p>(optional subparameter) Is used for VTAM AutoLogon sessions only and is discussed in “Defining the START Command for VTAM AutoLogon Sessions” on page 67.</p>
<p><code>UCB()</code></p>	<p>(optional subparameter) Is used for EXCP AutoLogon sessions only and is discussed in “Defining the START Command for EXCP AutoLogon Sessions” on page 79.</p>

Understanding LAS Initialization Parameters

The default LAS initialization parameters are contained in a list member called BBVLAS00 that resides in the *hl level . sysid . UBBPARM* data set (see [Figure 11 on page 26](#) for an example of the default member).

You can create additional LAS parameter list members containing other initialization parameters that also reside in UBBPARM (see [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#)). These other LAS parameter list members require a naming convention of BBVLASxx, where xx is a unique identifier. A BBVLASxx member can be used to initialize the LAS instead of the default BBVLAS00 member (see [“Controlling the LAS Initialization Parameters” on page 56](#) for more information).

The LAS initialization parameters that can be defined to any list member are shown in the following table.

Table 4. LAS Initialization Parameters

Parameter	Description
APPLID= { appl id BBVTAS }	(required) Defines the VTAM minor node root name assigned to BBV during customization. When a VTAM terminal is signed on, this parameter determines the name of the application where the LAS is to pass the user session. The default value of the VTAM minor node root name is BBVTAS, but this value can be changed during customization.
CAPS={ YES NO }	(optional) Determines if the LAS should function with or without Katakana terminal support. The default is NO. If you need Katakana terminal support, define YES. All BBV panels and messages are converted to uppercase. If you do not need Katakana terminal support, delete the CAPS=?CAPS parameter or define CAPS=NO.
GNAM= <i>generic</i>	(optional) Indicates the generic name used to define one or more VTAM Version 4 Release 4 or later applications. Using a generic name increases VTAM availability; if one VTAM is unavailable, the BBV logon request can be handled by another VTAM. Specify a one- to eight-character generic name. To use a generic name on the LOGON command, specify this parameter and substitute the generic name for <i>appl id</i> on the LOGON command (see page 57). To use a generic name with the abbreviated logon command, specify this parameter and substitute the generic name for ?VAPPL in the USS table entry (see page 19). If this parameter is specified, users can still log on to a particular VTAM application by using the <i>appl id</i> specified for the APPLID LAS initialization parameter.

Table 4. LAS Initialization Parameters

Parameter	Description
PROCNAME= {procname <u>BBVTAS</u> }	(required) Defines the TAS proc for the VTAM session that is initiated by the LAS and then passed off to a TAS. The default value of the TAS proc is <u>BBVTAS</u> , but this value can be changed during customization.
USS={ <u>YES</u> NO}	<p>(optional) Specifies whether BBV is configured with a USS table entry, instead of a logon interpret table entry. Sample entries for either table can be created during customization, but an entry must be added to the appropriate table and the table recompiled before customization is complete.</p> <p>The default is <u>YES</u>. If a USS table entry is customized to your VTAM configuration, either delete the USS=?USS parameter or define USS=<u>YES</u>. This enables the abbreviated logon command (see “Using the Abbreviated Logon Command” on page 60) defined in your USS table entry.</p> <p>If a logon interpret table entry is customized to your VTAM configuration, you must specify USS=NO. This enables the abbreviated logon command (see “Using the Abbreviated Logon Command” on page 60) defined in your logon interpret table entry; however, specifying NO also creates a requirement for defining extra characters in the DATA() parameter of the LOGON command when this command is issued from a VTAM terminal (see “Using the LOGON Command” on page 57).</p> <p>If no table entry exists, you can define <u>YES</u> or NO. If your site requires support of user logons from VTAM terminals, however, you may want to define <u>YES</u> so extra characters are not required in the DATA() parameter of the LOGON command.</p>

Verifying Your VTAM Configuration

If your site requires VTAM session access, some adjustments to your VTAM environment are required or may be necessary once BBV customization is complete. The following list shows the tasks that must be completed:

1. Verify that the VTAM application major node definition (default name `BBVTASA`) resides in a data set concatenated in the `VTAMLST` DD statement (see [Figure 6 on page 18](#) for an example of the sample member that ships with BBV).
2. Activate the VTAM application major node (see [“Activating the VTAM Application Major Node” on page 54](#)).
3. Optionally, add the VTAM application major node name (default name `BBVTASA`) to `SYS1.VTAMLST` member `ATTCONxx` (see your VTAM documentation for more information).
4. Optionally, update and recompile your site’s production USS table (`USSTAB`) with the sample `USSTAB` table entry provided or your own table entry (see [Figure 7 on page 19](#) for an example of the sample member that ships with BBV).
5. Optionally, update and recompile your site’s production logon interpret table (`LOGTAB`) with the sample `LOGTAB` table entry provided or your own table entry (see [Figure 8 on page 20](#) for an example of the sample member that ships with BBV).

Migrating from a Test to a Production Environment

This section assumes that your test system and production system have identical Boole & Babbage product mixes and that you typically migrate products by copying your test target libraries to your production target libraries.

Under this migration scenario, copying BBV from one system to another involves the following tasks:

1. Copy your test system target libraries to your production system target libraries. The libraries are:
 - `BBCLIB`
 - `BBLINK`
 - `BBPLIB`

For information about specific load modules associated with BBV, see [Appendix A, “Libraries and Load Modules” on page 89](#).

2. Copy `UBBSAMP` member `BBVINIT` and the MainView `CLIST` (default name `MAINVIEW`) to a `UBBSAMP` library on your production system, naming the new member `BBVINIT`.

See [Appendix A, “Libraries and Load Modules” on page 89](#) for information about other `UBBSAMP` members.

3. Copy the following `UBBPARM` members on your test system to a `UBBPARM` library on your production system:
 - `BBVLASxx`
 - `BBVTASxx` where `xx` is a unique identifier.

4. Copy the BBVTAS and BBVLAS (if needed) procs from your test system procedure library to your production system procedure library.
5. Copy the VTAM application major node name from your test system to your production system.

Note: If your test system and production system share the VTAMLST data set where the VTAM application major node resides and you want to run BBV concurrently on both systems, you must define a different major node definition (see [“Step 2: \(Optional\) Defining the VTAM Major and Minor Nodes”](#) on page 17) and TAS parameter list member(s) (see [“Step 5: Defining a Default TAS Initialization Parameter List”](#) on page 21).

You also may need to define a different BBVLAS proc and parameter list member, depending on your site requirements for user logons from VTAM terminals (see [“Step 7: \(Optional\) Defining a Default LAS Initialization Parameter List”](#) on page 25 and [“Step 8: \(Optional\) Defining a LAS Startup Procedure”](#) on page 27).

6. Determine whether you need to update the USS table or logon interpret table on your production system; see [“Step 3: \(Optional\) Defining a USS Table Entry”](#) on page 19 or [“Step 4: \(Optional\) Defining Logon Interpret Table Entries”](#) on page 20 for more information.

Chapter 3. Defining Security for MainView Alternate Access

MainView Alternate Access uses the standard MVS SAF interface to interact with your external security manager (ESM) and requires that each user ID for a session be defined to your ESM. A session does not initialize if the user ID is unknown to your ESM.

Other customization to support security may be required as well, depending on which ESM is installed at your site. For example:

- AutoLogon sessions may be established with user IDs that do not require manual entry of clear text passwords by customizing CA-ACF2 or RACF to define and protect a FACILITY(BOOLEBBV) resource or CA-TOP SECRET to define and protect an IBMFAC(BOOLEBBV) resource.
- By using a FACILITY(BOOLEBBV) or IBMFAC(BOOLEBBV) resource, sessions can be further restricted to be used only at specifically authorized terminals or only with MainView Alternate Access.

This type of customization eliminates any potential security exposure resulting from removing the requirement for manual entry of a clear text password.

This chapter discusses the security issues concerning MainView Alternate Access sessions and how to configure CA-ACF2 or RACF to support the FACILITY(BOOLEBBV) resource or CA-TOP SECRET to support an IBMFAC(BOOLEBBV) resource in the following sections:

- [“Understanding AutoLogon Session Security Issues”](#)
- [“Defining Additional Security for Terminals” on page 40](#)
- [“Defining Additional Security for Applications” on page 41](#)
- [“Configuring CA-ACF2” on page 42](#)
- [“Configuring RACF” on page 47](#)
- [“Configuring CA-TOP SECRET” on page 49](#)

Understanding AutoLogon Session Security Issues

AutoLogon sessions sign on a terminal directly to a product service at terminal address space (TAS) initialization, without requiring user intervention to enter a user ID or password.

AutoLogon sessions can be passed through to an application because the OPT() subparameter is passed at session initialization. For a session to initialize properly to support the OPT() subparameter, user ID and password (if required) values must be defined as startup parameters for the session. Because AutoLogon sessions require user ID and password values as startup parameters, security considerations dictate special procedures to prevent the unnecessary exposure of these values.

There are two ways to provide the user ID and a password at AutoLogon session initialization:

- In the START command
- In a BBVTAS_{xx} member

When the FACILITY(BOOLEBBV) or IBMFAC(BOOLEBBV) resource has not been defined to your ESM, the user ID and password must be manually specified in clear text in either the START command or the BBVTAS_{xx} member. Specifying the user ID and a password in clear text presents three potential security problems:

- The user ID and password are visible on the console when specified in a START command.
- The entire START command may be logged to SYSLOG when issued, making the user ID and password values visible in SYSLOG.
- The user ID and password are visible in a BBVTASxx member, which may be accessible by many other individuals.

To avoid the problem of clear-text password exposure for user IDs that are to be used with AutoLogon sessions, Boole & Babbage recommends the following two solutions:

- Define all BBVTASxx members containing user ID and/or password values in a read-secured data set. Additionally, this solution requires that
 - A separate TAS procedure be defined (see [“Step 6: Defining a TAS Startup Procedure” on page 23](#)) that concatenates the read-secured data set to the BBVPARAM DD statement in the procedure
 - The user ID to be used by the TAS when it is started be given access to this read-secured data set
- Define the resource FACILITY(B00LEBBV) to ACF2 or RACF or the IBMFAC(B00LEBBV) resource to TOP SECRET to eliminate the requirement to code clear text passwords for IDs that are authorized to access this resource.

Defining Additional Security for Terminals

MainView Alternate Access passes environmental information, such as the TERMID= parameter, to the ESM through RACROUTE to enable the installation to further restrict access.

The TERMID= parameter is an optional parameter that is specified in the RACROUTE VERIFY (signon) parameter list. The TERMID= is the VTAM LU or EXCP terminal address from which a MainView Alternate Access logon is being attempted. If the terminal is protected and the user ID is not allowed access from the terminal, the logon terminates after an error message is issued.

For RACF and CA-TOP SECRET, all VTAM LUs or addresses are protected using class TERMINAL. For CA-ACF2, terminal access authorization is handled through standard CA-ACF2 terminal/source group restrictions.

Specific examples for defining resources in class TERMINAL are provided in the following sections:

- [“Customizing Class TERMINAL Access Authorization for CA-ACF2” on page 45](#)
- [“Customizing Class TERMINAL Access Authorization for RACF” on page 48](#)
- [“Customizing Class TERMINAL Access Authorization for CA-TOP SECRET” on page 51](#)

Defining Additional Security for Applications

MainView Alternate Access passes environmental information, such as the APPL= parameter, to the ESM through RACROUTE to enable the installation to further restrict access. To use application names to control access to MainView Alternate Access, each application name must be defined as a resource in class APPL.

There are two instances where MainView Alternate Access uses resources in SAF class APPL to control access to MainView Alternate Access:

- The first instance is when the user ID attempting MainView Alternate Access logon is validated. This instance is intended to offer protection at the application level. The APPL= parameter is an optional parameter that is specified in the RACROUTE VERIFY (signon) parameter list. In this case, an application name of BBVLOGON is passed as the APPL= parameter when MainView Alternate Access validates the user ID. The security system uses this parameter automatically to validate application access.

If the installation wants to restrict MainView Alternate Access to only a certain subset of users, the resource entity name BBVLOGON in SAF resource class APPL may be defined. The ESM ensures, as part of user ID validation, that the user ID has at least read access to the BBVLOGON resource in class APPL. For CA-ACF2, SAF resource class APPL is usually mapped to a unique generalized resource rule type; the default for CA-ACF2 6.x is TYPE(APL).

- The second instance is designed to offer protection at the VTAM major node name level. Once the user ID has been validated (and access to the BBVLOGON application validated), BBV explicitly invokes the ESM to authorize access to a resource in SAF class APPL whose resource entity name is the VTAM major node name.

By controlling access to the VTAM major node name, the installation can, for example, allow a user to log on to MainView Alternate Access on specific MVS images only, or to log on to a test version of MainView Alternate Access, but not to a production version. For these examples to function as illustrated, the VTAM major node names used for MainView Alternate Access must be unique across different MVS images or between production and test versions.

Note: The VTAM major node name referred to here is the value specified for the APPLID= parameter in the BBVLASxx member in *hilevel.sysid*.UBBPARM.

When the MainView Alternate Access session is not using VTAM, but is instead an EXCP session, the constant, BBVEXCP, is used for the resource entity name instead of a VTAM major node name.

Specific examples for defining resources in class APPL are provided in the following sections:

- [“Customizing Class APPL Access Authorization for CA-ACF2” on page 45](#)
- [“Customizing Class APPL Access Authorization for RACF” on page 48](#)
- [“Customizing Class APPL Access Authorization for CA-TOP SECRET” on page 51](#)

Configuring CA-ACF2

The following customization is required to

- Define LOGONIDs for appropriate MainView Alternate Access started tasks and a FACILITY (BOOLEBBV) resource
- Allow access to the resource for LOGONIDs associated with user IDs to be used in AutoLogon sessions

Contact your CA-ACF2 system programmer or security administrator for help.

1. Define a LOGONID for the BBVTAS started task. This LOGONID is used during the initial phase of startup. The user's LOGONID is used for the session once the session initializes; therefore, users do not require additional or separate LOGONIDs to use MainView Alternate Access.

Note: The started task LOGONID requires access to the Boole & Babbage product data sets.

Define the LOGONID with the following attributes:

```
DUMPAUTH
STC
RESTRICT
SOURCE(STCINRDR)
```

Warning

If an CA-ACF2 LOGONID is defined with the NON-CNCL attribute, that LOGONID is allowed access to all resources. A BBV session can be initialized with a NON-CNCL LOGONID without entering a valid password; therefore, Boole & Babbage strongly recommends that NON-CNCL LOGONIDs not be permitted.

2. This optional step requires an IPL of your system. This step may not be applicable to your site.

If you use the ACF2 started task validation exit to assign a LOGONID to a started task:

- a. Add the LOGONID for the BBVTAS started task to the exit.
- b. Assemble and relink the exit.
- c. IPL your system to activate the revised exit.

3. Repeat the previous two steps for the BBVLAS started task, if necessary.
4. For CA-ACF2 5.2, update your CA-ACF2 SAFPROT list to include a control point for BBVTLOAD.

The following is a sample set of TSO commands to perform this update:

```
ACF
SET CONTROL(GSO)
INSERT SAFPROT.BBV CLASSES(-) CNTLPTS(-) SUBSYS(BBVTLOAD)
END
```

For CA-ACF2 6.x, define a SAFDEF record for BBVTLOAD.

The following is a sample set of TSO commands to perform this update:

```
ACF
SET CONTROL(GSO)
INSERT SAFDEF.BBVTLOAD ID(BBVTLOAD) +
      MODE(GLOBAL) REP +
      RACROUTE(SUBSYS=BBVTLOAD, REQSTOR=-)
END
```

5. For CA-ACF2 5.2, perform a CA-ACF2 refresh of the SAFPROT list.

Use the following operator command to refresh the SAFPROT list:

F ACF2, REFRESH(SAFPROT)

For CA-ACF2 6.x, perform a CA-ACF2 refresh of the SAFDEF records.

Use the following operator command to refresh all the SAFDEF records:

F ACF2, REFRESH(SAFDEF)

6. For CA-ACF2 5.2, optionally define a mapping between the SAF resource class FACILITY and a CA-ACF2 generalized resource rule of any type. If you define this mapping on CA-ACF2 5.2, you must update the SAFMAPS GSO record and refresh the in-storage SAFMAPS table.

- a. The following is a sample set of TSO commands to update the SAFMAPS GSO record for CA-ACF2 5.2, assuming you selected the generalized resource rule type FAC:

```
ACF
SET CONTROL(GSO)
CHANGE SAFMAPS ADD MAPS(FAC/FACILITY)
END
```

- b. Use the following operator command to refresh the in-storage SAFMAPS table for CA-ACF2 5.2:

F ACF2, REFRESH(SAFMAPS)

For CA-ACF2 6.x, the SAF resource class FACILITY is mapped automatically to the generalized resource rule type FAC, so no action is required.

7. Some MVS components may require that the rules for the rule type selected for use with SAF class FACILITY be resident. If you need to perform this customization

- And you have already migrated to the INFODIR record for either CA-ACF2 6.x or for CA-ACF2 5.2, update the INFODIR GSO record, refresh the INFODIR information in storage, and build the in-storage directory for the selected rule type.

- a. The following is a sample set of TSO commands to add rule type FAC to the INFODIR GSO record:

```
ACF
SET CONTROL(GSO)
CHANGE INFODIR ADD TYPES(R- RFAC)
END
```

- b. Use the following operator command to refresh the INFODIR data:

F ACF2, REFRESH(INFODIR)

- c. Use the following operator command to build the in-storage directory for the rule type FAC:

F ACF2, REBUILD(FAC)

- And your CA-ACF2 5.2 configuration uses the RESDIR GSO record instead of the INFODIR GSO record, update the RESDIR GSO record and refresh the RESDIR information in storage for the selected rule type. This will add rules of type FAC to the list from which in-storage directories are built at CA-ACF2 initialization.

- a. The following is a sample set of TSO commands to add rule type FAC to the RESDIR GSO record:

CHANGE RESDIR ADD TYPES(R- FAC)

- b. Use the following operator command to refresh the RESDIR data:

F ACF2, REFRESH(RESDIR)

- 8. Define a BOOLEBBV rule of type SAF or the type selected in [Step 7 on page 43](#) and permit access to the rule for each LOGONID to be used with an AutoLogon session. This is the step that allows LOGONIDs to skip the requirement to enter a clear text password manually.

- a. The following is a sample set of TSO commands to add a BOOLEBBV rule to the rule type FAC:

```
ACF
SET RESOURCE(FAC)
COMPILE * LIST STORE
SKEY(BOOLEBBV) TYPE(FAC)
  UID(UID string) ALLOW
  UID(-) PREVENT
END
END
```

The following is an example of how these commands might be defined for specific LOGONIDs:

```
ACF
SET RESOURCE(FAC)
COMPILE * LIST STORE
SKEY(BOOLEBBV) TYPE(FAC)
SUSERDATA( RESOURCE RULE TO ALLOW BBV AUTOLOGON )
  UID(152911131) ALLOW
  UID(155311115) ALLOW
  UID(155311117) ALLOW
  UID(155311119) ALLOW
  UID(333333333) ALLOW
  UID(-) PREVENT
END
END
```

- b. If the rule type is defined as resident, use the following operator command to rebuild the in-storage directory for the rule type FAC:

F ACF2, REBUILD(FAC)

Customizing Class TERMINAL Access Authorization for CA-ACF2

ACF2 support of the TERMID= parameter in the RACROUTE VERIFY parameter list is handled by standard CA-ACF2 source and source group restriction.

To authorize the LOGONID for system entry using MainView Alternate Access at a specific terminal, that terminal's VTAM LU name or EXCP address must appear in a source group whose name is specified by the SOURCE attribute for the LOGONID.

Customizing Class APPL Access Authorization for CA-ACF2

For CA-ACF2 support of MainView Alternate Access application authorization using the APPL= parameter in the RACROUTE VERIFY parameter list and VTAM major node name access authorization, review the following steps and perform any actions necessary for your installation:

1. For CA-ACF2 6.0 and later, you must define an additional SAFDEF record. The following TSO commands may be used to define the new record:

```
ACF
SET CONTROL(GSO)
INSERT SAFDEF.BBVTLOAD ID(BBVTAPPL) MODE(GLOBAL) REP +
      RB(BBVTLOAD) PROGRAM(BBVTLOAD) +
      RACROUTE(REQUEST=AUTH, CLASS=APPL)
END
```

Then, issue the following operator command to refresh the SAF definitions in storage:

F ACF2, REFRESH(SAFDEF)

2. Determine the generalized resource rule type that SAF resource class APPL has been mapped to
 - For CA-ACF2 5.2, this is defined by the SAFMAPS GSO records
 - For CA-ACF2 6.0 and later, this is defined by a CLASMAP GSO record

By default, SAF resource class APPL is mapped to resource type SAF. You may map class APPL to any generalized resource rule type. If you use a rule type other than SAF, you must add an entry to the SAFMAPS GSO record for CA-ACF2 5.2 or define an additional CLASMAP GSO record for CA-ACF2 6.0 and later.

Note: In the subsequent examples, the assumption is made that the SAF resource class APPL has already been mapped to the generalized resource rule type APL.

3. For CA-ACF2 5.2, if desired, update the SAFMAPS GSO record to map resource class APPL to (for example) rule type APL.

The following TSO commands may be used to update the existing SAFMAPS record:

```
ACF
SET CONTROL(GSO)
CHANGE SAFMAPS ADD MAPS(APL/APPL)
END
```

Then, issue the following operator command to refresh the SAFMAPS table in storage:

F ACF2, REFRESH(SAFMAPS)

4. For CA-ACF2 6.x, if desired, add a new CLASMAP GSO record that maps resource class APPL to (for example) rule type APL. The following TSO commands may be used to define the new record:

```
ACF
SET CONTROL(GSO)
INSERT CLASMAP.APPL RSRCTYPE(APL) RESOURCE(APPL) ENTITYLN(8)
END
```

Then, issue the following operator command to refresh the external CLASMAP table in storage:

F ACF2, REFRESH(CLASMAP)

5. Define a rule for resource BBVLOGON in type APL (for example) and specify READ(ALLOW) for each LOGONID that needs to log on to MainView Alternate Access. For example, you may use the following sample TSO commands:

```
ACF
SET RESOURCE(APL)
COMPILE * LIST STORE
SKEY(BBVLOGON) TYPE(APL)
  UID(-) PREVENT
  UID(bbv-user) ALLOW
END
```

6. For EXCP sessions only, define a rule for resource BBVEXCP in type APL (for example) and specify READ(ALLOW) for each LOGONID that needs to log on to MainView Alternate Access. For example, you may use the following sample TSO commands:

```
ACF
SET RESOURCE(APL)
COMPILE * LIST STORE
SKEY(BBVEXCP) TYPE(APL)
  UID(-) PREVENT
  UID(bbv-user) ALLOW
END
```

7. For each VTAM major node name to be used with MainView Alternate Access, define a rule in type APL (for example) whose resource name matches the VTAM major node name and specify READ(ALLOW) for each LOGONID that needs to log on to MainView Alternate Access using a VTAM terminal. For example, you may use the following sample TSO commands:

```

ACF
SET RESOURCE(APL)
COMPILE * LIST STORE
SKEY(1 u- name) TYPE(APL)
  UID(-) PREVENT
  UID(bbv- user) ALLOW
END

```

8. Rebuild the resident CA-ACF2 resource directory for resource type APL (for example), if necessary. Use the following sample operator command:

```
F ACF2, REBUILD(APL)
```

Configuring RACF

For RACF support of the FACILITY(BOOLEBBV) resource, the following customization is required. Contact your RACF system programmer or security administrator for help.

1. Define a generalized resource profile in class FACILITY with an entity name of BOOLEBBV. The universal access authority of this entity should be NONE.

The following is a sample RACF (TSO) command to add a BOOLEBBV entity to class FACILITY:

```
RDEFINE FACILITY (BOOLEBBV) UACC(NONE)
```

Important

If you have previously defined a generic profile that covers entity name BOOLEBBV but does not specify NONE as the universal access, you must perform this step. Unless a universal access of NONE is specified, any user ID can sign on through MainView Alternate Access without specifying a password.

2. Permit each USERID to be used with an AutoLogon session to have READ access to resource FACILITY(BOOLEBBV) .

The following is a sample RACF (TSO) command to permit each USERID:

```
PERMIT BOOLEBBV CLASS(FACILITY) ID(userID) ACCESS(READ)
```

3. List the BOOLEBBV profile to ensure that the access list is correct and complete.

Use the following sample RACF (TSO) command to list the access list:

```
RLIST FACILITY BOOLEBBV ALL
```

4. If necessary, activate the FACILITY resource class.

Use the following sample RACF (TSO) command to activate FACILITY:

```
SETROPTS CLASSACT(FACILITY)
```

5. If necessary, refresh any in-storage generic or RACLISTed profiles associated with the FACILITY resource class.

- Use the following sample RACF (TSO) command to refresh in-storage generic profiles:

```
SETR_OPTS REFRESH GENERIC(FACILITY)
```

- Use the following sample RACF (TSO) command to refresh in-storage RACLISTed profiles:

```
SETR_OPTS REFRESH RACLIST(FACILITY)
```

Customizing Class TERMINAL Access Authorization for RACF

For RACF support of terminal access authorization through the `TERMINID=` parameter in the `RACROUTE VERIFY` parameter list, review the following steps and perform any actions necessary for your installation:

1. Define the terminal ID (VTAM LU name or EXCP terminal address) as a protected resource to RACF in class `TERMINAL` by using the following sample command:

```
RDEFINE TERMINAL (terminal-id) UACC(NONE)
```

2. Permit each `USERID` that can log on from this terminal (LUname or address) using the following sample command:

```
PERMIT terminal-id CLASS(TERMINAL) ID(userid) ACCESS(READ)
```

3. List the profile to ensure that the access list is correct and complete using the following sample command:

```
RLIST TERMINAL terminal-id ALL
```

4. If necessary, activate the `TERMINAL` resource class by using the following sample command:

```
SETR_OPTS CLASSACT(TERMINAL)
```

5. If necessary, refresh the in-storage RACLISTed profiles for the `TERMINAL` class by using the following sample command:

```
SETR_OPTS REFRESH RACLIST(TERMINAL)
```

Customizing Class APPL Access Authorization for RACF

For RACF support of MainView Alternate Access application access authorization through the `APPL=` parameter in the `RACROUTE VERIFY` parameter list and/or through the VTAM major node name, review the following steps and perform any actions necessary for your installation:

1. Define the profile `BBVLOGON` in class `APPL` to RACF using the following sample command:

```
RDEFINE APPL (BBVLOGON) UACC(NONE)
```

2. If EXCP terminal sessions are to be used with MainView Alternate Access, also define the profile `BBVEXCP` in class `APPL` to RACF using the following sample command:

```
RDEFINE APPL (BBVEXCP) UACC(NONE)
```

3. Define a profile to RACF in class APPL for each VTAM major node name to be used with MainView Alternate Access using the following sample command:

```
RDEFINE APPL (major-node-name) UACC(NONE)
```

4. Permit each USERID that needs to log on to MainView Alternate Access to have READ access to the BBVLOGON resource in class APPL.

For example, you may use the following sample command:

```
PERMIT BBVLOGON CLASS(APPL) ID(userid) ACCESS(READ)
```

5. Permit each USERID that needs to log on to MainView Alternate Access using an EXCP session to have READ access to the BBVEXCP resource in class APPL.

For example, you may use the following sample command:

```
PERMIT BBVEXCP CLASS(APPL) ID(userid) ACCESS(READ)
```

6. Permit each USERID that needs to log on to MainView Alternate Access using a VTAM terminal to have READ access to the profile in class APPL whose name matches the major node name to be used by MainView Alternate Access.

For example, you may use the following sample command:

```
PERMIT major-node-name CLASS(APPL) ID(userid) ACCESS(READ)
```

7. List the profiles to ensure that the access list is correct and complete using the following sample commands:

```
RLIST APPL BBVLOGON ALL
RLIST APPL BBVEXCP ALL
RLIST APPL major-node-name ALL
```

8. If necessary, activate the APPL resource class by using the following sample command:

```
SETRPTS CLASSACT(APPL)
```

9. If necessary, refresh the in-storage RACLISTed profiles for the APPL class by using the following sample command:

```
SETRPTS REFRESH RACLIST(APPL)
```

Configuring CA-TOP SECRET

For CA-TOP SECRET (TSS) support of the IBMFAC(BOOLEBBV) resource, the following customization is required. Contact your CA-TOP SECRET system programmer or security administrator for help.

1. Create a separate TSS facility for MainView Alternate Access.

Use the following sample TSS (TSO) commands and TSSPARM updates to create a separate CA-TOP SECRET facility:

- a. Create a separate Facility Matrix entry for the TAS by renaming an existing entry in the TSS Facility Matrix. You can model this entry after the TSO Facility Matrix entry, except that NOTSOC must be specified. The LAS can initialize properly with the

standard STC Facility Matrix entry defaults, so a separate facility usually is not needed for the LAS.

Use the following sample TSSPARM statements to rename an existing entry:

```
FAC(USER8=NAME=BBV)
FAC(BBVTAS=PGM=BBV, ACTIVE, NOTSOC)
FAC(BBVTAS=SHRPRF, NOASUBM, ABEND, SUAS, NOXDEF)
FAC(BBVTAS=LUMSG, STMSG, SIGN(s), INSTDATA)
FAC(BBVTAS=NORNDPW, AUTHINIT, NOPROMPT, NOAUDIT)
FAC(BBVTAS=RES, WARNPW, LCFCMD, NOTRACE)
FAC(BBVTAS=DORMPW, NONPW, NOIMSXTND, MSGLC)
FAC(BBVTAS=MODE=FAIL)
```

- b. After an IPL or TSS restart with the new MainView Alternate Access Facility Matrix entry in place, define interconnections between this Facility Matrix entry, the BBVTAS started task, and its default STC ACID.

The following are sample TSS (TSO) commands to define these interconnections:

```
TSS CREATE(BBVTASA) NAME(' bbvtas stc acid' ) +
  FAC(STC) PAS(NOPW, 0) DEPT(department-acid) +
  MASTFAC(BBV)
```

```
TSS ADD(STC) PROC(BBVTAS) ACID(BBVTASA)
```

- c. If you do not have standard STC Facility Matrix entry defaults, as discussed in [Step a on page 49](#), you must also define the LAS ACID with MASTFAC specifying the new MainView Alternate Access entry.

The following are sample TSS (TSO) commands to define these interconnections:

```
TSS CREATE(BBVLASA) NAME(' bbvlas stc acid' ) +
  FAC(STC) PAS(NOPW, 0) DEPT(department-acid) +
  MASTFAC(BBV)
```

```
TSS ADD(STC) PROC(BBVLAS) ACID(BBVLASA)
```

2. Add the BOOLEBBV resource to the class IBMFAC.

The following is a sample TSS (TSO) command to add a BOOLEBBV resource to class IBMFAC:

```
TSS ADDTO(owner-acid) IBMFAC(BOOLEBBV)
```

3. Permit each ACID to be used with an AutoLogon session to have READ access to resource IBMFAC(BOOLEBBV).

Use the following sample TSS (TSO) command to permit ACID with READ access:

```
TSS PERMIT(ACID) IBMFAC(BOOLEBBV) ACCESS(READ)
```

Customizing Class TERMINAL Access Authorization for CA-TOP SECRET

For CA-TOP SECRET support of terminal access authorization using the `TERMID=` parameter in the `RACROUTE VERIFY` parameter list, review the following steps and perform any actions necessary for your installation:

1. Define the terminal ID (VTAM LU name or EXCP terminal address) as a protected resource to CA-TOP SECRET in class `TERMINAL` by using the following sample command:

```
TSS ADDTO(owner- aci d) TERMINAL(termin al - i d)
```

2. Permit each ACID that can log on from this terminal (LU name or address) using the following sample command:

```
TSS PERMIT(aci d) TERMINAL(termin al - i d) ACCESS(READ)
```

Customizing Class APPL Access Authorization for CA-TOP SECRET

For CA-TOP SECRET support of MainView Alternate Access application authorization using the `APPL=` parameter in the `RACROUTE VERIFY` parameter list and VTAM major node name access authorization, review the following steps and perform any actions necessary for your installation:

1. Define the resource `BBVLOGON` in class `APPL` to CA-TOP SECRET using the following sample command:

```
TSS ADDTO(owner- aci d) APPL(BBVLOGON)
```

2. If EXCP terminal sessions are to be used with MainView Alternate Access, also define the resource `BBVEXCP` in class `APPL` to CA-TOP SECRET using the following sample command:

```
TSS ADDTO(owner- aci d) APPL(BBVEXCP)
```

3. Define a resource to CA-TOP SECRET in class `APPL` for each VTAM major node name to be used with MainView Alternate Access using the following sample command:

```
TSS ADDTO(owner- aci d) APPL(maj or- node- name)
```

4. Permit each ACID that needs to log on to MainView Alternate Access to have `READ` access to the `BBVLOGON` resource in class `APPL`.

For example, you may use the following sample command:

```
TSS PERMIT(aci d) APPL(BBVLOGON) ACCESS(READ)
```

5. Permit each ACID that needs to log on to MainView Alternate Access using an EXCP session to have `READ` access to the `BBVEXCP` resource in class `APPL`.

For example, you may use the following sample command:

```
TSS PERMIT(aci d) APPL(BBVEXCP) ACCESS(READ)
```

6. Permit each ACID that needs to log on to MainView Alternate Access using a VTAM terminal to have READ access to the resource in class APPL whose name matches the major node name to be used by MainView Alternate Access.

For example, you may use the following sample command:

```
TSS PERMIT(acid) APPL(major-node-name) ACCESS(READ)
```

Chapter 4. Logging On and Off from a VTAM Terminal

A VTAM terminal is a terminal that is connected to your VTAM network. MainView Alternate Access can establish a VTAM session with your Boole & Babbage product through a VTAM terminal when you issue either the LOGON command or the abbreviated logon command directly from the terminal. However, before BBV can support any logons from VTAM terminals, the VTAM application major node for BBV must be activated and the LAS must be started. The procedure for activating BBV and logging on through a VTAM terminal is shown in [Figure 15](#).

This chapter discusses how to

- Activate the VTAM application major node (see page [54](#))
- Start the LAS, control the LAS parameters, and stop the LAS (see page [54](#))
- Use the LOGON command (see page [57](#))
- Use the abbreviated logon command (see page [60](#))
- Use the logon panels (see page [61](#))
- Log off a VTAM session (see page [64](#))

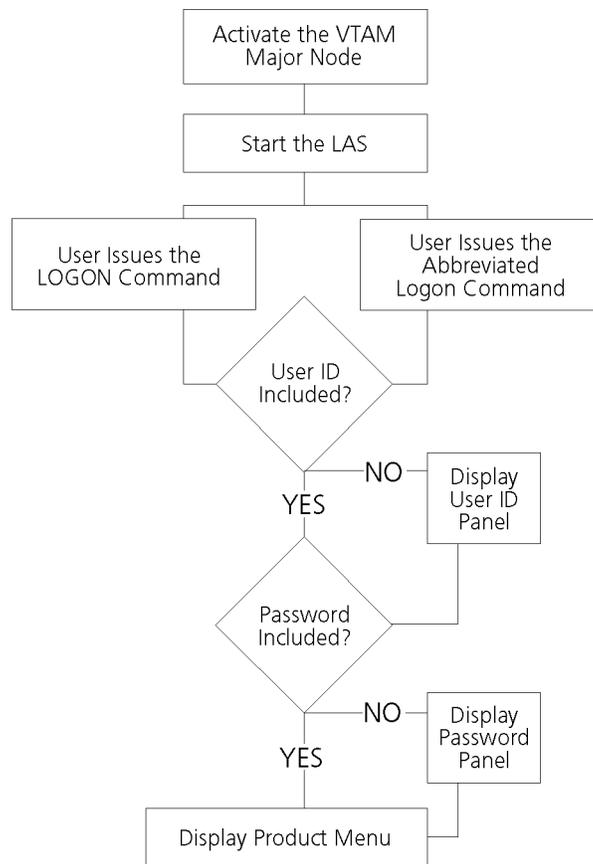


Figure 15. BBV Startup and Logon Procedure from VTAM Terminals

Activating the VTAM Application Major Node

To initialize, the LAS requires that VTAM be available and the VTAM application major node be active.

Boole & Babbage recommends placing the LAS in your IPL procedure; however, VTAM can take a long time to initialize at IPL and may not be available by the time the LAS initializes. If VTAM is not available, the following message is issued:

```
BBVL385I VTAM is inactive, waiting 15 seconds for retry
```

The LAS automatically waits 15 seconds and checks for VTAM again. The LAS performs this waiting function 20 times before terminating.

VTAM may be available but the major node may not be active. If the major node is not active when the LAS is started, the following message is issued:

```
BBVL485I VTAM applid applid is inactive; vary the major node active
```

The LAS automatically waits 15 seconds and attempts to connect again. The LAS performs this waiting function 20 times before terminating.

Note: You can configure the VTAM application major node name to be directly activated at VTAM startup by adding the major node name to SYS1.VTAMLST member ATTCONxx

To activate the VTAM major node for BBV:

1. Verify that VTAM is available.
2. Activate the VTAM application major node definition by issuing the following operator command:

```
V NET, ID=BBVTASA, ACT
```

This command example assumes that the default VTAM application major node name, BBVTASA, was not changed during customization.

Starting and Stopping the Logon Address Space (LAS)

The LAS must be running to support user logons from VTAM terminals. It is not used for VTAM or EXCP AutoLogon sessions. (For more information about how the LAS functions, see [“Logon Address Space \(LAS\) Functions” on page 4.](#))

This section includes

- Step-by-step instructions for starting the LAS (see [“Starting the LAS” on page 55.](#))
- Information about defining LAS initialization parameters at startup (see [“Controlling the LAS Initialization Parameters” on page 56.](#))
- Step-by-step instructions for stopping the LAS (see [“Stopping the LAS” on page 56.](#))

Starting the LAS

The LAS runs as a started task. The LAS proc must execute from a procedure library, such as SYS1.PROCLIB.

Issue the following START command to start the LAS using the default parameter list defined in the LAS PROC:

S BBVLAS

Notes:

- This command assumes that the default LAS proc name, BBVLAS, was not changed during customization.
- If a LAS named BBVLAS already is operational, this command fails. The first LAS must be stopped before a new LAS can be started; see [“Stopping the LAS” on page 56](#) for more information. If you want to run more than one LAS, see [“Starting Multiple LASs”](#).
- To define the LAS so it initializes at IPL, add this START command to an IPL procedure such as the COMMNDxx member of SYS1.PARMLIB.
- If you do not want to use the default parameters and you want to define a different parameter list for LAS initialization, see [“Controlling the LAS Initialization Parameters” on page 56](#).

Starting Multiple LASs

You might want to start one LAS for user access to the MainView Selection Menu and additional LASs for access directly to various MainView product menus. You can run more than one LAS on the same system at a time; however, two LASs cannot run concurrently using the same minor node root name. This means you must perform the following customization to support multiple LASs:

1. Define a different VTAM application major node and minor nodes, specifying a different minor node root name (see [“Step 2: \(Optional\) Defining the VTAM Major and Minor Nodes” on page 17](#)).
2. Define a LAS proc with a unique name (see [“Step 8: \(Optional\) Defining a LAS Startup Procedure” on page 27](#)).
3. Define a LAS initialization parameter list member named BBVLAS.xx (see [“Step 7: \(Optional\) Defining a Default LAS Initialization Parameter List” on page 25](#)). In this member, specify the
 - VTAM minor node root name created in Step 1 at the APPLID= parameter
 - The unique name of the proc created in Step 2 at the PROCNAME= parameter

See [“Understanding LAS Initialization Parameters” on page 35](#) for more information about these parameters.

Controlling the LAS Initialization Parameters

LAS initialization parameters are used to set the LAS environment when the LAS initializes (initialization parameters are described in [“Understanding LAS Initialization Parameters” on page 35](#)). These initialization parameters are defined in a parameter list member in *hl level . sysid*. UBBPARM. The default parameter list member is BBVLAS00. BBVLAS00 is created during customization, but you can define other parameter list members (see [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#)).

Which parameter list member is used to initialize the LAS is specified by the `SUF=xx` keyword on the BBVLAS PROC statement, where the `xx` value is a two-character suffix belonging to a BBVLAS`xx` member in UBBPARM (see [Figure 12 on page 27](#) for an example of the default LAS proc). BBVLAS00 member is the default member for the LAS because `SUF=00` is the default keyword value. When the LAS initializes, it reads the parameter list member that the `SUF=` keyword points to.

You can override the `SUF=` parameter value defined on the BBVLAS PROC statement at LAS initialization by specifying the `SUF=` parameter and the suffix of a different parameter list member in the LAS START command.

To initialize the LAS with a different parameter list member, specify the following START command:

```
S BBVLAS, SUF=xx
```

where `xx` is the two-character suffix belonging to a BBVLAS`xx` member in UBBPARM.

Note: This command assumes that the default LAS proc name, BBVLAS, was not changed during customization. For example, if you specify:

```
S BBVLAS, SUF=01
```

the LAS is started using the parameter list defined to member BBVLAS01 in UBBPARM.

Stopping the LAS

Once the LAS is operational, there should be no need to stop it unless you want to change the initialization parameters (see [“Controlling the LAS Initialization Parameters”](#) for more information). When you do stop the LAS, you do not affect any currently active VTAM terminal sessions, but new VTAM terminal users who want to log on through BBV cannot.

To stop the LAS, issue the following operator STOP command:

```
P BBVLAS
```

Using the LOGON Command

Once the VTAM application major node is activated and the LAS is initialized, the LOGON command can be used to log on to a VTAM session from a VTAM terminal. See [“Examples of Using the VTAM LOGON Command” on page 59](#) for examples of this command.

Note: You can also use the abbreviated logon command to log on to a VTAM session (see [“Using the Abbreviated Logon Command” on page 60](#) for more information). The abbreviated logon command is available only if your site has completed customization for either the VTAM USS table or the logon interpret table; check with your VTAM administrator.

To use the LOGON command, issue the following command at the VTAM terminal logon screen. The default configuration allows you to type the LOGON command in all uppercase characters or in all lowercase characters; however, your site may have changed this default during customization.

LOGON APPLID(*appl id*) [DATA(*user id*[/*password*)]]

where

APPLID(*appl id*) (required) Can be one of the following:

- The minor node root name defined in the VTAM application major node definition. The default value is BBVTAS.
- The VTAM generic name specified for the GNAME LAS initialization parameter, as described in [Table 4 on page 35](#).

DATA(*user id*/*password*) (optional) Defines the session user ID and password, where

user id

Is the user ID to be associated with your session. If you specify the DATA() parameter, you *must* define the *user id* operand. Any standard ID that TSO supports is supported by BBV; however, the ID must be defined to your external security manager to be accepted by BBV.

If you do *not* specify DATA(*user id*), the user ID panel, shown in [Figure 16 on page 62](#), is displayed. If you do define DATA(*user id*), the user ID panel is skipped.

If you are required to provide a password for the user ID, the password panel, shown in [Figure 17 on page 63](#), is displayed.

If you are *not* required to provide a password for the user ID, both the user ID and password panels are skipped and the default product menu is displayed.

/password

(optional) Specifies the password associated with the user ID. When you define the *password* value, you **must** define the *userid* value and separate *userid* with a / (slash) character.

When you define DATA(*userid/password*), both the user ID and password panels are skipped and the default product menu is displayed. However, you may not want to define the password because it is visible on the screen when specified in the LOGON command.

Note: If the LAS initializes with USS=NO (see [“Understanding LAS Initialization Parameters” on page 35](#)), be sure to define extra characters when specifying the DATA() parameter. The correct syntax is

DATA(' \$ *userid/password*')

The *password* value is optional, but a single quotation mark-dollar sign-blank space ('\$) must be specified before the *userid* value and an end single quotation mark (') must be specified at the end of the DATA() parameter value.

The following message appears when the LOGON command is issued:

MAINVIEW ALTERNATE ACCESS LOGON IN PROGRESS

Then, depending on how you issue the LOGON command, one or both of the BBV logon panels can appear (see [“Using the Logon Panels” on page 61](#)).

Examples of Using the VTAM LOGON Command

The following examples show how the LOGON command can be defined for a VTAM terminal session logon. The default values are used in these examples, but your site may have defined other values. See “Using the LOGON Command” on page 57 for more information.

Example 1: If you specify

```
LOGON APPLID(BBVTAS)
```

the user ID panel (see [Figure 16 on page 62](#)) is displayed.

Example 2: If you specify

```
LOGON APPLID(BBVTAS) DATA(JXM1)
```

the password panel (see [Figure 17 on page 63](#)) is displayed, unless JXM1 is an invalid user ID. If JXM1 is an invalid user ID, the logon terminates.

Example 3: If you specify

```
LOGON APPLID(BBVTAS) DATA(MAK1/MARCI E)
```

and MAK1 is a valid user ID, the default product menu is displayed, unless MARCI E is an invalid password. If MARCI E is an invalid password, the password panel, shown in [Figure 17 on page 63](#), is displayed so you can enter the correct password.

Example 4: If the LAS initializes with USS=NO and you specify

```
LOGON APPLID(BBVTAS) DATA(' $ SXC2' )
```

the password panel (see [Figure 17 on page 63](#)) is displayed, unless SXC2 is an invalid user ID. If SXC2 is an invalid user ID, the logon terminates.

Example 5: If the LAS initializes with USS=NO and you specify

```
LOGON APPLID(BBVTAS) DATA(' $ DRP1/DON' )
```

and DRP1 is a valid user ID, the default product menu is displayed, unless DON is an invalid password. If DON is an invalid password, the password panel, shown in [Figure 17 on page 63](#), is displayed so you can enter the correct password.

Using the Abbreviated Logon Command

An abbreviated logon command can be used to log on to a VTAM session from a VTAM terminal. See [“Examples of Using the Abbreviated Logon Command” on page 61](#) for examples of this command.

Your VTAM configuration may contain a USS table (USSTAB) entry or a logon interpret (LOGTAB) table entry that defines a shorter command for logging on than the required minimum LOGON APPLID(*applid*) command. The default abbreviated logon command is BBV.

During customization, sample LOGTAB and USSTAB entries can be created (see [Figure 7 on page 19](#) and [Figure 8 on page 20](#) for examples), but they must be added to production tables and the tables recompiled before the abbreviated command is usable. In addition, the USS= parameter must be configured properly in the LAS initialization parameter list to enable the table entry and the abbreviated command (see [“Understanding LAS Initialization Parameters” on page 35](#)). Check with your VTAM site administrator to find out if a table entry is configured or if the default value was changed during customization.

To use the abbreviated command, issue the following command at the terminal logon screen:

BBV [userid/password]

Notes:

- This command assumes that the default USS or logon interpret table entry, which specifies BBV, was not changed during customization.
- The **userid** and **password** parameters are identical to the operands of the DATA () parameter in the LOGON command; see [page 57](#) for more information.

The following message appears when the abbreviated logon command is issued:

MAINVIEW ALTERNATE ACCESS LOGON IN PROGRESS

Then, depending on how you issue the abbreviated logon command, one or both of the BBV logon panels will appear (see [“Using the Logon Panels” on page 61](#)).

Examples of Using the Abbreviated Logon Command

The following examples show how the abbreviated logon command can be defined for a VTAM terminal session logon. The default values are used in these examples, but your site may have defined other values. See [“Using the Abbreviated Logon Command” on page 60](#) for more information about the abbreviated command.

Example 1: If you specify

BBV

the user ID panel (see [Figure 16 on page 62](#)) is displayed.

Example 2: If you specify

BBV SXC2

the password panel (see [Figure 17 on page 63](#)) is displayed, unless SXC2 is an invalid user ID. If SXC2 is an invalid user ID, the logon terminates.

Example 3: If you specify

BBV DRP1/DON

and DRP1 is a valid user ID, the default product menu is displayed, unless DON is an invalid password. If DON is an invalid password, the password panel is displayed so you can enter the correct password.

Using the Logon Panels

There are two logon panels:

- User ID panel
- Password panel

Depending on how you specify the LOGON command or the abbreviated logon command, one or both or none of these panels are displayed at session log on.

This section provides an example of each logon panel and explains how to navigate and respond to each panel during a session logon.

Using the User ID Panel

The user ID panel, shown in [Figure 16](#), is displayed when the LOGON command or the abbreviated logon command is issued at the terminal logon screen without a user ID value specified.

```
BBVL748E ----- MainView Alternate Access ----- Boole & Babbage
Enter userid ==> jxm1
```

Figure 16. User ID Panel

There is only one field on this panel, the **Enter userid ==>** field. Your cursor is located at this field when the panel is displayed. Type your user ID and press Enter.

Note: Your user ID must be defined to the external security manager at your site; otherwise, BBV rejects your logon request.

The following message appears and then the password panel (see [Figure 17 on page 63](#)) is displayed:

```
BBVL701I Logon in progress for userid at hh:mm:ss on dd:mm:yyyy
```

If you press End at the user ID panel rather than type your user ID, the following message is displayed and then the terminal logon screen is displayed:

```
BBVL705I Logon terminated
```

Using the Password Panel

The password panel, shown in [Figure 17](#), is displayed when

- A user ID has been entered at the user ID panel
- The LOGON command is issued with a user ID but no password, and the user ID requires a password
- The LOGON command is issued with a user ID and a password, but the external security manager determined that the specified password is invalid

```
*****
***                                     ***
***           MainView Alternate Access           ***
***           (C) COPYRIGHT 1998, Boole & Babbage, Inc.           ***
***                                     ***
*****

BBVT105I Please enter current password

      Userid      JXM1
      Password ==>           New Password ==>

                               Press PF1 for HELP
```

Figure 17. Password Panel

There are four fields on this panel. Use the Tab key to move from field to field. You can access online help by pressing PF1.

Once you have completed the password panel fields, press Enter. No message is displayed, but the default product menu is displayed on the terminal screen and your session is established.

Note: BBV does not support the ATTN key; use PA1 to invoke the attention function.

If you press End at the password panel rather than continuing with your logon, the following message appears and the logon screen is displayed:

```
BBVT194W TAS logon session cancelled by user
```

The fields on the password panel are:

- | | |
|---------------------|--|
| Userid | Is a display field only. You cannot change the value in this field. The value shows the user ID for your session that was entered at the user ID panel or was defined to the LOGON or abbreviated logon command. |
| Password | Is a required input field. Type the password associated with the user ID. The password value is not displayed when you type it and can be one to eight alphanumeric characters. |
| New Password | Is an optional input field. You can change the password associated with the user ID or your external security manager may require you to change your password; otherwise, you can skip this field. |

To change your password:

1. Type your new password value. The password value is not displayed when you type it and can be one to eight alphanumeric characters.
2. Press Enter. A message is displayed prompting you to reenter your new password for verification.
3. Retype your new password value.
4. Press Enter again.

Your new password is now active.

Logging Off

To log off your session, simply press the End key until you have exited your terminal session.

Chapter 5. Configuring, Starting, and Terminating VTAM AutoLogon Sessions

AutoLogon sessions are for terminals that are configured to be directly logged on to a Boole & Babbage product at TAS initialization, without requiring user intervention at the terminal. To achieve this, AutoLogon sessions require the OPT() subparameter to be passed to ISPF at session initialization.

For the OPT() subparameter to be passed, other session parameters must be defined. Session parameters can be defined

- Directly to a START command
- To a BBVTAS_{xx} parameter list member in UBBPARM
- At both locations

See [Appendix B, “Cross-Reference of Parameters” on page 93](#) for a list of all TAS parameters and the locations where each may be defined.

Before BBV can support any VTAM AutoLogon terminal session, VTAM must be available and the VTAM application major node must be activated (see [“Activating the VTAM Application Major Node” on page 54](#)).

This chapter discusses

- Three methods for issuing START commands for VTAM AutoLogon sessions
- How to control the TAS initialization parameters for an AutoLogon session (see [page 66](#))
- How to define the START command for a VTAM AutoLogon session (see [page 67](#))
- How to define BBVTAS_{xx} members for VTAM AutoLogon sessions (see [page 73](#))
- How to terminate a TAS (see [page 75](#))

Understanding the Three Methods of VTAM AutoLogon Session Initialization

You can issue a START command at the operator’s console or customize START commands to your system configuration.

There are three ways to issue a START command to initialize a VTAM AutoLogon session.

Operator-entered: The operator-entered START command string can be up to 121 character spaces in length. If you cannot fit all the required parameters for your VTAM session onto the operator command line, you can

1. Define a BBVTAS_{xx} member (see [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#)) to contain the LOGON command and parameters.
2. Point to this member in the START command using the SUF= keyword (see [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for more information).

Automation or Scheduling Product: A START command configured to an automation or scheduling product typically can be up to 256 character spaces, depending on the product. See your automation or scheduling product documentation for more information.

IPL Procedure: Sixty-five bytes are provided for a START command in an IPL procedure, such as the COMMND_{xx} member of SYS1. PARMLIB. You must use the SUF= keyword that points to a BBVTAS_{xx} member in a START command configured to your system's IPL procedure. (See [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for more information about using the START command and the SUF= keyword; see [“Defining BBVTAS_{xx} Members for VTAM AutoLogon Sessions” on page 73](#) for more information about defining BBVTAS_{xx} members.)

If you want to include VTAM AutoLogon sessions in your IPL procedure, consider that VTAM can take a long time to initialize at IPL and may not be fully operational by the time a TAS attempts to initialize. If VTAM is not available, the TAS does not initialize.

Likewise, a session cannot become active unless the VTAM major node is active. You can configure the VTAM application major node name to be activated automatically at VTAM startup by adding the major node name to SYS1. VTAMLST member ATTCO_{xx}.

Controlling TAS Initialization Parameters

There are three locations where TAS parameters can be defined for AutoLogon sessions:

- Parameter list member in *hi level . sysid*. UBBPARM (see [“Understanding TAS Initialization Parameters” on page 31](#) for more information)
- EXEC statement parameters defined to the proc JCL (see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information)
- Keyword parameters specified in the START command

At TAS initialization, BBV reads parameters from all three locations in a specific sequence: first, the parameter list member; second, the proc JCL; and third, the START command. As each location is read in sequence, all parameters are merged. Duplicate values are overwritten as the merging sequence progresses. Default values are substituted for required parameters that are not defined or are invalid.

Note: If invalid parameter values are encountered during the merging sequence, initialization continues using default values and messages BBVT233E, BBVT234E, and BBVT235E are issued. These three messages specify which values or parameters are invalid.

Duplicate values can occur only when the START command contains one or both of its keyword parameters. The two keyword parameters supported in the TAS START command are SUF= and P= (see [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for more information). These keywords provide two different methods of starting AutoLogon TASs.

Both keywords are defined to the PROC statement in the TAS proc (default of BBVTAS) and symbolic equivalents to the PARM= parameter of the BBVTAS EXEC statement (see [Figure 10 on page 24](#) for an example of the TAS proc). These keywords can be used separately or together with the START command, as described in [Table 5 on page 67](#).

Defining the START Command for VTAM AutoLogon Sessions

Note: BBV does not support the Attn key; use PA1 to invoke the attention function.

The TAS proc must execute from a procedure library, such as SYS1.PROCLIB.

You can start a TAS for a VTAM AutoLogon session by issuing the START command. When the START command is issued, the following messages appear on the VTAM terminal screen:

```
MAINVIEW ALTERNATE ACCESS LOGON IN PROGRESS
```

```
MainView Alternate Access AUTOLOGON in progress
```

The syntax for the START command is shown here.

```
S BBVTAS. userid [, SUF=xx] [, P=' LOGON APPLID(applid) TERMI D(LUname)
          DATA(userid/password) OPT(x; x; x) : ACCOUNT=value: CAPS=x:
          DEVGROUP=device: SYSPREF=prefix ]
```

Each parameter and subparameter in the VTAM START command is explained in the following table.

Table 5. START Command Parameters (VTAM)

Parameter or Subparameter	Description
<i>userid</i>	<p>(required) Indicates the unique started task identifier for the TAS. Use this value to differentiate one TAS task from another that runs concurrently on your system.</p> <p>Note: When displaying active address spaces in MVS, the <i>userid</i> appears in the procedure name field.</p> <p>The <i>userid</i> value should be the user ID to be associated with your session, which is the same value defined to the DATA(<i>userid</i>) subparameter.</p>

Table 5. START Command Parameters (VTAM) (Continued)

Parameter or Subparameter	Description
SUF=	<p>This keyword defines a UBBPARAM(BBVTAS$_{xx}$) member to a START command. The BBVTAS$_{xx}$ member must contain the LOGON command and subparameters; otherwise, the VTAM session is not logged on.</p> <p>Use the SUF= keyword to point to a BBVTAS$_{xx}$ member when the LOGON command does not fit into the amount of space you have for defining the START command, or when you want to customize and save the LOGON command and subparameters for an AutoLogon session. BBVTAS$_{xx}$ members provide a means of expanding the amount of space necessary to specify the required LOGON command subparameters with the START command.</p> <p>Each BBVTAS$_{xx}$ member for AutoLogon sessions must contain a user ID; therefore, each AutoLogon session started using the SUF= keyword must have a separate BBVTAS$_{xx}$ member, or possible ISPF profile data set conflicts can result. See “Step 5: Defining a Default TAS Initialization Parameter List” on page 21 and “Defining the START Command for VTAM AutoLogon Sessions” on page 67 for information about creating BBVTAS$_{xx}$ members.</p> <p>The following is an example of how SUF= is used in the START command:</p> <pre>S BBVTAS. <i>user i d</i>, SUF=<i>xx</i></pre> <p>See “Defining BBVTAS$_{xx}$ Members for VTAM AutoLogon Sessions” on page 73 for more information about BBVTAS$_{xx}$ members; see “Controlling TAS Initialization Parameters” on page 66 for more information about using the SUF= keyword.</p>
P=	<p>(optional) Changes keywords and LOGON command subparameters specified in the BBVTAS$_{xx}$ member used to initialize the session, or defines keywords and subparameters not specified in BBVTAS$_{xx}$.</p> <p>To define more than one value to P=, separate each value with a : (colon). For example, if you define two keywords or the LOGON command and a keyword, you must use a : (colon) to separate the values.</p> <p>Single quotation marks must surround the entire value defined to the P= parameter.</p> <p>See “Controlling TAS Initialization Parameters” on page 66 for more information about using the P= parameter. The P= parameter allows AutoLogon sessions to be started by an operator, without requiring additional customization. It also provides a method of specifying different session values, such as user ID, application menu, or caps support for Katakana terminals, when starting an AutoLogon session.</p> <p>When using the P= parameter, the length of the START command and the entire LOGON command string may be too long to use with some methods of issuing the START command (see “Understanding the Three Methods of VTAM AutoLogon Session Initialization” on page 65).</p> <p>The following is an example of how P= is used in the START command:</p> <pre>S BBVTAS. <i>user i d</i>, P=' <i>parm: parm: parm. . .</i> '</pre>

Table 5. START Command Parameters (VTAM) (Continued)

Parameter or Subparameter	Description
SUF= and P=	<p>Use these keywords together in a single START command to</p> <ul style="list-style-type: none"> Temporarily modify one or more parameter values defined to a BBVTASxx member without having to edit the member Define parameters or keywords that are not specified in the BBVTASxx member without having to edit the member <p>By defining both the SUF= keyword and the P= keyword in the START command, any duplicate parameter values defined in the operator START command override the similar parameters defined in the BBVTASxx member (see Appendix B, “Cross-Reference of Parameters” on page 93 for more information).</p> <p>The following is an example of how SUF= and P= are used together in the START command:</p> <pre>S BBVTAS. user i d, SUF=xx, P=' parm: parm: parm. . . '</pre>
LOGON	<p>Required when defining the TERMID(), APPLID(), DATA(), and OPT() subparameters.</p> <p>When specified in the START command, the LOGON command and its subparameters must be defined to the P= keyword.</p> <p>When defined to a BBVTASxx member, the LOGON command and its subparameters must be pointed to in the START command by the SUF= keyword. The LOGON command itself must begin in column 1 and no comments may be coded on the same line as the LOGON command.</p>
APPLID(<i>applid</i>)	<p>Required subparameter of the LOGON command and must be defined for VTAM AutoLogon sessions, either in a START command or in the BBVTASxx member. The <i>applid</i> value is the minor node root name defined in the VTAM application major node definition (see “Example of the BBSAMP Member BBVTASA” on page 18 for an example of this definition).</p>
TERMID(<i>LUname</i>)	<p>Required subparameter of the LOGON command and must be defined for VTAM AutoLogon sessions, either in a START command or in a BBVTASxx member. The TERMID() subparameter indicates to the TAS which terminal session (known as an LU to VTAM) should be initialized and logged on.</p> <p>The <i>LUname</i> value is the VTAM address of the terminal device.</p>

Table 5. START Command Parameters (VTAM) (Continued)

Parameter or Subparameter	Description
<p>DATA(<i>user id</i>/<i>password</i>)</p>	<p>Required subparameter of the LOGON command and must be defined for AutoLogon sessions, either in a START command or in a BBVTASxx member. The DATA() subparameter has two operands:</p> <p><i>user id</i> This required operand is the user ID to be associated with your session. The <i>user id</i> must be defined or the session will not initialize. Any standard ID that TSO supports is supported by BBV; however, the ID must be defined to your external security manager to be accepted by BBV.</p> <p><i>/password</i> The required <i>password</i> operand is the password associated with the <i>user id</i> operand. When defined, <i>password</i> must be separated from <i>user id</i> by a / (slash) character.</p> <p>Note: Your <i>user id</i> may not require a <i>/password</i>, depending on how security is defined for BBV and <i>user id</i>. If a <i>/password</i> is not required for <i>user id</i>, then <i>/password</i> is not a required operand of the DATA() subparameter.</p> <p>If you define <i>/password</i>, you should be aware of the security issues concerning this operand; see “Defining Security for MainView Alternate Access” on page 39 for more information.</p> <p>If you do not provide <i>/password</i> and the <i>user id</i> specified requires that one be entered, the password panel shown in Figure 17 on page 63 is displayed when the TAS initializes and your AutoLogon session is interrupted and does not pass the OPT() subparameter.</p> <p>If you do provide <i>/password</i>, the password panel is skipped and you are logged on directly to your session.</p>

Table 5. START Command Parameters (VTAM) (Continued)

Parameter or Subparameter	Description
OPT (x; x; x)	<p>Optional subparameter of the LOGON command and can be defined for AutoLogon sessions, whether on a START command or in a BBVTASxx member. The OPT() subparameter contains a character string that is passed by BBV to the application and gets executed once the session is initialized</p> <p>The OPT() subparameter value, (x;x;x), can be any alphanumeric character string that could be entered in the command field of the MainView Selection Menu, where each x is a command or menu option. Each x value in the character string must be separated by one or more ; (semicolons). Other delimiter characters may not be valid.</p> <p>The maximum length of OPT(x;x;x) and its variable string is 80 bytes, but you may have fewer bytes available depending on where this subparameter is defined. See “Understanding the Three Methods of VTAM AutoLogon Session Initialization” on page 65 for information about limitations when defined to a START command; see “Defining BBVTASxx Members for VTAM AutoLogon Sessions” on page 73 for information about defining subparameters in a BBVTASxx member.</p> <p>Each x value must be a MainView product menu option, view, screen, or command; for example, OPT(1; 1) for AutoOPERATOR or OPT(4; 1) for IMF. Additionally, access to a MainView product results in a CONNECTING. . . message that requires that you define multiple consecutive semicolons (;) in your OPT() string at the point where this message occurs; for example, OPT(7; 1; ; SYSSUM; ASU 5) for MainView for OS/390 or OPT(7; 2; ; ARD; LOCK) for CMF MONITOR.</p> <p>Boole & Babbage recommends that you test your (x;x;x) value <i>before</i> defining it to the OPT() subparameter. You can test (x;x;x) by displaying the respective menu and entering the character string to verify that it displays the view, screen, product, or service you expect. See “Examples of the START Command for VTAM AutoLogon Sessions” on page 72 for examples of how the OPT() subparameter can be configured.</p>
ACCOUNT= <i>value</i>	<p>(optional keyword) Defines a default account value used for job accounting; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, ACCOUNT= must be defined to the P= keyword.</p>
CAPS={ YES NO }	<p>(optional keyword) Specifies Katakana terminal support; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, CAPS= must be defined to the P= keyword.</p>
DEVGROUP= <i>device</i>	<p>(optional keyword) Specifies an esoteric device name for ISPF log, list, and profile data sets; see “Understanding TAS Initialization Parameters” on page 31. for more information. When specified in the START command, DEVGROUP= must be defined to the P= keyword.</p>
SYSPREF= <i>prefix</i>	<p>(optional keyword) Specifies a prefix for the ISPF profile data sets that is different from the user ID or the prefix defined in the profile for the user ID in the external security manager; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, SYSPREF= must be defined to the P= keyword.</p>

Examples of the START Command for VTAM AutoLogon Sessions

The following examples show how the START command can be defined for a VTAM AutoLogon session. The default values are used in these examples, but your site may have defined other values; see [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for more information. These examples also assume that VTAM and the BBV major node are active, and that all VTAM terminals are varied active on the network.

Example 1: If you specify

```
S BBVTAS.MAK1,P=' LOGON APPLID(BBVTAS) TERMD(F099J206) DATA(MAK1 OPT(7; 1) '
```

the initial view or screen defined to the MAK1 profile is displayed or, if there is no initial display value assigned to the MAK1 profile, the default MVMVS screen for MainView for OS/390 is displayed on the VTAM terminal with an LU name of F099J206.

The following must be valid for this AutoLogon session to initialize:

- The MAK1 user ID must have at least read access to the FACILITY(BOOLEBBV), so that MAK1 does not require a clear text password (see [“Defining Security for MainView Alternate Access” on page 39](#) for more information).
- This session initializes using the parameter list member BBVTAS00 (unless some other two-character suffix value is defined to the SUF= parameter on the PROC statement in the BBVTAS proc; see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information).

Example 2: If you specify

```
S BBVTAS.SXC2,SUF=SC,P=' LOGON OPT(7; 1; ; ; ; SCR OVERVIEW; ASU 5) '
```

the OVERVIEW screen is displayed in automatic screen update (ASU) mode set at 5 seconds, assuming user SXC2 has previously created a MainView for OS/390 screen named OVERVIEW.

The following must be valid for this AutoLogon session to initialize:

- This session initializes using the parameter list member BBVTASSC, even if some other two-character suffix value is defined to the SUF= parameter on the PROC statement in the BBVTAS proc (see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information).
- The BBVTASSC member must contain the LOGON command with the following subparameters defined:
 - APPLID(BBVTAS)
 - TERMD(LUname), where LUname is a valid VTAM terminal LU name
 - DATA(SXC2/SHARI), where SXC2 is a valid user ID defined to the external security manager (ESM) and SHARI is the valid password value for SXC2
- The OPT(7; 1; ; ; ; SCR OVERVIEW; ASU 5) subparameter is passed to ISPF even if an OPT subparameter is defined to the LOGON command in the BBVTASSC parameter list member. Any parameter defined in the START command overrides duplicate parameters defined in either the TAS proc or the parameter list member (see [Appendix B, “Cross-Reference of Parameters” on page 93](#) for more information).

- Multiple ; (semicolons) are defined consecutively in the OPT subparameter value to adjust for the amount of time a particular system may typically require to process the Connecting . . . message that is displayed when accessing the MainView window interface.

Example 3: If you specify

```
S BBVTAS. LRP1, SUF=LP
```

the product service or display defined to the OPT subparameter is displayed.

The following must be valid for this AutoLogon session to initialize:

- This session initializes using the parameter list member BBVTASLP, even if some other two-character suffix value is defined to the SUF= parameter on the PROC statement in the BBVTAS proc (see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information).
- The BBVTASLP member must contain the LOGON command with the following subparameters defined:
 - APPLID(BBVTAS)
 - TERMD(LUname), where LUname is a valid VTAM terminal LU name
 - DATA(LRP1), where LRP1 is a valid user ID defined to the external security manager (ESM) and has at least read access to the FACILITY(BOOLEBBV) so that LRP1 does not require a clear text password (see [“Defining Security for MainView Alternate Access” on page 39](#) for more information)
 - OPT(x; x; x), where x; x; x is a valid option string

Defining BBVTASxx Members for VTAM AutoLogon Sessions

A BBVTASxx member can contain a LOGON command and subparameters for a VTAM AutoLogon session. Any member is pointed to at TAS initialization by specifying the SUF= keyword with the START command.

The LOGON command contained within a BBVTASxx member is used to specify the following subparameters for an AutoLogon session:

- APPLID(BBVTAS)
- TERMD(LUname), where LUname is a valid VTAM terminal LU name
- DATA(userid/password), where
 - userid* Is a valid user ID defined to the external security manager (ESM), and can have at least read access to the FACILITY(BOOLEBBV) so that *userid* does not require a clear text password (see [“Defining Security for MainView Alternate Access” on page 39](#) for more information).
 - /password* Is a valid password for *userid*, if *userid* requires a password.
- OPT(x; x; x), where x; x; x is a valid option string

To define BBVTASxx members for VTAM AutoLogon sessions:

1. Read the information in [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#) and follow the instructions in [“Step 5: Defining a Default TAS Initialization Parameter List” on page 21](#) to create a new BBVTASxx member in UBBPARM.
 2. Use the following syntax rules to define the LOGON command and subparameters in a BBVTASxx member:
 - Each line must contain the LOGON command.
 - A single line cannot exceed 72 bytes.
 - Always begin the LOGON command in column 1.
 - Do not code comments on the same line as a LOGON command.
 - If the LOGON command and subparameter string exceeds 72 bytes, continue on additional lines by defining the LOGON command on each line that contains valid subparameters.
 - A single subparameter must be coded on one line; it cannot be started on one line and continued on another.
- Note:** See [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for a description of each keyword and LOGON command subparameter.
3. Use the SUF= keyword in the START command to initialize your AutoLogon session; see [“Controlling TAS Initialization Parameters” on page 66](#) for a description of this keyword.

See [“Examples of Defining the LOGON Command in a BBVTASxx Member”](#) for more information.

Examples of Defining the LOGON Command in a BBVTASxx Member

The following examples show how the LOGON command can be defined to a BBVTASxx member for a VTAM AutoLogon session. The default values are used in these examples, but your site may have defined other values; see [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for more information.

Example : If you specify

```
LOGON APPLID(BBVTAS) TERMI D(S00A1432) DATA(VEA1/VI RGINI A)
LOGON OPT(7; 1; ; ; ; SCR OVERVIEW; ASU 5)
```

the OVERVIEW screen is displayed in automatic screen update (ASU) mode set at 5 seconds, assuming user VEA1 has previously created a MainView for OS/390 screen named OVERVIEW.

Multiple ; (semicolons) are defined consecutively in the OPT subparameter value to adjust for the amount of time a particular system may typically require to process the Connect i ng. . . message that is displayed when accessing the MainView window interface.

To invoke this AutoLogon session, you would issue the following START command:

S BBVTAS. VEA1, SUF=xx

where **xx** is the two-character suffix assigned to the BBVTASxx member containing the LOGON command.

Terminating a VTAM AutoLogon Session

Once started, there are three ways to end an AutoLogon TAS session:

- By logging off the terminal session

To log off your session, simply press the End key until you have exited your terminal session.

- By issuing an operator STOP command

When a TAS is stopped, the user of that session loses all communications with the host.

To stop a TAS using an operator STOP command, issue

P *userid* (, A=*asid*)

where

userid Is a unique identifier for the started task. When displaying active address spaces in MVS, the *userid* value appears in the procedure name field.

A=*asid* Is an optional parameter if the *userid* value is being used in only one active session. If you have a BBV and a TSO session running under the same *userid* concurrently, you must specify the A= parameter, where *asid* is the ASID value for the TAS session as shown in the DA display of SDSF.

Note: You do not want to issue a STOP command against the TAS proc name, such as P BBVTAS, because this command is ignored.

- By issuing an operator CANCEL command

Note: Boole & Babbage recommends that you use the MVS STOP command rather than the MVS CANCEL command to stop a TAS, but you can use the CANCEL command if necessary.

To cancel a TAS using an operator CANCEL command, issue

C U=*userid* (, A=*asid*)

where

userid Is a unique identifier for the started task. When displaying active address spaces in MVS, the *userid* value appears in the procedure name field.

A=*asid* Is an optional parameter if the *userid* value is being used in only one active session. If you have a BBV and a TSO session running under the same *userid* concurrently, you must specify the A= parameter, where *asid* is the ASID value for the TAS session as shown in the DA display of SDSF.

Chapter 6. Configuring, Starting, and Terminating EXCP AutoLogon Sessions

AutoLogon sessions are for terminals that are configured to be directly logged on to a Boole & Babbage product at TAS initialization, without requiring user intervention at the terminal. To achieve this, AutoLogon sessions require the OPT() subparameter to be passed to ISPF at session initialization.

For the OPT() subparameter to be passed, other session parameters must be defined. Required session parameters can be defined:

- Directly to a START command
- To a BBVTAS_{xx} parameter list member in UBBPARAM
- Both locations See [Appendix B, “Cross-Reference of Parameters” on page 93](#) for a list of all TAS parameters and the locations where each may be defined.

EXCP terminals must be started as AutoLogon sessions because you cannot initiate an EXCP session from an EXCP terminal. Logon services are not provided for EXCP communications (see [“Understanding an EXCP Terminal Session”](#) for more information), so a TAS for any EXCP terminal session must be initialized by issuing a START command.

You can run up to 32 EXCP terminal sessions concurrently.

This chapter explains

- What an EXCP terminal session is
- How to inactivate a VTAM terminal to support an EXCP session (see page [78](#))
- What the three methods for issuing START commands for EXCP AutoLogon sessions are (see page [78](#))
- How to control the TAS initialization parameters for an AutoLogon session (see page [79](#))
- How to define the START command for an EXCP AutoLogon session (see page [79](#))
- How to define BBVTAS_{xx} members for EXCP AutoLogon sessions (see page [85](#))
- How to terminate a TAS (see page [87](#))

Understanding an EXCP Terminal Session

An EXCP terminal is a 3270 display station that is channel-attached to your host through a non-SNA control unit. This means the terminal is directly accessed through a BBV EXCP session using channel programming rather than being connected through your VTAM network.

A directly connected terminal can communicate with the host using the EXCP access method. This type of low-level communications requires the user session to manage the communication protocols and does not support any form of user services, such as logon panels. Without logon support, an EXCP session must be initialized by an operator START command. (For BBV, this is accomplished by starting a TAS for the terminal’s unit address.)

On the other hand, when you use a terminal connected through your VTAM network, VTAM manages the communication protocols, logon functions, and other services for your user session. For this reason, non-SNA terminals are typically connected to the VTAM network so users can log on or perform other functions without requiring operator assistance. Usually, you can tell whether your terminal is channel-attached through a non-SNA local control unit by the presence of an A character in the second column of the status line on the terminal.

Inactivating a VTAM Terminal for EXCP Communications

Any 3270 display station that is channel-attached to your host through a non-SNA control unit must be varied off the VTAM network before it can be logged on to an EXCP session. Because non-SNA terminals are typically connected to VTAM networks, you may need to inactivate a terminal before it can be started as an EXCP AutoLogon session.

To inactivate a VTAM terminal, issue this operator command:

V NET, ID=*nodename*, INACT

where ***nodename*** is the one- to eight-character VTAM minor node name.

Understanding the Three Methods of EXCP AutoLogon Session Initialization

You can issue a START command at the operator's console or customize START commands to your system configuration.

There are three ways to issue a START command to initialize an EXCP AutoLogon session.

Operator-entered: The operator-entered START command string can be up to 121 character spaces in length. If you cannot fit all the required parameters for your EXCP session onto the operator command line, you can

1. Define a BBVTAS`xx` member (see [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#)) to contain the LOGON command and parameters.
2. Point to this member in the START command using the SUF= keyword (see [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#) for more information).

Automation or Scheduling Product: A START command configured to an automation or scheduling product typically can be up to 256 character spaces, depending on the product. You may want to configure one or more EXCP terminals to be initialized automatically whenever VTAM or JES fails, or when these subsystems are brought down for maintenance. See your automation or scheduling product documentation for more information.

IPL Procedure: Sixty-five bytes are provided for a START command in an IPL procedure, such as the COMMND`xx` member of SYS1. PARMLIB. You must use the SUF= keyword that points to a BBVTAS`xx` member in a START command configured to your system's IPL procedure. (See [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#) for more information about using the START command and the SUF= keyword; see [“Defining BBVTAS`xx` Members for EXCP AutoLogon Sessions” on page 85](#) for more information about defining BBVTAS`xx` members.)

Controlling the TAS Initialization Parameters

There are three locations where TAS parameters can be defined for AutoLogon sessions:

- Parameter list member in *hi level . sysid . UBBPARM* (see “[Understanding TAS Initialization Parameters](#)” on page 31 for more information)
- EXEC statement parameters defined to the proc JCL (see “[Step 6: Defining a TAS Startup Procedure](#)” on page 23 for more information)
- Keyword parameters specified in the START command

At TAS initialization, BBV reads parameters from all three locations in a specific sequence: first, the parameter list member; second, the proc JCL; and third, the START command. As each location is read in sequence, all parameters are merged. Duplicate values are overwritten as the merging sequence progresses. Default values are substituted for required parameters that are not defined or are invalid.

Note: If invalid parameter values are encountered during the merging sequence, initialization continues using default values and messages BBVT233E, BBVT234E, and BBVT235E are issued. These three messages specify which values or parameters are invalid.

Duplicate values can occur only when the START command contains one or both of its keyword parameters. The two keyword parameters supported in the TAS START command are SUF= and P= (see “[Defining the START Command for EXCP AutoLogon Sessions](#)” for more information). These keywords provide two different methods of starting AutoLogon TASs.

Both keywords are defined to the PROC statement in the TAS proc (default of BBVTAS) and symbolic equivalents to the PARM= parameter of the BBVTAS EXEC statement (see [Figure 10 on page 24](#) for an example of the TAS proc). These keywords can be used separately or together with the START command, as described in [Table 6 on page 80](#).

Defining the START Command for EXCP AutoLogon Sessions

The TAS proc must execute from a procedure library, such as SYS1.PROCLIB.

You can start a TAS for an EXCP AutoLogon session by issuing this START command:

```
S BBVTAS.userid[, SUF=xx][, P=' LOGON UCB(ucb)DATA(userid/password)
OPT(x; x; x) : ACCOUNT=value: CAPS=x: DEVGROUP=device: SYSPREF=prefix' ]
```

Note: If JES and VTAM are not active, you can still activate an EXCP terminal session by adding, SUB=MSTR to the end of the START command for an EXCP session.

When the START command is issued, the following messages appear on the EXCP terminal screen:

```
MAINVIEW ALTERNATE ACCESS LOGON IN PROGRESS
```

```
MaInVi ew Al ternate Access AUTOLOGON in progress
```

Note: BBV does not support the ATTN key; use PA1 to invoke the attention function.

Each parameter and subparameter in the EXCP START command is explained in the following table.

Table 6. START Command Parameters (EXCP)

Parameter or Subparameter	Description
<i>userid</i>	<p>This required parameter is the unique started task identifier for the TAS. You use this value to differentiate one TAS task from another that run concurrently on your system.</p> <p>Note: When displaying active address spaces in MVS, the <i>userid</i> appears in the procedure name field. The <i>userid</i> value should be the user ID to be associated with your session, which is the same value defined to the DATA(<i>userid</i>) subparameter.</p>
SUF= <i>xx</i>	<p>This optional parameter defines a BBVTAS<i>xx</i> member to a START command, where <i>xx</i> is the member name suffix. The BBVTAS<i>xx</i> member must contain the LOGON command and subparameters; otherwise, the EXCP session is not logged on.</p> <p>Use the SUF= to point to a BBVTAS<i>xx</i> member when the length of LOGON command exceeds the space you have for defining the START command, or when you want to customize and save the LOGON command and subparameters for an AutoLogon session. BBVTAS<i>xx</i> members provide a means of expanding the amount of space necessary to specify the required LOGON command subparameters with the START command.</p> <p>Each BBVTAS<i>xx</i> member for an AutoLogon session must contain a user ID; therefore, each AutoLogon session started using the SUF= keyword must have a separate BBVTAS<i>xx</i> member, or possible ISPF profile data set conflicts can result. See “Step 5: Defining a Default TAS Initialization Parameter List” on page 21 and “Defining the START Command for EXCP AutoLogon Sessions” on page 79 for information about creating BBVTAS<i>xx</i> members.</p> <p>See “Defining BBVTAS<i>xx</i> Members for EXCP AutoLogon Sessions” on page 85 for more information about BBVTAS<i>xx</i> members; see “Controlling the TAS Initialization Parameters” on page 79 for more information about using the SUF= keyword.</p> <p>The following is an example of how SUF= is used in the START command:</p> <p>S BBVTAS. <i>userid</i>, SUF=<i>xx</i></p>

Table 6. START Command Parameters (EXCP) (Continued)

Parameter or Subparameter	Description
P=	<p>(optional) Changes the keywords and LOGON command subparameters specified in the BBVTASxx member used to initialize the session, or defines keywords and subparameters not specified in BBVTASxx.</p> <p>If you define more than one value to P=, you must separate each with a : (colon). For example, if you define two keywords or the LOGON command and a keyword, then you must use a : (colon) to separate the values.</p> <p>Single quotation marks must surround the entire value defined to the P= keyword.</p> <p>This parameter allows AutoLogon sessions to be started by an operator, without requiring additional customization. It also provides a method of specifying different session values, such as user ID, application menu, or caps support for Katakana terminals, when starting an AutoLogon session.</p> <p>The only restriction to using the P= keyword is that the START command and the entire LOGON command string may be too long to use P= with some methods of issuing the START command (see “Understanding the Three Methods of EXCP AutoLogon Session Initialization” on page 78).</p> <p>The following is an example of how P= is used in the START command:</p> <pre>S BBVTAS. user i d, P=' parm: parm: parm: . . . '</pre> <p>See “Controlling the TAS Initialization Parameters” on page 79 for more information about using the P= keyword.</p>
SUF= and P=	<p>You can use these two keywords together in a single START command to</p> <ul style="list-style-type: none"> Temporarily modify one or more parameter values defined to a BBVTASxx member without having to edit the member Define parameters or keywords that are not specified in the BBVTASxx member without having to edit the member <p>By defining both the SUF= keyword and the P= keyword in the START command, any duplicate parameter values defined in the operator START command override the similar parameters defined in the BBVTASxx member (see Appendix B, “Cross-Reference of Parameters” on page 93 for more information).</p> <p>The following is an example of how SUF= and P= are used together in the START command:</p> <pre>S BBVTAS. user i d, SUF=xx, P=' parm: parm: parm: . . . '</pre>

Table 6. START Command Parameters (EXCP) (Continued)

Parameter or Subparameter	Description
LOGON	<p>This command is required when defining the UCB(), DATA() and OPT() subparameters.</p> <p>When specified in the START command, the LOGON command and its subparameters must be defined to the P= keyword.</p> <p>When defined to a BBVTASxx member, the LOGON command and its subparameters must be pointed to in the START command by the SUF= keyword. The LOGON command itself must begin in column 1 and no comments may be coded on the same line as the LOGON command.</p>
UCB(<i>ucb</i>)	<p>Required subparameter of the LOGON command and must be defined for EXCP AutoLogon sessions, whether in a START command or a BBVTASxx member. The <i>ucb</i> value defines the unit address of the channel-attached terminal device. The <i>ucb</i> value is a four-character address; do not define a / (slash) as part of the address value.</p>
DATA(<i>useri d</i> / <i>password</i>)	<p>Required subparameter of the LOGON command and must be defined for EXCP AutoLogon sessions, whether in a START command or a BBVTASxx member. The DATA() subparameter has two operands:</p> <p><i>useri d</i> Is the user ID to be associated with your session. The <i>useri d</i> must be defined or your session does not initialize. Any standard ID that TSO supports is supported by BBV; however, the ID must be defined to your external security manager to be accepted by BBV.</p> <p><i>/password</i> Is the password associated with the <i>useri d</i> operand. When defined, <i>password</i> must be separated from <i>useri d</i> by a / (slash) character.</p> <p>Note: Your <i>useri d</i> may not require a <i>/password</i>, depending on how security is defined for BBV and <i>useri d</i> (see “Defining Security for MainView Alternate Access” on page 39 for more information). If a <i>/password</i> is not required for <i>useri d</i>, then <i>/password</i> is not a required operand of the DATA() subparameter.</p> <p>If you define <i>/password</i>, you should be aware of the security issues concerning this operand; see “Defining Security for MainView Alternate Access” on page 39 for more information.</p> <p>If you do not provide <i>/password</i> and the <i>useri d</i> specified requires that one be entered, the password panel shown in Figure 17 on page 63 is displayed when the TAS initializes and your AutoLogon session is interrupted and does not pass the OPT() subparameter.</p> <p>If you do provide <i>/password</i> the password panel is skipped and you are logged on directly to your session.</p>

Table 6. START Command Parameters (EXCP) (Continued)

Parameter or Subparameter	Description
OPT(<i>x; x; x</i>)	<p>Optional subparameter of the LOGON command and can be defined for EXCP AutoLogon sessions, whether in a START command or a BBVTASxx member. The OPT() subparameter contains a command string that is passed by BBV to the application and gets executed once the session is initialized.</p> <p>The OPT() subparameter value, (<i>x;x;x</i>), can be any alphanumeric character string that could be entered in the command field of the MainView Selection Menu, where each <i>x</i> is a command or menu option. Each <i>x</i> value in the character string must be separated by one or more ; (semicolons). Other delimiter characters may not be valid.</p> <p>The maximum length of OPT(<i>x;x;x</i>) and its variable string is 80 bytes, but you may have fewer bytes available depending on where this subparameter is defined. See “Understanding the Three Methods of EXCP AutoLogon Session Initialization” on page 78 for information about limitations when defined to a START command; see “Defining BBVTASxx Members for EXCP AutoLogon Sessions” on page 85 for information about defining subparameters in a BBVTASxx member.</p> <p>Each <i>x</i> value must be a MainView product menu option, view, screen, or command; for example, OPT(1; 1) for AutoOPERATOR or OPT(4; 1) for IMF. Additionally, access to a MainView product results in a CONNECTING. . . message that requires that you define multiple consecutive semicolons (;) in your OPT() string at the point where this message occurs; for example, OPT(7; 1; ; SYSSUM; ASU 5) for MainView for OS/390 or OPT(7; 2; ; ARD; LOCK) for CMF MONITOR. Boole & Babbage recommends that you test your (<i>x;x;x</i>) value <i>before</i> defining it to the OPT() subparameter. You can test (<i>x;x;x</i>) by displaying the menu and entering the character string to verify that it displays the view, screen, product, or service you expect. See “Examples of the START Command for EXCP AutoLogon Sessions” on page 84 for examples of how the OPT() subparameter can be configured.</p>
ACCOUNT= <i>value</i>	(optional keyword) Defines a default account value used for job accounting; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, ACCOUNT= must be defined to the P= keyword.
CAPS={ YES NO }	(optional keyword) Specifies Katakana terminal support; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, CAPS= must be defined to the P= keyword.
DEVGROUP= <i>device</i>	This optional keyword specifies an esoteric device name for ISPF log, list, and profile data sets; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, DEVGROUP= must be defined to the P= keyword.
SYSPREF= <i>prefix</i>	(optional keyword) Specifies a prefix for the ISPF profile data sets that is different from the user ID or the prefix defined in the profile for the user ID in the external security manager; see “Understanding TAS Initialization Parameters” on page 31 for more information. When specified in the START command, SYSPREF= must be defined to the P= keyword.

Examples of the START Command for EXCP AutoLogon Sessions

The following examples show how the START command can be specified for an EXCP AutoLogon session. The default values are used in these examples, but your site may have defined other values; see [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#) for more information about the EXCP START command. These examples also assume that the EXCP terminal is varied inactive on the VTAM network.

Example 1: If you specify

```
S BBVTAS.MAK1,P=' LOGON UCB(C10) DATA(MAK1) OPT(7;1)'
```

the initial view or screen defined to the MAK1 profile is displayed or, if there is no initial display value assigned to the MAK1 profile, the default MVMVS screen for MainView for OS/390 is displayed on the EXCP terminal with a UCB unit address of C10.

The following must be valid for this AutoLogon session to initialize:

- The MAK1 user ID must have at least read access to the FACILITY(B00LEBBV), so that MAK1 does not require a clear text password (see [“Defining Security for MainView Alternate Access” on page 39](#) for more information).
- This session initializes using the parameter list member BBVTAS00 (unless some other two-character suffix value is defined to the SUF= parameter on the PROC statement in the BBVTAS proc; see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information).

Example 2: If you specify

```
S BBVTAS.SXC2,SUF=SC,P=' LOGON OPT(7;1;;;;SCR OVERVIEW;ASU 5)'
```

the OVERVIEW screen is displayed in automatic screen update (ASU) mode set at 5 seconds, assuming user SXC2 has previously created a MainView for OS/390 screen named OVERVIEW.

The following must be valid for this AutoLogon session to initialize:

- This session initializes using the parameter list member BBVTASSC, even if some other two-character suffix value is defined to the SUF= parameter on the PROC statement in the BBVTAS proc (see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information).
- The BBVTASSC member must contain the LOGON command with the following subparameters defined:
 - UCB(*ucb*), where *ucb* is a valid EXCP terminal unit address
 - DATA(SXC2/SHARI), where SXC2 is a valid user ID defined to the external security manager (ESM) and SHARI is the valid password value for SXC2
- The OPT(7;1;;;;SCR OVERVIEW;ASU 5) subparameter is passed to ISPF even if an OPT subparameter is defined to the LOGON command in the BBVTASSC parameter list member. Any parameter defined in the START command overrides duplicate parameters defined in either the TAS proc or the parameter list member (see [Appendix B, “Cross-Reference of Parameters” on page 93](#) for more information).

- Multiple ; (semicolons) are defined consecutively in the OPT subparameter value to adjust for the amount of time a particular system may typically require to process the Connecting . . . message that is displayed when accessing the MainView window interface.

Example 3: If you specify

S BBVTAS.LRP1, SUF=LP

the product service or display defined to the OPT subparameter is displayed.

The following must be valid for this AutoLogon session to initialize:

- This session initializes using the parameter list member BBVTASLP, even if some other two-character suffix value is defined to the SUF= parameter on the PROC statement in the BBVTAS proc (see [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for more information).
- The BBVTASLP member must contain the LOGON command with the following subparameters defined:
 - UCB(*ucb*) , where *ucb* is a valid EXCP terminal unit address
 - DATA(LRP1) , where LRP1 is a valid user ID defined to the external security manager (ESM) and has at least read access to the FACILITY(BOOLEBBV) so that LRP1 does not require a clear text password (see [“Defining Security for MainView Alternate Access” on page 39](#) for more information)
 - OPT(*x; x; x*) , where *x; x; x* is a valid option string

Defining BBVTASxx Members for EXCP AutoLogon Sessions

A BBVTASxx member can contain a LOGON command and subparameters for an EXCP session. Any member is pointed to at TAS initialization by specifying the SUF= keyword with the START command.

The LOGON command contained within a BBVTASxx member

- UCB(*ucb*) , where *ucb* is a valid EXCP terminal unit name
- DATA(*user id/password*) , where
 - user id* Is a valid user ID defined to the external security manager (ESM), and can have at least read access to the FACILITY(BOOLEBBV) so that *user id* does not require a clear text password (see [“Defining Security for MainView Alternate Access” on page 39](#) for more information).
 - /password* Is a valid password for *user id*, if *user id* requires a password.
- OPT(*x; x; x*) , where *x; x; x* is a valid option string

To define BBVTASxx members for EXCP AutoLogon sessions:

1. Read the information in [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#) and follow the instructions in [“Step 5: Defining a Default TAS Initialization Parameter List” on page 21](#) to create a new BBVTASxx member in UBBPARAM.

2. Use the following syntax rules to define the LOGON command and subparameters in a BBVTASxx member:
 - Each line must contain the LOGON command.
 - A single line cannot exceed 72 bytes.
 - Always begin the LOGON command in column 1.
 - Do not code comments on the same line as a LOGON command.
 - If the LOGON command and subparameter string exceeds 72 bytes, continue on additional lines by defining the LOGON command on each line that contains valid subparameters.
 - A single subparameter must be coded on one line; it cannot be started on one line and continued on another.

Note: See [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#) for a description of each keyword and LOGON command subparameter.
3. Use the SUF= keyword in the START command to initialize your AutoLogon session; see [“Controlling the TAS Initialization Parameters” on page 79](#) for a description of this keyword. See [“Examples of Defining the LOGON Command in a BBVTASxx Member”](#) for more information.

Examples of Defining the LOGON Command in a BBVTASxx Member

The following example shows how the LOGON command can be defined to a BBVTASxx member for an EXCP AutoLogon session. The default values are used in these examples, but your site may have defined other values; see [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#) for more information.

Example : If you specify

```
LOGON UCB(J43) DATA(VEA1/VIRGINIA)
LOGON OPT(7; 1; ; ; ; SCR OVERVIEW; W2. LO; W4. LO; ASU 5; W3. DOWN)
```

the OVERVIEW screen is displayed with windows 2 and 4 locked, the remaining windows in automatic screen update (ASU) mode set at 5 seconds, and the data in window 3 paged down once, assuming user VEA1 has previously created a MainView for OS/390 screen named OVERVIEW.

Multiple ; (semicolons) are defined consecutively in the OPT subparameter value to adjust for the amount of time a particular system may typically require to process the Connecting . . . message that is displayed when accessing the MainView window interface.

To invoke this AutoLogon session, you would issue the following START command:

```
S BBVTAS. VEA1, SUF=xx
```

where **xx** is the two-character suffix assigned to the BBVTASxx member containing the LOGON command.

Terminating an EXCP AutoLogon Session

Once started, there are three ways to end an AutoLogon TAS session:

- By logging off the terminal session

To log off your session, simply press the End key until you have exited your terminal session.

- By issuing an operator STOP command

When a TAS is stopped, the user of that session loses all communications with the host.

To stop a TAS using an operator STOP command, issue

P *userid* (, A=*asid*)

where

userid Is a unique identifier for the started task. When displaying active address spaces in MVS, the *userid* value appears in the procedure name field.

A=*asid* Is an optional parameter if the *userid* value is being used in only one active session. If you have a BBV and a TSO session running under the same *userid* concurrently, you must specify the A= parameter, where *asid* is the ASID value for the TAS session as shown in the DA display of SDSF.

Note: You do not want to issue a STOP command against the TAS proc name, such as P BBVTAS, because this command is ignored.

- By issuing an operator CANCEL command

Note: Boole & Babbage recommends that you use the MVS STOP command rather than the MVS CANCEL command to stop a TAS, but you can use the CANCEL command if necessary.

To cancel a TAS using an operator CANCEL command, issue

C U=*userid* (, A=*asid*)

where

userid Is a unique identifier for the started task. When displaying active address spaces in MVS, the *userid* value appears in the procedure name field.

A=*asid* Is an optional parameter if the *userid* value is being used in only one active session. If you have a BBV and a TSO session running under the same *userid* concurrently, you must specify the A= parameter, where *asid* is the ASID value for the TAS session as shown in the DA display of SDSF.

Appendix A. Libraries and Load Modules

This appendix contains a list of libraries and sample members that are shipped with BBV and used for installation and customization. It also lists the BBV load modules.

Load Modules

The following load modules are installed for BBV. These load modules must reside in an APF-authorized library that is concatenated to the STEPLIB DD statements in both the LAS and TAS procs:

BBVCLOAD Used for common services that support both LAS and TAS functions.

BBVJLOAD Used for TSO services required for TAS functions.

BBVLLLOAD Used by the LAS for user logons to VTAM sessions.

BBVSLOAD Used for local SVC screening services that support TAS functions.

BBVTLOAD Used for TAS initializations.

Libraries

The following libraries are shipped in BBV:

- BBCLIB
- BBPARM
- BBPLIB
- BBSAMP
- BBTLIB

The BBCLIB, BBPLIB, and BBTLIB libraries contain modules specifically for customizing BBV using AutoCustomization.

The BBPARM and BBSAMP libraries contain sample members that are used for either AutoCustomization or manual customization. These members are never the target of SMP operations. To apply maintenance against BBPARM and BBSAMP members, you must first apply all PTFs and then perform the AutoCustomization or manual customization steps that create or change the affected members.

BBPARM Data Set Members

The following table lists customizable members that are shipped in BBPARM and are copied to a user parameter library (default is *hi level . sysid . UBBPARM*) for modifications during customization. The user parameter library must be concatenated by a BBVPARM DD statement in both the LAS and TAS procs.

Table 7. BBPARM Library Members

BBPARM member name	Description
BBVLAS00	Is a customizable member used to define the session environment at startup for each user that logs on to a VTAM session. This member is required only for VTAM session access, not EXCP sessions. The BBVLAS00 member is the default member. There can be other parameter list members using the naming convention of BBVLASxx.
BBVTAS00	Is a customizable member used to define the session environment at startup for each session. This member is required for both VTAM and EXCP sessions. The BBVTAS00 member is the default member. There can be other parameter list members using the naming convention of BBVTASxx. In fact, an AutoLogon session started by an automation product or during the IPL procedure requires its own parameter list member which contains a LOGON command.

BBSAMP Data Set Members

The following table lists sample members that are shipped in BBSAMP and copied for modification to a user sample library (default is *hi level . UBBSAMP*), unless otherwise specified. The user sample library must be concatenated by the SYSPROC DD statement in the TAS proc.

Table 8. BBSAMP Library Members

BBSAMP member name	Description
BBVINIT	Is a REXX EXEC that allocates the ISPF environment for each TAS session. During customization, this sample member is copied for modification to a site-specified data set (default is <i>hi level</i> . UBBSAMP) that must be concatenated by the SYSPROC DD statement in the TAS proc.
BBVINITC	Is a CLIST that creates the ISPF environment for a TAS session if your site does not run REXX. Boole & Babbage strongly recommends that you use the BBVINIT REXX EXEC if your site does run REXX. Follow the instructions at the top of this sample CLIST to customize it to your site requirements.
BBVISPF	Is a member created and used during AutoCustomization to customize BBVINIT. This member is re-created if you need to re-customize the ISPF environment AutoCustomization step because of ISPF maintenance. You do not need to maintain this member in your UBBSAMP data set.
BBVLAS	Is a sample started task procedure for initializing the LAS. The LAS is used only for user logons to VTAM sessions and not for EXCP sessions. During customization, this sample member is copied for modification to a site-specified data set (default is <i>hi level</i> . UBBSAMP), and, once modified, copied or moved to a site-wide procedure library.
BBVLIT	Is a sample logon interpret table entry that you can add to your logon interpret table and recompile the table to shorten the VTAM user logon command. The BBVLIT member is used optionally for VTAM access customization, if required and appropriate to your site. During customization, this sample member is copied for modification to a site-specified data set (default is <i>hi level</i> . UBBSAMP).
BBVLVER	Is a verification job to check that all CSECTS for all BBV load modules are present. It creates output showing the results of the verification procedures as well as the PTF maintenance level for each CSECT.
BBVTAS	Is a sample started task procedure for initializing each TAS. The TAS proc is used for all VTAM and EXCP session initializations. During customization, this sample member is copied for modification to a site-specified data set (default is <i>hi level</i> . UBBSAMP), and, once modified, copied or moved to a site-wide procedure library.
BBVTASA	<p>Is a default VTAM application major node definition, containing a minor node root name with 10 minor nodes, that identifies BBV to VTAM. The application major node definition is required only for VTAM session access.</p> <p>During customization, this sample member is copied for modification to either a site-specified data set (default is <i>hi level</i>. UBBSAMP) or a data set that is concatenated by the VTAMLST DD statement. If copied to a user library for modification, the BBVTASA member must be moved to a VTAMLST DD data set or the user library must be concatenated to VTAMLST DD.</p>
BBVUSST	Is a sample USS table entry that you can add to your USS table and recompile the table to shorten the VTAM user logon command. The BBVUSST member is used optionally for VTAM access customization, if required and appropriate to your site. During customization, this sample member is copied for modification to a site-specified data set (default is <i>hi level</i> . UBBSAMP).

Appendix B. Cross-Reference of Parameters

BBV parameters for sessions can be defined to four different places:

- Parameter list members

These are UBBPARM members with a naming convention of `BBVxASnn`, where

`x` Is either L for the LAS or T for the TAS.

`nn` Is a unique two-character alphanumeric value.

See [“Defining Other LAS and TAS Initialization Parameter Lists” on page 31](#) for information about creating and naming TAS and LAS parameter list members; see [“Step 5: Defining a Default TAS Initialization Parameter List” on page 21](#) or [“Step 7: \(Optional\) Defining a Default LAS Initialization Parameter List” on page 25](#) for information about customizing TAS and LAS parameter list members; see [“Understanding TAS Initialization Parameters” on page 31](#) or [“Understanding LAS Initialization Parameters” on page 35](#) for information about specific TAS and LAS parameters.

- Started task EXEC or PROC statements

You can specify a default set of session initialization parameters for each session by defining a parameter list member’s unique two-character value as `nn` to `SUF=` on the PROC statement of either the LAS or the TAS startup procedures or to `SUFFIX=` on the TAS EXEC statement.

The procs may reside in `SYS1.PROCLIB` or some other procedure library.

See [“Step 6: Defining a TAS Startup Procedure” on page 23](#) for TAS or [“Step 8: \(Optional\) Defining a LAS Startup Procedure” on page 27](#) for LAS started task customization information.

- START command

You can point to a different parameter list member in the START command by overriding the value specified in the PROC statement `SUF=` parameter by issuing an `S BBVxAS, SUF=nn` command.

For VTAM sessions, see [“Defining the START Command for VTAM AutoLogon Sessions” on page 67](#) for information about starting AutoLogon sessions or [“Starting and Stopping the Logon Address Space \(LAS\)” on page 54](#) for information about starting the LAS, which supports the terminal user issued LOGON or abbreviated logon commands.

For EXCP sessions, see [“Defining the START Command for EXCP AutoLogon Sessions” on page 79](#) for information about starting AutoLogon sessions.

- Terminal user issued LOGON or abbreviated logon command (VTAM access only)

VTAM session users can issue the LOGON or abbreviated logon command from a terminal connected to the VTAM network and can define a subset of the session initialization parameters. Namely, users can specify the VTAM minor node name in the `APPLID()` parameter and the user ID/password combination in the `DATA()` parameter.

See [“Using the LOGON Command” on page 57](#) for information about the LOGON command or [“Using the Abbreviated Logon Command” on page 60](#) for information about the abbreviated logon command. At TAS or LAS initialization, BBV reads parameters from all four locations in a specific sequence:

1. Parameter list member
2. Proc JCL
3. Operator-entered `START` command or the VTAM terminal logon command

As the parameters are read:

- They are merged as each location is read in sequence.
- Duplicate parameters that are defined in more than one location are overwritten as the merging sequence progresses.
- Default values are substituted for required but undefined parameters.

The cross-references of TAS and LAS parameters in the following tables are designed to help you understand where each parameter can be defined and overwritten.

Table 9. TAS and LAS Parameter Cross-Reference

TAS parameter	Parmlib member	Pg	Proc statement (PROC or EXEC)	Pg	Operator START command	Pg	VTAM terminal logon	Pg	Used with VTAM, EXCP, or BOTH
ACCOUNT=	X	32			X	67 79			BOTH
CAPS=	X	32			X	67 79			BOTH
DEVGROU P=	X	33			X	67 79			BOTH
LOGON	X	33			X	67 79	X	57 60	BOTH
LOGON APPLI D	X	69			X	67	X	57 60	VTAM
LOGON DATA	X	70 83			X	67 79	X	82	BOTH
LOGON OPT	X	71 83			X	67 79			BOTH
LOGON TERMI D	X	69			X	67			VTAM
LOGON UCB	X	82			X	79			EXCP
P=			PROC	68 80	X	67 79			BOTH
SUF=			PROC	68 80	X	67 79			BOTH
SUFFI X=			EXEC	23					BOTH
SYSPREF=	X	34			X	67 79			BOTH
useri d					X	67 80			BOTH
APPLI D=	X	35					X	57 60	VTAM
CAPS=	X	35							VTAM
PROCNAME=	X	36							VTAM
SUF=	X		PROC	56	X	55			VTAM
USS=	X	36							VTAM

Appendix C. Customer Support

BMC Software offers customers technical support for this product through the San Jose, California Customer Response Center.

Contacting Customer Support

To contact Customer Support from the *USA or Canada*, call or fax our San Jose, California office using the numbers listed below. For *all other areas*, contact your local BMC Software support office:

Telephone (within the US) 800 538 1872

Telephone (outside the US, but within North America) 408 526 3040

Fax requests 408 526 3897

Hours of Operation

Normal business hours are Monday through Friday from 5:00 am to 5:00 pm, Pacific Time. Technical Support Analysts are available during these hours.

After-Hours

Our after-hours support uses an on-call paging system. A recording requests you to choose one of three product groups: MVS, Client/Server, or Storage Division. Leave your name, phone number, company name, and the product for which you are experiencing problems. A Customer Service Representative will contact you within the hour.

Mailing Address

If requested by Customer Support to send documentation, mail it to:

Customer Support
BMC Software, Inc.
3131 Zanker Road
San Jose, CA 95134-1933
Incident # _____

For Product Information

For information on BMC Software products from within North America, call **Chris Parker** at **408 526 3484** (within California), or **800 544 2152** (outside California). For areas outside of North America, contact your local sales office.

Or visit us on the web at <http://www.bmc.com>

Glossary

This glossary defines Boole & Babbage terminology. Other companies in the computer industry, such as IBM, publish dictionaries and glossaries that may be used in conjunction with this glossary.

Since this glossary pertains to all Boole & Babbage-related products, some of the terms defined here may not appear in this book but may appear in other books documenting those products.

To help you find the information you need, this glossary uses the following cross-references:

Contrast with	Indicates a term that has an opposite or contradictory meaning.
Same as	Indicates one or more terms that have the same meaning.
See	Indicates an entry that contains additional or expanded information.
See also	Indicates an entry that contains related information.

A

AutoCustomization. Boole & Babbage online facility for customizing the installation of products. AutoCustomization provides an ISPF panel interface that presents customization steps in sequence and provides current status information about the progress of the install. *Contrast with* manual customization

AutoLogon. Function of MainView Alternate Access that allows VTAM or EXCP sessions to be started and logged directly on a Boole & Babbage product by passing of the OPT subparameter in one of three ways: as part of the IPL procedure, by an automation product, or by issuing an operator START command.

AutoOPERATOR. Boole & Babbage online product that provides tools, techniques, and facilities used to improve an operator's ability to provide high availability through error minimization, improved productivity, and problem prediction and prevention.

B

BBV. Component code for MainView Alternate Access.

BBVINIT. BBSAMP member shipped with MainView Alternate Access that contains a sample REXX EXEC that allocates the ISPPROF profile data set dynamically for each user session. BBVINIT is used during customization and is required for both VTAM and EXCP access.

BBVLAS. (1) BBSAMP member shipped with MainView Alternate Access that contains a sample startup procedure for the Logon Address Space (LAS) that is used during customization if VTAM terminal users will be required to log on from the terminal. (2) Default name of the LAS proc.

BBVLAS00. (1) BBPARM member shipped with MainView Alternate Access that contains a sample initialization

parameter list which defines the LAS environment. BBVLAS00 is used during customization and is required only if VTAM terminal users will be required to log on from the terminal. (2) Name of the default LAS initialization parameter list member that must reside in a user parameter library.

BBVLASxx. User-defined LAS initialization parameter list member which can be used when a different environment is required from that specified by the default parameters contained in the BBVLAS00 member. Any user-defined initialization parameter list member must reside in a user parameter library and is necessary only if VTAM terminal users will be required to log on from the terminal.

BBVLIT. BBSAMP member shipped with MainView Alternate Access that contains a sample logon interpret table entry that is optionally used during customization if VTAM terminal users will be required to log on from the terminal. This sample entry must be added to the site logon interpret table and the table recompiled. *See* logon interpret table

BBVLVER. BBSAMP member shipped with MainView Alternate Access that contains a sample verification program that checks the CSECTs in the load modules and indicates the PTF level of each CSECT. If a CSECT is in error, this is reported in output produced by the program.

BBVTAS. (1) BBSAMP member shipped with MainView Alternate Access that contains a sample startup procedure for the Terminal Address Space (TAS). BBVTAS is used during customization and is required for both VTAM and EXCP access. (2) Default name of the TAS proc. (3) Default minor node root name for minor nodes associated with the VTAM application major node.

BBVTASA. 1) BBSAMP member shipped with MainView Alternate Access that contains a sample VTAM application major node definition that is used during customization if VTAM access is required. 2) Default name of the VTAM application major node definition used for VTAM access. It must reside in a data set concatenated by the VTAMLST DD statement and be activated before VTAM access is enabled.

This application major node can be added to SYS1.VTAMLIST member ATTCONxx so it is activated at IPL.

BBVTAS00. (1) BBPARM member shipped with MainView Alternate Access that contains a sample initialization parameter list which defines the TAS environment. BBVTAS00 is used during customization and is required for both VTAM and EXCP access. (2) Name of the default TAS initialization parameter list member that must reside in a user parameter library.

BBVTASxx. 1) User-defined TAS initialization parameter list member residing in a user parameter library that can be used when a different environment is required from that specified by the default parameters contained in the BBVTAS00 member. 2) User-defined initialization parameter list member residing in a user parameter library that can contain a unique user ID and other logon parameters for a specific EXCP or VTAM terminal for an AutoLogon session.

BBVUSST. BBSAMP member shipped with MainView Alternate Access that contains a sample USS table entry that is optionally used during customization if VTAM terminal users will be required to log on from the terminal. This sample entry must be added to the site USS table and the table recompiled. *See* USS table

BBV(YES). AutoCustomization startup parameter that is required only by CMF MONITOR customers to display the BBV steps in AutoCustomization.

BOOLEBBV. Default resource class entity name that must be defined under the class FACILITY if added security is required for AutoLogon sessions.

C

CA-ACF2. External security manager that provides security for a system by intercepting resource requests prior to the actual access.

CA-TOP SECRET. External security manager that controls user access through established parameters defined by the system administrator to the security environment.

CMF MONITOR. Boole & Babbage's *Comprehensive Management Facility*, which is a platform for collecting and reporting system performance information. CMF MONITOR measures and reports on all critical system variables, such as CPU, channel, and device usage; memory, paging, and swapping activity; and domain and performance group characteristics.

command. Request to execute a particular program or initiate an action.

CPO. Customized Product Offering (CPO) delivery and installation technique developed by Boole & Babbage that allows any combination of Boole & Babbage SMP-maintainable products to be distributed on a product tape to a customer and installed quickly. The CPO product tape contains SMP target libraries required for product customization and execution, plus SMP distribution libraries and data sets needed for application of SMP maintenance.

CSA. Common service area.

D

DASD ADVISOR. Interactive software tool that diagnoses DASD performance problems and makes recommendations to reduce overall service time. This tool measures and reports on the operational performance of IBM and IBM-compatible devices.

E

EXCP. (1) Execute channel program. (2) Environment that provides a communications to a device attached directly to the host system.

F

FMID. Element with a logical grouping of elements that comprise the modules belonging to Boole & Babbage products. *Same as* functional module identifier component

functional module identifier component. *See* FMID

H

hilevel. Value that is the high level data set qualifier required by a site's naming conventions.

I

initialization parameter list. For the task of initialization, a list of values that provides a means of associating addressability of data defined in a called program with data in the calling program. It contains parameter names and the order in which they are to be associated.

Installation Checklist Letter. Checklist sent with all Boole & Babbage products outlining procedures for installation and customization.

L

LAS. *See* Logon Address Space

logoff. (1) Procedure by which a user ends a terminal session. (2) In ACF/VTAM programs, an unformatted session-termination request.

logmode. In VTAM programs, a subset of session parameters specified in a logon mode table for communication with a logical unit.

logon. (1) Procedure by which a user begins a terminal session. (2) In ACF/VTAM programs, an unformatted session-initiation request for a session between two logical units.

Logon Address Space. Address space that is accessed by a VTAM terminal user when the user types the LOGON command or abbreviated logon command at the terminal to access a Boole & Babbage product through MainView Alternate Access. This address space is not used for EXCP access or VTAM AutoLogon sessions.

logon interpret table. In VTAM programs, an installation-defined correlation list that translates an argument into a string of eight characters and translates logon data into the name of an application program for which the logon is intended. A BBVLIT sample table entry ships with MainView Alternate Access, which can be added to the logon interpret table and the table recompiled so that the command string that VTAM terminal users enter to log on through MainView Alternate Access from a terminal is shortened. *See* BBVLIT

logon mode. *See* logmode

LOGTAB. *See* logon interpret table

M

MainView. Boole & Babbage's integrated systems management architecture. Currently, there are two versions of this architecture, BBI-2 and BBI-3.

MainView Selection Menu. Common menu from which to access all Boole & Babbage MainView products.

MainView Alternate Access. Product component that provides EXCP and VTAM interfaces for terminal access to Boole & Babbage products.

manual customization. Process of customizing a Boole & Babbage product by following the instructions for modifying sample members shipped with the product, using ISPF services. *Contrast with* AutoCustomization

minor node. In VTAM programs, a uniquely defined resource within a major node.

MainView for OS/390. Boole & Babbage performance management tool designed to gather and store information on the performance of your system in real, interval, and session time.

P

proc. Startup procedure.

PERFORMANACE MANAGER. Set of panels and functions common to more than one Boole & Babbage product that runs under ISPF. The Boole & Babbage products are integrated into the PERFORMANACE MANAGER menu and its workbench of services.

R

RACF. Resource access control facility, which is an external security manager that provides security for a system by intercepting resource requests prior to the actual access.

RESOLVE PLUS. Boole & Babbage realtime software tool that detects, diagnoses, and corrects problems in an MVS environment.

RMBBVUJV. BBSAMP member shipped with MainView Alternate Access that contains a job that relinks IEFUJV to remove VTAM/EXCP Terminal Monitor's BBVUJV exit. *See* VTAM/EXCP Terminal Monitor

S

SAF. System Authorization Facility. MainView Alternate Access uses a SAF interface for security.

session. (1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection. (2) Period of time during which a user of a terminal can communicate with an interactive system, usually elapsed time between logon and logoff.

SMP. *See* System Modification Program

START command. Command issued by an operator, automation product, or the IPL procedure to initiate the LAS or TAS in MainView Alternate Access.

started task. Task initiated at the system console.

STOP command. Command issued by an operator to terminate the LAS.

SYS1.PROCLIB. Procedure library used to execute procedures such as startup.

System Modification Program. Product packaging method that provides installation and servicing functions for Boole & Babbage products.

T

TAS. *See* Terminal Address Space

Terminal Address Space. Address space that supports a session that has access to a Boole & Babbage product through either the VTAM or EXCP communications protocols. A Terminal Address Space can be started by a user issuing a logon command at a VTAM terminal or a START command being issued for either a VTAM or EXCP terminal session.

TSO subsystem. TSO subordinate or secondary system, capable of operating independently of, or asynchronously with, a controlling system.

U

user ID. String of characters that uniquely identifies a user to a system.

Unformatted system services (USS) table. In SNA products, a system services control point (SSCP) facility that translates a character-coded request, such as the BBV

abbreviated logon command, into a field-formatted request for processing by formatted system services. A BBVUSST sample table entry ships with MainView Alternate Access, which can be added to the USS table and the table recompiled so that the command string that VTAM terminal users enter to log on through MainView Alternate Access from a terminal is shortened. *See* BBVUSST

USSTAB. *See* unformatted system services table

V

Virtual Telecommunications Access Method. IBM program product that links host-resident applications to terminals. It supports SNA. *Same as* VTAM

VTAM. *See* Virtual Telecommunications Access Method

VTAM application major node definition. Within the VTAM program, a set of resources that can be activated and deactivated as a group. If VTAM access is required through MainView Alternate Access, then a VTAM application major node definition must be defined with some number of minor nodes, and the major node must be activated prior to logon attempts. A BBVTASA sample application major node member ships with MainView Alternate Access. *See* BBVTASA

VTAM minor node root name. Within the VTAM program, the name by which the host knows the object. The minor node root name is also called an application prefix name. The default name for MainView Alternate Access VTAM sessions is BBVTAS and is defined in the VTAM application major node definition. A BBVTASA sample application major node member ships with MainView Alternate Access. *See* BBVTAS and BBVTASA

VTAM/EXCP Terminal Monitor. Boole & Babbage product that has been replaced by MainView Alternate Access.

Index

A

- abbreviated logon command
 - defining in logon interpret table 20
 - defining in USS table 19
 - using 60
- activating
 - LAS 54
 - TAS for EXCP session 79
 - TAS for VTAM session 67
- address space
 - LAS 4
 - TAS 5
- APF-authorized load modules 89
- architecture
 - limitations 6
 - overview 2
- ATTN key, limitation 7
- AutoCustomization
 - BBV steps 10
 - bypassing BBV customization 11
 - for CMF MONITOR 9
 - help 10
 - LAS 11
 - startup 9
 - TAS 11
- AutoLogon
 - defined 5
 - EXCP session
 - BBVTASxx member 85
 - initialization methods 78
 - START command 79
 - TAS parameters 79
 - terminating 87
 - security issues 39
 - VTAM session
 - BBVTASxx member 73
 - initialization methods 65
 - START command 67
 - TAS parameters 66
 - terminating 75
- AutoOPERATOR and AutoLogon 1

B

- BBPARM members 90
- BBSAMP members 90
- BBV
 - AutoCustomization steps 10
 - component code 1
 - functions 1
- BBVINIT 6
 - description of 91
 - example 15
 - overview of REXX EXEC 6
 - updating 8

- BBVINITC 91
- BBVISPF 91
- BBVLAS
 - customizing 27
 - description of 91
 - example 27
 - using 56
- BBVLAS00
 - defining 25
 - description of 90
 - example 26
 - parameters 35
 - using 56
- BBVLIT
 - customizing 20
 - description of 91
- BBVLVER
 - description of 91
 - using 29
- BBVTAS
 - description of 91
 - example 24
- BBVTAS00
 - description of 90
 - sample 22
- BBVTASA
 - customizing 17
 - description of 91
 - VTAM major node name 17
- BBVUSST
 - description of 91
 - example 19
- bypassing AutoCustomization 11

C

- CA-ACF2
 - APPL access 45
 - considerations 2
 - MainView Alternate Access 42
 - release requirements 8
 - TERMINAL access authority 45
- CAPS=
 - LAS 35
 - START command for EXCP 83
 - START command for VTAM 71
 - TAS 32
- CA-TOP SECRET
 - APPL access authority 51
 - considerations 2
 - MainView Alternate Access 49
 - release requirement 8
 - TERMINAL access authority 51

CLIST

- BBVINITC 91
- MAINVIEW 6
- CMF MONITOR and AutoCustomization 9
- concurrent TSO and BBV sessions
 - defining log and list data sets 14
 - defining profile data sets 14
- CSA requirements 8
- customization
 - manual 13
 - LAS initialization parameters 25
- customizing 9

D

- DASD requirements 7

G

- generic name for VTAM 35
- GNAME parameter 35

H

- hardware requirements 7
- help while using AutoCustomization 10
- hilevel qualifier 9

I

- initialization
 - controlling LAS parameters 56
 - controlling TAS parameters 66
 - default LAS parameter list 25
 - default TAS parameter list 21
 - defining additional parameter lists 31
 - LAS parameters 35
 - TAS parameters 31
- IPL
 - AutoLogon sessions 1, 3
 - procedure for EXCP AutoLogon session 78
- ISPF
 - maintaining 8
 - manually customizing 13
- ISPF requirement 8
- ISPPROF profile data set 6

K

- Katakana support (CAPS=)
 - LAS 35
 - START command for EXCP 83
 - START command for VTAM 71
 - TAS 32

L

LAS

- activating VTAM major node 54
- customizing 11, 25
- functions 4
- initialization parameters 56
- sample proc 91
- starting 54, 55
- starting more than one 55
- startup procedure 27
- stopping 54, 56
- libraries shipped with BBV 89
- limitations 6
- load modules
 - list of 89
 - verifying with BBVLVER 29
- logging off 64
- logging on
 - abbreviated command 60
 - to VTAM with LOGON command 57
- Logon Address Space
 - see LAS
- LOGON command 57
- logon interpret table 20
- LOGONID for CA-ACF2 42
- LOGTAB 20

M

- maintenance 8
- MainView Alternate Access
 - applications security 41
 - architecture 3
 - AutoLogon sessions 39
 - CA-ACF2 42
 - CA-TOP SECRET 49
 - component code 1
 - functions 1
 - RACF 47
 - securing 39–52
 - security 2
 - terminal security 40
- MAINVIEW CLIST 6
- major node
 - activating 54
 - defining 17
- manual customization
 - defining ISPF environment 13
 - TAS 21
 - using 13
 - USSTAB entry 19
- migrating from test to production 37
- minor node, defining 17

N

- NETVIEW 17

O

overview of architecture 2

P

parameters

- LAS initialization 35
- LOGON command 57
- START command for EXCP 79
- START command for VTAM 67
- TAS initialization 31

password

- EXCP START command 82
- on LOGON command 58
- panel 63
- VTAM START command 70

priority of TAS 7

procedures

- startup for LAS 27
- startup for TAS 23

Product Customization Steps panel 10

production migration 37

R

RACF

- APPL access authority 48
- considerations 2
- MainView Alternate Access 47
- release requirement 8
- TERMINAL access authority 48

requirements

- hardware 7
- software 8
- TAS priority 7

REXX EXEC

- BBVINIT 91
- BBVINITC 91

S

SAF interface 2

security

- CA-ACF2 release requirement 8
- CA-TOP SECRET release requirement 8
- considerations 2
- implementing 39–52
- RACF release requirement 8

software requirements 8

START command

- EXCP AutoLogon 79
- LAS 55
- VTAM AutoLogon 67

stopping

- EXCP AutoLogon session 87
- LAS 56
- VTAM AutoLogon session 75

T

TAS

- customizing 11, 21
- defining startup procedure 23
- functions 5
- initialization parameters 5
- priority of 7
- sample startup procedure 91
- starting EXCP session 79
- starting VTAM session 67

Terminal Address Space

see TAS

terminal security, MainView Alternate Access 40

terminating

- EXCP AutoLogon session 87
- LAS 56
- VTAM AutoLogon session 75

test to production migration 37

TSO subsystem requirement 8

U

unformatted systems services table 19

user ID panel 62

USSTAB entry

- description of 91
- manual customization 19

V

verifying load modules 29

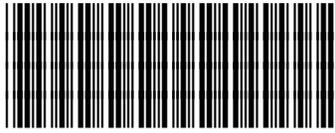
virtual storage requirement 7

VTAM

- abbreviated logon 60
- activating major node 54
- LAS functions 4
- logging on through 4
- logging on to terminal 53
- minor node root name 17
- requirement 8
- using generic names 35

VTAM/EXCP Terminal Monitor 1

Notes



100020658